THE UNIVERSITY OF MANITOBA

DEVELOPMENT AND APPLICATION OF THE UNSYMMETRIC LANCZOS REDUCTION METHOD

BY

HENIAN LI

DISSERTATION
PRESENTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

WINNIPEG, MANITOBA
December 1996

ISBN 0-612-16191-9

Canada

Name Henian Li

Dissertation Abstracts International and *Masters Abstracts International* are arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation or thesis. Enter the corresponding four-digit code in the spaces provided.

SUBJECT TERM

Applied Mathematics and Geological Engineering

0405 UMI

SUBJECT CODE

0543
0775

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Design and Decorative Arts 0389
Fine Arts 0357
Information Science 0723
Journalism 0391
Landscape Architecture 0390
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Educational Psychology 0525
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291
- Rhetoric and Composition 0681

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- Caribbean 0360
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Church 0330
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Fisheries and Aquaculture 0792
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Range Management 0777
- Soil Science 0481
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Animal Physiology 0433
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Plant Physiology 0817
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Dentistry 0567
- Education 0350
- Administration, Health Care 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Safety 0354
- Oncology 0992
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Rehabilitaion and Therapy 0382
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Condensed Matter 0611
- Electricity and Magnetism 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984
Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Environmental 0775
- Industrial 0546
- Marine and Ocean 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Cognitive 0633
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

DEVELOPMENT AND APPLICATION OF THE UNSYMMETRIC LANCZOS REDUCTION METHOD

BY

HENIAN LI

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Abstract

The purpose of this dissertation is to further develop the Unsymmetric Lanczos Reduction (ULR) method as a practical approach to solve the advection dispersion equation. After spatial discretization by Finite Element (FE) method or Finite Difference (ED) method, the ULR method uses two-sided Gram-Schmidt M-biorthogonalization procedure to reduce the discretized system to a very small system. Thus, large computational savings over classical time integration can be achieved. Unlike other successful modal reduction methods, which use a one-sided Gram-Schmidt M-orthogonalization procedure, it is known that the ULR method sometimes fails because of (1) breakdown problems or (2) the numerical instability problems during the time stepping scheme for solving the reduced system.

In this thesis, the Maximum-Pivot New-Start Vector (MPNSV) method and the Switch method are developed for overcoming breakdown problems. When breakdown occurs, the MPNSV method generates a new starting vector which has a maximum pivot. If the breakdown is pathological, that is the maximum pivot is still less than specified pivot tolerance, the Switch method can be used to change over the ULR method to the one-sided Gram-Switch M-orthogonalization method to continue the solution process.

The Eigenvalue Translation (ET) technique is also developed to stabilize the reduced system. Because the reduced system is sometimes numerically unstable due to the eigenvalues with negative real part, the ET method translates these eigenvalues to the right half complex plane while keeping the others unchanged. This approach ensures that the translated system is stable.

A robust ULR algorithm is therefore achieved that includes the algorithm for monitoring and terminating the ULR procedure by evaluating the relative

residual error bounds. All of these improvements make the ULR method much more efficient than the classic Crank-Nicolson method for solving the advection dispersion equation.

Numerical experiments are also presented in this dissertation. The analyses of the Root-Mean Square (RMS) error and the maximum error with respect to the results from the classic approach yield insight into the efficiency and accuracy of the ULR method. Experiments showed that the ULR method can save 95% of the execution time of other standard methods.

The main contribution of the ULR method is the application of the method to the multi-species decay chain problem. This method makes the "one-step" approach which deals with advection, dispersion, decay and species transforming in one step to be a easy task. However, there is a convergence problem in that a common starting vector is required to ensure convergence for all species. In this thesis, this problem is overcome by analyzing the residual errors and the right hand side vectors of the semi-discretized equations of each species. All of the simulations show tremendous saving in computation time over other methods, and storage saving particularly for the multi-species decay chain problem as well.

Acknowledgements

I am very grateful to my supervisors, Dr. Allan D. Woodbury and Dr. Peter Aitchison, for their encouragement and guidance from research work to the actual proof-reading of my thesis. I have benefited greatly in many ways from working with them.

Sincere thanks to Dr. Matthew Yedlin (University of British Columbia), Dr. Dunbar Scott (University of British Columbia), and Dr. R. SriRanjan for their careful reading of this thesis and their helpful suggestions. Dr. Blair W. Nakka (Atomic Energy of Canada Limited (AECL) Whiteshell Laboratories) has also been very kind in providing materials which are very useful for my thesis.

I am especially grateful to my wife Yimin Gu and my daughter Liyun Li for their support that have made my harder days seemed worthwhile. It is impossible to complete this thesis successfully without such support and love during some of the most stressful times of my life.

This thesis was financially supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Postgraduate Scholarship, the University of Manitoba Graduate Fellowship, and the Atomic Energy of Canada Limited (AECL) Whiteshell Research Establishment through contract.

Contents

List of Figures

List of Tables

Part I

Preliminaries

Chapter 1

Introduction

1.1 Introduction

The research presented in this thesis focuses on unsymmetric modal reduction methods for solving the time-dependent advection dispersion equation. This equation describes the mass transport processes of advection, diffusion, dispersion, absorption, and decay in the subsurface. More specifically, in the absence of other sources/sinks and absorption in a solution devoid of chemical reactions, the governing equation of the problems thus can be expressed as

$$\nabla \cdot (\mathbf{D} \cdot \nabla C_i) - \mathbf{v} \cdot \nabla C_i - \mu_i C_i + \sum_{t=1}^{\ell} \xi_{it} \mu_t C_t = \frac{\partial C_i}{\partial t}, \tag{1.1}$$

where C_i denotes mass concentration per unit volume of the i'th species, $\mathbf{D}$ a rank two tensor of the coefficient of the hydrodynamic dispersion, $\mathbf{v}$ the transport velocity vector which depends on the spatial coordinates, μ_i the decay constant of radionuclide species i, ξ_{it} the fraction of (parent) component t transforming into (daughter) component i and the upper limit of the summation ℓ is the number of parent components transforming into component i (see [30]). The transport of

single species without decay is mainly considered in this thesis. General radionuclide decay chain problems are discussed in Chapter 7.

The numerical method to solve equation (1.1) of single species with initial and boundary conditions involves establishing a semi-discretized linear system by using an appropriate finite element or finite difference method, that is

$$\mathbf{M}\dot{\mathbf{c}} + \mathbf{K}\mathbf{c} = \mathbf{f}, \tag{1.2}$$

where $\mathbf{c}$ is the concentration vector of n unknowns at the nodes of the mesh, $\dot{\mathbf{c}}$ the derivative with respect to time t, $\mathbf{f}$ a time-dependent force vector containing the effects of the sources and boundary conditions, $\mathbf{M}$ and $\mathbf{K}$ are $n \times n$ "capacity" and "conductivity" matrices, respectively. Here, $\mathbf{M}$ is positive-definite symmetric and $\mathbf{K}$ is unsymmetric. In this thesis, $\mathbf{f}$ is assumed to be of the form $\mathbf{f} = \mathbf{b}\mu(t)$ where $\mathbf{b}$ is a time independent vector and $\mu(t)$ is a scalar function of t. A detailed formulation on a general form of $\mathbf{f}$ was given by [15] and is also discussed in section 7.6.

Common numerical methods to solve equation (1.2) consist of various time-stepping schemes [52], such as the Crank-Nicolson method. This method discretizes equation (1.2) in time, and then obtains solutions by recursive substitutions. For instance, the Crank-Nicolson algorithm gives

$$\tilde{\mathbf{M}}\mathbf{c}^{s+1} = \tilde{\mathbf{K}}\mathbf{c}^{s} + \frac{1}{2}\Delta t(\mathbf{f}^{s+1} + \mathbf{f}^{s}), \tag{1.3}$$

where $\tilde{\mathbf{M}} = \mathbf{M} + \Delta t\mathbf{K}/2$, $\tilde{\mathbf{K}} = \mathbf{M} - \Delta t\mathbf{K}/2$, Δt the time step and s is the time level. Provided the time step remains constant during the simulation, the recur-

sive solutions to the above equation can be obtained by keeping the left hand side matrix in the factored form, $\tilde{\mathbf{M}} = LU$, followed by repeating the forward and the backward substitution procedures. Because there are constraints on Δt to maintain numerical stability, a large number of time steps may be required to obtain accurate solutions [52]. In addition, because of these constraints, small spatial discretizations must be used and consequently (1.2) is usually a large time-dependent linear unsymmetric system [10]. Thus, computational "cost" is one of the main concerns in solving these equations. Recently, there has been research into reduction methods to solve these problems, for example the Symmetric Lanczos Reduction (SLR) method and the Arnoldi Reduction (AR) method. These methods project the large system into a very small Krylov subspace constructed using M-orthogonal bases, giving a small-sized system of the first order differential equations of the form

$$\mathbf{T_m}\dot{\mathbf{w}} + \mathbf{G_m}\mathbf{w} = \mathbf{g}, \tag{1.4}$$

where $\mathbf{T}_m$ and $\mathbf{G}_m$ are $m \times m$ matrices, $\mathbf{g}$ is m-vector and $m \ll n$. Approximate solutions can then be obtained by solving this reduced system. Either method is economical in computation for solutions of (1.4) using a time-stepping scheme. The SLR method applies the symmetric Lanczos algorithm [35] to the symmetric matrix $(\mathbf{K} + \mathbf{K}^*)^{-1}$, using a three term recurrence to construct the tridiagonal matrix $\mathbf{T}_m$ and the full matrix $\mathbf{G}_m$. Here, the superscript $*$ denotes the transposed matrix. The AR method applies the Arnoldi algorithm [2] to the unsymmetric matrix $\mathbf{K}^{-1}$. In the AR algorithm, the recurrence involves all previously produced vectors, (unlike the SLR algorithm where only three terms are used), resulting in

matrices $\mathbf{T}_m$ which is the upper Hessenberg and $\mathbf{G}_m$ which is an identity matrix. Details can be seen in [15, 42, 43, 44, 45, 65]. Similar ideas have been used previously in different applications, (for instance, [54, 20, 50, 41]).

Alternatively, in this thesis, the Unsymmetric Lanczos Reduction (ULR) method is explored. This method applies a two-sided Gram-Schmidt M-biorthogonalization process to the unsymmetric matrix $\mathbf{K}^{-1}$ and forms two Krylov subspaces based on M-biorthogonal vectors [35]. Two-sided three-term recurrence equations are used to form these vectors, resulting in a matrix $\mathbf{T}_m$ which is tridiagonal and $\mathbf{G}_m$ which is an identity matrix. Like the SLR and AR methods, the main advantage of the ULR method is the large savings of computational cost over the other time-stepping schemes, particularly for long-time period prediction.

The use of the Lanczos method for unsymmetric problems was discussed in the early papers of Lanczos [35, 36], which is essentially an eigenvalue approximation method. The detailed procedure can be seen in [63, 24]. Various algorithms can be seen in the works of [25, 34, 46, 53]. Also, the Lanczos method has been developed to solve system of linear equations by using residuals lying in a Krylov subspace associated with the unsymmetric coefficient matrix of the system. There are several algorithms variable, such as Generalized Conjugate Gradients (GCG) [3, 4, 31, 32], Orthomin [59], Orthodir [68], Manteuffel-Chebyshev [38, 39], Bi-Conjugate Gradients (BCG) [18], Induced Dimension Reduction [61], and hybrid combinations of CG generalizations and Chebyshev [17]. Others can been seen in [12, 40, 49, 56, 62].

However, in the past, a number of mathematical challenges have hindered the application of this type of method to the system of linear equations discretized

from the time dependent Partial Differential Equations (PDE), such as advection dispersion equations. One of these is the "breakdown" or "near breakdown" problem which occurs when a zero or near-zero pivot (i.e., a zero or near-zero value in a division) is produced. A zero pivot will cause the algorithm to fail and a near-zero pivot may cause numerical instability [21]. This situation can not occur in either the SLR or AR processes.

Another difficulty arises when a reduced system loses accuracy during the time-stepping process (instability). Even when the original semi-discretized problem (1.2) is stable with respect to time, (i.e., all the eigenvalues of the system (1.2) have positive real parts [11]), the reduced system may not be stable. Note that modal reduction methods are basically eigenvalue approximation methods. If a 'true' eigenvalue is located near or on the imaginary axis, then an approximate eigenvalue may be relocated to the left half of the complex plane. An example can be seen in Chapter 5, Figure 5.2. However, generally, this situation can not occur in the AR method; a detailed discussion can be seen in §4.1 later or [20].

The contributions of this thesis are:

- development of a Maximum-Pivot New-Start Vector (MPNSV) method to overcome breakdown or near breakdown situations in the ULR approach,
- a solution for the case of a "pathological breakdown" by switching to an Arnoldi reduction-based method from ULR process,
- analysis of relative residual error bound for single and multi-species, which can be used to monitor and terminate the ULR process.
- development of an Eigenvalue Translation (ET) technique to overcome in-

stability problems in time,

- successful applications of the ULR method to single and multi-species decay problems and analyses of these applications including error comparison, advantages, limitations and
- a solution for the starting vector of the ULR method applied to the multi-species decay chain problems.

The main contributions are more fully discussed below.

Various methods of trying to overcome the breakdown and near breakdown problems have been developed. For example, the well-known look-ahead Lanczos algorithm (LAL) was provided in [51]. A "nongeneric" Lanczos algorithm was developed in [26, 27] and an alternate method called the New-Start Vector method was given in [66]. Some new algorithms that substantially mitigate the problem of breakdown for nonsymmetric systems of linear equations were given in [31].

In this thesis, when breakdown occurs, a new starting vector with the maximum pivot is constructed and replaces the old Lanczos vector. A new Krylov subspace is then formed by adding the subspace generated by this vector to the old Krylov subspace. The resulting reduced system becomes a "special band matrix", with lower band width of one and an irregular upper band. This approach is a modification of the New-Start Vector method of [66], so it is referred as Maximum-Pivot New-Start Vector (MPNSV) method.

Unfortunately, the breakdown may be pathological; i.e., the maximum pivot is very small or less than the pivot tolerance. It is then impossible to find a new starting vector within the subspace which is orthogonal to the generated right

Krylov subspace, because that the pivot of the new starting vector produced by the MPNSV method is the maximum. If this situation occurs, the breakdown is considered to be pathological and the Switch method is used to change over to an Arnoldi Reduction method in order to continue the solution process.

Our implementation, which constructs a new starting vector, is different from the one in [66]. There, a new starting vector is chosen randomly so that the pivot may not necessarily be improved. If this situation happens, the method chooses another new starting vector randomly. After a number of unsuccessful trials, the method decreases the pivot tolerance to force the process to continue. Decreasing the pivot tolerance could lose accuracy of the solution. The experiment in §5.2 shows that there is a balance between pivot tolerance and accuracy. Moreover, if the right Lanczos vectors are almost linearly dependent, the maximum pivot must be very small (discussion in §3.2). Thus decreasing the pivot tolerance will cause numerical instability. A detailed discussion about linear dependence can be seen in [19, 60]. The intention in developing the MPNSV method and the Switch method is to develop a robust algorithm that can handle these situations.

In order to stabilize the reduced system which is generated by the ULR method, the Eigenvalue Translation (ET) technique is used to translate all eigenvalues from the left half complex plane to the right half while keeping all others in the right plane, thus making the translated system stable. Although this approach to the problem is new, the idea translating eigenvalues to stabilize a linear system has been around for a long time. Eigenvalue translation has been exploited in [33] to translate eigenvalues located far from the point $(1, 0)$ in the complex plane into a vicinity of that point in order to improve the convergence rate of the GMRES(k)

method. In [55], unstable eigenvalues are translated from one half complex plane to the other half to ensure a stable algorithm for partial pole assignment in linear state feedback using Projection and Deflation methods. Another approach for overcoming this instability is described by Grimme at al [23] based on an Implicitly Restarted Lanczos method to stabilize a system in which unstable eigenvalues are discarded.

To test and demonstrate the ULR method, this thesis gives the results of numerical experiments performed by the method, including error analyses and comparisons of the ULR results against the classic Crank-Nicolson method. The classic Crank-Nicolson method is defined as the directly applying Crank-Nicolson method to equation (1.2) without reduction. Applications to the simple one and two-dimensional examples are demonstrated. The representation of physical field problems is studied.

Also, applications of the ULR method to the multi-species radionuclide decay chain problems are developed. The existing "one-step" approach as referred in the text [14] to solve these problems is numerically complicated. In this reference, it is commented that "Most of the more general implementations of this scheme have been for one-dimensional transport. Extending this approach to two-dimensional problems is a formidable task". In this thesis, the application of the ULR method enables the extension of the "one-step" approach to higher dimensional problems. The application includes a new method for choosing a common starting vector for all species in order to ensure the ULR method is convergent. This convergence is ensured for the single species model (see [16, 43]), but that result does not generalize to the multi-species very well.

These applications have showed the tremendous advantages in computation time and storage saving. It can be concluded that the ULR method together with the ET technique is an effective alternative for solving contaminant transport problems in porous media, and in particular for the radionuclide decay chain problems.

1.2 Organization of the thesis

This thesis is organized as follows. In the next chapter, a basic overview of the SLR and AR methods is introduced. Chapter 3 deals with the Unsymmetric Lanczos Reduction (ULR) method. The implementation of the MPNSV method for handling the breakdown or near breakdown situation is discussed. The Switch method for overcoming a pathological breakdown is also described, and the relative residual error bound which is used to monitor and determine the termination of the process is given. Chapter 4 explores some of the theoretical aspects of instability. This analysis includes a new method of Eigenvalue Translation (ET) technique. In Chapter 5, 6 and 7, the ULR method is applied to the transport problems in subsurface hydrology. The results of numerical experiments of these applications are presented. Chapter 5 analyzes simple one and two-dimensional problems. Chapter 6 illustrates the application of the methods to field problems and Chapter 7 discusses radionuclide decay chain problems. Finally, the thesis concludes with some remarks in Chapter 8.

Householder notational conventions with a few noted exceptions are used in this thesis. Bold upper case and lower case Roman letters denote matrices and column vectors, respectively. Bold zero,$\mathbf{0}$, is a matrix or vector with all zero value

entries. Lower case Greek denotes scalars. The superscript $*$ denotes the conjugate transpose of a complex matrix or vector, or just transpose of a real matrix or vector. m, n, ℓ are integers for dimensions of matrices and i, j, k, τ are index variables. t is the time variable. $\mathbf{I}_n = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n]$ is the identity matrix with dimension n, where the columns $\mathbf{e_i}$, $i = 1, 2, \ldots, n$ is n-vector whose components are zero except that the i'th is one. $\mathbf{S_{Q_j}}$ denotes the subspace spanned by the column vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j$, of the matrix $\mathbf{Q}_j$, i. e. $\mathbf{S_{Q_j}} = span\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j\}$. $\mathbf{S}_{\mathbf{Q}_j}^{\perp}$ is the corresponding complementary subspace. $\Re^n$ is the usual linear vector space over the real numbers with dimension n. $\|\cdot\|$ in this thesis represents the M-norm defined by $\|\mathbf{x}\| = (\mathbf{x}^*\mathbf{M}\mathbf{x})^{1/2}$, unless otherwise stated. Throughout this paper, n is the size of the original system (1.2) and m is the size of the reduced system so that $m \leq n$.

Chapter 2

The Symmetric Lanczos Reduction and the Arnoldi Reduction methods

2.1 The Symmetric Lanczos Reduction (SLR) method

In this chapter, the SLR and the AR methods are reviewed. Similar algorithms and proofs of these results can be seen from [21, 15, 16, 42, 43, 44, 45, 65].

The SLR process begins by changing (1.2) to

$$\mathbf{A}^{-1}\mathbf{M}\dot{\mathbf{c}} + \mathbf{c} + \mathbf{A}^{-1}\mathbf{S}\mathbf{c} = \mathbf{A}^{-1}\mathbf{b}\mu(t), \tag{2.1}$$

where $\mathbf{A}$ is the symmetric part of $\mathbf{K}$, i. e., $\mathbf{A} = (\mathbf{K} + \mathbf{K}^*)/2$, $\mathbf{S}$ is the skew–symmetric part, i. e., $\mathbf{S} = (\mathbf{K} - \mathbf{K}^*)/2$, $\mathbf{b}$ is a time independent vector and $\mu(t)$ is a scalar such that $\mathbf{f}(t) = \mathbf{b}\mu(t)$. The method generates a set of Lanczos vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m$ $(m \ll n)$ to form a M-orthogonal matrix, $\mathbf{Q}_m = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m)$ (i. e., $\mathbf{Q}_m^*\mathbf{M}\mathbf{Q}_m = \mathbf{I}_m$), as follows:

Algorithm 2.1 *Start from an initial vector* $\mathbf{r}_0 = \mathbf{A}^{-1}\mathbf{b}$, *then calculate for* $j = 0,1,2,\ldots,m-1$, *recursively,*

$$
\begin{aligned}
(1) \quad \beta_{j+1} &= (\mathbf{r}_j^* \mathbf{M} \mathbf{r}_j)^{\frac{1}{2}}, \\
(2) \quad \mathbf{q}_{j+1} &= \frac{1}{\beta_{j+1}} \mathbf{r}_j, \\
(3) \quad \alpha_{j+1} &= \mathbf{q}_{j+1}^* \mathbf{M} \mathbf{A}^{-1} \mathbf{M} \mathbf{q}_{j+1}, \\
(4) \quad \mathbf{r}_{j+1} &= \begin{cases} \mathbf{A}^{-1}\mathbf{M}\mathbf{q}_{j+1} - \alpha_{j+1}\mathbf{q}_{j+1} - \beta_{j+1}\mathbf{q}_j, & j \neq 0, \\ \mathbf{A}^{-1}\mathbf{M}\mathbf{q}_{j+1} - \alpha_{j+1}\mathbf{q}_{j+1}, & j = 0. \end{cases}
\end{aligned}
$$

Thus, the matrix $\mathbf{Q}_m = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m]$ and the tridiagonal matrix

$$
\mathbf{T}_m = \begin{bmatrix}
\alpha_1 & \beta_2 & & & & \\
\beta_2 & \alpha_2 & \beta_3 & & & \\
 & \beta_3 & \alpha_3 & \ddots & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \ddots & \alpha_{m-1} & \beta_m \\
 & & & & \beta_m & \alpha_m
\end{bmatrix} \tag{2.2}
$$

are formed.

From the above algorithm, the following equations hold

$$
\mathbf{A}^{-1}\mathbf{M}\mathbf{Q}_m = \mathbf{Q}_m \mathbf{T}_m + \mathbf{r}_m e_m^* \tag{2.3}
$$

and

$$
\mathbf{Q}_m^* \mathbf{M} \mathbf{Q}_m = \mathbf{I}_m, \quad \text{and} \quad \mathbf{Q}_m^* \mathbf{M} \mathbf{r}_m = \mathbf{0}, \tag{2.4}
$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix and the m-vector $\mathbf{e}_m$ is the m'th column of matrix $\mathbf{I}_m$. Two equations in (2.4) are referred to as the M-orthogonality property.

Based on equations (2.3) and (2.4), we apply the standard Rayleigh-Ritz reduction procedure to (2.1). Substituting the approximate transformation

$$\mathbf{c} = \mathbf{Q}_m \mathbf{w} \tag{2.5}$$

into (2.1), where $\mathbf{w}$ is the m-vector, and then left-multiplying $\mathbf{Q}_m^* \mathbf{M}$ to the both sides of the result, it follows that

$$\mathbf{Q}_m^* \mathbf{M} \mathbf{A}^{-1} \mathbf{M} \mathbf{Q}_m \dot{\mathbf{w}} + \mathbf{Q}_m^* \mathbf{M} \mathbf{Q} \mathbf{w} + \mathbf{Q}_m^* \mathbf{M} \mathbf{A}^{-1} \mathbf{S} \mathbf{Q}_m \mathbf{w} = \mathbf{Q}_m^* \mathbf{M} \mathbf{A}^{-1} \mathbf{b} \mu(t).$$

Considering equations (2.3) and (2.4) and $\mathbf{A}^{-1}\mathbf{b} = \mathbf{r}_0 = \beta_1 \mathbf{q}_1$, the above equation becomes

$$\mathbf{T}_m \dot{\mathbf{w}} + (\mathbf{I}_m + \mathbf{E}_m)\mathbf{w} = \beta_1 \mathbf{e}_1 \mu(t), \tag{2.6}$$

where $\mathbf{E}_m = \mathbf{Q}_m^* \mathbf{M} \mathbf{A}^{-1} \mathbf{S} \mathbf{Q}_m$, and $\mathbf{T}_m$ is a tridiagonal matrix (2.2).

The initial conditions for the above equation can also be constructed accordingly. Once the solution is obtained by solving the reduced system (2.6), the approximate solution to the original equation (2.1) can be obtained by substituting the solution $\mathbf{w}$ back into (2.5).

2.2 The Arnoldi Reduction (AR) method

The AR method can be also used to reduce the size of system (1.2) to a small set of equations. Unlike the SLR method where the Krylov subspace is formed based on the symmetric matrix $\mathbf{M}\mathbf{A}^{-1}\mathbf{M}$, the AR method generates the subspace

directly based on the unsymmetric matrix $\mathbf{MK}^{-1}\mathbf{M}$. Thus, (1.2) is changed to

$$\mathbf{K}^{-1}\mathbf{M}\dot{\mathbf{c}} + \mathbf{c} = \mathbf{K}^{-1}\mathbf{b}\mu(t). \tag{2.7}$$

The reduction procedure is nearly identical to the SLR method.

Algorithm 2.2 *Starting from a vector* $\mathbf{r}_0 = \mathbf{K}^{-1}\mathbf{b}$, *calculate recursively for* $j = 0, 1, 2, \ldots, m-1$,

$$\begin{aligned}
&(1) \quad \beta_{j+1} = (\mathbf{r}_j^t \mathbf{M} \mathbf{r}_j)^{\frac{1}{2}} \\
&(2) \quad \mathbf{q}_{j+1} = \frac{1}{\beta_{j+1}} \mathbf{q}_j \\
&(3) \quad \alpha_{j+1} = \mathbf{q}_{j+1}^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_{j+1} \\
&(4) \quad \gamma_{j+1}^{(j+1-i)} = \mathbf{q}_i^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_{j+1}, \quad i = 1, 2, \ldots, j, \\
&(5) \quad \mathbf{r}_{j+1} = \begin{cases} \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_{j+1} - \sum_{i=1}^{j} \gamma_{j+1}^{(i)} \mathbf{q}_{j+1-i} - \alpha_{j+1}\mathbf{q}_{j+1} & j \neq 0 \\ \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_1 - \alpha_1 \mathbf{q}_1 & j = 0 \end{cases}
\end{aligned}$$

Thus the M-orthogonal matrix $\mathbf{Q}_m = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m)$ and the Hessenberg matrix

$$\mathbf{H}_m = \begin{bmatrix} \alpha_1 & \gamma_2^{(1)} & \gamma_3^{(2)} & & \gamma_{m-1}^{(m-2)} & \gamma_m^{(m-1)} \\ \beta_2 & \alpha_2 & \gamma_3^{(1)} & & \cdot & \cdot \\ & \beta_3 & \alpha_3 & & \cdot & \cdot \\ & & \ddots & \ddots & \gamma_{m-1}^{(1)} & \gamma_m^{(1)} \\ & & & \ddots & \alpha_{m-1} & \gamma_m^{(m-1)} \\ & & & & \beta_m & \alpha_m \end{bmatrix} \tag{2.8}$$

are formed.

Analogous to the SLR method, we also have the following results

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m = \mathbf{Q}_m\mathbf{H}_m + \mathbf{r}_m e_m^* \tag{2.9}$$

and

$$\mathbf{Q}_m^*\mathbf{M}\mathbf{Q}_m = \mathbf{I}_m \quad \text{and} \quad \mathbf{Q}_m^*\mathbf{M}\mathbf{r}_m = \mathbf{0}, \tag{2.10}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix and the m-vector $\mathbf{e}_m$ is the mth column of matrix $\mathbf{I}_m$.

The same Rayleigh-Ritz reduction process as in the SLR method is used. Multiplying $\mathbf{Q}_m^*\mathbf{M}$ to (2.7) and substituting the approximate transformation

$$\mathbf{c} = \mathbf{Q}_m\mathbf{w} \tag{2.11}$$

into the result, and using (2.9) and (2.10),

$$\mathbf{H}_m\dot{\mathbf{w}} + \mathbf{w} = \mathbf{e}_1\beta_1\mu(t) \tag{2.12}$$

follows, where $\mathbf{H}$ is the Hessenberg matrix (2.8). Similar to the SLR method, the initial conditions for the above equation can also be constructed accordingly. Once the solution is obtained by solving the reduced system (2.12), we may obtain the approximate solution to the original equation (2.7) by substituting the solution $\mathbf{w}$ back into (2.11).

2.3 Conclusions

In this chapter, the SLR and AR methods are briefly reviewed. Both methods

are based on the Gram-Schmidt M-orthogonalization process to construct an M-orthogonal basis for a Krylov subspace. The basic difference between them is that the Gram-Schmidt process is applied to a symmetric matrix in the SLR method, and a unsymmetric matrix in the AR method. Accordingly, the SLR method uses a three-term recurrence, while in the AR method all produced vectors are presented in the recurrence. Both methods tremendously reduce the size of the system and are very economical in computation time for solving the linear system (2.6) or (2.12). Thus, these methods stand out as potentially important ways of solving transport problems.

Numerical experiments have shown that the either method can fail to perform well if the M-orthogonality is lost. Error analyses, e.g. in [8] indicate that the rounding error significantly affects the M-orthogonality when applying the Gram-Schmidt orthogonalization process ((4) in Algorithm 2.1 and (5) in Algorithm 2.2). Various modifications to the Gram-Schmidt method have been suggested to improve the M-orthogonality. For instance, the Modified Gram-Schmidt (MGS) process [21], the Householder orthogonalizations [60], the Iterated Modified Gram-Schmidt (IMGS) [29] and recently, the Ordering Modified Gram-Schmidt (OMGS) orthogonalization [47]. Based on the theoretical analysis and the examples presented in [47], OMGS appears to provide a powerful alternative to other orthogonalization methods.

The basic concept discussed in this chapter will be generalized and developed in the ULR method in the following chapters.

Part II

Theoretical Aspects of the ULR Method

Chapter 3

Unsymmetric Lanczos Reduction (ULR) method

3.1 Introduction

In this chapter the ULR method is explored for solving the equation

$$\mathbf{K}^{-1}\mathbf{M}\dot{\mathbf{c}} + \mathbf{c} = \mathbf{K}^{-1}\mathbf{b}\mu(t). \tag{3.1}$$

This equation may be obtained by multiplying both sides of equation (1.2) by $\mathbf{K}^{-1}$. Note that $\mathbf{K}^{-1}\mathbf{M}$ is an unsymmetric matrix. Unlike the SLR, this method uses the two-sided Gram-Schmidt biorthogonalization method ([21, 35]) to generate biorthogonal bases of a pair of Krylov subspaces. Unlike the AR method, the ULR method is able to use three-term recurrence to produce the Lanczos vectors.

The current chapter is organized as follows. This section briefly introduces the two-sided Gram-Schmidt biorthogonalization method and the New-Start Vector method [66] which are the bases of the ULR method. The New-Start Vector method has been used to overcome breakdown problems. Then in §3.2 and §3.3, the Maximum-Pivot New-Start Vector (MPNSV) method and the Switching

method are developed to modify the New-Start Vector method. §3.4 presents the algorithm of calculating the relative residual error bound which is used to monitor and terminate the ULR process. §3.5 gives the general algorithm for the ULR method. Finally, §3.6 concludes with some remarks.

3.1.1 The two-sided Gram-Schmidt biorthogonalization method

The discussion of the ULR method begins with the Gram-Schmidt biorthogonalization process. Starting with the left and the right starting vectors $\mathbf{q}_0 = \mathbf{p}_0 = \mathbf{0}$,

$$\mathbf{q}_1 = \mathbf{p}_1 = \frac{\mathbf{K}^{-1}\mathbf{b}}{\|\mathbf{K}^{-1}\mathbf{b}\|}, \tag{3.2}$$

and scalars $\gamma_1 = \beta_1 = 0$, the Gram-Schmidt biorthogonalization process generates a pair of sequences

$$\mathbf{Q}_m = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m\}, \quad \text{and} \quad \mathbf{P}_m = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m\}, \tag{3.3}$$

which are defined in terms of the three-term relations, i. e.,

$$\beta_{j+1}\mathbf{q}_{j+1} = \mathbf{r}_j = \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j - \alpha_j\mathbf{q}_j - \gamma_j\mathbf{q}_{j-1}, \tag{3.4}$$

$$\gamma_{j+1}\mathbf{p}^*_{j+1} = \mathbf{s}^*_j = \mathbf{p}^*_j\mathbf{M}\mathbf{K}^{-1} - \alpha_j\mathbf{p}^*_j - \beta_j\mathbf{p}^*_{j-1}, \tag{3.5}$$

$j = 1, 2, \ldots, m$, where

$$\alpha_j = \mathbf{p}^*_j\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j, \tag{3.6}$$

and β_{j+1}, γ_{j+1} are calculated from the formulae

$$\beta_{j+1} = \|\mathbf{r}_j\|, \quad \gamma_{j+1} = \frac{\mathbf{s}_j^* \mathbf{M} \mathbf{r}_j}{\beta_{j+1}}. \tag{3.7}$$

$\|\cdot\|$ is the M-norm. $(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m)$ and $(\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m)$ are referred as the left and right Lanczos sequences, and

$$\{\mathbf{p}_1^*, \mathbf{p}_1^* \mathbf{M}\mathbf{K}^{-1}, \mathbf{p}_1^* (\mathbf{M}\mathbf{K}^{-1})^2, \ldots, \mathbf{p}_1^* (\mathbf{M}\mathbf{K}^{-1})^{m-1}\}$$

and

$$\{\mathbf{q}_1, \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_1, (\mathbf{K}^{-1}\mathbf{M})^2 \mathbf{q}_1, \ldots, (\mathbf{K}^{-1}\mathbf{M})^{m-1} \mathbf{q}_1\},$$

are the left and right Krylov subspaces, respectively.

In matrix form, it is easy to show that the following properties are true [43].

$$\mathbf{P}_m^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m = \mathbf{T}_m, \quad \mathbf{P}_m^* \mathbf{M}\mathbf{Q}_m = \mathbf{I}_m, \quad \mathbf{P}^* \mathbf{M}\mathbf{r}_m = \mathbf{0} \tag{3.8}$$

and

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m = \mathbf{Q}_m \mathbf{T}_m + \mathbf{r}_m \mathbf{e}_m^*, \tag{3.9}$$

where $\mathbf{I}_m$ is the identity matrix and $\mathbf{T}_m$ is a tridiagonal matrix

$$\mathbf{T}_m = \begin{bmatrix} \alpha_1 & \gamma_2 & & & & \\ \beta_2 & \alpha_2 & \gamma_3 & & & \\ & \beta_3 & \alpha_3 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \alpha_{m-1} & \gamma_m \\ & & & & \beta_m & \alpha_m \end{bmatrix}.$$

The property of $\mathbf{P}_m^*\mathbf{M}\mathbf{Q}_m = \mathbf{I}_m$ is referred as the M-biorthogonality. Equation (3.9) can be obtained by changing equation (3.4) into the form

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j = \gamma_j\mathbf{q}_{j-1} + \alpha_j\mathbf{q}_j + \beta_{j+1}\mathbf{q}_{j+1},$$

and assembling it from $j = 1$ to $j = m$ in matrix form.

$\mathbf{P}_m$ is the trail matrix (so called in [44]) and $\mathbf{Q}_m$ is used in the Rayleigh-Ritz reduction. By substituting the approximate transformation

$$\mathbf{c} = \mathbf{Q}_m\mathbf{w} \tag{3.10}$$

into (3.1), pre-multiplying the resulting by $\mathbf{P}_m^*\mathbf{M}$, and by taking account of relations of (3.2), (3.8) and (3.9), the reduced system is obtained, i.e.,

$$\mathbf{T}_m\dot{\mathbf{w}} + \mathbf{w} = \mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{b}\mu(t) = \beta_1\mathbf{e}_1\mu(t), \tag{3.11}$$

where $\beta_1 = \|\mathbf{K}^{-1}\mathbf{b}\|$ and $\mathbf{e}_1$ is the first column of identity matrix $\mathbf{I}_m$. The last equality holds because of $\mathbf{K}^{-1}\mathbf{b} = \beta_1\mathbf{q}_1$ and biorthogonality of (3.8). The above reduced system is a tridiagonal of size m, where $m \ll n$. Solution of (3.11) is computationally trivial [45].

3.1.2 New-Start Vector method

Analysis of equations (3.4), (3.5) and (3.7) show that the Gram-Schmidt process could prematurely terminate at step j ($j < m$) if:

(1). $\mathbf{r}_j \neq \mathbf{0}$, $\mathbf{s}_j \neq \mathbf{0}$ and

$$\varpi_j = \frac{|\mathbf{s}_j^* \mathbf{M} \mathbf{q}_{j+1}|}{\|\mathbf{s}_j\|}$$

vanishes, or is numerically small. ϖ_j is referred to as the j'th pivot as in [66].

(2). $\mathbf{r}_j \neq \mathbf{0}$ but $\mathbf{s}_j = \mathbf{0}$.

(3). $\mathbf{r}_j = \mathbf{0}$.

Termination in case (3) is an ideal situation, because the right Lanczos sequences define an invariant subspace and the approximation (3.10) is an exact solution. Cases (1) and (2) represent serious problems. If $\varpi_j = 0$ or $\mathbf{s}_j = \mathbf{0}$, the algorithm must terminate because non-zero vectors $\mathbf{p}_{j+1}$ and $\mathbf{q}_{j+1}$ cannot be computed. This is referred as the "breakdown" problem. Moreover, it might be expected that the numerical process is unstable due to rounding error if ϖ_j is very small, (see for example [63]). This is referred as the near-breakdown problem.

Note that the Lanczos and Arnoldi sequences in the SLR and AR methods are M-orthogonal. Hence it is easy to see that both methods are special cases of the two-sided unsymmetric Lanczos method described above. The pivots in each step of the SLR and AR methods are always equal to one, and therefore there are no breakdown or near-breakdown difficulties in these approaches.

In the ULR method, when case (1) breakdown occurs, i.e., $\varpi_j = 0$ or ϖ_j is very small, $\mathbf{r}_j \neq \mathbf{0}$ and $\mathbf{s}_j \neq \mathbf{0}$, the New-Start Vector method chooses a random vector $\mathbf{x}$ and then biorthogonalizes it against $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_j$. This can be achieved

by

$$\mathbf{s}_j = \mathbf{x} - \delta_1\mathbf{p}_1 - \delta_2\mathbf{p}_2 - \ldots - \delta_j\mathbf{p}_j, \tag{3.12}$$

where $\delta_i = \mathbf{q}_i^*\mathbf{M}\mathbf{x}$ for $1 \leq i \leq j$. If the new pivot ϖ_j greater than the prefixed tolerance, the new left Lanczos vector $\mathbf{p}_{j+1}$ may be computed by the formula

$$\mathbf{p}_{j+1} = \frac{\mathbf{s}_j}{\mathbf{s}_j^*\mathbf{M}\mathbf{q}_{j+1}}. \tag{3.13}$$

Otherwise the method uses another random vector. If after a certain number of trials are unsuccessful, the method decreases the tolerance.

After the introduction of the new starting vector $\mathbf{p}_{j+1}$, the method continues to construct biorthogonal bases. To this end, the Gram-Schmidt process should be modified. Generally, the left and right Lanczos sequences are generated by the formulae

$$\beta_{j+1}\mathbf{q}_{j+1} = \mathbf{r}_j = \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j - \alpha_j\mathbf{q}_j - \sum_{t=1}^{k_r^j}\gamma_j^{(t)}\mathbf{q}_{j-t}, \tag{3.14}$$

$$\begin{aligned}\gamma_{j+1}^{(k_s^j+1)}\mathbf{p}_{j+1}^* = \mathbf{s}_j = \mathbf{p}_{j-k_s^j}^*\mathbf{M}\mathbf{K}^{-1} - \alpha_{j-k_s^j}\mathbf{p}_{j-k_s^j}^* - \beta_{j-k_s^j}\mathbf{p}_{j-k_s^j-1}^* \\ - \sum_{t=1}^{k_s^j}\gamma_{j-k_s^j+t}^{(t)}\mathbf{p}_{j-k_s^j+t}^*,\end{aligned} \tag{3.15}$$

where k_r^j and k_s^j are denoted as the number of terms involving γ in the construction of $\mathbf{r}_j$ and $\mathbf{s}_j^*$ at step j, respectively, $\gamma_{j+1}^{(k_s^j+1)} = \gamma_{j+1}^{(k_r^{j+1})}$ is used in the next step, and

$$\alpha_j = \mathbf{q}_j^*\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j, \qquad \gamma_j^{(t)} = \mathbf{p}_{j-t}^*\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j, \tag{3.16}$$

where $t = 1, 2, \ldots, k_r^j$. If no previous breakdown occurs, $k_r^j = 1$ and $k_s^j = 0$. Equations (3.14) and (3.15) therefore are reduced to (3.4) and (3.5). If no breakdown occurs at step j, the New-Start Vector algorithm sets

$$k_r^{j+1} = k_s^j + 1 \qquad \text{and} \qquad k_s^{j+1} = k_s^j. \tag{3.17}$$

Then equations (3.14) and (3.15) are repeatedly computed for $j+1$. If the breakdown occurs at this step, $\mathbf{s}_j$ and $\mathbf{p}_{j+1}$ are reconstructed from (3.12) and (3.13). The algorithm sets

$$k_r^{j+1} = k_s^j + 1 \qquad \text{and} \qquad k_s^{j+1} = k_s^j + 1, \tag{3.18}$$

and goes to the next step.

When case (2) breakdown or near breakdown occurs, i.e., $\mathbf{r}_j \neq \mathbf{0}$ but $\mathbf{s}_j = \mathbf{0}$, there may be two subcases, $k_s^j = 0$ or $k_s^j \neq 0$. If $k_s^j \neq 0$, the new vector $\mathbf{s}_j^*$ can still be constructed by using the subspace S_{P_j}. That is, the algorithm sets

$$k_s^j = k_s^j - 1 \tag{3.19}$$

and then uses (3.15) to compute $\mathbf{s}_j^*$ again. If $k_s^j = 0$, the above approach can not be used because k_s^j becomes negative. Instead, the algorithm randomly chooses a vector and then constructs a new starting vector by the formulae (3.12) and (3.13).

Similar to the previous section, matrices $\mathbf{Q}_m$ and $\mathbf{P}_m$ are denoted by (3.3).

Changing equation (3.14) into the form

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j = \sum_{t=1}^{k_r^j} \gamma_j^{(t)} \mathbf{q}_{j-t} + \alpha_j \mathbf{q}_j + \beta_{j+1}\mathbf{q}_{j+1},$$

and assembling it from $j = 1$ to $j = m$ results in the matrix form equation (3.9). Note that $\mathbf{T}_m$ is the $m \times m$ "special band" matrix with lower band width of one and irregular upper band, whose i'th column for $i = 1, 2, \ldots, m$ is of the form

$$(0, \ldots, 0, \gamma_i^{(k_r^i)}, \gamma_i^{(k_r^i-1)}, \ldots, \gamma_1^{(1)}, \alpha_i, \beta_{i+1}, 0, \ldots, 0)^*,$$

where α_i is the diagonal entry. It can also be verified that the relations (3.8) are valid [66], which will be introduced by Lemma 3.1 in §3.3.2.

According to the above algorithm, if the new pivot ϖ_j is less than the prefixed tolerance, this algorithm utilizes another random vector. After a certain number of unsuccessful trials, the algorithm decreases the tolerance [66]. However, decreasing the pivot tolerance will result in a loss of accuracy in the solution. Table 5.6 in §5.3 shows an example when the pivot tolerance is decreased from 10^{-5} to 10^{-10}, the Root-Mean Square (RMS) errors (defined in §5.1) increase 3-6 orders of magnitude. So, according to an accuracy perspective, decreasing the pivot tolerance may not be a useful approach in some problems. Moreover, if the maximum pivot is very small, §3.3 shows that the right Lanczos vectors are almost linearly dependent. This will cause a loss of M-biorthogonality. [8, 60] have given a detailed discussion about this problem in the orthogonalization algorithm. On the other hand, even if decreasing the pivot tolerance is adopted, one cannot

ascertain how many trials should be taken before decreasing the tolerance. If the maximum pivot of new starting vector is less than prefixed pivot tolerance, any further trial is futile.

Because of these problems, a "Maximum-Pivot New-Start Vector (MPNSV)" method and a "Switch Method" are developed to improve the above problems. The MPNSV method produces a new starting vector whose pivot is the maximum. In other words, if the new pivot is less than a specified tolerance, any further trials that randomly chose a new starting vector are unnecessary. This is referred as a pathological breakdown. The Switch method switches the process to the AR algorithm so as the pivots after switching will always be equal to one. The MPNSV method and Switch method will be discussed in §3.2 and §3.3 respectively.

3.2 The Maximum-Pivot New-Start Vector (MPNSV) method

3.2.1 New starting vector with maximum pivot

Analysis of the New-Start Vector method indicates that any non-zero vector $\mathbf{x}$ which is M-biorthogonal to the subspace S_{Q_j}, that is $\mathbf{x}^*\mathbf{M}\mathbf{q}_j = 0$ for any $\mathbf{q}_j \in \mathbf{Q}_j$, can be chosen to be a new starting vector. In this thesis, the MPNSV method tries to find a new starting vector $\hat{\mathbf{s}}_j$ such that

$$\hat{\varpi}_j = \frac{|\hat{\mathbf{s}}_j^*\mathbf{M}\mathbf{q}_{j+1}|}{\|\hat{\mathbf{s}}_j\|} = \max_{\mathbf{x}\in S_{\hat{Q}_j}^{\perp}} \frac{|\mathbf{x}^*\mathbf{M}\mathbf{q}_{j+1}|}{\|\mathbf{x}\|},$$

where $\hat{\varpi}_j$ denotes the maximum pivot at step j. Then $\mathbf{p}_{j+1} = \hat{\mathbf{s}}_j/\hat{\mathbf{s}}_j^*\mathbf{M}\mathbf{q}_{j+1}$.

It is well known that if $\mathbf{q}_{j+1} \neq \mathbf{0}$, $\mathbf{q}_{j+1}$ is the sum of two unique vectors $\mathbf{g}_1$

and $\mathbf{g}_2$, where $\mathbf{g}_1 \in \S_{Q_j}^{\perp}$ and $\mathbf{g}_2 \in \S_{Q_j}$. It is shown in [66] that $\mathbf{g}_1$ is the vector whose pivot is the maximum. Therefore $\mathbf{g}_1$ can be chosen to be the new starting vector $\hat{\mathbf{s}}_j$. Thus $\hat{\mathbf{s}}_j$ can be expressed as a linear combination of $\mathbf{q}_{j+1}$ minus the base vectors in $\S_{Q_j}$:

$$\hat{\mathbf{s}}_j = \mathbf{g}_1 = \mathbf{q}_{j+1} - \sum_{k=1}^{j} \delta_k \mathbf{q}_k. \tag{3.20}$$

Since $\mathbf{q}_i^* \mathbf{M} \hat{\mathbf{s}}_j = 0$, for $i = 1, 2, \ldots, j$, the δ_k, $k = 1, 2, \ldots, j$ are the solution of equations

$$\sum_{k=1}^{j} \delta_k \mathbf{q}_i^* \mathbf{M} \mathbf{q}_k = \mathbf{q}_i^* \mathbf{M} \mathbf{q}_{j+1}, \quad i = 1, 2, \ldots, j,$$

which can be expressed in the form of matrices

$$\mathbf{Q}_j^* \mathbf{M} \mathbf{Q}_j \mathbf{d} = \mathbf{Q}_j^* \mathbf{M} \mathbf{q}_{j+1} = \mathbf{h}, \tag{3.21}$$

where the transposed vector $\mathbf{d}^* = (\delta_1, \delta_2, \ldots, \delta_j)$ and $\mathbf{h}$ is a column vector. Since $\mathbf{Q}_j^* \mathbf{M} \mathbf{Q}_j$ is a positive definite matrix there is a unique solution to (3.21) and pivoting is unnecessary in the Cholesky factorization of the matrix.

Extra computational effort results mainly from the Cholesky factorization of $\mathbf{Q}_j^* \mathbf{M} \mathbf{Q}_j = \mathbf{L}\mathbf{L}^*$, where $\mathbf{L}$ is the lower triangular matrix. Suppose that two successive breakdown or near breakdown conditions occur at step j_1 and j_2. Denoting $\hat{\jmath} = j_2 - j_1$ and $\mathbf{Q}_{j_2} = [\mathbf{Q}_{j_1}, \mathbf{Q}_{\hat{\jmath}}]$, then the Cholesky factorizations are $\mathbf{Q}_{j_1}^* \mathbf{M} \mathbf{Q}_{j_1} = \mathbf{L}_{j_1} \mathbf{L}_{j_1}^*$ and $\mathbf{Q}_{j_2}^* \mathbf{M} \mathbf{Q}_{j_2} = \mathbf{L}_{j_2} \mathbf{L}_{j_2}^*$. Hence,

$$\mathbf{Q}_{j_2}^* \mathbf{M} \mathbf{Q}_{j_2} = \begin{bmatrix} \mathbf{Q}_{j_1}^* \\ \mathbf{Q}_{\hat{\jmath}}^* \end{bmatrix} \mathbf{M} [\, \mathbf{Q}_{j_1} \quad \mathbf{Q}_{\hat{\jmath}} \,] = \begin{bmatrix} \mathbf{L}_{j_1} \mathbf{L}_{j_1}^* & \mathbf{Q}_{j_1}^* \mathbf{M} \mathbf{Q}_{\hat{\jmath}} \\ \mathbf{Q}_{\hat{\jmath}}^* \mathbf{M} \mathbf{Q}_{j_1} & \mathbf{Q}_{\hat{\jmath}}^* \mathbf{M} \mathbf{Q}_{\hat{\jmath}} \end{bmatrix} = \mathbf{L}_{j_2} \mathbf{L}_{j_2}^*.$$

$\mathbf{L}_{j_2}$ can be written by

$$\mathbf{L}_{j_2} = \begin{bmatrix} \mathbf{L}_{j_1} & 0 \\ B & \mathbf{L}_{\hat{\jmath}} \end{bmatrix},$$

where $\mathbf{B}$ is of size $\hat{\jmath} \times j_1$ and $\mathbf{L}_{\hat{\jmath}}$ is the lower triangular matrix with size $\hat{\jmath} \times \hat{\jmath}$. $\mathbf{B}$ and $\mathbf{L}_{\hat{\jmath}}$ may be obtained by Cholesky algorithm on the new blocks $\mathbf{Q}_{\hat{\jmath}}^* \mathbf{M} \mathbf{Q}_{j_1}$ and $\mathbf{Q}_{\hat{\jmath}}^* \mathbf{M} \mathbf{Q}_{\hat{\jmath}}$ only, since

$$\mathbf{L}_{j_2} \mathbf{L}_{j_2}^* = \begin{bmatrix} \mathbf{L}_{j_1} \mathbf{L}_{j_1}^* & \mathbf{L}_{j_1} B^* \\ \mathbf{B} \mathbf{L}_{j_1}^* & \mathbf{B}\mathbf{B}^* + \mathbf{L}_{\hat{\jmath}} L_{\hat{\jmath}}^* \end{bmatrix}.$$

The algorithm for the MPNSV method is presented below.

3.2.2 Algorithm

Algorithm 3.1 *Suppose that j_1 and j_2 $(j_1 < j_2)$ are the numbers of steps at which two successive breakdowns occur (if only one breakdown occurs, $j_1 = 0$). $\mathbf{q}_i$, $i = 1, 2, \ldots, j_2 + 1$ are right Lanczos vectors and $\mathbf{Q}_{j_1} \mathbf{M} \mathbf{Q}_{j_1} = \mathbf{L}_{j_1} \mathbf{L}_{j_1}^*$, where $\mathbf{Q}_{j_1} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{j_1}]$ and*

$$\mathbf{L}_{j_1} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{j_1 1} & l_{j_2 1} & \cdots & l_{j_1 j_1} \end{bmatrix}.$$

Do:

(1) $l_{ij} = \mathbf{q}_i \mathbf{M} \mathbf{q}_j, \quad i = j_1 + 1, j_1 + 2, \ldots, j_2; \;\; j = 1, 2, \ldots, j_2$

(2) $h_i = \mathbf{q}_i \mathbf{M} \mathbf{q}_{j_2+1}, \quad i = 1, 2, \ldots, j_2$

(3) For $i = j_1 + 1, \ldots, j_2$ *and* $j = 1, 2, \ldots, j_1$

$l_{i,j} = l_{i,j} - l_{i,k} * l_{j,k}, \quad k = 1, 2, \ldots, j-1$

(4) For $j = j_1 + 1, \ldots, j_2$ *(without pivoting)*

$l_{j,j} = l_{j,j} - l_{j,k} * l_{j,k}, \quad k = 1, 2, \ldots, j-1$

If $l_{j,j} \leq 0$

$\mathbf{Q}_{j_2}^* \mathbf{M} \mathbf{Q}_{j_2}$ *is not a positive definite matrix or is ill-condition.*

Then stop the process.

else

$l_{i,j} = l_{i,j}/l_{j,j}$

$l_{j,j} = \sqrt{l_{j,j}}$

end if

(5) For $j = j_1 + 1, \ldots, j_2$ *and* $i = j_1 + 1, j_1 + 2, \ldots, j_2$

$l_{i,j} = l_{i,j} - l_{i,k} * l_{j,k}, \quad k = 1, 2, \ldots, j-1$

(6) Form the lower triangular matrix $\mathbf{L}_{j_2}$

(7) Solve for $\mathbf{d}$ *from the equation* $\mathbf{L}_{j_2} \mathbf{L}_{j_2}^* \mathbf{d} = \mathbf{h}$ *by forward and forward substitution, where* $\mathbf{h}^* = \{h_1, h_2, \ldots, h_{j_2}\}$ *(see (3.21))*

(8) Set $j_1 = j_2$ *for next breakdown use.*

(9) Return to the main algorithm (Algorithm 3.4).

3.2.3 Summary and remarks

(1.) Computation and storage.

Suppose that there are several breakdowns or near breakdowns occurring during the process and the last breakdown occurs at step j_m ($j_m < m \ll n$). The total cost of computation of the MPNSV method will mainly come from the Cholesky factorization of one matrix $\mathbf{Q}_{j_m}^* \mathbf{M} \mathbf{Q}_{j_m}$ and forward and backward sub-

stitutions. Each previous Cholesky factorization is only a corresponding part of the whole factorization of $\mathbf{Q}_{j_m}^* \mathbf{M} \mathbf{Q}_{j_m}$. Noting that the fact $m \ll n$ and that $\mathbf{Mq}_1, \mathbf{Mq}_2, \ldots, \mathbf{Mq}_{j_m}$ have already been calculated and stored during the computations of $\mathbf{r}_j$ and $\mathbf{s}_j$, $j = 1, 2, \ldots, j_m$, the computation of the δs is inexpensive in comparison with the method which picks up the new starting vector randomly [66]. From (3.12), that method also needs to compute $\mathbf{q}_i^* \mathbf{Mx}$ and store $\mathbf{q}_i^* \mathbf{M}$ for $1 \leq i \leq j_m$.

(2.) Condition number

An 'ill' condition number will cause the Cholesky factorization to fail. Wilkinson pointed this out in his paper ([64]). Golub and Van Loan also gave an example, where the dimension of the positive definite matrix is only 3×3. Nevertheless the process breaks down, if it is rounded off to 2 decimal places (see [21]). This is a disadvantage of the MPNSV method comparison with the method which choses a new starting vector ramdonly. However, the conditioning of the matrices of system (1.2) is always an important issue, because the accuracy of LU factorization of $\mathbf{K}$ depends on its condition number. Detailed discussion can be seen in [21]. The condition numbers of the "capacity" and "conductivity" matrices are important to consider, no matter what method is used to chose a new starting vector when applying the ULR method.

(3.) Pathological breakdown

It is possible that the new pivot computed from Algorithm 3.1 is still less than a specified tolerance. In addition, if the new pivot is very small, (3.20) shows that the right Lanczos vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{j+1}$ are almost linearly dependent. Thus the M-biorthogonality, $\mathbf{P}_{j+1}^* \mathbf{M} \mathbf{Q}_{j+1} = \mathbf{I}_{j+1}$ is lost. Because the new pivot produced

by the MPNSV method is the maximum, any further trial is unnecessary. Such situation is referred as "pathological breakdown". If this situation occurs, the ULR process can be switch to the AR algorithm. This process will be discussed in the next section. In short, the advantage of the MPNSV method is that one can quickly check the situation without additional computational effort.

3.3 The Switch method

3.3.1 Switching to the AR method

The Switch method switches the ULR process to the AR algorithm so that $\mathbf{p}_j$ is always equal to $\mathbf{q}_j$ in each step after switching. Consequently, the corresponding pivot is always equal to one. However, this will partially lose the M-biorthogonality. For example, if this situation occurs at step $\bar{m}$, it is considered that the pathological breakdown occurs or $\mathbf{q}_{\bar{m}+1}$ is an almost linear combination of $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{\bar{m}}$. Thus, $\mathbf{q}_{\bar{m}+1}$ should be reconstructed again. To this end, let $\mathbf{p}_{\bar{m}} = \mathbf{q}_{\bar{m}}$. Then $\mathbf{q}_{\bar{m}+1}$ is reconstructed using formula (3.14) where j is replaced by $\bar{m}$. $\gamma_{\bar{m}}^{(t)}$, $t = 1, 2, \ldots, k_r^{\bar{m}}$ are still formed from (3.16), but

$$\alpha_{\bar{m}} = \mathbf{p}_{\bar{m}}^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_{\bar{m}} - \sum_{t=1}^{k_r^{\bar{m}}} \gamma_{\bar{m}}^{(t)} \mathbf{p}_{\bar{m}}^* \mathbf{M}\mathbf{q}_{\bar{m}-t}.$$

Therefore, set $\mathbf{p}_{\bar{m}+1} = \mathbf{q}_{\bar{m}+1}$. Thus, it is proved in Theorem 3.1 that

$$\mathbf{p}_{\bar{m}}^* \mathbf{M}\mathbf{q}_j \neq 0, \qquad \mathbf{p}_{\bar{m}+1}^* \mathbf{M}\mathbf{q}_j \neq 0, \qquad 1 \leq j \leq \bar{m} - 1,$$

but

$$\mathbf{p}_i^* \mathbf{M} \mathbf{q}_{\bar{m}} = 0, \qquad \mathbf{p}_i^* \mathbf{M} \mathbf{q}_{\bar{m}+1} = 0, \qquad 1 \le i \le \bar{m} - 1,$$

and

$$\mathbf{p}_{\bar{m}}^* \mathbf{M} \mathbf{q}_{\bar{m}+1} = 0, \qquad \mathbf{p}_{\bar{m}+1}^* \mathbf{M} \mathbf{q}_{\bar{m}} = 0.$$

After that $\mathbf{p}_i$ and $\mathbf{q}_j$, $i, j > \bar{m} + 1$, continue to be formed according to this idea. The detailed algorithm follows. For the sake of simplicity, $k_r^{\bar{m}}$ is replaced by k_r in the algorithm.

Algorithm 3.2 : *Set* $\mathbf{p}_{\bar{m}} = \mathbf{q}_{\bar{m}}$. *Then, do for* $j = \bar{m}, \bar{m}+1, \ldots, m$.

(1) $\gamma_j^{(j-i)} = \mathbf{p}_i^* \mathbf{M} \mathbf{K}^{-1} \mathbf{M} \mathbf{q}_j, \qquad i = \bar{m} - k_r, \bar{m} - k_r + 1, , \ldots, \bar{m} - 1$

(2) $\gamma_j^{(j-i)} = \mathbf{p}_i^* \mathbf{M} \mathbf{K}^{-1} \mathbf{M} \mathbf{q}_j - \sum_{t=j-\bar{m}+1}^{j-(\bar{m}-k_r)} \gamma_j^{(t)} \mathbf{p}_i^* \mathbf{M} \mathbf{q}_{j-t},$

$i = \bar{m}, \bar{m}+2, \cdots, j-1$

(3) $\alpha_j = \mathbf{p}_j^* \mathbf{M} \mathbf{K}^{-1} \mathbf{M} \mathbf{q}_j - \sum_{t=j-\bar{m}+1}^{j-(\bar{m}-k_r)} \gamma_j^{(t)} \mathbf{p}_j^* \mathbf{M} \mathbf{q}_{j-t}$

(4) If $j = m - \bar{m}$ *go to (11)*

(5) $\mathbf{r}_j = \beta_{j+1} \mathbf{q}_{j+1} = \mathbf{K}^{-1} \mathbf{M} \mathbf{q}_j - \alpha_j \mathbf{q}_j - \sum_{t=1}^{j-(\bar{m}-k_r)} \gamma_j^{(t)} \mathbf{q}_{j-t}$

(6) $\beta_{j+1} = \|\mathbf{r}_j\|$

(7) Check the relative residual bound (see §3.4, (3.39)).

If it is less than tolerance, go to (11)

(8) $\mathbf{q}_{j+1} = \dfrac{\mathbf{r}_j}{\beta_{j+1}}$

(9) $\mathbf{p}_{j+1} = \mathbf{q}_{j+1}$

(10) Go to (1)

(11) Form the reduced system (3.11) and solve it

Note the following points concerning this algorithm. Denoting

$$[\mathbf{p}_1\ \mathbf{p}_2\ \cdots\ \mathbf{p}_{\bar{m}-1}] = \mathbf{P}_1,\ [\mathbf{p}_{\bar{m}}\ \mathbf{p}_{\bar{m}+1}\ \cdots\ \mathbf{p}_m] = \mathbf{P}_2,\ [\mathbf{P}_1\ \mathbf{P}_2] = \mathbf{P}_m,$$

$$[\mathbf{q}_1\ \mathbf{q}_2\ \cdots\ \mathbf{q}_{\bar{m}-1}] = \mathbf{Q}_1,\ [\mathbf{q}_{\bar{m}}\ \mathbf{q}_{\bar{m}+1}\ \cdots\ \mathbf{q}_m] = \mathbf{Q}_2,\ [\mathbf{Q}_1\ \mathbf{Q}_2] = \mathbf{Q}_m,$$

in the next subsection, it is proved in Theorem 3.1 that

$$\mathbf{P}_1^*\mathbf{M}\mathbf{Q}_1 = \mathbf{I}_1, \qquad \mathbf{P}_2^*\mathbf{M}\mathbf{Q}_2 = \mathbf{I}_2 \qquad \text{and} \qquad \mathbf{P}_1^*\mathbf{M}\mathbf{Q}_2 = \mathbf{0},$$

where $\mathbf{I}_1$ and $\mathbf{I}_2$ are $(\bar{m}-1)\times(\bar{m}-1)$ and $(m-\bar{m}+1)\times(m-\bar{m}+1)$ identity matrices and $\mathbf{0}$ is $(\bar{m}-1)\times(m-\bar{m}+1)$ $\mathbf{0}$ matrix. However, $\mathbf{P}_2^*\mathbf{M}\mathbf{Q}_1$ is not $\mathbf{0}$. Furthermore, it is also proved that

$$\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m = \Gamma\mathbf{T}_m,$$

where Γ satisfies

$$\Gamma = \mathbf{P}_m^*\mathbf{M}\mathbf{Q}_m = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{\Pi} & \mathbf{I}_2 \end{bmatrix},$$

where

$$\mathbf{\Pi} = \mathbf{P}_2^*\mathbf{M}\mathbf{Q}_1,$$

and $\mathbf{T}_m$ has the form of

$$\mathbf{T}_m =$$

$$\left[\begin{array}{ccc|ccccc}
\ddots & & & & & & & \\
\ddots & \gamma_{\bar{m}-1}^{(k_r^{\bar{m}-1})} & & & & & & \\
\ddots & \gamma_{\bar{m}-1}^{(k_r^{\bar{m}-1}-1)} & & \gamma_{\bar{m}}^{(k_r)} & \gamma_{\bar{m}+1}^{(k_r+1)} & \cdots & \cdots & \gamma_m^{(k_r+m-\bar{m})} \\
\ddots & \gamma_{\bar{m}-1}^{(k_r^{\bar{m}-1}-2)} & & \gamma_{\bar{m}}^{(k_r-1)} & \gamma_{\bar{m}+1}^{(k_r)} & \cdots & \cdots & \gamma_m^{(k_r+m-\bar{m}-1)} \\
\ddots & \gamma_{\bar{m}-1}^{(k_r^{\bar{m}-1}-3)} & & \gamma_{\bar{m}}^{(k_r-2)} & \gamma_{\bar{m}+1}^{(k_r-1)} & \cdots & \cdots & \gamma_m^{(k_r+m-\bar{m}-2)} \\
 & \vdots & & \vdots & \vdots & & & \vdots \\
\ddots & \alpha_{\bar{m}-1} & & \gamma_{\bar{m}}^{(1)} & \gamma_{\bar{m}+1}^{(2)} & \ddots & \ddots & \cdot \\
-- & -- & & -- & -- & -- & -- & -- \\
 & \beta_{\bar{m}} & & \alpha_{\bar{m}} & \gamma_{\bar{m}+1}^{(1)} & \ddots & \ddots & \cdot \\
 & & & \beta_{\bar{m}+1} & \alpha_{\bar{m}+1} & \ddots & \ddots & \cdot \\
 & & & & \beta_{\bar{m}+2} & \ddots & \ddots & \cdot \\
 & & & & & \ddots & \ddots & \gamma_m^{(1)} \\
 & & & & & & \cdot & \alpha_m
\end{array}\right] . \tag{3.22}$$

where α_i for all $i \leq m$ are diagonal entries, all γs in the upper-right corner are computed from (1) in Algorithm 3.2, while these in the lower-right corner are computed from (2) of the algorithm.

The proof of the Switch method is presented in the next subsection.

3.3.2 Proof of the Switch method

If no pathological breakdown occurs, [66] gave the theorem that the New-Start

Vector method ensures equations (3.8) and (3.9) are valid, i.e.,

$$\mathbf{P}_m^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m = \mathbf{T}_m, \quad \mathbf{P}_m^* \mathbf{M}\mathbf{Q}_m = \mathbf{I}_m, \quad \mathbf{P}^* \mathbf{M}\mathbf{r}_m = \mathbf{0} \tag{3.23}$$

and

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m = \mathbf{Q}_m \mathbf{T}_m + \mathbf{r}_m \mathbf{e}_m^*. \tag{3.24}$$

This subsection shows the corresponding results for the Switch method. In order to accomplish the proof, the following lemmas are introduced. Similar results of Lemma 3.2 and 3.3 can be found in [66, pp. 187]. Here provides another approach.

Lemma 3.1 *The two-sided Gram-Schmidt method with New-Start Vector method produces M-biorthogonal sequences* $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{q}_m\}$, $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m\}$ $(m \ll n)$ *and n-vector* $\mathbf{r}_m$. $\mathbf{P}_m$ *and* $\mathbf{Q}_m$ *are* $n \times m$ *rectangle matrices defined by* $\mathbf{Q}_m = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m]$ *and* $\mathbf{P}_m = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{q}_m]$. *If no pathological breakdown occurs, then equations (3.23) and (3.24) hold, where* $\mathbf{T}_m$ *is an* $m \times m$ *special band matrix with lower band width of one and irregular upper band, and its* i, j *entry satisfy*

$$\mathbf{T}_{ij} = \mathbf{p}_i^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j = \begin{cases} \alpha_j, & i = j, \\ \gamma_j^{(j-i)}, & 0 < j - i \le k_r^j, \\ \beta_i, & j = i - 1, \\ 0, & otherwise. \end{cases} \tag{3.25}$$

Lemma 3.2 *Under the conditions of Lemma 3.1, let* j *be integer with* $j \le m$, *and let* k_s^j *be the number of terms involving* γ *in the formula for* $\mathbf{s}_j^*$ *in equation (3.15). Then* $j - k_s^j$, $j = 1, 2, \cdots, m-1$ *is a non-decreasing integer sequence starting at* $1 - k_s^1 = 1$ *and ending at* $(m-1) - k_s^{m-1}$.

PROOF: According to equations (3.17), (3.18) and (3.19), k_s^j is determined twice, in step $j-1$ and step j. There are two situations about process going from step $j-1$ to step j. (a) no breakdown occurs at step $j-1$, then $k_s^j = k_s^{j-1}$, and (b) breakdown of case (1) or case (2) with $k_s^{j-1} = 0$ occurs at step $j-1$, then $k_s^j = k_s^{j-1} + 1$. If breakdown of case (2) with $k_s^j \neq 0$ occurs at step $j-1$, $k_s^j = k_s^j - 1$. This situation could be repeated μ times until one of cases of (a) and (b) occurs and then goes to the next step. Thus, $k_s^j = k_s^j - \mu \geq 0$. The increment of $j - k_s^j$ is given in Table 3.1 below.

Table 3.1: Increment of $j - k_s^j$

Step $j-1$	Step j	Increment
no breakdown	no breakdown or breakdown of case (1) or of case (2) with $k_s^j = 0$	1
	breakdown of case (2) with $k_s^j \neq 0$ repeated μ times	$\mu + 1$
breakdown of case (1) breakdown of case (2) with $k_s^{j-1} = 0$	no breakdown or breakdown of case (1) or of case (2) with $k_s^j = 0$	0
	breakdown of case (2) with $k_s^j \neq 0$ repeated μ times	μ

□

Lemma 3.3 *Under the conditions of Lemma 3.2, for each positive integer* $i = 1, 2, \ldots, (m-1) - k_s^{m-1}$, *there exists a positive integer* $j \leq m-1$ *such that*

$$\mathbf{p}_i^* \mathbf{M} \mathbf{K}^{-1} = \gamma_{j+1}^{(k_s^j+1)} \mathbf{p}_{j+1}^* + \alpha_i \mathbf{p}_i + \beta_i \mathbf{p}_{i-1}^* + \sum_{t=1}^{j-i} \gamma_{i+t}^{(t)} \mathbf{p}_{i+t}^*; \tag{3.26}$$

or

$$\mathbf{p}_i^* \mathbf{M} \mathbf{K}^{-1} = \alpha_i \mathbf{p}_i^* + \beta_i \mathbf{p}_{i-1}^* + \sum_{t=1}^{j-i} \gamma_{i+t}^{(t)} \mathbf{p}_{i+t}^*. \tag{3.27}$$

PROOF: Lemma 3.2 indicates that there are two cases: (I) There is a unique integer j or there are several successive integers js ($j \leq m$) such that $i = j - k_s^j$. (II) There is a unique integer $j < m$ and a some positive integer ν such that $(j-1) - k_s^{j-1} < i$ and $i + \nu = j - k_s^j$.

For case (I), if there are several successive positive integers js satisfying $i = j - k_s^j$, the largest one is chosen such that $i = j - k_s^j < (j+1) - k_s^{j+1}$. From Table 3.1, there are three subcases matching this situation.

(a). No breakdown occurs at step j and any situation occurs at next step.

(b). Breakdown of case (1) or case (2) with $k_s^j = 0$ occurs at step j and followed by case (2) with $k_s^{j+1} \neq 0$.

(c). Breakdown of case (2) with $k_s^j \neq 0$ repeats several times until one of above occurs.

If (a) occurs, from (3.15), it is obvious that (3.26) holds with $i = j - k_s^j$. If (b) occurs, $\mathbf{p}_{j+1}$ is not produced from (3.15), but from New-Start Vector method, and (3.26) does not hold any more. According to equation (3.18), in the next step, $\bar{k}_s^{j+1} = k_s^j + 1$, here $\bar{k}_s^{j+1}$ is temporarily used. Then s_{j+1} is calculated using (3.15). That breakdown of case (2) with $k_s^{j+1} \neq 0$ occurs in step $j+1$ means $\mathbf{s}_{j+1} = \mathbf{0}$. Therefore, in this temporary stage, $(j+1) - \bar{k}_s^{j+1} = j - k_s^j = i$. Substituting $\mathbf{s}_{j+1} = \mathbf{0}$ into (3.15), replacing j by $j+1$, k_s^j by $\bar{k}_s^{j+1}$, and then replacing $(j+1) - \bar{k}_s^{j+1}$ by i, Finally, replacing $j+1$ by j, therefore the upper

limit $\bar{k}_s^j = j - i$, (3.27) follows. For (c), the situation eventually is one of the two subcases, (a) or (b).

Case (II) implies that the repeated breakdown of case (2) with $k_s^j \neq 0$ occurs in step j, i.e., $\mathbf{s}_j$ is temporarily calculated to be zero from equation (3.15). The algorithm sets $k_s^j = \bar{k}_s^j - 1$ immediately after when $\mathbf{s}_j = \mathbf{0}$. If the situation occurs again, k_s^j becomes $\bar{k}_s^j$. Set $k_s^j = \bar{k}_s^j - 1$ again. Thus $\mathbf{s}_j = \mathbf{0}$ at any temporary stage until such k_s^j that $\mathbf{s}_j \neq \mathbf{0}$. If $(j-1) - k_s^{j-1} < i < j - k_s^j$, there must exist a temporary stage at which $j - \bar{k}_s^j = i$ and $\mathbf{s}_j = \mathbf{0}$ which is formed by replacing k_s^j by $\bar{k}_s^j$ in equation (3.15). Substituting $\mathbf{s}_j = \mathbf{0}$ into equation (3.15), replacing $j - k_s^j$ by i and k_s^j by $\bar{k}_s^j$, finally replacing $\bar{k}_s^j$ in the upper limit of the summation by $j - i$, (3.27) follows. This completes the proof. □

Theorem 3.1 *During the two-sided Gram-Schmidt biorthogonalization with New-Start Vector process, assume that a pathological breakdown occurs at step $\bar{m}$. Let $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{\bar{m}-1}$ and $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{\bar{m}-1}$ be produced from (3.14) and (3.15), and $\mathbf{p}_{\bar{m}}, \mathbf{p}_{\bar{m}+1}, \ldots, \mathbf{p}_m, \mathbf{q}_{\bar{m}}, \mathbf{q}_{\bar{m}+1}, \ldots, \mathbf{q}_m$ and $\mathbf{r}_m$ be produced from algorithm 3.2. Denote*

$$[\mathbf{p}_1 \ \mathbf{p}_2 \ \cdots \ \mathbf{p}_{\bar{m}-1}] = \mathbf{P}_1, \quad [\mathbf{p}_{\bar{m}} \ \mathbf{p}_{\bar{m}+1} \ \cdots \ \mathbf{p}_m] = \mathbf{P}_2, \quad [\mathbf{P}_1 \ \mathbf{P}_2] = \mathbf{P}_m,$$

$$[\mathbf{q}_1 \ \mathbf{q}_2 \ \cdots \ \mathbf{q}_{\bar{m}-1}] = \mathbf{Q}_1, \quad [\mathbf{q}_{\bar{m}} \ \mathbf{q}_{\bar{m}+1} \ \cdots \ \mathbf{q}_m] = \mathbf{Q}_2, \quad [\mathbf{Q}_1 \ \mathbf{Q}_2] = \mathbf{Q}_m,$$

where $\mathbf{P}_m$ and $\mathbf{Q}_m$ are $n \times m$ matrices. Then

$$\mathbf{\Gamma} = \mathbf{P}_m^* \mathbf{M} \mathbf{Q}_m = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{\Pi} & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{P}_m^* \mathbf{M} \mathbf{r}_m = \mathbf{0}, \tag{3.28}$$

where, $\mathbf{\Pi} = \mathbf{P}_2^*\mathbf{M}\mathbf{Q}_1$, *and* $\mathbf{P}_1^*\mathbf{M}\mathbf{Q}_1 = \mathbf{I}_1$, $\mathbf{P}_2^*\mathbf{M}\mathbf{Q}_2 = \mathbf{I}_2$ *are identity matrices. Furthermore*

$$\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m = \mathbf{\Gamma}\mathbf{T}_m, \tag{3.29}$$

where $\mathbf{T}_m$ *is an* $m \times m$ *matrix whose lower-right corner is a Hessenberg submatrix (3.22), i. e.,*

$$\mathbf{T}_m(i,j) = \begin{cases} \alpha_i, & i = j, \\ \beta_i, & i = j+1, \\ \gamma_j^{(j-i)}, & 0 < j - i \le k_r^j, \qquad\qquad j < \bar{m}, \\ \gamma_j^{(j-i)}, & 0 < j - i \quad and \quad i \ge \bar{m} - k_r^{\bar{m}}, \quad j \ge \bar{m}, \\ 0, & otherwise. \end{cases}$$

Here k_r^j *is the number of the terms involving* γ *for forming* $\mathbf{r}_j$ *in (3.14). When* $j < \bar{m}$, α_i, β_i *and* $\gamma_j^{(j-i)}$ *are calculated from New-Start Vector process (3.25); when* $j \ge \bar{m}$, *all of them are calculated from Algorithm 3.2.*

PROOF: Before step $\bar{m}$, $\mathbf{p}_1$, $\mathbf{p}_2$, ..., $\mathbf{p}_{\bar{m}-1}$ and $\mathbf{q}_1$, $\mathbf{q}_2$, ..., $\mathbf{q}_{\bar{m}-1}$ are produced by the two-sided Gram-Schmidt biorthogonalization with the New-Start Vector method, Lemma 3.1 ensures that $\mathbf{P}_1^*\mathbf{M}\mathbf{Q}_1 = \mathbf{I}_1$.

For $\mathbf{P}_1^*\mathbf{M}\mathbf{Q}_2 = \mathbf{0}$ and $\mathbf{P}_2^*\mathbf{M}\mathbf{Q}_2 = \mathbf{I}_2$, it is easy to see $\mathbf{p}_i\mathbf{M}\mathbf{q}_i = 1$ for all $i \ge \bar{m}$, from (6), (8) and (9) in Algorithm 3.2. The equation

$$\mathbf{p}_i^*\mathbf{M}\mathbf{q}_j = 0, \tag{3.30}$$

for any $i = 1, 2, \cdots, m$ and $j = \bar{m}, \bar{m}+1, \cdots, m+1$ when $i \ne j$ can be proved by mathematical induction.

Initially, it can be proved that (3.30) is valid for (I) $j = \bar{m}$, $i < j$, (II) $j = \bar{m}+1$, $i < j$ and (III) $j = \bar{m}$, $i = \bar{m}+1$.

For case (I), $\mathbf{q}_{\bar{m}}$ is produced from (3.14). Lemma 3.1 ensures that (3.30) is valid for $i < j$ because of equation (3.23), i.e., $\mathbf{P}_1^*\mathbf{M}\mathbf{r}_{\bar{m}-1} = \mathbf{0}$, and $\mathbf{q}_{\bar{m}} = \mathbf{r}_{\bar{m}-1}/\beta_{\bar{m}}$. For case (II), $j = \bar{m}+1$, $\mathbf{r}_{\bar{m}}$ is not produced from (3.14) but from algorithm 3.2. Therefore, the approach to proving (3.30) is different. In terms of (5) of Algorithm 3.2, it is obtained by replacing j by $\bar{m}$ such that

$$\begin{aligned}
&\mathbf{p}_i^*\mathbf{M}\mathbf{q}_{\bar{m}+1} \\
&\quad = \frac{1}{\beta_{\bar{m}+1}}\mathbf{p}_i^*\mathbf{M}\left(\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_{\bar{m}} - \alpha_{\bar{m}}\mathbf{q}_{\bar{m}} - \sum_{t=1}^{k_r^{\bar{m}}}\gamma_{\bar{m}}^{(t)}\mathbf{q}_{\bar{m}-t}\right). \\
&\quad = 0
\end{aligned}$$

It can be proved that the above equation is equal to zero, noting that the M-biorthogonality when $1 \le i, j \le \bar{m}-1$,

$$\mathbf{p}_i^*\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_{\bar{m}} = \begin{cases} \gamma_{\bar{m}}^{(\bar{m}-i)}, & \bar{m} - k_r^{\bar{m}} \le i, \\ 0, & 1 \le i \le \bar{m} - k_r^{\bar{m}}, \end{cases}$$

from equation (3.25) and equation (3) in Algorithm 3.2. For case (III), since $\mathbf{p}_{\bar{m}} = \mathbf{q}_{\bar{m}}$ and $\mathbf{p}_{\bar{m}+1} = \mathbf{q}_{\bar{m}+1}$,

$$\mathbf{p}_{\bar{m}+1}^*\mathbf{M}\mathbf{q}_{\bar{m}} = \mathbf{p}_{\bar{m}}^*\mathbf{M}\mathbf{q}_{\bar{m}+1} = 0,$$

due to the result of case (II).

According to the principle of mathematical induction, under the assumption

that there is a positive integer $\ell > \bar{m}$ such that (3.30) is true for $1 \le i \le \ell$, $\bar{m}+1 \le j \le \ell$, and $i \ne j$, it should be proved that equation (3.30) is also true for two cases (a) $1 \le i \le \ell$, $j = \ell+1$, (b) $i = \ell+1$ and $\bar{m}+1 \le j \le \ell$.

In the case (a), the integer set $\{1, 2, \ldots, \ell\}$ is divided into three ranges: $\{1, 2, \ldots, \bar{m} - k_r^{\bar{m}} - 1\}$, $\{\bar{m} - k_r^{\bar{m}}, \bar{m} - k_r^{\bar{m}} + 1, \ldots, \bar{m} - 1\}$ and $\{\bar{m}, \bar{m}+1, \ldots, \ell\}$. For the first range, noting that (3.17) and (3.18), i.e., $k_r^{\bar{m}} = k_r^{\bar{m}-1} + 1$, and taking account of Lemma 3.3, there is an integer $\tau \le \bar{m} - 2$ such that

$$\begin{aligned} &\mathbf{p}_i^* \mathbf{M} \mathbf{K}^{-1} \mathbf{M} \mathbf{q}_\ell \\ &= \left(\bar{\gamma} \mathbf{p}_{\tau+1}^* + \alpha_i \mathbf{p}_i^* + \beta_i \mathbf{p}_{i-1}^* + \sum_{t=1}^{\tau-i} \gamma_{i+t}^{(t)} \mathbf{p}_{i+t}^* \right) \mathbf{M} \mathbf{q}_\ell = 0, \end{aligned} \tag{3.31}$$

where $\bar{\gamma}$ equals to $\gamma_{\tau+1}^{k_s^\tau+1}$ or 0 in equations (3.26) and (3.27) respectively. Noting that i is in the first range, the subscript $i+t$ in (3.31) satisfies $i+t \le \tau \le \bar{m}-2$ and the subindex of the first term $\mathbf{p}_{\tau+1}^*$ satisfies $\tau+1 \le \bar{m}-1$. Thus the last equality holds due to the assumption. This result is used to verify the following equation:

$$\begin{aligned} &\mathbf{p}_i^* \mathbf{M} \mathbf{q}_{\ell+1} \\ &= \frac{1}{\beta_{\ell+1}} \mathbf{p}_i^* \mathbf{M} \left(\mathbf{K}^{-1} \mathbf{M} \mathbf{q}_\ell - \alpha_\ell \mathbf{q}_\ell - \sum_{t=1}^{\ell-(\bar{m}-k_r^{\bar{m}})} \gamma_\ell^{(t)} \mathbf{q}_{\ell-t} \right) \\ &= 0. \end{aligned} \tag{3.32}$$

All terms of $\mathbf{p}_i^* \mathbf{M} \mathbf{q}_j$ in the above equation are equal to 0, where i is in the first range and $j = \bar{m} - k_r^{\bar{m}}, \bar{m} + 1 - k_r^{\bar{m}}, \ldots, \ell$, according to Lemma 3.1 when $j = \bar{m} - k_r^{\bar{m}}, \bar{m} - k_r^{\bar{m}} + 1, \ldots, \bar{m}$ and the assumption when $\bar{m} < j \le \ell$.

For i in the second range, $\bar{m} - k_r^{\bar{m}} \leq i \leq \bar{m} - 1$, equation (3.32) also holds, according to the assumption, $\mathbf{p}_i^* \mathbf{M} \mathbf{q}_j = 0$ when $\bar{m} \leq j \leq \ell$, from Lemma 3.1, the M-biorthogonality when $\bar{m} - k_r^{\bar{m}} \leq i, j \leq \bar{m} - 1$, and from (1) in algorithm 3.2, $\mathbf{p}_i^* \mathbf{M} \mathbf{K}^{-1} \mathbf{M} \mathbf{q}_\ell = \gamma_\ell^{(\ell - i)}$.

For i in the third range, $\bar{m} \leq i \leq \ell$, equation (3.32) holds due to the assumption and formulae (2) and (3) of Algorithm 3.2. For case (b), a similar approach to the initial condition (III) can be taken. Thus, (3.30) is proved. This completes the proof of the first equation of (3.28).

Noting that when $j = m + 1$, (3.30) implies the second equation of (3.28), $\mathbf{P}_m^* \mathbf{M} \mathbf{r}_m = \mathbf{0}$, hence complete equation (3.28).

Finally, (3.29) is proved by using (3.28). Rearranging formulae (3.14) and (5) of Algorithm 3.2, assembling them for $j = 1, 2, \ldots, m$, it follows that the relation

$$\mathbf{K}^{-1} \mathbf{M} \mathbf{Q}_m = \mathbf{Q}_m \mathbf{T}_m + [\mathbf{0}, \ldots, \mathbf{0}, \mathbf{r}_m] = \mathbf{Q}_m \mathbf{T}_m + \mathbf{r}_m \mathbf{e}_m^*, \tag{3.33}$$

where $\mathbf{0}$ is the $\mathbf{0}$ vector and $\mathbf{e}_m$ is the identity vector with entries zero except the m'th one. Pre-multiply the above equation by $\mathbf{P}_m^* \mathbf{M}$ and use (3.28) to establish (3.29). Thus the theorem is completed. □

3.3.3 Summary and remarks

The pathological breakdown or almost linearly dependent of right Lanczos vectors is due to the rounding error of computer machine and is closely related to the condition number of the matrix $\mathbf{M} \mathbf{K}^{-1} \mathbf{M}$, which can be seen in [60, 8]. In practice, this situation is very rare if the pivot tolerance is set very small. The intention of developing the Switch method is to develop a robust algorithm that

can handle all breakdowns.

It is easy to see that the Switch method can also be used to handle any kinds of breakdown. The ULR process can be switched to the AR process immediately after breakdown occurs.

From (3.22), it is easy to see that when $\bar{m} = 1$, (3.22) is a Hessenberg matrix. This fact can also been seen from Algorithm 3.2. When $\bar{m} = 1$, from §3.1.1, $k_r = 0$. Thus, in Algorithm 3.2, step (1) and the summation in steps (2), (3) do not need to be executed, because the upper limit is less than lower limit there. Hence, Algorithm 3.2 reduces to Algorithm 2.2. It is also obvious that when $\bar{m} > m$, or the pathological breakdown never occurs, Theorem 3.1 is just Lemma 3.1.

3.4 Rayleigh-Ritz process and termination criterion

If no breakdown occurs, the reduced system (3.11) has been derived in §3.1.2, i.e.,

$$\mathbf{T}_m \dot{w} + \mathbf{w} = \beta_1 \mathbf{e}_1 \mu(t). \tag{3.34}$$

If no pathological breakdown occurs, Lemma 3.1 ensures a similar procedure to derive equation (3.34). If the pathological breakdown occurs, it can be proved that the same form of system (3.34) can be obtained. Substituting the approximate transformation

$$\mathbf{c} = \mathbf{Q}_m \mathbf{w} \tag{3.35}$$

into equation (3.1), and noting that, $\mathbf{K}^{-1}\mathbf{b} = \beta_1\mathbf{q}_1 = \beta_1\mathbf{Q}_m\mathbf{e}_1$,

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m\dot{\mathbf{w}} + \mathbf{Q}_m\mathbf{w} = \mathbf{Q}_m\beta_1\mathbf{e}_1\mu(t). \tag{3.36}$$

results. Pre-multiplying (3.36) by $\mathbf{\Gamma}^{-1}\mathbf{P}_m^*\mathbf{M}$, where $\mathbf{\Gamma}$ is the non-singular matrix given in (3.28), and using (3.29), (3.34) follows. The approximate solution of (1.2), $\mathbf{c} \approx \mathbf{Q}_m\mathbf{w}$ is obtained by solving (3.34) for $\mathbf{w}$.

In order to terminate the process, a residual error function is defined by substituting the transformation (3.35) into the original equation (3.1), i. e.,

$$\theta_m(t) = \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m\dot{\mathbf{w}} + \mathbf{Q}_m\mathbf{w} - \mathbf{K}^{-1}\mathbf{b}\mu(t), \tag{3.37}$$

where $\mathbf{w}$ is the solution to the reduced system (3.34). The following theorem is an immediate consequence of result in reference [43, p. 225].

Theorem 3.2 *In the ULR algorithm, after m steps, the residual error, defined above, is bounded by*

$$\|\theta_m(t)\| \leq |\dot{\mathbf{w}}_m|\|\mathbf{r}_m\|, \tag{3.38}$$

where $\dot{\mathbf{w}}_m$ is the derivative of the last component of $\mathbf{w}$, and $\mathbf{r}_m$ is produced by (3.14) or (5) in Algorithm 3.2 with $j = m$ and $\|\cdot\|$ is any kind of norm.

It is easy to see that when $\mathbf{r}_m = \mathbf{0}$, the corresponding approximation is the exact solution regardless of round-off error. Furthermore, if the computations are exact, the exact solution can be obtained in n steps, because $\mathbf{r}_n = \mathbf{0}$ in this case.

The error of (3.38) is a function of time t and depends on the solution of the reduced system (3.34). If the system (3.34) is stable, the error decreases as time

t increases . Therefore, $||\theta_m(0)||$ can be used to terminate the process.

From (3.34)

$$\dot{\mathbf{w}}(\mathbf{0}) = \mathbf{T}_m^{-1}[\beta_1 \mathbf{e}_1 \mu(0) - \mathbf{w}(\mathbf{0})].$$

Usually, the initial condition $\mathbf{c}(\mathbf{0})$ of equation (1.2), and consequently $\mathbf{w}(\mathbf{0})$ is zero. Thus

$$\dot{\mathbf{w}}(\mathbf{0}) = \mathbf{T}_m^{-1}\beta_1 \mathbf{e}_1 \mu(0),$$

and for all t, the following is satisfied,

$$||\theta_m(t)|| \leq ||\dot{\mathbf{w}}_m(\mathbf{0})\mathbf{r}_m|| = |\beta_1||(\mathbf{T}_m^{-1}\mathbf{e}_1)_m||\mu(0)|||\mathbf{r}_m||.$$

Defining the relative residual error bound at each step j as

$$\delta_j = \frac{||\theta_j(\mathbf{0})||}{||\mathbf{K}^{-1}\mathbf{b}\mu(0)||}, \tag{3.39}$$

it can be easily shown that

$$\delta_j = |\beta_{j+1}||(\mathbf{T}_j^{-1}\mathbf{e}_1)_j|, \tag{3.40}$$

where the $||\cdot||$ is M-norm. Thus δ_j can be used to monitor the process of generating the Lanczos vectors, $j = 1, 2, \cdots, m$. If δ_j is less than a prefixed tolerance, then process is terminated. Further, $|(\mathbf{T}_j^{-1}\mathbf{e}_1)_j|$ can be simply obtained by QR factorization of $\mathbf{T}_j$ [21]. If $\mathbf{T}_j = \mathbf{A}_j\mathbf{B}_j$ is its QR factorization, where $\mathbf{A}_j$ is

orthogonal and $\mathbf{B}_j$ is upper triangular, and

$$\mathbf{A}_j = [\ldots, \mathbf{a}_j] \quad \text{and} \quad \mathbf{B}_j = \begin{bmatrix} \cdot & \cdots & \cdot \\ & \ddots & \vdots \\ & & \xi_j \end{bmatrix}, \tag{3.41}$$

where $\mathbf{a}_j$ denotes the j'th column vector in $\mathbf{A}_j$ and ξ_j is the j'th diagonal entry of $\mathbf{B}_j$, then by substitution, it is easy to show that

$$(\mathbf{T}_j^{-1}\mathbf{e}_1)_j = \frac{a_{1j}}{\xi_j}, \tag{3.42}$$

where a_{1j} is the first entry of the vector of $\mathbf{a}_j$.

From (3.40) and (3.42), it can be seen that only the column vector of $\mathbf{A}_j$, $\mathbf{a}_j$ and lower-right corner element, ξ_j, are used to compute the relative residual error δ_j, and $\mathbf{a}_j$ and ξ_j can be easily updated recursively from $\mathbf{a}_{j-1}$ and ξ_{j-1} in terms of QR factorization and Householder Reflection (see [21]).

Set the initial values $\mathbf{a}_1 = 1$ and $\xi_1 = \alpha_1$. The ULR method gives

$$\mathbf{T}_j = \left[\begin{array}{cccc|c} & & \mathbf{T}_{j-1} & & \mathbf{u} \\ -- & -- & -- & -- & -- \\ 0 & \cdots & 0 & \beta_j & \alpha_j \end{array}\right],$$

where $\mathbf{u}$ is a $(j-1)$-vector and

$$\left[\begin{array}{c} \mathbf{u} \\ -- \\ \alpha_j \end{array}\right]$$

is the last column vector of $\mathbf{T}_j$. Note that $\mathbf{T}_{j-1} = \mathbf{A}_{j-1}\mathbf{B}_{j-1}$, hence

$$\begin{bmatrix} \mathbf{A}_{j-1}^* & \mathbf{0} \\ \mathbf{0}^* & 1 \end{bmatrix} \mathbf{T}_j = \left[\begin{array}{cccc|c} & & \mathbf{B}_{j-1} & & \mathbf{A}_{j-1}^*\mathbf{u} \\ -- & -- & -- & -- & -- \\ 0 & \cdots & 0 & \beta_j & \alpha_j \end{array}\right]$$

$$= \begin{bmatrix} \cdot & \cdots & \cdot & \cdot & \cdot \\ & \ddots & \ddots & \cdot & \cdot \\ & & \ddots & \cdot & \cdot \\ & & & \xi_{j-1} & \mathbf{a}_{j-1}^*\mathbf{u} \\ & & & \beta_j & \alpha_j \end{bmatrix}.$$

Defining a Householder Reflection matrix

$$\mathbf{H}_2 = \frac{1}{\sqrt{\xi_{j-1}^2 + \beta_j^2}} \begin{bmatrix} -\xi_{j-1} & -\beta_j \\ -\beta_j & \xi_{j-1} \end{bmatrix},$$

$\mathbf{A}_j$ and $\mathbf{B}_j$ may be obtained by

$$\mathbf{A}_j = \begin{bmatrix} \mathbf{A}_{j-1} & \mathbf{0} \\ \mathbf{0}^* & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{j-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix},$$

where $\mathbf{I}_{j-2}$ is an identity matrix, and

$$\mathbf{B}_j = \begin{bmatrix} \mathbf{I}_{j-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \cdot & \cdots & \cdot & \cdot & \cdot \\ & \ddots & \ddots & \cdot & \cdot \\ & & \ddots & \cdot & \cdot \\ & & & \xi_{j-1} & \mathbf{a}_{j-1}^*\mathbf{u} \\ & & & \beta_j & \alpha_j \end{bmatrix}.$$

It is easy to verify that $\mathbf{A}_j\mathbf{B}_j = \mathbf{T}_j$, $\mathbf{A}_j$ is orthogonal and $\mathbf{B}_j$ is upper triangular

matrix. Finally, the updated row vector $\mathbf{a}_j$ and scalar ξ_j are obtained, i.e.,

$$\mathbf{a}_j^* = \left[-\frac{\beta_j}{\sqrt{\xi_{j-1}^2 + \beta_j^2}}\mathbf{a}_{j-1}^*, \quad \frac{\xi_{j-1}}{\sqrt{\xi_{j-1}^2 + \beta_j^2}} \right] \tag{3.43}$$

and

$$\xi_j = \frac{\alpha_j \xi_{j-1} - \beta_j(\mathbf{a}_{j-1}^* \mathbf{u})}{\sqrt{\xi_{j-1}^2 + \beta_j^2}}. \tag{3.44}$$

Now the Algorithm for calculation of this relative residual error bound follows.

Algorithm 3.3 *Let* $j = 1$, $a(1) = 1$ *and* $\xi_1 = \alpha_1$. *Compute recursively for* $j = 2, 3, \ldots, m$,

(1) $u(k) = T_j(k, j)$, *for* $k = 1, 2, \ldots, j-1$,

($T_j(k,j)$ IS THE k, j ELEMENT OF MATRIX $\mathbf{T}_j$)

(2) $\xi_2 = \dfrac{\alpha_j \xi_1 - \beta_j \mathbf{a}^* \mathbf{u}}{\sqrt{\xi_1^2 + \beta_j^2}}$

(3) $a(k) = -\dfrac{\beta_j}{\sqrt{\xi_1^2 + \beta_j^2}} a(k)$, *for* $k = 1, 2, \ldots, j-1$

(ξ IS ξ_j SEE (3.41) (3.44))

(4) $a(j) = \dfrac{\xi_1}{\sqrt{\xi_1^2 + \beta_j^2}}$ ($a(k)$,$k = 1, 2, \ldots, j$: COLUMN VECTOR a_j IN MATRIX $\mathbf{A}_j$ (3.41) (3.43))

(5) $(\mathbf{T}_j^{-1} e_1)_j = \dfrac{a(1)}{\xi_2}$ (SEE (3.42))

(6) $\delta_j = |\beta_{j+1}||(\mathbf{T}_j^{-1}\mathbf{e}_1)_j|$ (SEE (3.40))

(7) $\xi_1 = \xi_2$

3.5 The General ULR algorithm

The general algorithm of the ULR method is summarized in this section. The Maximum-Pivot algorithm and Switch algorithm are simply called without details. Note that they are presented in §3.2 and §3.3 respectively. The algorithm for termination is also discussed in §3.4.

Given coefficient matrices $\mathbf{K}$ and $\mathbf{M}$, right hand side vector $\mathbf{f} = \mathbf{b}\mu(t)$ in equation (1.2), the ULR algorithm can be expressed as the following Algorithm 3.4. The integers k_r and k_s denote the numbers of terms involving γ in the formulae forming $\mathbf{r}_j$ and $\mathbf{s}_j$ at current step j, respectively (note that in the previous sections, the superscript index j for k_r^j and k_s^j is used to indicate the step number. In the algorithm below, it is unnecessary to do so). The positive real number ϵ_1 is the termination criterion. If the value $x < \epsilon_1$, x is considered to be zero. Another positive real number, ϵ_2 is the pivot tolerance. The positive integer j_2 is the number of steps at which the current breakdown occurs (see Algorithm 3.1).

Algorithm 3.4 : *Starting with summation indices values* $k_r = k_s = 0$, $j_2 = 0$ *and logical variable afterbreak=.false., calculate* $\mathbf{r}_0 = \mathbf{K}^{-1}\mathbf{b}$, $\beta_1 = \|\mathbf{r}_0\|$, $\mathbf{p}_1 = \mathbf{q}_1 = \mathbf{r}_0/\beta_1$. *Then compute recursively for* $j = 1, 2, \ldots, m$*:*

(1) $\alpha_j = \mathbf{q}_j^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j$

(2) $\gamma_j^{(j-i)} = \mathbf{p}_i^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j, \quad i = j-1, j-2, \ldots, j-k_r$

(3) $\mathbf{r}_j = \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j - \alpha_j \mathbf{q}_j - \sum_{t=1}^{k_r} \gamma_j^{(t)} \mathbf{q}_{j-t}$

(4) $\beta_{j+1} = \|\mathbf{r}_j\|$

(5) If $(\beta_{j+1} < \epsilon_1)$

go to (9) (breakdown case (1), terminate process)

else

$$\mathbf{q}_{j+1} = \frac{\mathbf{r}_j}{\beta_{j+1}} \text{ (right Lanczos vector)}$$

end if

(6) $\mathbf{s}_j^* = \mathbf{p}_{j-k_s}^* \mathbf{M}\mathbf{K}^{-1} - \alpha_{j-k_s}\mathbf{p}_{j-k_s}^* - \beta_{j-k_s}\mathbf{p}_{j-k_s-1}^* - \sum_{t=1}^{k_s} \gamma_{j-k_s+t}^{(t)} \mathbf{p}_{j-k_s+t}^*$

(7) *If* $\|\mathbf{s}_j\| < \epsilon_1$ *(breakdown of case (2))*

if (afterbreak)

Call Switch method (see Algorithm 3.2)

else

if $k_s = 0$

go to (8)

else

$k_s = k_s - 1$ *(3.19)*

go to (6)

end if

end if

else

$\varpi_j = \dfrac{|\mathbf{s}_j^* \mathbf{M}\mathbf{q}_{j+1}|}{\|\mathbf{s}_j\|}$ *(pivot)*

If $(\varpi_j > \epsilon_2)$

$\mathbf{p}_{j+1} = \dfrac{\mathbf{s}_j}{\mathbf{s}_j^* \mathbf{M}\mathbf{q}_{j+1}}$ *(left Lanczos vector)*

$k_r = k_s + 1$ *(3.17)*

if (afterbreak)

$k_s = k_s + 1$ *(3.18)*

afterbreak=.FALSE.

end if

δ_j *(relative residual error bound, see (3.39) and Algorithm 3.3)*

if $(\delta_j < \epsilon_1)$

go to (9) (terminate process)

else

$j = j + 1$

go to (1)

end if

else (breakdown or near breakdown of case (1) occurs)

if (afterbreak)

call Switch method (see Algorithm 3.2)

else

go to (8)

end if

end if

end if

(8) $j_2 = j$

call MPNSV method (reconstruct $\mathbf{s}_j$*, see Algorithm 3.1)*

afterbreak=.TRUE.

go to (7)

(9) Form the $\mathbf{T}_m$ *and solve the reduced equation*

3.6 Conclusions

The ULR method is based on the two-sided Gram-Schmidt biorthogonalization

method, the New-Start Vector method and the Rayleigh-Ritz process. Due to the possibility of breakdown, near breakdown and pathological breakdown, the MPNSV method, Switch method and their algorithms have developed in this chapter. Thus the ULR method can recover from any breakdown.

This chapter has also defined the relative residual error bound and developed an algorithm to compute the bound iteratively in order to monitor and terminate the ULR process.

Solving the reduced system produced by the ULR method can be unstable. In the next chapter, the Eigenvalue Translation (ET) technique is developed to overcome this problem.

Chapter 4

Eigenvalue Translation (ET) technique

4.1 Introduction

Although it is assumed in §1.1 that the semi-discretized linear system (1.2) is stable with respect to time, the reduced system (3.11) may not be. Those eigenvalues that are near or on the imaginary axis can be relocated into the left half of the complex plane, for example see Figure 5.2. This relocation causes instability in time, as $\mathbf{w}$ can possess exponentially increasing terms. Therefore, the residual error function (3.38) could exponentially increase as the time increases. If some eigenvalues are purely imaginary, the error could be oscillating. On the other hand, because the non-zero eigenvalues of the matrix $\mathbf{K}^{-1}\mathbf{M}$ are reciprocals of the corresponding eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$, those eigenvalues of $\mathbf{K}^{-1}\mathbf{M}$ which are near to the imaginary axis and whose modulus are larger than one play the dominant role in the solution of (1.2).

The ET method translates those eigenvalues of the reduced system (3.11) which are in the left half of the complex plane and those eigenvalues which are on the

imaginary axis to the right half and near to the imaginary axis, thus stabilizing the system. The translation satisfies the requirements:

1. After translation, all the eigenvalues of new system must have positive real parts, i.e., the new system is stable in time.

2. The translation does not lead to significant changes in those eigenvalues of $\mathbf{T}_m$ whose real parts are positive.

3. The new residual error (defined by (4.4)) is a minimum.

To this end, a right multiplicative transformation of the matrix $\mathbf{T}_m$ is used to change (3.11) into equation:

$$\mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\mathbf{S}\tilde{\mathbf{D}}_U\mathbf{S}^{-1} - \mathbf{I}_\ell)\mathbf{Y}_U^*]\dot{\mathbf{w}} + w = \beta\mathbf{e}_1\mu(t), \tag{4.1}$$

which is shown in Theorem 4.1 to satisfy above 1 and 2. In equation (4.1), ℓ is the number of eigenvalues to be translated, $\mathbf{X}_U$ and $\mathbf{Y}_U$ are $m \times \ell$ real rectangular matrices formed by the right and left eigenvectors corresponding to the eigenvalues with negative real parts, $\mathbf{D}_U$ and $\tilde{\mathbf{D}}_U$ are $\ell \times \ell$ real matrices constructed from the original and translated eigenvalues respectively (see (4.7) and (4.9)), The $\ell \times \ell$ non-singular matrix $\mathbf{S}$ is yet to be determined, and the system matrix is

$$\mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\mathbf{S}\tilde{\mathbf{D}}_U\mathbf{S}^{-1} - \mathbf{I}_\ell)\mathbf{Y}_U^*]. \tag{4.2}$$

In the next section, it is shown that $\mathbf{S}$ can be chosen as the identity, so that the matrix becomes

$$\mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\tilde{\mathbf{D}}_U - \mathbf{I}_\ell)\mathbf{Y}_U^*], \tag{4.3}$$

and the new system has a small residual error

$$\tilde{\theta}_m(t) = \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m\dot{\mathbf{w}} + \mathbf{Q}_m\mathbf{w} - \mathbf{K}^{-1}\mathbf{f}. \tag{4.4}$$

thus satisfying 3.

4.2 Eigenvalue translation

Let $\{\lambda_j, \mathbf{x}_j, \mathbf{y}_j\}$, $j = 1, 2, \ldots, m$ be the eigentriples of the matrix $\mathbf{T}_m$, satisfying,

$$\begin{cases} \mathbf{T}_m\mathbf{x}_j &= \lambda_j\mathbf{x}_j \\ \mathbf{y}_j^*\mathbf{T}_m &= \lambda_j\mathbf{y}_j^*. \end{cases}$$

We assume that $\mathbf{x}_j$ and $\mathbf{y}_j$ are normalized by

$$\begin{cases} \|\mathbf{x}_j\| &= 1, \\ \mathbf{y}_i^*\mathbf{x}_j &= \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \end{cases} \tag{4.5}$$

where $\|\cdot\|$ denotes the Euclidean vector norm. Without loss of generality, let the first ℓ eigenvalues be those with non-positive real parts, i.e., λ_i, where $i \in L$ and $L = \{1, 2, \ldots, \ell\}$. However, the multiplicity of any eigenvalue is larger than 1, i.e., for some i and j $\lambda_i = \lambda_j$, then only one is included in this set. Clearly, the eigenvalues with non-zero imaginary parts appear in the subset L in complex conjugate pairs. Let $\mathbf{\Lambda}_L$ be the diagonal $\ell \times \ell$ matrix whose diagonal entries are equal to λ_j, $j \in L$.

Let $\mathbf{X}_L$ and $\mathbf{Y}_L$ be the $m \times \ell$ matrices whose columns are the right and left eigenvectors, respectively, $\mathbf{x}_j$ and $\mathbf{y}_j$ for $j \in L$, with the columns of these matrices

arranged according to the numbering of the eigenvalues λ_j in the matrix $\mathbf{\Lambda}_L$. From (4.5), it is easy to prove that the normalization of $\mathbf{X}_L$ and $\mathbf{Y}_L$ is

$$\mathbf{Y}_L^* \mathbf{X}_L = \mathbf{I}_\ell, \tag{4.6}$$

where $\mathbf{I}_\ell$ is the $\ell \times \ell$ identity matrix.

We form real matrices $\mathbf{D}_U$, $\mathbf{X}_U$ and $\mathbf{Y}_U$ corresponding to the complex matrices $\mathbf{\Lambda}_L$, $\mathbf{X}_L$ and $\mathbf{Y}_L$ as follows.

Define the 2×2 unitary matrix

$$\mathbf{U}_b = \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix},$$

which can be easily shown to satisfy the following equalities,

$$\mathbf{U}_b^* \mathbf{U}_b = \mathbf{U}_b \mathbf{U}_b^* = \mathbf{I}_2$$

$$\mathbf{U}_b \begin{bmatrix} \lambda & 0 \\ 0 & \bar{\lambda} \end{bmatrix} \mathbf{U}_b^* = \begin{bmatrix} R(\lambda) & -I_m(\lambda) \\ I_m(\lambda) & R(\lambda) \end{bmatrix},$$

$$[\mathbf{x} \ \bar{\mathbf{x}}] \mathbf{U}_b^* = [R(\mathbf{x}) + I_m(\mathbf{x}) \quad R(\mathbf{x}) - I_m(\mathbf{x})],$$

for any complex number λ and vector $\mathbf{x}$, where and $\bar{\lambda}$ and $\bar{\mathbf{x}}$ denote complex conjugate, and $R(\cdot)$ and $I_m(\cdot)$ denote respectively the real part and imaginary part of vector or scalar. The right hand side of these equations contains only real matrices. Let $\mathbf{U}$ be the unitary block diagonal $\ell \times \ell$ matrix with non-zero 1×1 and 2×2 diagonal blocks. The diagonal 1×1 blocks corresponding to the real eigenvalues in the same position in $\mathbf{\Lambda}_L$ are equal to 1, while diagonal 2×2 blocks

corresponding to the eigenvalues with non-zero imaginary parts are $\mathbf{U}_b$. Define the real matrices

$$\begin{cases} \mathbf{D}_U &= \mathbf{U}\mathbf{\Lambda}_L\mathbf{U}^*, \\ \mathbf{X}_U &= \mathbf{X}_L\mathbf{U}^*, \\ \mathbf{Y}_U &= \mathbf{Y}_L\mathbf{U}^*. \end{cases} \tag{4.7}$$

It is easy to prove that the following equalities are true,

$$\begin{cases} \mathbf{T}_m\mathbf{X}_U &= \mathbf{X}_U\mathbf{D}_U, \\ \mathbf{Y}_U^*\mathbf{X}_U &= \mathbf{I}_\ell. \end{cases} \tag{4.8}$$

Now, let us construct the matrix $\tilde{\mathbf{D}}_U$ in (4.2). Let $\tilde{\mathbf{\Lambda}}_L$ be a diagonal matrix made up of the translated eigenvalues that are chosen freely by the user to minimize the residual error of (4.4) as discussed at the end of this section. Similarly, we construct the unitary matrix $\tilde{\mathbf{U}}$ based on the matrix $\tilde{\mathbf{\Lambda}}_L$ as described above and define the real matrix $\tilde{\mathbf{D}}_U$ by

$$\tilde{\mathbf{D}}_U = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}_L\tilde{\mathbf{U}}^*. \tag{4.9}$$

We have the following theorem for the translation matrix (4.2).

Theorem 4.1 *With* $\mathbf{\Lambda}_L$, $\tilde{\mathbf{\Lambda}}_L$, $\mathbf{U}$, $\tilde{\mathbf{U}}$, $\mathbf{X}_U$, $\mathbf{D}_U$, $\tilde{\mathbf{D}}_U$ *and* $\mathbf{Y}_U$ *defined as above, for any non-singular* $\ell \times \ell$ *matrix* $\mathbf{S}$, *denote* $\tilde{\mathbf{S}} = \mathbf{U}^*\mathbf{S}\tilde{\mathbf{U}}$. *Then the pair of matrices* $(\mathbf{X}_L\tilde{\mathbf{S}}, \tilde{\mathbf{\Lambda}}_L)$ *contains the eigenvectors and corresponding eigenvalues for the matrix (4.2). Those right eigenvalues of* $\mathbf{T}_m$ *with positive real parts are still right eigenvalues of (4.2).*

PROOF: Post-multiplying (4.2) by $\mathbf{X}_U\mathbf{S}$ and taking account (4.8), we obtain

$$\mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\mathbf{S}\tilde{\mathbf{D}}_U\mathbf{S}^{-1} - \mathbf{I}_\ell)\mathbf{Y}_U^*]\mathbf{X}_U\mathbf{S} = \mathbf{X}_U\mathbf{S}\tilde{\mathbf{D}}_U.$$

Post-multiplying both sides of the above equation by $\tilde{\mathbf{U}}$ and noting (4.7) and (4.9), we obtain

$$\mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\mathbf{S}\tilde{\mathbf{D}}_U\mathbf{S}^{-1} - \mathbf{I}_\ell)\mathbf{Y}_U^*]\mathbf{X}_L\tilde{\mathbf{S}} = \mathbf{X}_L\tilde{\mathbf{S}}\tilde{\mathbf{\Lambda}}_L.$$

This is the first assertion of the theorem.

The second assertion holds due to the orthogonality of the right and left eigenvalues (4.5) and the definition of $\mathbf{Y}_U$ (4.7), i. e.,

$$\mathbf{Y}_U^*\mathbf{x}_j = \mathbf{0},$$

where $\mathbf{x}_j$, $j \notin L$, is a right eigenvector of $\mathbf{T}_m$ with positive real part. Therefore

$$\mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\mathbf{S}\tilde{\mathbf{D}}_U\mathbf{S}^{-1} - \mathbf{I}_\ell)\mathbf{Y}_U^*]\mathbf{x}_j = \mathbf{T}_m\mathbf{x}_j = \lambda_j\mathbf{x}_j.$$

This completes the theorem. □

Now, let us investigate the residual error of the system. Substituting (3.33) into equation (4.4) and using (3.34), we have

$$\tilde{\theta}_m(t) = \mathbf{Q}_m(\mathbf{T}_m\dot{\mathbf{w}} + \mathbf{w} - \beta\mathbf{e}_1\mu(t)) + \mathbf{r}_m\mathbf{e}_m^*\dot{\mathbf{w}}.$$

Since $\mathbf{w}$ is the solution of (4.1), $\tilde{\theta}_m(t)$ can be simplified to

$$\tilde{\theta}_m(t) = [\mathbf{r}_m \mathbf{e}_m^* - \mathbf{Q}_m \mathbf{T}_m \mathbf{X}_U (\mathbf{D}_U^{-1} \mathbf{S} \tilde{\mathbf{D}}_U \mathbf{S}^{-1} - \mathbf{I}_\ell) \mathbf{Y}_U^*] \dot{\mathbf{w}} = \mathbf{E} \dot{\mathbf{w}}, \tag{4.10}$$

where $\mathbf{E}$ is an $m \times m$ constant matrix.

We can minimize the residual bound from (4.10), which depends on the choice of translated eigenvalues $\tilde{\mathbf{D}}_U$ and the non-singular matrix $\mathbf{S}$.

To this end, we denote $\|\cdot\|$ as the spectral norm on rectangular matrices. From (4.10) and (4.7),

$$\begin{aligned} \|\tilde{\theta}_m\| &\leq \left(\|\mathbf{r}_m\| + \|\mathbf{Q}_m\|\|\mathbf{T}_m\|\|\mathbf{X}_L\|(\|\mathbf{D}_U^{-1}\|\|\tilde{\mathbf{D}}_U\|\kappa(\mathbf{S}) + 1)\|\mathbf{Y}_L\|\right) \|\dot{\mathbf{w}}\| \\ &= (\|\mathbf{r}_m\| + c)\|\dot{\mathbf{w}}\|, \end{aligned} \tag{4.11}$$

where $\kappa(\mathbf{S}) = \|\mathbf{S}\|\|\mathbf{S}^{-1}\|$ is the condition number of the matrix $\mathbf{S}$, c is a constant. To minimize the residual error bound based on this estimate, $\max|\tilde{\lambda}_i|$ can be chosen as being very small and $\mathbf{S} = \mathbf{I}_\ell$, in which case $\min \kappa(\mathbf{S}) = \kappa(\mathbf{I}_\ell) = 1$ and $\|\tilde{\mathbf{D}}_U\| = \max|\tilde{\lambda}_i|$.

4.3 ET technique algorithm

Based on Theorem 4.1, an algorithm for the Eigenvalue Translation Technique (ETT) is presented as follows:

Algorithm 4.1

1. *Compute the eigenvalues of the matrix* $\mathbf{T}_m$, *and find an eigenvalue set* $\{\lambda_1, \lambda_2, \ldots, \lambda_\ell\}$ *consisting of eigenvalues with non-positive real part and* $\lambda_i \neq$

λ_j. *If there are multiple such eigenvalues, only one is included in this set. Compute the corresponding right and left eigenvectors. Normalize the eigenvectors to satisfy (4.5) and form the matrices* $\mathbf{X}_L$, $\mathbf{Y}_L$ *and* $\mathbf{\Lambda}_L$. *If there are any complex eigentriples, transform them into real matrices* $\mathbf{X}_U$, $\mathbf{Y}_U$ *and* $\mathbf{D}_U$ *as in (4.7).*

2. *Chose translated eigenvalues* $\{\tilde{\lambda}_1, \tilde{\lambda}_2, \cdots, \tilde{\lambda}_\ell\}$ *to form the matrix* $\tilde{\mathbf{\Lambda}}_L$ *and then the real matrix* $\tilde{\mathbf{D}}_U$, *where* $\tilde{\lambda}_i \neq \tilde{\lambda}_j$, $i \neq j$, *and* $\max_{i\in L} |\tilde{\lambda}_i|$ *is very small. Experiments show that* $0.1 < \bar{\lambda} < 0.9$ *is appropriate.*

3. *Finally, form matrix (4.3), and the translated system is*

$$\mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\tilde{\mathbf{D}}_U - \mathbf{I}_\ell)\mathbf{Y}_U^*]\dot{\mathbf{w}} + \mathbf{w} = \beta\mathbf{e}_1\mu(t). \tag{4.12}$$

4. *If there are multiple eigenvalues of matrix* $\mathbf{T}_m$ *with non-positive real part, let*

$$\mathbf{T}_m = \mathbf{T}_m[\mathbf{I}_m + \mathbf{X}_U(\mathbf{D}_U^{-1}\tilde{\mathbf{D}}_U - \mathbf{I}_\ell)\mathbf{Y}_U^*],$$

and repeat above steps. Noting that those eigenvalues of new $\mathbf{T}_m$ *with non-positive real part are the same eigenvalues of the old* $\mathbf{T}_m$, *repeatedly computing the eigenvalues of the new* $\mathbf{T}_m$ *is not necessary in step 1.*

4.4 Concluding remarks

This chapter has developed the ET technique and its algorithm to stabilize the reduced system. Thus the ULR method is complete theoretically. Some remarks follow.

(1.) Computation

In the ET technique, all eigenvalues of reduced system need to be computered. Because the reduced system is always small and already upper Hessenberg in form, the computation is inexpensive, (calculating nonsymmetric eigenvalues usually needs such stages: balancing, reducing to upper Hessenberg form, further reducing to Schur form and then obtaining eigenvalues from the diagonal entries of the Schur matrix (see [1, 21])).

(2.) Stability of the AR method

The stability of the AR method is discussed here. The conclusion is that, in equation (1.2), if

$$\mathbf{K}^{-1} + (\mathbf{K}^{-1})^* \tag{4.13}$$

is positive definite, then the reduced system (2.12) from the AR method is stable in time. The same result can also be seen in the paper of Gallopoulos and Saad [20]. Demonstrated below is a proof in different context.

First, it can be seen that if (4.13) is positive definite, then so is $\mathbf{H}_m + \mathbf{H}_m^*$. This is because from (2.9) and (2.10),

$$\begin{aligned} \mathbf{H}_m + \mathbf{H}_m^* &= \mathbf{Q}_m^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m + (\mathbf{Q}_m^* \mathbf{M}\mathbf{K}^{-1}\mathbf{M}Q_m)^* \\ &= (\mathbf{M}\mathbf{Q}_m)^*(\mathbf{K}^{-1} + (\mathbf{K}^{-1})^*)\mathbf{M}\mathbf{Q}_m. \end{aligned}$$

As $\mathbf{M}\mathbf{Q}_m$ is a full rank matrix, $\mathbf{H}_m + \mathbf{H}_m^*$ is positive definite as well. Therefore, it can be proved from the following Lemma which can be found in Young [67] that all eigenvalues of $\mathbf{H}$ have positive real part,

Lemma 4.1 *A real matrix* $\mathbf{A}$ *is a matrix whose eigenvalues are all in the right half complex plane if and only if there exists a real positive definite matrix* $\mathbf{B}$ *such that the matrix* $\mathbf{C}$ *given by*

$$\mathbf{C} = \mathbf{AB} + \mathbf{BA}^*$$

is positive definite.

Applying Lemma 4.1 with $\mathbf{A} = \mathbf{H}_m$ and $\mathbf{B} = \mathbf{I}$, $\mathbf{H}_m$ is a such matrix that all its eigenvalues are in the right half complex plane.

(3.) Other stabilization techniques

Other methods have developed to deal with the unstable problem, for example, those in [55, 23]). The former work translates unstable eigenvalues from one half complex plane using Projection and Deflation methods. The latter work for overcoming this instability problem is based on an Implicitly Restarted Lanczos method in which unstable eigenvalues are discarded. The comparison of the ET technique to these methods is an important topic for further work.

Part III

Numerical Experiments with the ULR Method

Chapter 5

Simple one and two-dimensional examples

5.1 Introduction

Numerical experiments performed using the algorithms described in Part II are presented in this and the next two chapters. Various problems are tested. The Galerkin finite element method is used to spatially discretize the governing equation. Then the ULR method is carried out to reduce the size of the result system and therefore obtain an approximate solution in terms of the Crank-Nicolson time-stepping scheme (1.3). During the ULR process, reorthogonalization is performed. All these computations were run on a SUN Sparc2 workstation at the University of Manitoba.

These results give us some idea of the comparative effectiveness of the ULR method with respect to the classic Crank-Nicolson solver, i.e., directly applying the Crank-Nicolson time-stepping scheme to the equation (1.2) which is produced by the Galerkin finite element method. The Root-Mean Square (RMS) error is

designed to compare these results. The RMS error is defined as

$$RMS = \sqrt{\frac{\sum_{k=1}^{n}(c(k) - c_0(k))^2}{n-1}}, \tag{5.1}$$

where $c(k)$ is the approximation solution on each node, $c_0(k)$ is the solution from the classic Crank-Nicolson solver and n is the number of total nodes.

In this chapter, some basic models of the one and two dimensional advection dispersion equations are used to examine and verify the suitability of the ULR method. In the next chapter, the method is applied to two-dimensional field problems, for the purpose of evaluating the overall effectiveness of the method. In chapter 6, the radionuclide decay chain problems will be solved by the method. All these applications indicate that the ULR method is highly efficient.

5.2 One–dimensional example

Consider the semi-discretization of the one-dimensional advection dispersion equation

$$D\frac{\partial^2 C}{\partial x^2} - v\frac{\partial C}{\partial x} = \frac{\partial C}{\partial t}, \qquad x \in [0, l], \quad t \in [0, \infty),$$

with Dirichlet boundary conditions

$$C(0,t) = \phi(t), \qquad C(l,t) = \psi(t),$$

and initial conditions

$$C(x,0) = \begin{cases} \phi(0) & x = 0 \\ 0 & otherwise, \end{cases}$$

where C is the mass concentration, v is the velocity and D is the coefficient of hydrodynamic dispersion.

Assume that the flow velocity is 1 m/yr and dispersion coefficient is 10 m^2/yr. The Galerkin method is used to yield a semidiscretized system (see equation (1.2)) of size $n = 499$, where $\mathbf{M}$ is a tridiagonal positive definite matrix and $\mathbf{K}$ is the tridiagonal unsymmetric.

Mesh size (m)	0.006	0.0075	0.015	0.03	0.06	0.1	0.2	0.3
N_s	86	58	66	65	95	120	97	186
Reduced size	Stability							
50	s	u	s	s	s	u	s	s
100	s	u	s	u	s	s	s	s
150	s	s	s	s	s	s	s	s
200	s	s	s	s	s	s	s	s

Table 5.1: Breakdown and Stability behavior of the ULR method. Pivot tolerance $\epsilon = 0.001$. N_s denotes the step number at which the first breakdown occurs. u stands for unstable while s for stable.

Table 5.1 shows the step number at which the first breakdown occurs (N_s), and the stability of the ULR method in various reduced sizes. The pivot tolerance is $\epsilon = 0.001$. This table shows that the breakdown can easily occur in the ULR method. The table also indicates that there are 4 out of 32 cases which are unstable. For those unstable cases, there is only one eigenvalue whose real part is negative when the reduced size is 50; while there are 2 such eigenvalues when

the reduced size is 100. Obviously, the greater the size of the reduced system, the less chance the instability phenomenon occurs.

	ULR without ET	
Reduced size	50	100
Time steps	RMS	
1	0.178D-2	0.219D-6
10	0.603D-3	0.154D-6
100	0.532D-4	0.213D-10
1500	0.821D-9	0.934D-13
3000	0.162D-12	0.161D-12

Table 5.2: RMS error for the cases where the reduced systems are stable (mesh size is 0.06 m).

	ULR with ET			
Mesh size (m)	0.0075	0.0075	0.03	0.1
Reduced size	50	100	100	50
Time steps	RMS			
1	0.771D-3	0.244D-10	0.261D-5	0.324D-2
10	0.334D-2	0.191D-10	0.156D-4	0.255D-2
100	0.391D-3	0.297D-8	0.308D-14	0.930D-3
1500	0.269D-12	0.275D-12	0.133D-12	0.187D-4
3000	0.266D-12	0.275D-12	0.135D-12	0.938D-7

Table 5.3: RMS error for the cases where the reduced systems are unstable.

Table 5.2 shows the RMS error of the stable cases to which the ULR method without ET technique is applied. Table 5.3 shows the RMS error of the unstable cases to which the ULR method with ET technique is applied. It is apparent that when the evolution period increases in time, the RMS error decreases and

approaches to zero. This is consistent with the analysis of residual errors of (3.38) and (4.10), because the solutions $\dot{\mathbf{w}}$ and $\dot{\tilde{\mathbf{w}}}$ are stable in these cases.

Table 5.4 shows an execution time comparison between the ULR method and classic Crank-Nicolson solver. For the stable case, the execution time is 29 second and 1 minute 20 seconds when the reduced size is 50 and 100 respectively. While the execution time of classic Crank-Nicolson solver is 13 minutes 22 seconds. For the unstable case, the execution times are longer than stable case. It is due to the additional efforts to translate these eigenvalues which have a negative real part. However, overall time comparison shows superior behavior to the classic Crank-Nicolson solver.

	Reduced size	Stable case	unstable case
Mesh size (m)		0.06	0.0075
ULR solver	50	0 : 29	1 : 24
	100	1 : 20	4 : 48
Classic CN	499	13 : 22	12 : 51

Table 5.4: Comparison of execution time between the ULR and classical Crank-Nicolson (CN) solvers (minute: second).

5.3 Two-dimensional example

A simple two-dimensional example of the time-dependent advection dispersion equation is designed to test the accuracy of the method for simulating transport in a saturated system. The mathematical model is given by

$$D_{xx}\frac{\partial^2 C}{\partial x^2} + D_{xz}\frac{\partial^2 C}{\partial x \partial z} + D_{zx}\frac{\partial^2 C}{\partial z \partial x} + D_{zz}\frac{\partial^2 C}{\partial z^2} - v_x\frac{\partial C}{\partial x} - v_z\frac{\partial C}{\partial z} - \frac{\partial C}{\partial t}$$
$$= 0, \tag{5.2}$$

where C is mass concentration of species, $D_{xx}, D_{xz}, D_{zx}, D_{zz}$ are components of hydrodynamic dispersion and v_x, x_z are components of groundwater velocity. In this example, only one species is considered with no retardation or radionuclide decay. The components of the coefficient of the hydrodynamic dispersion for an isotropic medium are given by [7],

$$\begin{aligned} D_{xx} &= \alpha_l \frac{v_x^2}{|\mathbf{v}|} + \alpha_t \frac{v_z^2}{|\mathbf{v}|} + D_d\tau, \\ D_{zz} &= \alpha_t \frac{v_x^2}{|\mathbf{v}|} + \alpha_l \frac{v_z^2}{|\mathbf{v}|} + D_d\tau, \\ D_{xz} &= D_{zx} = \frac{(\alpha_l - \alpha_t)v_x v_z}{|\mathbf{v}|}, \end{aligned}$$

where α_l and α_t are longitudinal and transversal dispersivities, respectively, $|\mathbf{v}| = \sqrt{v_x^2 + v_z^2}$, D_d is the free-solution diffusion coefficient of the solute, and τ is the tortuosity of the medium.

An example in the paper [57] is chosen for a test. This example involves uniform horizontal flow in a rectangular domain. A continuous patch source, represented as a first-type (Direchlet) boundary condition, is placed on the upstream boundary. A representation of the problem is provided in Figure 5.1 below. A Dirichlet boundary condition is imposed at the left boundary: $C = C_0$ for $0 \leq z \leq 0.5$ m and $C = 0$ for $z > 0.5$ m. The Neuman type boundary condition is imposed on

the remaining three boundaries with $\partial C/\partial \vec{N} = 0$, where $\vec{N}$ is the outward normal vector of the boundary. The longitudinal groundwater velocity, v_x, is 0.1 m/day, and the transverse velocity, v_z, is 0. The longitudinal dispersivity, α_l, is 0.05 m, and the transverse dispersivity, α_t, is 0.005 m.

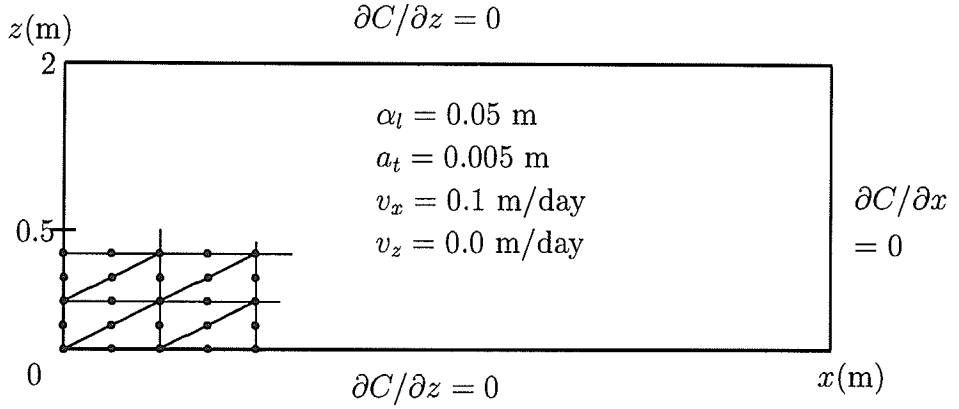

Figure 5.1: Domain of the two-dimensional example

The Galerkin finite element method is used to discretize the problem, with uniform quadratic (six points) triangular elements. The code was written in Fortran 77, using the BLAS and LAPACK packages [1]. In order to study the efficiency of the ULR method, several examples were tested with various domain sizes adhering to 1071 nodes. The possibility of the breakdown or near breakdown, the behavior of the convergence and the execution time were investigated. The relative residual error bound (3.39) or (3.40) is used to terminate the recursion and the tolerance is set to 10^{-8}. The maximum number of Lanczos vectors is set to 100. The termination of the process depends on either of the above requirements being satisfied.

In all these samples, the ULR processes are terminated at step 100. So, the reduced sizes of all these examples are 100. Table 5.5 lists the step number at which the first breakdown occurs (N_s) and stability for three different pivot tolerances $\epsilon = 10^{-5}, 10^{-10}, 10^{-15}$ and different domain sizes $x \times z$ (m). The symbol (–) means that no breakdown occurs during the process and $*$ indicates that the reduced system is unstable and the ET technique is applied in these cases. It is clear to see that the larger the pivot tolerance, the more likely a breakdown problem is encountered. Furthermore, the larger the grid size, the more likely the instability problem is encountered. This is evidence that the stability of the reduced system is closely related to the stability condition of the semi-discretized system (1.2). It is known that there is the grid Peclet number which is the indicator of the stability of the semi-discretized system (see[11, 30]) and the grid Peclet number is direct proportion to the grid size. A larger grid Peclet number of a mesh, the semi-discretized system is more likely to suffer numerical oscillation, and consequently the reduced system is more likely to be unstable. Definition and detailed discussion about the grid Peclet number will be given in the next chapter.

Grid size ($x \times z$ (m))	25×2	25×4	50×2	75×2	100×2	100×4
ϵ	N_s					
10^{-5}	16	16	14	17*	25*	13
10^{-10}	22	31	65	–*	–*	38
10^{-15}	58	–	–	–*	–*	–*

Table 5.5: Investigation of breakdown and stability. N_s is the step number at which the first breakdown occurs. The symbol (–) means that no breakdown occurs during the process and the $*$ means that the reduced system is unstable.

	Behavior				RMS error				
ϵ	N_s	S	E_t	δ	t_1	t_{100}	t_{300}	t_{1000}	t_{3000}
10^{-5}	17	u	7:22	0.3D-4	0.8D-6	0.3D-5	0.9D-5	1.0D-5	0.3D-14
10^{-10}	–	u	6:53	0.2D-2	0.8D-2	0.2D-2	0.3D-3	0.3D-4	0.4D-8

Table 5.6: Results from the example with the domain size 75×2 (m). ϵ denotes the pivot tolerance, N_s is the step number at which the first breakdown occurs, S stands for stability, u stands for unstable, E_t stands for execution time, δ is the relative residual error. t_1, t_{100}, t_{300}, t_{1000} and t_{3000} stand for t =1, 100, 300, 1000 and 3000 days respectively.

The typical examples presented in Table 5.6 and Table 5.7 show the influence of the pivot tolerance ϵ on the accuracy of the approximation and the execution time. The first part of each table lists the behavior of breakdown, the step number at which the first breakdown occurs (N_s), stability (S) where u and s denote unstable and stable respectively, execution time (E_t) and the relative residual error (δ) of equation (3.39). The second part of the table lists RMS errors at various times, $t = 1$ (t_1), $t = 100$ (t_{100}), $t = 300$ (t_{300}), $t = 1000$ (t_{1000}) and $t = 3000$ (t_{3000}) days. The RMS error is used to measure the difference between the approximations obtained from the ULR method and that from the classic Crank-Nicolson solver.

	Behavior				RMS error				
ϵ	N_s	S	E_t	δ	t_1	t_{100}	t_{300}	t_{1000}	t_{3000}
10^{-5}	14	s	8:54	0.7D-5	0.6D-7	0.2D-5	0.4D-4	0.5D-15	0.0
10^{-10}	65	s	7:10	0.4D-2	0.9D-5	0.4D-3	0.3D-2	0.4D-4	0.2D-8
10^{-15}	–	s	3:39	0.8D-2	0.2D-4	0.4D-3	0.5D-3	0.2D-3	0.6D-6

Table 5.7: Results from the example with the domain size 50×2 (m). The meaning of the symbols can be seen in Table 5.6 except for s which is stands for stable.

Table 5.6 shows the case with the domain size $75 \times 2(m)$ with two different pivot tolerances. When the pivot tolerance ϵ is equal to or lower than 10^{-10}, no breakdown occurs. Both situations are unstable and the ET technique is used. In Table 5.7 where the domain size is $50 \times 2(m)$, all three situations are stable but two cases suffer breakdown. Both tables show that the larger the tolerance ϵ, the earlier the ULR process encounters breakdown, but the accuracy of the solution improves. A large saving in execution time can be obtained in the absence of breakdown and instability. This behavior can be observed for the case with $\epsilon = 10^{-15}$ in Table 5.7. Together with consideration of Table 5.5, it is obvious that there is also balance between accuracy and breakdown. Decreasing the pivot tolerance will postpone or avoid the breakdown but the accuracy in the solutions is lowered. These tables also show that the RMS errors are not greater than the relative residual error. This means that it is appropriate to use the relative residual error to monitor and check the termination of the process, and it can be easily computed. Furthermore, from Table 5.6 and 5.7, it only take less than 10 seconds to solve for the concentration at $t = 3000$ days, while the classic Crank-Nicolson solver needs more than 35 seconds to solve the same problem. When $t = 3000$ days, the RMS errors are lower than 10^{-6}. It is concluded that the ULR method is very efficient for long time period prediction.

Figure 5.2 and Figure 5.3 show the eigenvalue distribution comparisons for the case where the domain size is 75×2 (m) and the pivot tolerance $\epsilon = 10^{-10}$. Figure 5.2 shows these distributions before and after the ULR method. These plots show that the eigenvalues approximated by the ULR method seem to converge from the outside of the convex spectrum. There are conjugate pair eigenvalues,

which are plotted with $\diamond$, relocated in the left half plane after ULR method, which cause instability. Figure 5.3 plots the eigenvalue distributions before and after the ET method. After translation, these unstable eigenvalues are transformed into the right half: the real parts are set to be 0.1 and the imaginary parts keep the same. And all other eigenvalues do not change.

Figures 5.4-5.6 show comparisons to the classic Crank-Nicolson solver. Figure 5.4 shows two comparison profiles between solutions of equation (5.2) by the ULR method and classic Crank-Nicolson method in the longitudinal direction, at the location $z = 0$ and at various time. The transversal direction profiles for the same comparison purpose are shown in Figures 5.5 and 5.6 respectively. The comparisons are given at various x locations and various time evolution steps.

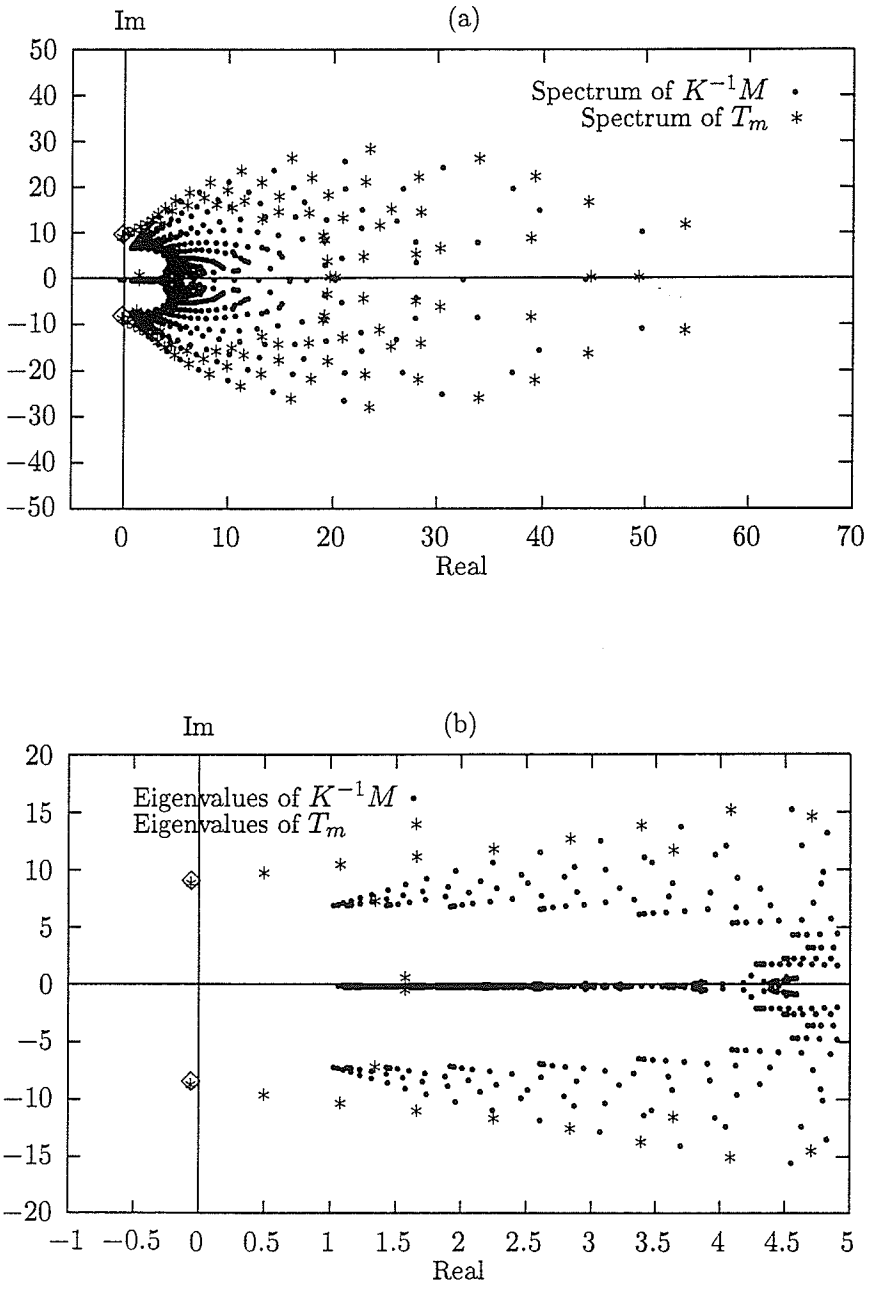

Figure 5.2: Eigenvalue distribution comparison between before and after the ULR method for the case with domain size $75m \times 2m$. (a) Spectrum. (b) Focus on the area $\{\ -1 \leq x \leq 5\ \}$. $\diamond$ is the plot of eigenvelue with negative real part.

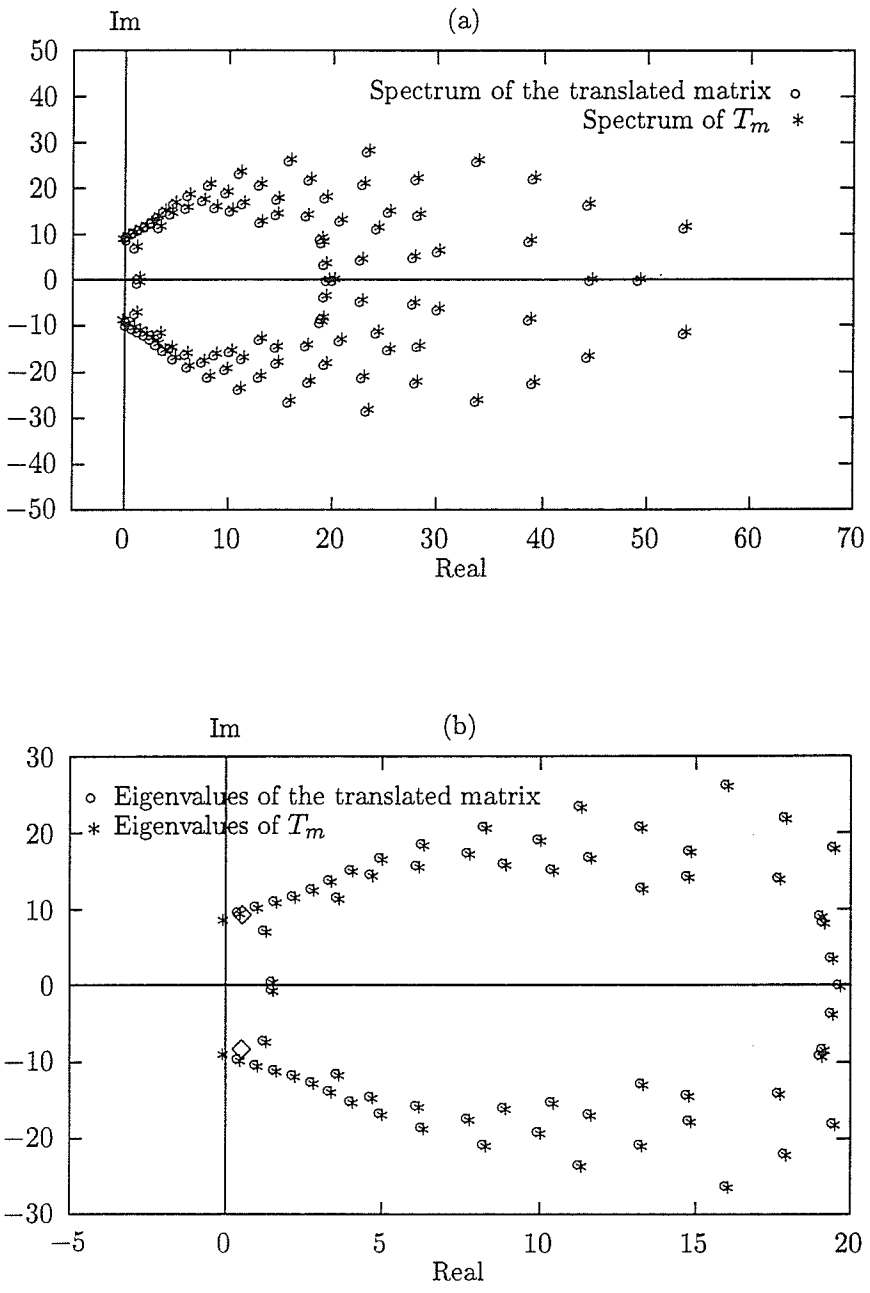

Figure 5.3: Eigenvalue distribution comparison between before and after the ET technique for the case with domain size $75m \times 2m$. (a) Spectrum. (b) Focus on the area $-5 \leq x \leq 20$. $\diamond$ is plot of translated eigenvalue.

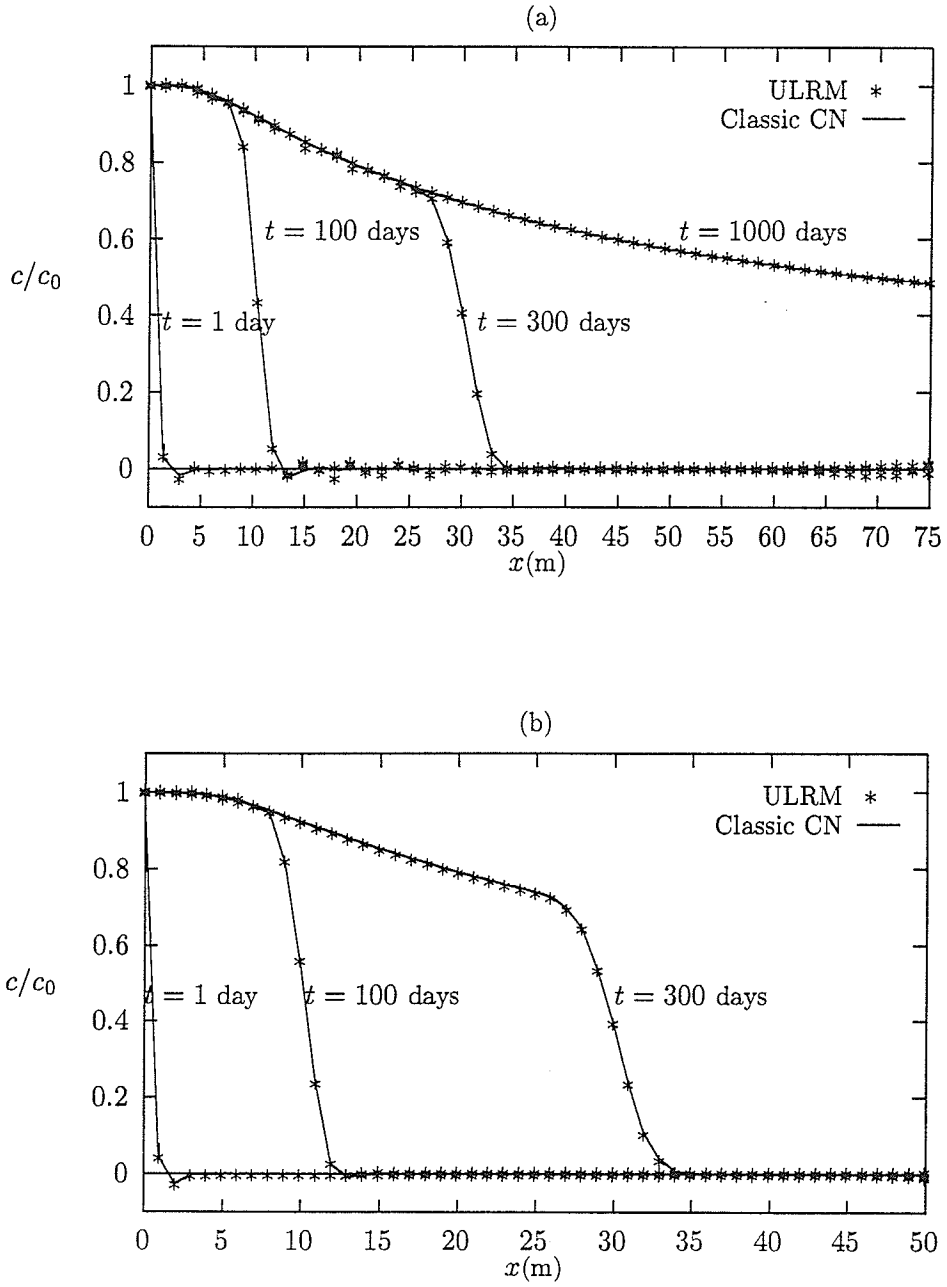

Figure 5.4: Comparison of the concentration solutions of equation (5.2) between the ULR method and classic Crank-Nicolson (CN) method in longitudinal direction at location $z = 0$ and different time steps. Domain size is (a) $75m \times 2m$ (b) $50m \times 2m$.

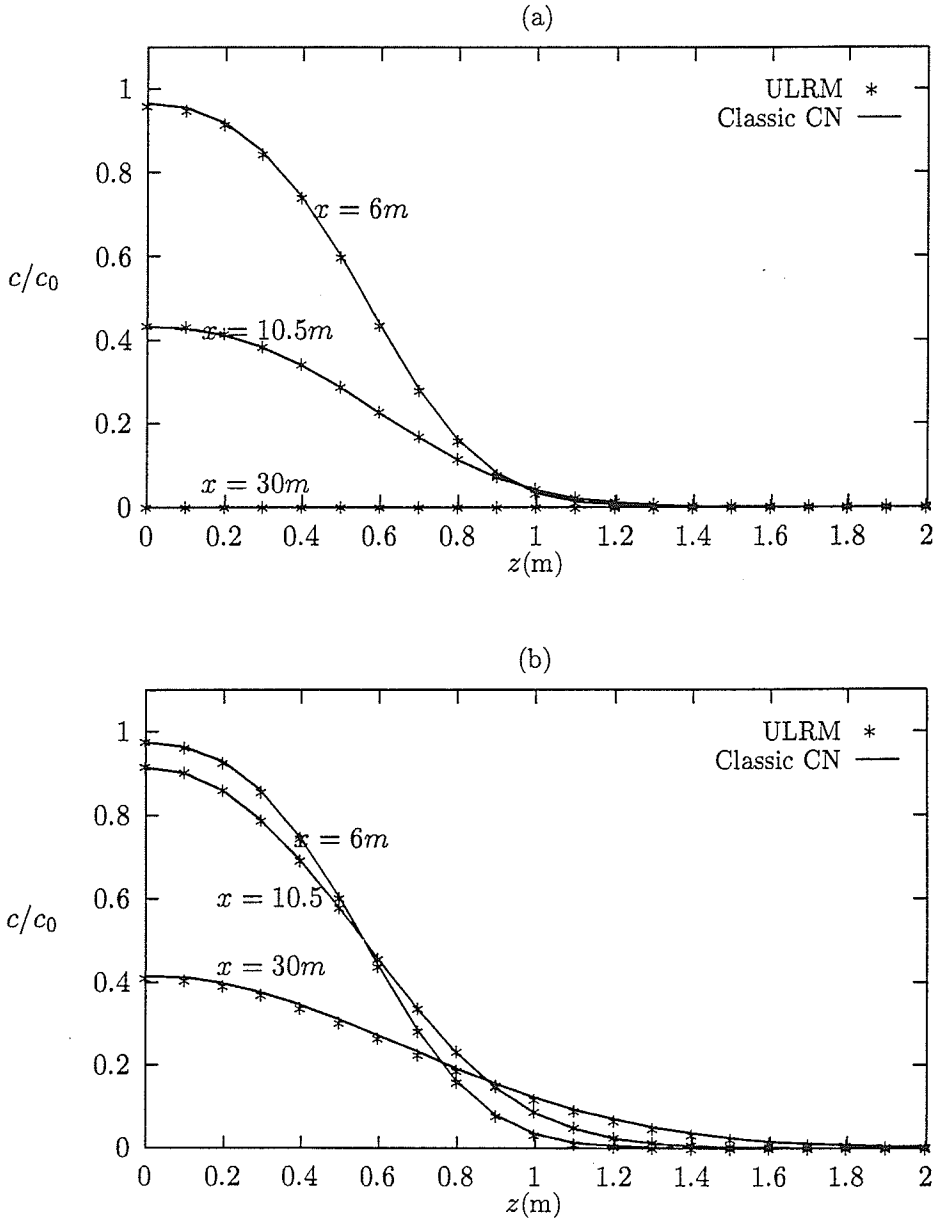

Figure 5.5: Comparison of the concentration solutions of equation (5.2) between the ULR method and the classic Crank-Nicolson (CN) method in transversal direction at different x coordinates and at (a) $t = 100$ days (b) $t = 300$ days. Domain size is $75m \times 2m$.

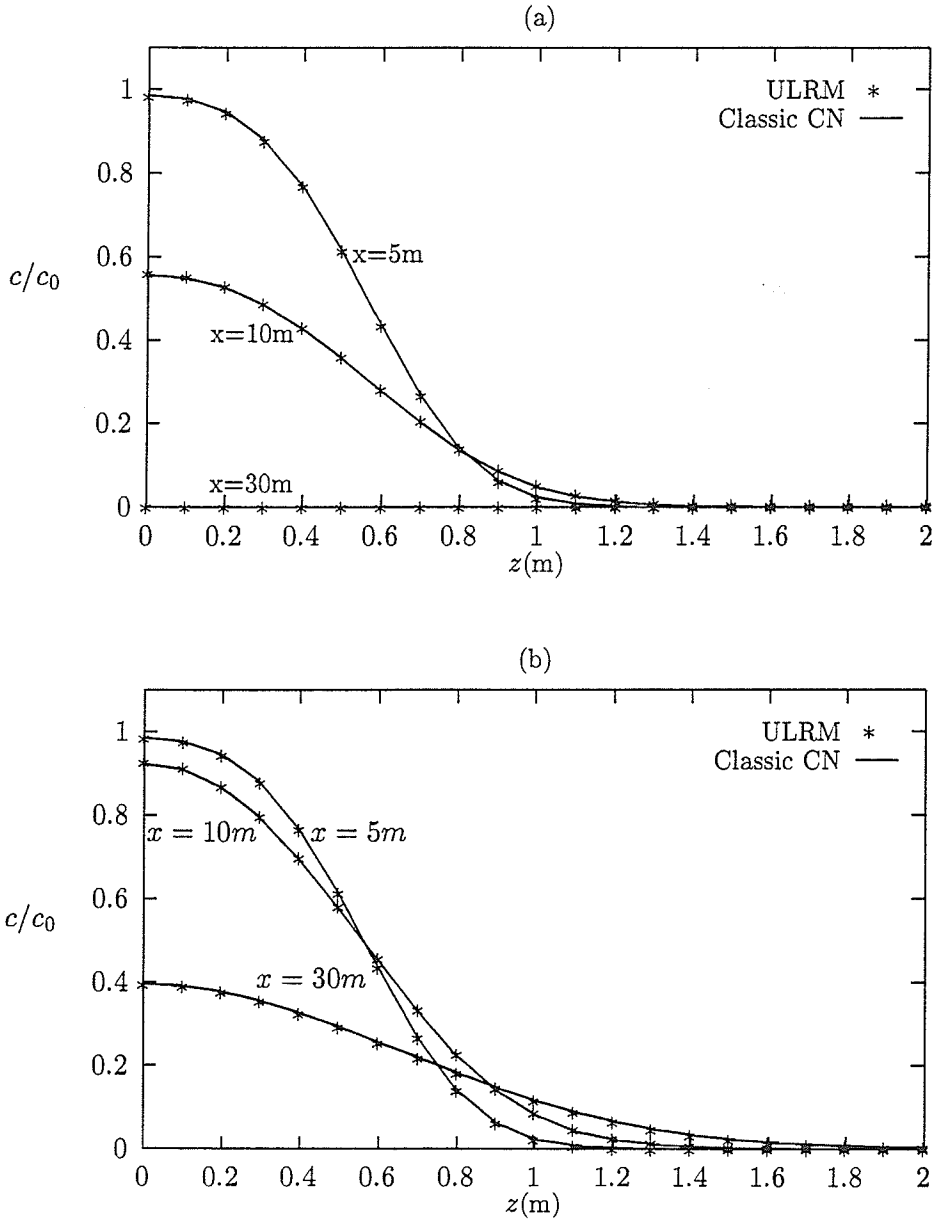

Figure 5.6: Comparison of the concentration solutions of equation (5.2) between the ULR method and the classic Crank-Nicolson (CN) method in transversal direction at different x coordinates and at (a) $t = 100$ days (b) $t = 300$ days. Domain size is $50m \times 2m$.

5.4 Conclusions

This concludes the presentation of numerical tests of the ULR method on one and two-dimensional examples. The above results show that the ULR method including the ET technique is effective in overcoming the problems of breakdown and instability in time. The comparison with respect to the classic Crank-Nicolson method shows that the ULR method is efficient, particularly for long term period prediction. The following are some specific remarks.

(1.) Setting of the pivot tolerance and the relative residual error bound tolerance

Tests show a balance of the pivot tolerance between breakdown and the accuracy of the approximate solutions. Higher pivot tolerance can result in better accuracy. However, lower tolerance can postpone or avoid breakdown, but obtain lower accuracy. Tests also show that the RMS errors are much smaller than the relative residual error bound. Based on these tests, the following choices for the parameters are appropriate and are used to next two practical applications: pivot tolerance $\epsilon = 10^{-8}$ and relative residual error bound tolerance $\delta = 10^{-3}$. However, much is still unknown about the relation between these tolerances and the size of problem (1.2). Nevertheless, it is recommanded that these settings be used for applications of the ULR method to the transport problems.

(2.) The grid Peclet number

Tests also show that it is important to chose an appropriate grid mesh (or the grid Peclet number) to balance the stability and the condition number. A coarse grid with a large grid Peclet number can produce an unstable semi-discretized

system (1.2), and consequently an unstable reduced system by the ULR method. On the other hand, the coarse grid will improve the condition number of the system (1.2). A detailed discussion on the relation between the grid size and the condition number in the FE method can be seen in [9].

(3.) Storage limitation

The ULR method needs extra storage to store the matrix $\mathbf{Q}_m$ to construct the Ritz vector, and matrices $\mathbf{Q}_m$ and $\mathbf{P}_m$ to reorthogonalize right and left Lanczos sequences (The SLR and AR methods have the same problem). This is one of the biggest disadvantages of all reduction methods. It is recommended that a sparse storage and a corresponding sparse solver be introduced ([37, 48, 70]) for large problems. Storage is no longer a limitation for the decay chain problem, and the conventional method requires more storage than the ULR method. Details is discussed in Chapter 7. After all, given the inexpensive nature RAM and new storage devices such as the digital video disc, storage is not considered a limitation.

In the next chapter, the application of the ULR method to the complex two-dimensional field problem is presented.

Chapter 6

Two-dimensional field study

6.1 Introduction

6.1.1 Conceptual hydrogeological model of the site

This example demonstrates the effectiveness of the ULR method in solving complex field problems. These applications focus on a conceptual model loosely based on the Whiteshell Research Area (WRA), which is shown on Figure 6.1. It is located on the Lac du Bonnet granite batholith in southeast Manitoba, Canada. A series of cross-section generic models, based on hypothetical situations, are designed and simulated. The geological features are similar to those in WRA, see [13], but are purposely kept simple in order to be of general use in understanding the ULR method being studied.

The groundwater flow equation representing the model for the steady-state saturated groundwater flow is simulated and solved. This produces a velocity distribution. The results from the flow equation are then used to solve the solute transport equation, i.e., the advection dispersion equation (4.2). The governing equation for steady-state saturated groundwater flow in an inhomogeneous

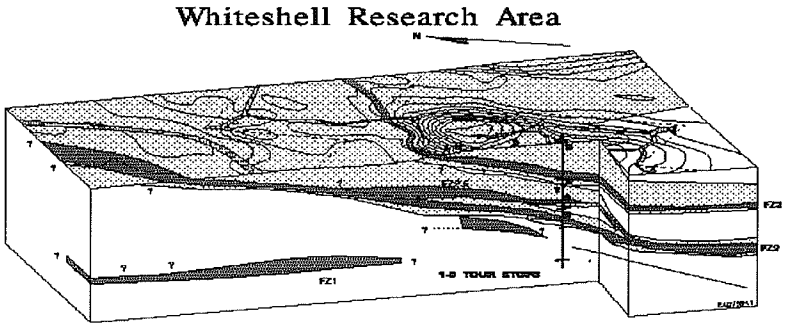

Figure 6.1: Three-dimensional block view of WRA

incompressible fluid in a cross-section is ([6])

$$(6.1)\quad \frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial x}\left(K_{xz}\frac{\partial h}{\partial z}\right)+\frac{\partial}{\partial z}\left(K_{zz}\frac{\partial h}{\partial z}\right)+\frac{\partial}{\partial z}\left(K_{zx}\frac{\partial h}{\partial x}\right)=0,$$

where h is the hydraulic head and K_{xx}, K_{xz}, K_{zx} and K_{zz} are hydraulic conductivities. The flow velocity (the coefficient of the solute equation (4.2)) can be obtained from Darcy's law

$$v_x=-\frac{K_{xx}}{\theta}\frac{\partial h}{\partial x}-\frac{K_{xz}}{\theta}\frac{\partial h}{\partial z}\qquad\text{and}\qquad v_z=-\frac{K_{zz}}{\theta}\frac{\partial h}{\partial z}-\frac{K_{zx}}{\theta}\frac{\partial h}{\partial x},$$

where θ is the effective porosity.

All model simulations are carried out in the same domain which is divided into five areas according to the different intensities of open fractures (Figure 6.2): Area 1 (grey granite) is a sparsely fractured rock, which is assumed to be isotropic with hydraulic conductivity 6.5×10^{-11} m/s in both directions. Area 3 (pink granite) is a moderately fractured rock with the hydraulic conductivity in the horizontal direction two orders higher than area 1, and in the vertical direction

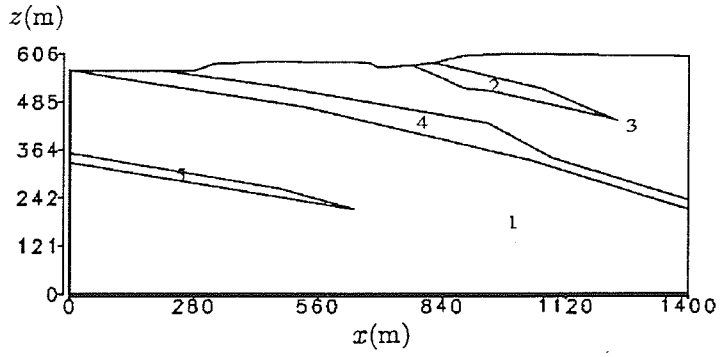

Figure 6.2: The cross-section model with five areas according to the different intensities of open fractures: Area 1, grey granite, is sparsely fractured rock. Area 3, pink granite, is moderately fractured rock. Areas 2, 4 and 5 are fracture zones, which are volumes of intensely fractured rock.

five times that value. Areas 2, 4 and 5 are fracture zones with the hydraulic conductivities in the horizontal direction two orders higher than in the moderately fractured rock horizontal direction, and half this value in the vertical direction. The hydraulic conductivities and corresponding effective porosities are listed in Table 6.1 except that $K_{xz} = K_{zx} = 0$. For the solute transport problem, the longitudinal dispersivity is set to 15 m and a transverse dispersivity is 1.5 m. The molecular diffusion coefficient is set at 0.15×10^{-8} m^2/s, see Table 6.2. All these setting are obtained from [13].

Rock mass area	Conductivity (m/s)		Effective porosity
	Horizontal	Vertical	
1	6.50D-11	6.50D-11	4.0D-3
2	6.50D-7	3.25D-7	1.0D-1
3	6.50D-9	3.25D-8	5.0D-3
4	6.50D-7	3.25D-7	1.0D-1
5	6.50D-7	3.25D-7	1.0D-1

Table 6.1: Conductivity and porosity value in five areas.

For these simulations, various boundary conditions are considered. For the fluid problem, two different boundary conditions are assumed. In the first case, the base of the model is considered to be impermeable because geologic evidence suggests no vertical flow at that depth. The top boundary of the model has prescribed hydraulic head values equal to the estimated water table elevations. The hydraulic head values on the left and the right side boundaries are specified equal to the corresponding value at the top point of that side. In the second simulation, both the base and the right side boundaries are assumed to be impermeable. On the left and the top boundaries, the hydraulic head is set to the same as in the first simulation. For the solute transport equation, four subcases based upon different source locations are considered. All of the sources are placed on the right boundary and they are at a depth of 160 m $< z <$ 180 m, 260 m $< z <$ 280 m, 360 m $< z <$ 380 m and 460 m $< z <$ 480 m, respectively. It is also assumed that there is no solute concentration gradients across the other boundaries.

6.1.2 Finite grid and Galerkin's finite element solver

Galerkin's finite element method ([30, 52, 69]) is used to discretize both the flow and solute transport equations, under the same domain grid mesh of 2567 non-uniform triangular elements. The grid is refined along the inner boundaries where the hydraulic head gradients are comparatively high and the vicinity of the source disposal sites where contaminant concentration gradients are expected to be comparatively high, see Figure 6.3.

AQUIFEM-1 [58], a versatile two-dimensional finite element code, is used to model the groundwater flow system. AQUIFEM-1 employs the Galerkin finite

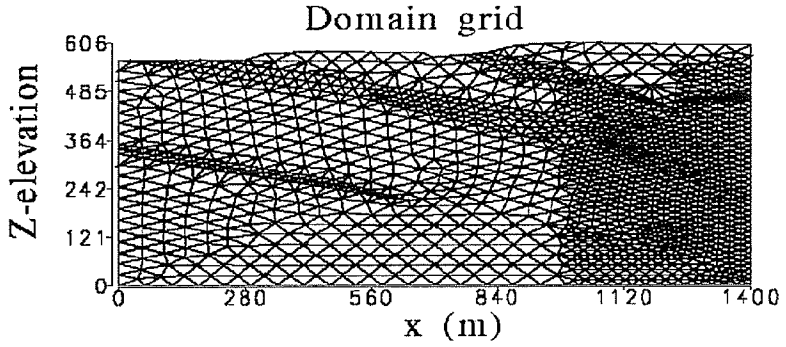

Figure 6.3: A view of the two-dimensional finite element mesh for the WRA cross-section model.

element technique with linear interpolation functions and triangular elements. Next, the ULR code is used for solute transport problem. The ULR code also employs the Galerkin finite element method to spatially discretize the equation and uses the ULR method to obtain an approximate solute concentration distribution. This code uses a six-point triangular mesh, consequently, three extra nodes per element have to be inserted and the nodes must be renumbered as to minimize the band width of resulting matrices. As the result, 5264 nodes are used for solute transport model.

6.1.3 Parameters for the ULR solver

The model parameters for the ULR solver are given in Table 6.2. Termination of the ULR process occurs if more than 100 Lanczos vectors are required to satisfy a relative residual error of 0.001. Of course, increasing the number of vectors can produce more accurate solutions, but with more computational cost. Hence there is a balance of accuracy with computation cost. Experience shows

Parameter	Value
Number of nodes	5264
Number of elements	2567
Time step Δt	1 year
Maximum reduced size	100
Relative residual error criterion	0.001
Breakdown pivot tolerance	10^{-8}
Longitudinal dispersivity	15 m
Transversal dispersivity	1.5 m
Molecular diffusion coefficient	0.15×10^{-9} m^2/s

Table 6.2: Model parameters for the ULR solver

that a maximum size of 100 is appropriate for a problem with an original size of approximately 5000 nodes. A detailed discussion about the relative residual error can be found in §3.4 and §5.4. Experience also shows the RMS error is much lower than the relative residual error. In this model, the relative residual error bound tolerance is set to 10^{-3}. The pivot tolerance is set to 10^{-8}. As it was pointed out in §5.2, a larger value of pivot tolerance can produce a more accurate solution, but there will be more breakdowns. In practice, it is often important to choose an appropriate pivot tolerance to balance accuracy with breakdowns.

The following sections will detail the performance and accuracy of these simulations.

6.2 Case 1: bottom boundary is impermeable

6.2.1 Performance in solving the flow equation

In this case, the bottom boundary is impermeable, i.e.,

$$\frac{\partial h}{\partial z} = 0, \qquad \text{on} \quad z = 0.$$

where $h(x, z)$ is the hydraulic head (the solution of the flow equation (6.1)). Heads on the top boundary are equal to the water elevation, i.e.,

$$h(x, z) = z, \qquad (x, z) \in \text{top boundary}.$$

Heads on the left and right boundaries are equal to the corresponding values at the top point of that side, i.e.,

$$h(0, z) = h_1, \quad \text{and} \quad h(1400, z) = h_2,$$

where $h_1 = 564$ m and $h_2 = 600$ m, the water elevations of the top points of these boundaries respectively.

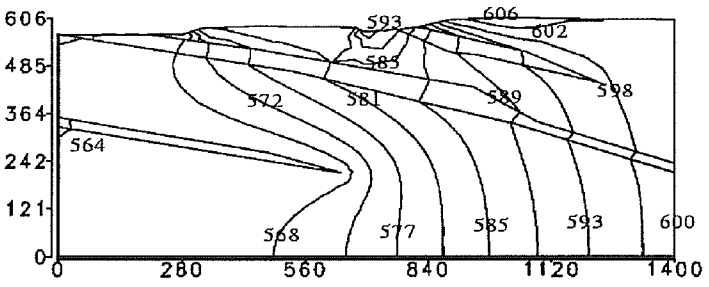

Figure 6.4: Hydraulic head contour for case 1.

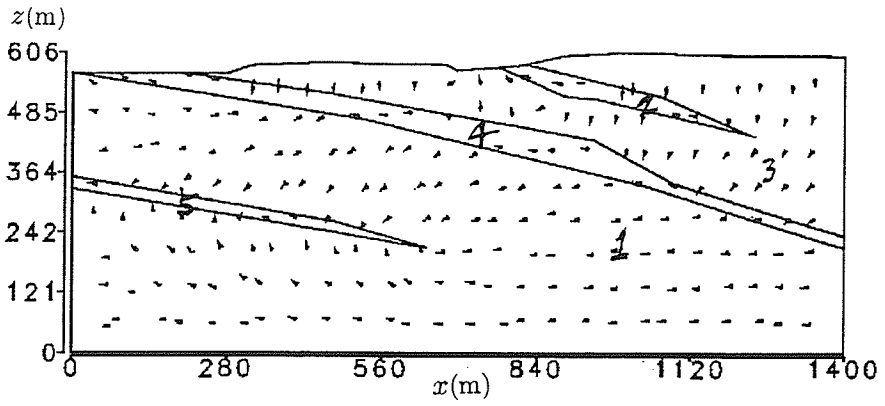

Figure 6.5: Velocity vector distribution for case 1.

The computed flow field can be seen in Figure 6.4 and Figure 6.5. Figure 6.4 shows the hydraulic head contours and Figure 6.5 is the corresponding velocity vector distribution. It can be seen from Figure 6.5 that the groundwater flow is essentially downward in area 3, and horizontal in area 1 (see also Figure 6.2). Representative average velocities in the x and z directions are shown in Table 6.3 for regions shown in Figure 6.6. Here, a weighted area average is used to compute average velocities for each region. If there are n elements in a region with area A, v_k represents the velocity v_x or v_y in the kth element and A_k is the area of

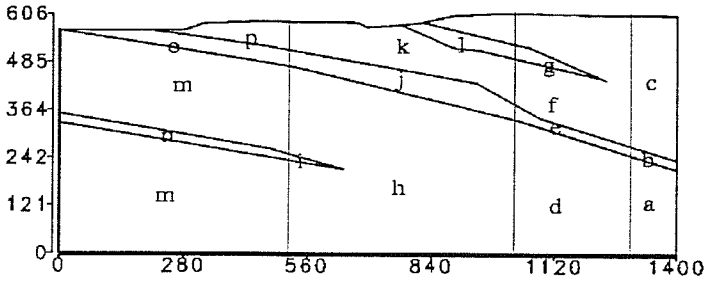

Figure 6.6: Regions for the average velocity and grid Peclet numbers.

Region	Average velocity (m/a)	
	x	z
a	-0.16D-1	-0.64D-3
b	-0.44D+1	+0.13D+1
c	-0.51D+0	-0.11D+1
d	-0.18D-1	-0.15D-2
e	-0.50D+1	+0.18D+1
f	-0.10D+1	-0.78D+1
g	-0.29D+1	+0.20D+0
h	-0.21D-1	-0.49D-2
i	-0.31D-1	+0.73D-2
j	-0.35D+1	+0.86D+0
k	-0.12D+1	-0.61D+1
l	-0.11D+2	+0.21D+1
m	-0.58D-2	-0.16D-2
n	-0.29D-1	+0.45D-2
o	-0.72D+1	+0.14D+1
p	-0.27D+1	-0.26D+2

Table 6.3: Average velocity in the x and z directions for regions specified by Figure 6.6.

the k'th element, then the corresponding average velocity $\bar{v}$ on the whole region is defined as

$$\bar{v} = \frac{1}{A}\sum_{k=1}^{n} v_k A_k.$$

In Figure 6.6, the domain is vertically divided into four blocks and regions are labeled alphabetically from `a` to `p`. In the block with regions `a-c`, 1300.0 m $< x <$ 1400.0 m, in the block with regions `d-g`, 1000.0 m $< x <$ 1300.0 m, in the block with regions `h-l`, 500.0 m $< x <$ 1000.0 m, in the block with regions `m-p`, 0.0 m $< x <$ 500.0 m.

The regions `a, b, c, d, e, f` shown in Figure 6.6, which are near to the con-

taminant source sites, are of special interest. The average groundwater velocities in regions b and e are respectively -4.4, -5.0 (m/a) in the x direction and 1.3, 1.8 (m/a) in the z direction. In regions a and d, they are -0.016, -0.018 (m/a) in the x direction and -0.00064, -0.0015 (m/a) in the z direction, and in regions c and f, they are -0.51, -1.0 (m/a) in the x direction, -1.1 and -7.8 in the z direction. The average velocities in regions a and d are much lower than those in the other regions. In the next section, it can be seen, from Figure 6.7 to 6.10, region a is a pottentially good area to place a contaminant source.

6.2.2 Performance in solving the transport equation

The grid Peclet number is a good indicator for the onset of numerical oscillations in the Galerkin finite element solution, or the stability of the semi-discretized system (1.2). Table 6.4 lists these numbers for each region. The grid Peclet number is defined as

$$Pe_x = \frac{v_x \Delta x}{D_{xx}} \quad \text{and} \quad Pe_z = \frac{v_z \Delta z}{D_{zz}},$$

where v_x, v_z are element velocities and D_{xx} and D_{zz} are coefficients of hydrodynamic dispersion, (given in (1.1) and (5.2)). Δx and Δz are defined as the maximum length in the x and z directions for the element, respectively.

For one-dimensional problems with linear basis functions on a uniform mesh, Carey and Sepehrnoori [11] proved that the discretization of an advection dispersion equation is numerically stable when the grid Peclet number is less than 2. For higher dimensions with non-uniform irregular meshes, the proper grid Peclet numbers are difficult to ascertain. Huyakorn and Pinder in [30] point out that "acceptable numerical solutions with very mild oscillations are achieved even when

Region	grid Peclet number	
	x	z
a	0.14D+1	0.11D+0
b	0.16D+1	0.11D+0
c	0.20D+1	0.92D+0
d	0.15D+1	0.26D+0
e	0.17D+1	0.13D+1
f	0.24D+1	0.10D+1
g	0.17D+1	0.66D+1
h	0.31D+1	0.12D+1
i	0.15D+1	0.47D+0
j	0.19D+1	0.15D+1
k	0.30D+1	0.18D+1
l	0.18D+1	0.84D+0
m	0.19D+1	0.79D+0
n	0.16D+1	0.56D+0
o	0.16D+1	0.19D+1
p	0.18D+1	0.12D+1

Table 6.4: Average grid Peclet number in the x and z directions for regions specified by Figure 6.6.

the grid Peclet number is as high as 10". Nevertheless, the numerical stability of the discretization of solute transport equation is sensitive to the grid Peclet number. As the number increases, numerical results become unstable which is considered to be an artifact of the discretization. In [45], it is pointed out that "the constraints of the grid Peclet number may be relaxed in the regions where the mass front exhibits small gradients". In regions where the velocity is very low and which are not expected to cover the transport path, a grid can be coarser and the grid Peclet numbers can be larger. Furthermore, as pointed in section §5.4, there is a balance of the grid number, i.e., under the constraints of the stability, coarser grid can improve the condition number of resulting matrices. More com-

ments on the grid Peclet number can be found in [45]. In our simulations, the highest average grid Peclet number is 3.1 in x in region h and 6.6 in z in region g. The result of the performance shows the grid is appropriate.

Average grid Courant number		
Area	x	z
1	0.46D-3	0.18D-3
2	0.28D+0	0.10D+0
3	0.45D-1	0.42D+0
4	0.17D+0	0.82D-1
5	0.11D-2	0.54D-3
Region	x	z
a	0.67D-3	0.47D-4
b	0.19D+0	0.11D+0
c	0.27D-1	0.84D-1
d	0.68D-3	0.11D-3
e	0.21D+0	0.15D+0
f	0.51D-1	0.33D+0
g	0.13D+0	0.41D-1
h	0.44D-3	0.37D-3
i	0.14D-2	0.11D-2
j	0.13D+0	0.62D-1
k	0.61D-1	0.78D+0
l	0.43D+0	0.17D+0
m	0.12D-3	0.15D-3
n	0.97D-3	0.41D-3
o	0.23D+1	0.66D-1
p	0.55D-1	0.30D+1

Table 6.5: Average Courant number of each area specified by Figure 6.2 and for regions specified by Figure 6.6.

There is another criterion for the time step selection in the Crank-Nicolson

method. This is the grid Courant number, which is defined as

$$Cr_x = \frac{v_x \Delta t}{\Delta x} \qquad \text{and} \qquad Cr_z = \frac{v_z \Delta t}{\Delta z}.$$

The time step size Δt should be selected so that the grid Courant numbers are less than or equal to 1, see [30, 10]. In this model, the average grid Courant numbers for each area and for each region shown on Figure 6.2 and Figure 6.6 are listed in Table 6.5, respectively. These constraints are essentially satisfied.

Various subcases with different boundary conditions are tested. Denote $\Omega_1 = \{x = 1400,\ 460\ m < z < 480\ m\}$, $\Omega_2 = \{x = 1400,\ 360\ m < z < 380\ m\}$, $\Omega_3 = \{x = 1400,\ 260\ m < z < 280\ m\}$, and $\Omega_4 = \{x = 1400,\ 160\ m < z < 180\ m\}$. Then for subcase i, $i = 1, 2, 3, 4$, the boundary conditions are defined as

$$\begin{cases} C(x,z) = 1, & (x,z) \in \Omega_i, \\ \dfrac{\partial C}{\partial z} = 0 & (x,z) \text{ on the top and bottom boundaries}, \\ \dfrac{\partial C}{\partial x} = 0 & (x,z) \text{ on the left boundary}, \\ \dfrac{\partial C}{\partial x} = 0, & (x,z) \text{ on the right boundary, but } \notin \Omega_i. \end{cases}$$

Figures 6.7, 6.8, 6.9 and 6.10 show concentrations evaluated at $t = 100$, 500, 1000, and 3000 years for each of the subcases. All plumes are obtained using the ULR method with the setting in Table 6.2. In Figures 6.7, 6.8 and 6.9, where the source sites are in region c, the zone of high concentrations moves downward until entering the intensely fractured rock (area 4). The solute then travels along the intensely fractured rock (area 4) with a higher speed. The leading edge of the

plume eventually exits area 4 and spreads in a diverging pattern. In Figure 6.10, the deposit site is in the region a where the groundwater flows laterally at the velocities which are much slower than that in region b. At $t = 3000$ years, the peak concentration has migrated only a few meters and the leading edge of plume has entered the intensely fractured area, area 4. It is obvious that area 1 is a potentially safer place for contaminant source deposit as the lowest velocities are present there. Area 4 is the least suitable where the velocities are higher, and contaminant solute eventually enters and travels along that area. The best source site is in the area 1 and far from area 4.

6.2.3 Behavior of the ULR method

Table 6.6 gives the behavior of the ULR method applied to the subcases. It lists the size of reduced system (Sr), the relative residual error (δ) calculated at the termination step of the ULR process, the step number at which the first breakdown occurs (N_s), the number of eigenvalues with negative real part (# Eg), the execution time (or system time) comparison, the RMS error comparison and maximum error comparison. All comparisons are with respect to the classic Crank-Nicolson solver.

The first breakdowns occur between 40 and 50 recursive steps. Two subcases have eigenvalues with negative real part , and ET stabilization technique is applied in these cases. The results of the comparisons are very encouraging. The lowest RMS error is of the order of 10^{-8} at 3000 years, while the highest error is of 10^{-3} at 100 years. The highest maximum error is the order of 10^{-2} at 100 years, while the lowest maximum error is the order of 10^{-7} at 3000 years. These results

Behavior	Figure 6.7	Figure 6.8	Figure 6.9	Figure 6.10
S_r	100	100	100	100
δ	0.95D-1	0.16D+00	0.29D+00	0.66D-02
N_s	48	47	46	43
# Eg	2	Non	1	Non
RMS Error				
100 years	0.470D-03	0.998D-03	0.141D-02	0.193D-04
500 years	0.173D-04	0.300D-04	0.502D-04	0.708D-05
1000 years	0.493D-05	0.188D-05	0.730D-05	0.193D-05
3000 years	0.330D-06	0.330D-07	0.181D-06	0.148D-06
Maximum error				
100 years	0.957D-02	0.173D-01	0.543D-01	0.447D-03
500 years	0.216D-03	0.536D-03	0.481D-03	0.206D-03
1000 years	0.862D-04	0.217D-04	0.680D-04	0.362D-04
3000 years	0.613D-05	0.333D-06	0.274D-05	0.196D-05
Execution time (hour:minute:second)				
ULR method	0:19:56	0:25:14	0:23:01	0:20:48
Classic CN	6:40:32	6:42:58	7:49:23	6:49:38

Table 6.6: Behavior of the ULR method and comparison with respect to the classic Crank-Nicolson solver for case 1. S_r denotes the size of the reduced system, δ is the relative residual error calculated at the termination step of the ULR process, N_s stands for the step number at which the first breakdown occurs, # Eg is the number of eigenvalues with negative real part. Classic CN stands for classic Crank-Nicolson method.

presented in Table 6.6 suggest that ULR method requires only about 5% execution time of the classic Crank-Nicolson solver. The advantage of ULR method in the computation time saving is promising.

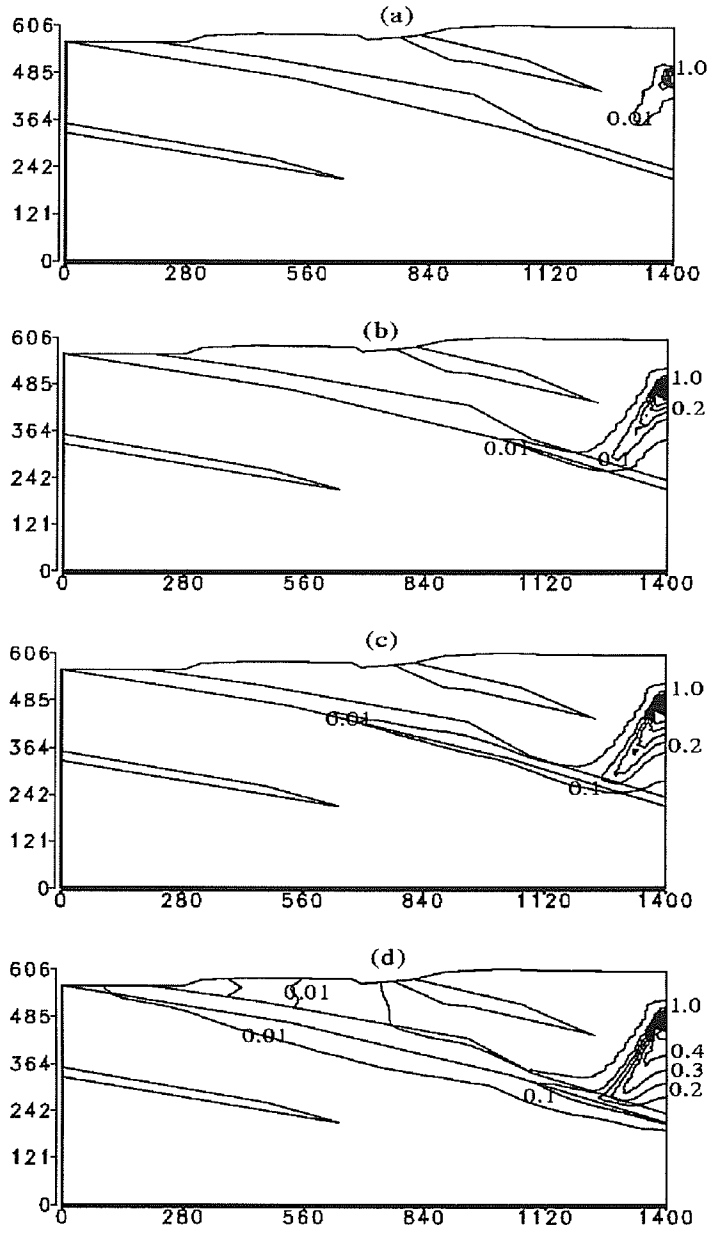

Figure 6.7: Contaminant plumes for case 1, subcase 1, (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

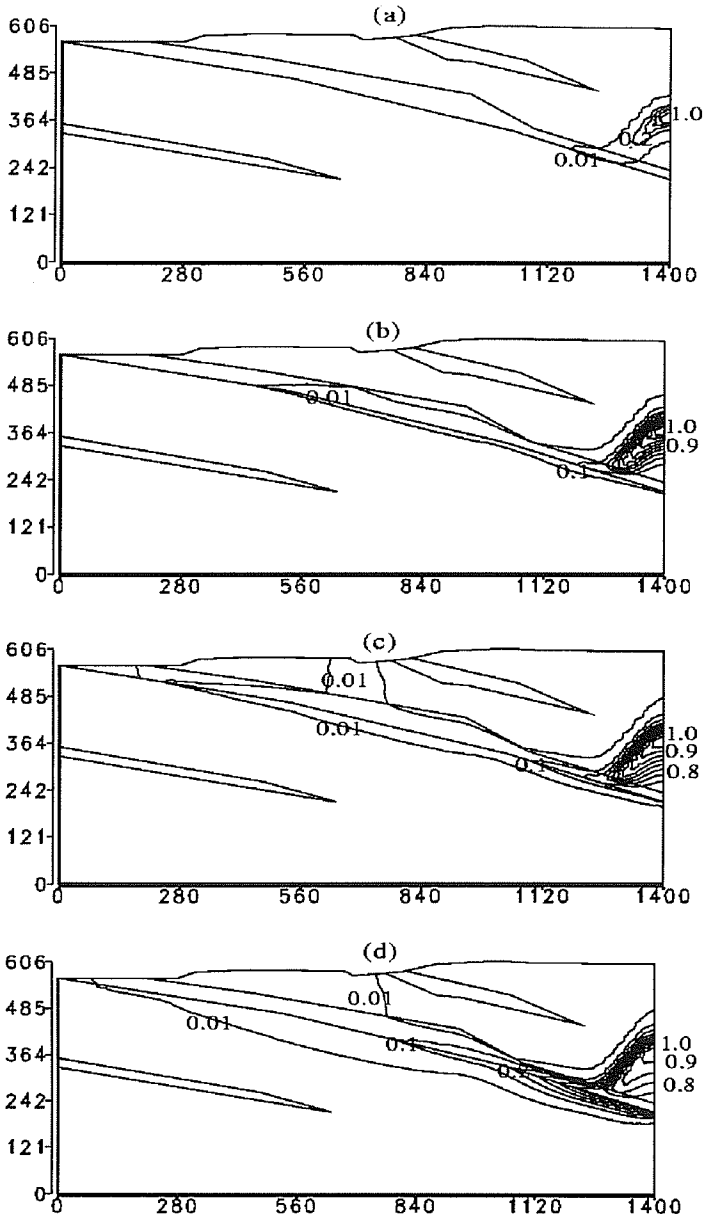

Figure 6.8: Contaminant plumes for case 1, subcase 2, (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

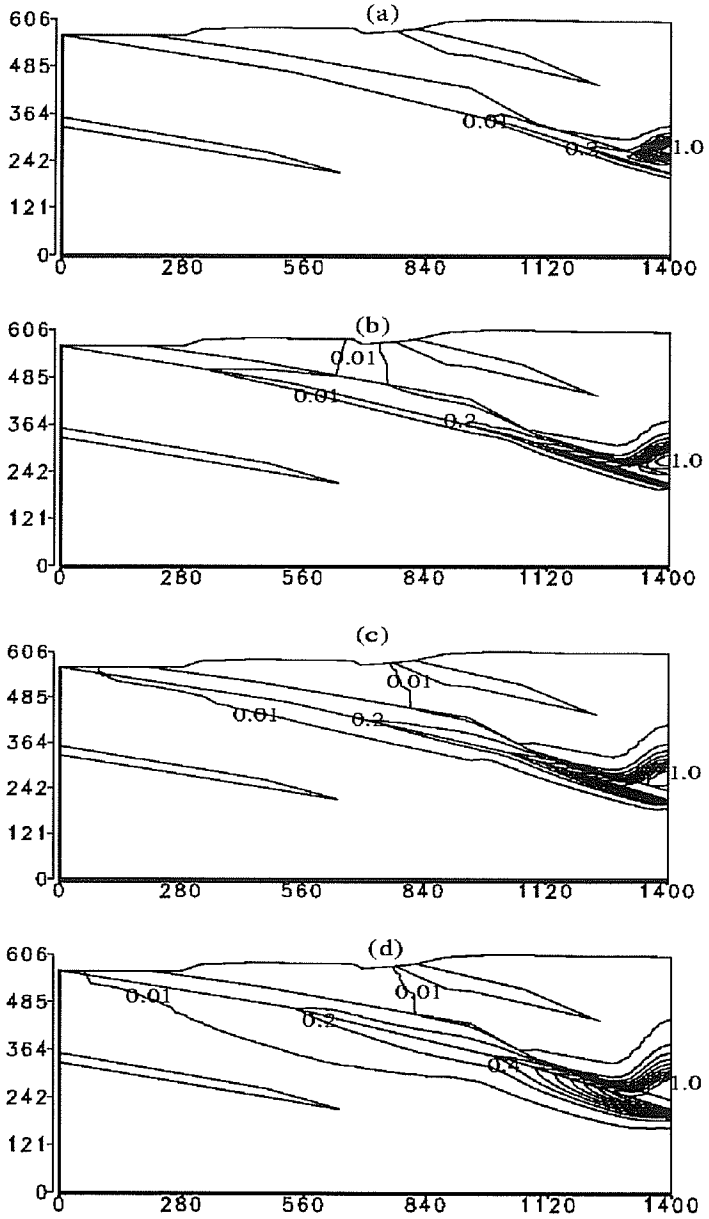

Figure 6.9: Contaminant plumes for case 1, subcase 3 (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

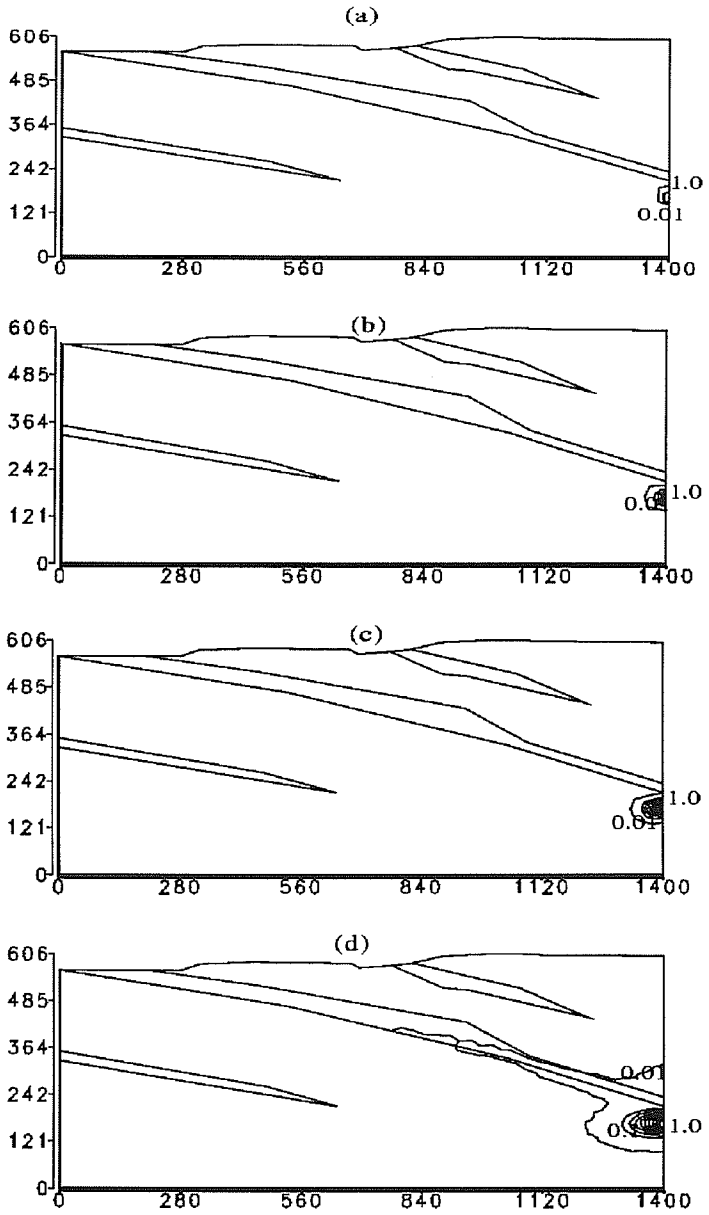

Figure 6.10: Contaminant plumes for case 1, subcase 4 (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

6.3 Case 2: bottom and right boundaries are impermeable

6.3.1 Performance in solving the flow equation

In this case, the bottom and right boundaries are impermeable, and the prefixed heads on the top and left boundaries are the same as in case 1. i.e.,

$$\begin{cases} \dfrac{\partial h}{\partial z} = 0, & \text{on} \quad z = 0 \text{ or } x = 1400, \\ h(x,z) = z, & (x,z) \in \text{top boundary}, \\ h(0,z) = 564. \end{cases}$$

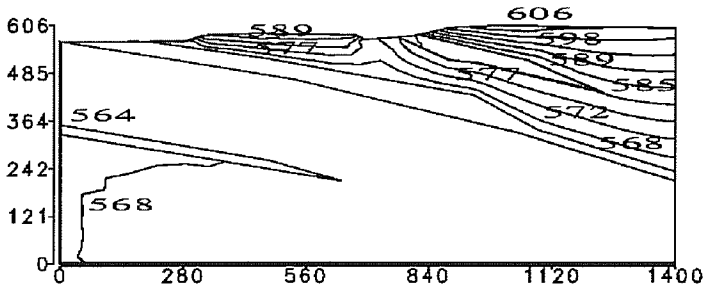

Figure 6.11: Hydaulic head contour for case 2.

Figure 6.11 and Figure 6.12 show the hydraulic head contour and the corresponding velocity vector distribution. It can be seen from Figure 6.11 that the gradient of the hydraulic head in area 1 (see Figure 6.2) is smooth. It is expected that in area 1 the velocities are low. From Figure 6.12, it can also be seen that the groundwater in area 3 flows downward and enters area 4. It eventually exits from area 3 and enters area 1. In area 1, the groundwater flows essentially laterally

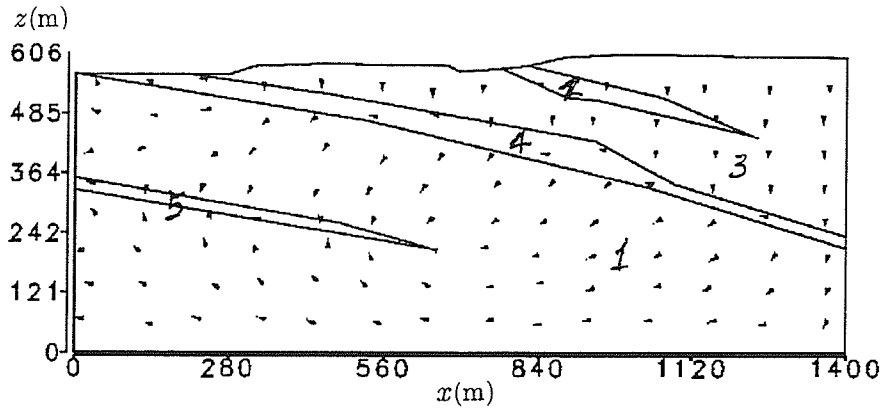

Figure 6.12: Velocity vector distribution for case 2.

and eventually enters area 5 or exits across the left boundary. Table 6.7 gives the quantitative details of velocities and the grid Peclet number for each region.

In regions a and d (see Figure 6.6), the velocities are of the order of 10^{-6}. In regions c and f, velocities are more than four orders higher than in region a and d, and those in vertical direction are at least ten times higher than those in horizontal direction. Even in the intensely fracture rock, region b and e, the velocities are as low as -0.013, -0.080 m/a in the x direction and 0.001 and 0.026 m/a in the z direction. Because of the low velocities, the constraints of grid Peclet number and Courant number are easily satisfied as seen in Table 6.7 and 6.8.

6.3.2 Performance in solving the transport equation

With the parameter settings by listed in Table 6.2 and the same source locations tests as in Case 1, four subcases are presented. The solute migrations for each subcase are depicted in Figures 6.13, 6.14, 6.15 and 6.16. When the source locations are chosen in region c (see Figures 6.13, 6.14 and 6.15) the zone of peak concentration moves slowly downward along the fluid flow and enters area 4. The

Region	Average velocity (m/a)		Average grid Peclet number	
	x	z	x	z
a	-0.88D-6	-0.19D-5	0.45D-3	0.51D-3
b	-0.13D-1	+0.11D-2	0.11D+1	0.31D+0
c	-0.30D-2	-0.16D+0	0.31D+0	0.89D+0
d	-0.37D-5	-0.20D-5	0.21D-2	0.57D-3
e	-0.80D-1	+0.26D-1	0.16D+1	0.11D+1
f	-0.11D-1	-0.22D+0	0.90D+0	0.95D+0
g	-0.24D-1	-0.73D-2	0.15D+1	0.61D+0
h	-0.84D-5	-0.51D-5	0.91D-2	0.38D-2
i	-0.19D-4	+0.44D-5	0.11D-1	0.84D-3
j	-0.13D+0	+0.23D-1	0.18D+1	0.11D+1
k	-0.12D-1	-0.31D+0	0.15D+1	0.17D+1
l	-0.14D+1	-0.15D+0	0.18D+1	0.81D+0
m	-0.21D-5	-0.17D-5	0.22D-2	0.15D-2
n	-0.16D-4	+0.28D-5	0.90D-2	0.73D-3
o	-0.29D+0	+0.59D-1	0.14D+1	0.12D+1
p	-0.16D-4	+0.28D-5	0.90D-2	0.73D-3

Table 6.7: Average velocity and grid Peclet number in the x and z directions for regions specified by Figure 6.6 for case 2.

solute merely travels a short distance in area 4 rock unit and then exits into area 1. Figure 6.16 shows the plume when the source position is placed in region a where the velocities are very low. So, the transport in this region is diffusion dominant (note that, in general, equation (4.2) is not diffusion dominant, because the velocities in the other regions are not low enough to be ignored). After 3000 years, the solute has only traveled a few miles. The migration patterns can be seen clearly in Figure 6.17 which depict the plumes of four subcases at one million years. It can be seen that the peak concentration ($0.2 \sim 1.0$) of each subcase has traveled a few miles, in the vicinity of the deposit site even after one million years

while the scattered solute (concentration $0.001 \sim 0.2$) disperses in a wide area in area 1. In Figure 6.17 (a), because the deposit site is near the surface, the solute eventually exits from the ground surface.

Average grid Courant number		
Area	x	z
1	0.11D-6	0.17D-6
2	0.32D-1	0.23D-1
3	0.37D-3	0.14D-1
4	0.54D-2	0.20D-2
5	0.66D-6	0.34D-6
Region	x	z
a	0.36D-7	0.14D-7
b	0.57D-3	0.19D-3
c	0.16D-3	0.12D-1
d	0.14D-6	0.15D-6
e	0.32D-2	0.21D-2
f	0.46D-3	0.14D-1
g	0.11D-2	0.69D-3
h	0.17D-6	0.30D-6
i	0.89D-6	0.66D-6
j	0.48D-2	0.18D-2
k	0.51D-3	0.15D-1
l	0.63D-1	0.45D-1
m	0.46D-7	0.10D-6
n	0.61D-6	0.26D-6
o	0.93D-2	0.28D-2
p	0.34D-3	0.25D-1

Table 6.8: Average Courant number for areas specified by Figure 6.2 and for regions specified by Figure 6.6 for case 2.

6.3.3 Behavior of the ULR method

Table 6.9 lists the behavior and error comparisons for the ULR method in these

subcases. There are no breakdowns and no eigenvalues with negative real part in the reduced system in the subcases depicted in Figures 6.14, 6.15 and 6.16. The example of Figure 6.16, where the velocities in the surrounding of the source site are of the order of 10^{-6} (region a), the size of reduced system is only 69. The highest RMS error is the order of 10^{-4}, the lowest is 10^{-6}. The highest maximum error is the order of 10^{-2}, and lowest is the order of 10^{-5}. A great execution time saving can also be obtained. Like case 1, the execution time of ULR method for time of 3000 years is only about 5% of that in the classic Crank-Nicolson solver. For time of one million years, the largest execution time for subcases (b), (c) and (d) is only 39 seconds, while (a) needs 7 hours and 54 seconds. This is because that, in subcases (b), (c) and (d), no breakdown occurs during the ULR process and there is no eigenvalues with negative real part in the reduced system. Thus the reduced system is tridiagonal and the time-stepping solver is very economical. Particularly, in the subcase (d), the reduced size is only 69 and the saving of the execution time is the greatest. The comparisons with the classic Crank-Nicolson solver were not made, but according to the comparison for 3000 years, it is expected that the classic Crank-Nicolson solver for one million years would require 140 hours CPU time. These results sufficiently demonstrate the advantage of the ULR method.

Behavior	Figure 6.13	Figure 6.14	Figure 6.15	Figure 6.16
Sr	100	100	100	69
δ	0.70D-02	0.59D-01	0.28D-01	0.30D-3
Bk No.	30	Non	Non	Non
# Eg	3	Non	Non	Non
RMS error				
100 years	0.616D-03	0.883D-03	0.296D-03	0.318D-04
500 years	0.169D-03	0.132D-03	0.563D-04	0.125D-04
1000 years	0.592D-04	0.655D-04	0.171D-04	0.152D-04
3000 years	0.304D-05	0.515D-04	0.137D-05	0.181D-05
Maximum error				
100 years	0.760D-02	0.322D-01	0.103D-01	0.666D-03
500 years	0.363D-02	0.416D-02	0.254D-02	0.438D-03
1000 years	0.911D-03	0.226D-02	0.749D-03	0.292D-03
3000 years	0.312D-04	0.270D-03	0.532D-04	0.435D-04
Execution time at 3000 years(hour: minute: second)				
ULR method	0:29:08	0:17:46	0:21:59	0:17:10
Classic CN	8:33:17	7:02:05	7:13:42	6:43:14
Execution time at one million years (Figure 6.17)				
	(a)	(b)	(c)	(d)
ULR method	7:54:56	0:38:55	0:39:30	0:27:30

Table 6.9: Behavior of the ULR method and comparison with respect to the classic Crank-Nicolson solver for case 2. Symbols, see Table 6.6.

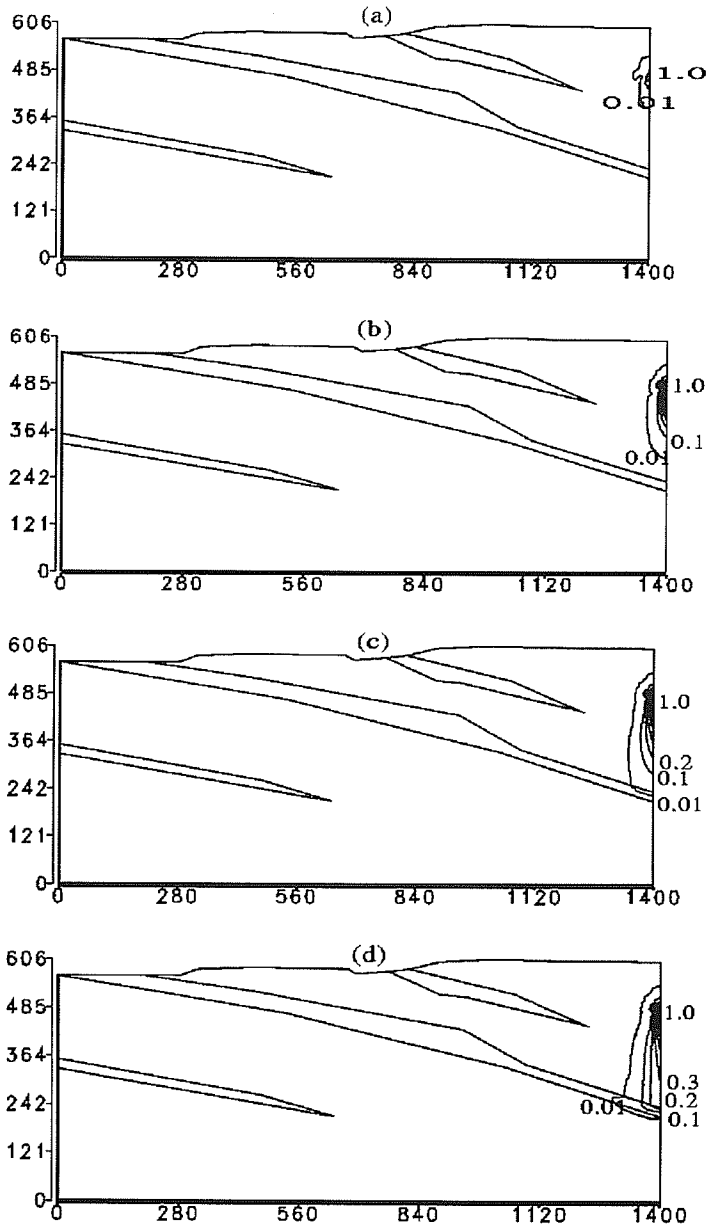

Figure 6.13: Contaminant plumes for case 2, subcase 1, (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

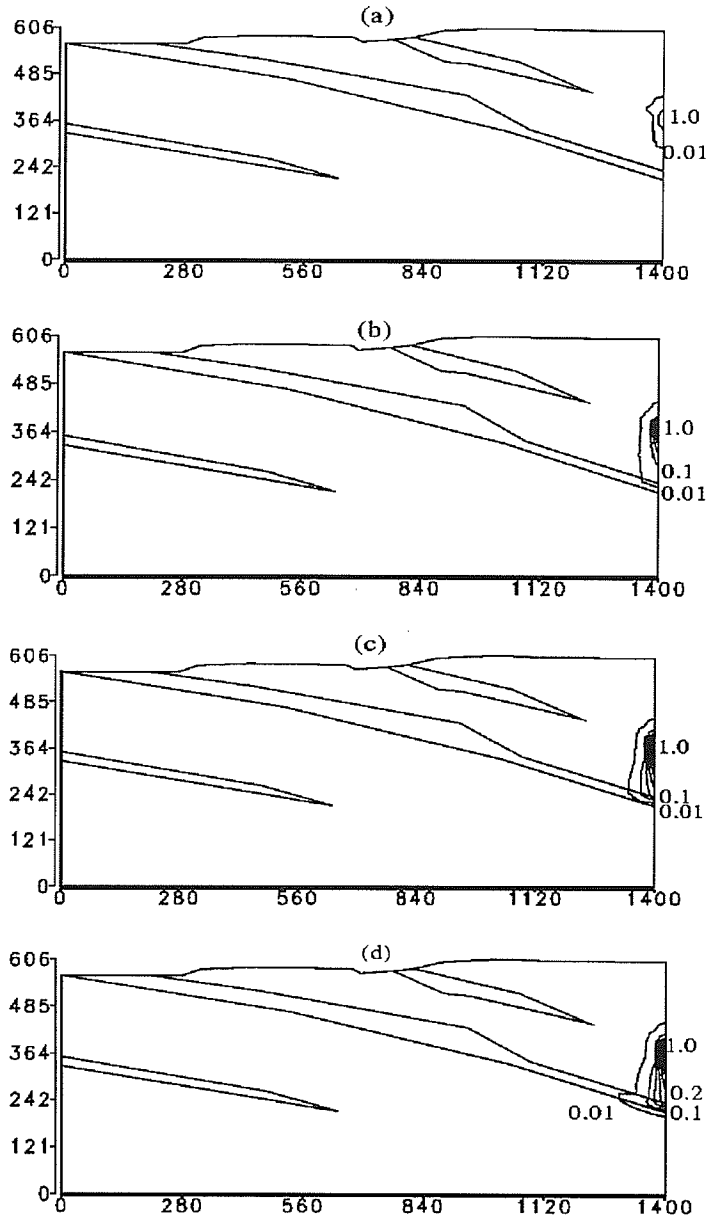

Figure 6.14: Contaminant plumes for case 2, subcase 2, (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

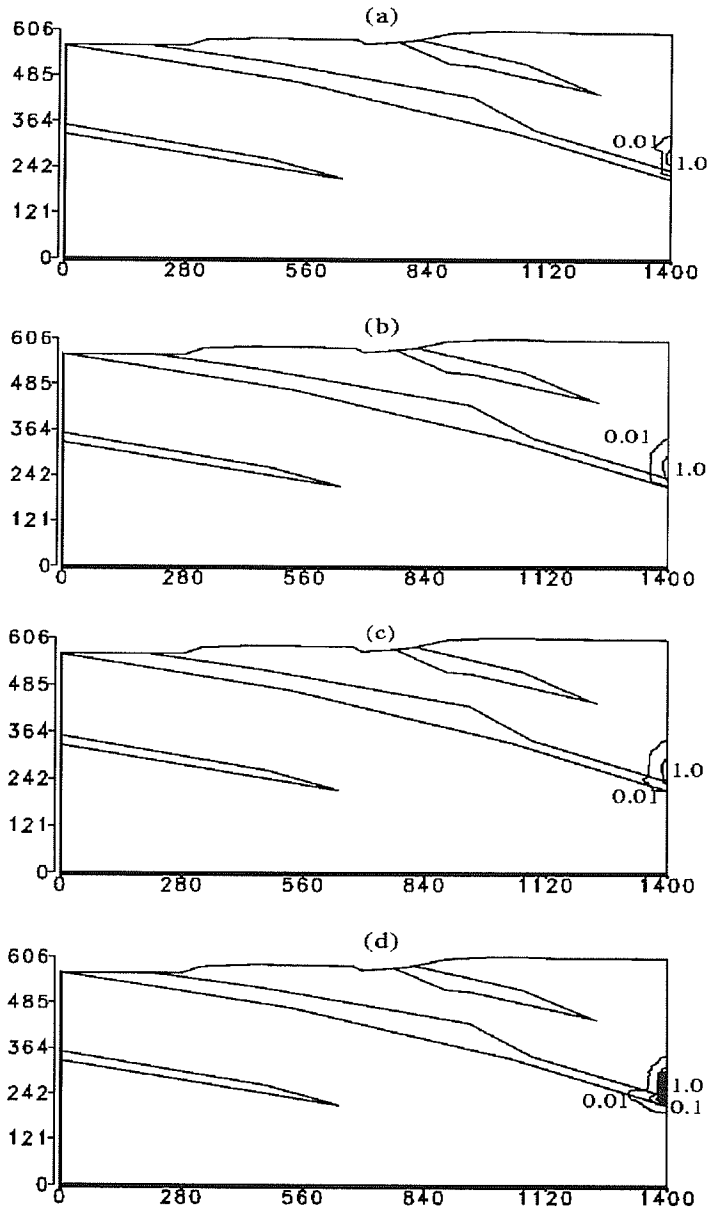

Figure 6.15: Contaminant plumes for case 2, subcase 3, (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

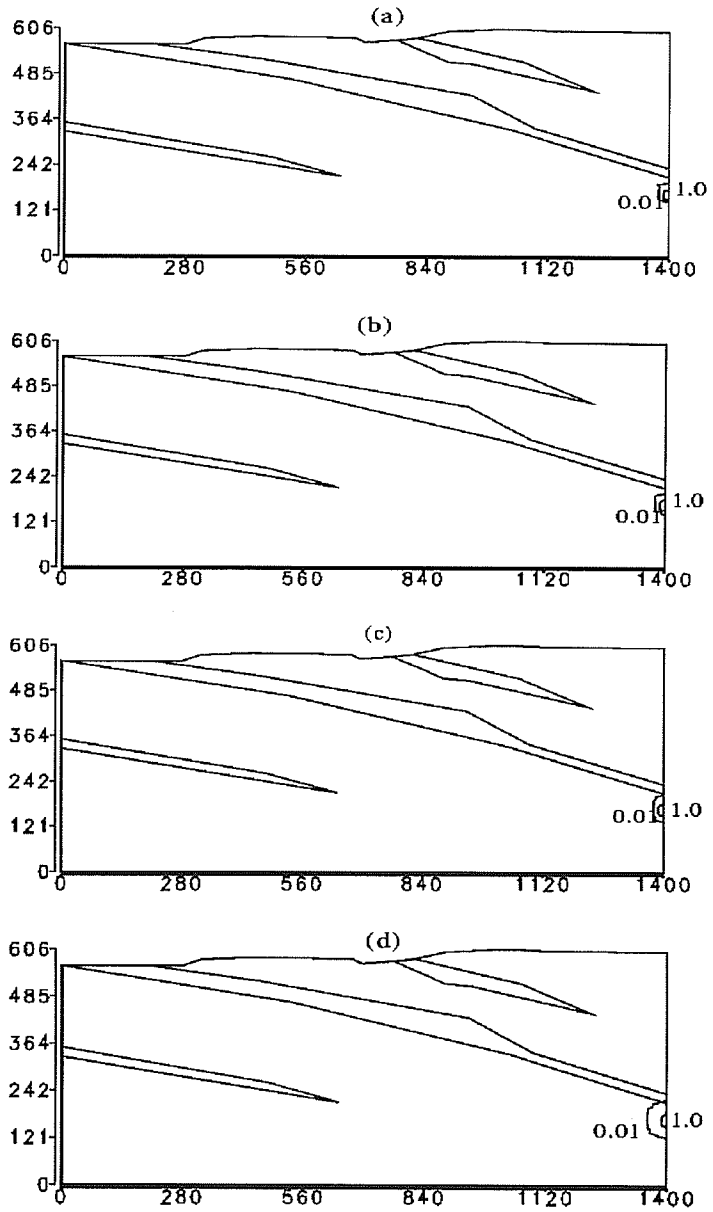

Figure 6.16: Contaminant plumes for case 2, subcase 4, (a) t=100 years (b) t=500 years (c) t=1000 years (d) t=3000 years.

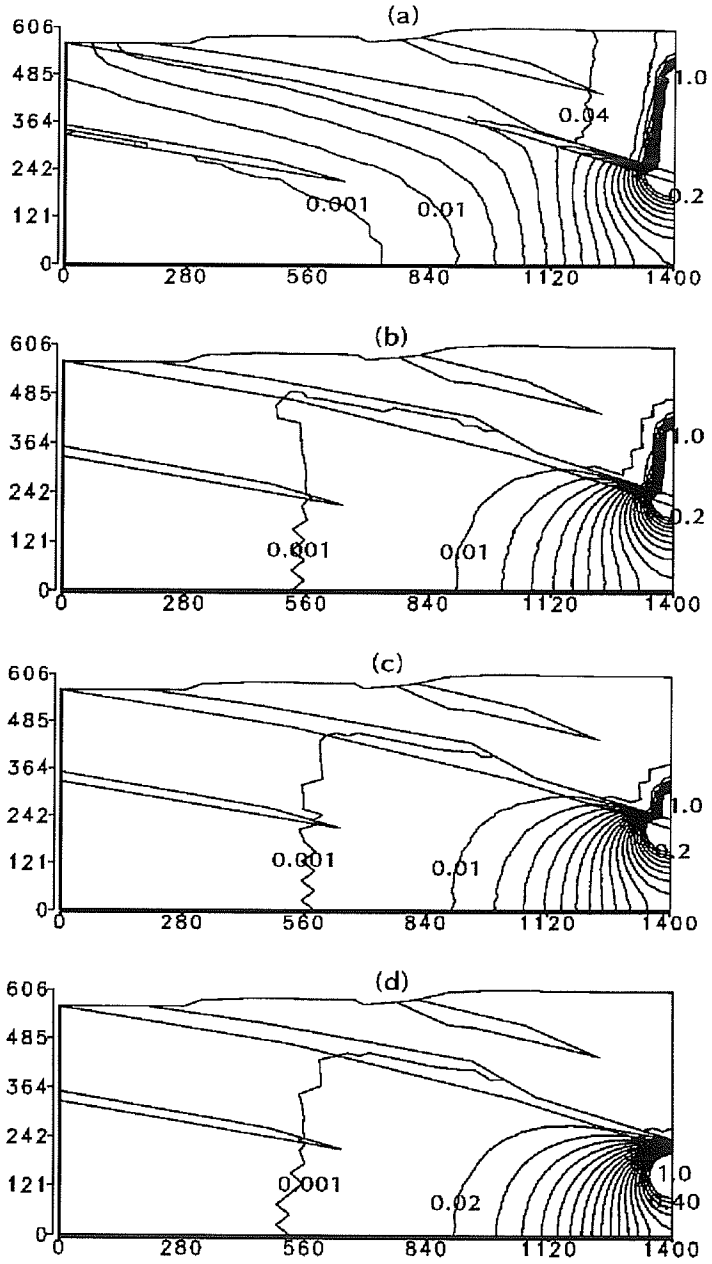

Figure 6.17: Contaminant plumes for case 2, four subcases at time of one million years, (a) subcase 1 contoured from 0.001 to 0.2 by 0.01, (b) subcase 2 contoured from 0.001 to 0.2 by 0.01, (c) subcase 3 contoured from 0.001 to 0.2 y 0.01, (d) subcase 4 contoured from 0.001 to 0.4 by 0.02.

6.4 Concluding remarks

This chapter describes the simulation of a groundwater field problem with the ULR method. Tremendous computational saving (about 95%) compared to the classic Crank-Nicolson method can be obtained. The simulation of the transport at one million years gives a good insight into the advantage of the ULR method.

Experiments show that the refinement of the grid is necessary, but there is a balance of the condition number and stability of the semi-discretized system (1.2). A finer mesh grid could cause larger condition numbers but can improve the grid Peclet number. A study of this tradeoff is worth further investigation.

Experiments also show in these simulations that the boundary conditions of the hydraulic head gradient determine the steady-state fluid flow patterns, and consequently determine the migration patterns of the solute, if the hydraulic properties of the media are fixed. When the bottom and the right boundaries are imposed as impermeable (case 2), the flow velocities are decreased, especially in the bottom-right corner. Therefore, the bottom-right corner is the best place for the deposit site of a contaminant source.

The simulations in this Chapter are undertaken in the absence of the radionuclide decay and generation owing to parent-to-daughter transformation of components, which is referred as decay chain problem. Because the governing equations for each species in the decay chain problems have the analogous form, it can be expected that a big advantage can be obtained from solving those problems by ULR method. This is studied in the next chapter.

Chapter 7

Two-species radionuclide decay chain problems

7.1 Introduction to decay chain problems

Radionuclide transport involves species decay and generation owing to parent-to-daughter transformation of components. For example, simulating transportation of two or more species, C_1 and C_2, the decay chain is of the type $C_1 \rightarrow C_2 \rightarrow C_3$, etc. The mathematical model for two species transport can be expressed by a system of two advection dispersion equations

$$\frac{\partial C_1}{\partial t} = \nabla \cdot (\mathbf{D} \nabla C_1) - \mathbf{v} \cdot \nabla C_1 - \mu_1 C_1, \tag{7.1}$$

$$\frac{\partial C_2}{\partial t} = \nabla \cdot (\mathbf{D} \nabla C_2) - \mathbf{v} \cdot \nabla C_2 - \mu_2 C_2 + \mu_1 C_1, \tag{7.2}$$

where μ_i is the decay constant of radionuclide species C_i ($\mathbf{D}$ and $\mathbf{v}$, see equation (1.1)). The decay constant μ_i is defined in terms of the species' half-life, $(t_{1/2})_i$ as

$$\mu_i = \ln 2/(t_{1/2})_i. \tag{7.3}$$

Equations (7.1) and (7.2) represent a simple case where the ratio of parent component decaying into daughter component to the initial amount of parent component is one ($\xi_{it} = 1$ in equation (1.1)). The description of general decay chain problems in given in [30].

The radionuclide source itself also involves parent-to-daughter transformation of components and species decay. Thus the boundary conditions are also transient. Denoting $\bar{C}_i(t)$ as the concentration of species i, for $i = 1, 2$ on the boundary Γ_1, and write

$$C_i(\mathbf{x}, t) = \bar{C}_i(t), \qquad \mathbf{x} \in \Gamma_1,$$

then $\bar{C}_i(t)$ can be described by a set of mass-balance equations:

$$\frac{d\bar{C}_1}{dt} = -\mu_1 \bar{C}_1, \tag{7.4}$$

$$\frac{d\bar{C}_2}{dt} = -\mu_2 \bar{C}_2 + \mu_1 \bar{C}_1. \tag{7.5}$$

In the above equations, $-\mu_1\bar{C}_1$ and $-\mu_2\bar{C}_2$ are species decay terms and $\mu_1\bar{C}_1$ in equation (7.5) is the species generating term from the $C_1 \rightarrow C_2$ transformation.

It is not difficult to obtain the solutions for these ordinary differential equations (7.4) and (7.5), i.e.,

$$\bar{C}_1 = C_1^0 e^{-\mu_1 t}, \tag{7.6}$$

$$\bar{C}_2 = \begin{cases} \left(C_2^0 + \dfrac{C_1^0\mu_1}{\mu_1 - \mu_2}\right) e^{-\mu_2 t} + \dfrac{C_1^0\mu_1}{\mu_2 - \mu_1} e^{-\mu_1 t}, & \mu_1 \neq \mu_2 \\ (C_2^0 + C_1^0\mu_1 t)e^{-\mu_1 t} & \mu_1 = \mu_2, \end{cases} \tag{7.7}$$

where initial values C_i^0 are constants. The general equations of the boundary conditions like (7.4) and (7.5) are known as Bateman's equations. The analytical solutions of them is given in [5]. More detailed discussions of the boundary conditions can be found in Harada et al, [28].

7.2 Application of the ULR method to decay chain problems and analysis of the residual error

In this section, an application of the ULR method applying to the decay chain problem is described. The major advantages of this method are large savings in computation time and storage.

7.2.1 Solving the decay chain problems by the ULR method

Using a common finite element grid, equations (7.1) and (7.2) are discretized to a set of the first order differential equations

$$\mathbf{M}\dot{\mathbf{c}}_1 + \mathbf{K}\mathbf{c}_1 + \mu_1\mathbf{M}\mathbf{c}_1 = \mathbf{f}_1, \tag{7.8}$$

$$\mathbf{M}\dot{\mathbf{c}}_2 + \mathbf{K}\mathbf{c}_2 + \mu_2\mathbf{M}\mathbf{c}_2 - \mu_1\mathbf{M}\mathbf{c}_1 = \mathbf{f}_2, \tag{7.9}$$

where $\mathbf{c}_i$ is the vector of species i concentrations on unknown nodes, $\mathbf{M}$ and $\mathbf{K}$ are "capacity" and "conductivity" matrices respectively as introduced in §1.1, and $\mathbf{f}_i$ is a time dependent vector formed by boundary conditions and other source/sinks. The ULR algorithm 3.4 is applied, using the starting vector $\mathbf{K}^{-1}\mathbf{a}$, where $\mathbf{a}$ is an arbitrary vector. Note that the starting vector for the ULR method is not the

simple $\mathbf{f} = \mathbf{b}\mu(t)$ as described in equation (1.2). A discussion of starting vector a will be presented in §7.3. From Theorem 3.1, the important relations like (3.28) and (3.29) are established, i.e.,

$$\mathbf{\Gamma} = \mathbf{P}_m^* \mathbf{M} \mathbf{Q}_m = \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{\Pi} & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{P}_m^* \mathbf{M} \mathbf{r}_m = \mathbf{0}, \tag{7.10}$$

$$\mathbf{P}_m^* \mathbf{M} \mathbf{K}^{-1} \mathbf{M} \mathbf{Q}_m = \mathbf{\Gamma} \mathbf{T}_m, \tag{7.11}$$

and

$$\mathbf{K}^{-1} \mathbf{M} \mathbf{Q}_m = \mathbf{Q}_m \mathbf{T}_m + \mathbf{r}_m \mathbf{e}_m^*, \tag{7.12}$$

where, $\mathbf{\Pi}$, $\mathbf{P}_m$, $\mathbf{Q}_m$, $\mathbf{T}_m$ and $\mathbf{r}_m$ are defined in Theorem 3.1, and m is the size of reduced system. The Rayleigh-Ritz process as introduced in §3.4 is used. By replacing $\mathbf{c}_i$ in equation (7.8) and (7.9) with $\mathbf{Q}_m \mathbf{w}_i$, pre-multiplying both sides of these equations by $\mathbf{\Gamma}^{-1} \mathbf{P}_m^* \mathbf{M} \mathbf{K}^{-1}$ and taking account of (7.10), (7.11) and (7.12), the reduced equations are obtained simultaneously

$$\mathbf{T}_m \dot{\mathbf{w}}_1 + \mathbf{w}_1 + \mu_1 \mathbf{T}_m \mathbf{w}_1 = \mathbf{\Gamma} \mathbf{P}_m^* \mathbf{M} \mathbf{K}^{-1} \mathbf{f}_1, \tag{7.13}$$

$$\mathbf{T}_m \dot{\mathbf{w}}_2 + \mathbf{w}_2 + \mu_2 \mathbf{T}_m \mathbf{w}_2 = \mathbf{\Gamma} \mathbf{P}_m^* \mathbf{M} \mathbf{K}^{-1} \mathbf{f}_2 + \mu_1 \mathbf{T}_m \mathbf{w}_1. \tag{7.14}$$

7.2.2 Computation and storage saving

The great advantage explored here is that only one ULR process is required to reduce two or more equations, including computing the LU factorization for one $n \times n$ matrix $\mathbf{K}$ in the ULR process and two small $m \times m$ matrices of the form $\mathbf{T}_m + \Delta t(\mathbf{I}_m + \mu_1 \mathbf{T}_m)$ and $\mathbf{T}_m + \Delta t(\mathbf{I}_m + \mu_2 \mathbf{T}_m)$ in the Crank-Nicolson scheme.

The Crank-Nicolson scheme requires the storage of four small $m \times m$ matrices of the form $\mathbf{T}_m + \Delta t(\mathbf{I}_m + \mu_1 \mathbf{T}_m)$, $\mathbf{T}_m - \Delta t(\mathbf{I}_m + \mu_1 \mathbf{T}_m)$, $\mathbf{T}_m + \Delta t(\mathbf{I}_m + \mu_2 \mathbf{T}_m)$ and $\mathbf{T}_m - \Delta t(\mathbf{I}_m + \mu_2 \mathbf{T}_m)$, where $m \ll n$. The classic Crank-Nicolson solver on equations (7.8) and (7.9) requires the LU factorization for two $n \times n$ matrices of the form, $\mathbf{M} + \Delta t(\mathbf{K} + \mu_1 \mathbf{M})$ and $\mathbf{M} + \Delta t(\mathbf{K} + \mu_2 \mathbf{M})$. and storage for four $n \times n$ matrices of the form, $\mathbf{M} + \Delta t(\mathbf{K} + \mu_1 \mathbf{M})$, $\mathbf{M} - \Delta t(\mathbf{K} + \mu_1 \mathbf{M})$, $\mathbf{M} + \Delta t(\mathbf{K} + \mu_2 \mathbf{M})$ and $\mathbf{M} - \Delta t(\mathbf{K} + \mu_2 \mathbf{M})$. Obviously, if more decay species are involved in the problem, then a larger saving of computation time and storage may be achieved.

7.2.3 Analysis of residual error

One problem however arises. That is how to choose the starting vector for the ULR process. The choice of a starting vector has a large effect on the accuracy and the convergence speed of the ULR method. In equation (7.8), when $\mathbf{K}^{-1}\mathbf{f}_1$, the steady-state solution is chosen as the starting vector, a faster convergence can be observed than any other starting vector. A discussion of this issue can be found in [43] and a recent paper [16]. This assertion can be investigated by the analysis of the residual error. The residual error of equation (7.8) and (7.9) are defined by substituting $\mathbf{c}_1 = \mathbf{Q}_m \mathbf{w}_1$ and $\mathbf{c}_2 = \mathbf{Q}_m \mathbf{w}_2$ into equation (7.8) and (7.9), and then rearranging the results, namely,

$$\theta_1^m(t) = \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m \dot{\mathbf{w}}_1 + \mathbf{Q}_m \mathbf{w}_1 + \mu_1 \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m \mathbf{w}_1 - \mathbf{K}^{-1}\mathbf{f}_1, \tag{7.15}$$

$$\begin{aligned} \theta_2^m(t) = {} & \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m \dot{\mathbf{w}}_2 + \mathbf{Q}_m \mathbf{w}_2 + \mu_2 \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m \mathbf{w}_2 \\ & - \mu_1 \mathbf{K}^{-1}\mathbf{M}\mathbf{Q}_m \mathbf{w}_1 - \mathbf{K}^{-1}\mathbf{f}_2, \end{aligned} \tag{7.16}$$

where m is the reduced size. Substituting (7.12) into (7.15) and (7.16) , considering (7.10) and (7.11), and that $\mathbf{w}_1$ and $\mathbf{w}_2$ are the solutions of (7.13) and (7.14) respectively, equation (7.15) and (7.16) become

$$\theta_1^m = \mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{f}_1 - \mathbf{K}^{-1}\mathbf{f}_1 + \mathbf{r}_m\mathbf{e}_m^*(\dot{\mathbf{w}}_1 + \mathbf{w}_1) \tag{7.17}$$

$$\theta_2^m = \mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{f}_2 - \mathbf{K}^{-1}\mathbf{f}_2 + \mathbf{r}_m\mathbf{e}_m^*(\dot{\mathbf{w}}_2 + \mu_2\mathbf{w}_2 - \mu_1\mathbf{w}_1). \tag{7.18}$$

The last term of the above equations decreases exponentially as t increases. The ULR method with ET technique guarantees that the reduced equation is stable in time. However, the first two terms on the right side of the above equations affect the convergence of θ_1^m and θ_2^m if their norms are large. If the steady-state solution is chosen as the starting vector for equation (7.8) only, it follows that

$$\mathbf{K}^{-1}\mathbf{f}_1 = \beta_1\mathbf{q}_1 = \beta_1\mathbf{Q}_m\mathbf{e}_1, \tag{7.19}$$

where $\beta_1 = \|\mathbf{K}^{-1}\mathbf{f}_1\|$. Thus the sum of the first two terms of right hand side of equation (7.17) vanishes by substituting (7.19) into it, i.e.,

$$\mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{f}_1 - \mathbf{K}^{-1}\mathbf{f}_1 = \beta_1(\mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{Q}_m\mathbf{e}_1 - \mathbf{Q}_m\mathbf{e}_1) = \mathbf{0}. \tag{7.20}$$

For decay chain problems involving more than one species such as equation (7.8) and (7.9), it is impossible to choose a suitable starting vector as $\mathbf{f}_1 \neq \mathbf{f}_2$. This problem can be solved by further investigating the structures of $\mathbf{f}_1$ and $\mathbf{f}_2$, as discussed in the next section.

7.3 Analysis of the starting vector for the ULR method

This section investigates the structures of the right hand side vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ and finds that these two vectors are parallel each other. Therefore, it is possible to choose a suitable starting vector for both equations (7.8) and (7.9) and apply the ULR method. Here, the simple case is discussed where the 2nd-type (Neumman) boundary condition is zero and there are no other source/sink sources. The more general conditions will be discussed in §7.6.

Equations (7.8) and (7.9) resulting from Galerkin's finite element method are only applicable at unknown nodes, i.e., the 1st-type (Dirichlet) boundary nodes are excluded from these equations. During the finite element process, the following equations are first derived, i.e.,

$$\tilde{\mathbf{M}}\dot{\tilde{\mathbf{c}}}_1 + \tilde{\mathbf{K}}\tilde{\mathbf{c}}_1 + \mu_1\tilde{\mathbf{M}}\tilde{\mathbf{c}}_1 = \tilde{\mathbf{f}}_1, \tag{7.21}$$

$$\tilde{\mathbf{M}}\dot{\tilde{\mathbf{c}}}_2 + \tilde{\mathbf{K}}\tilde{\mathbf{c}}_2 + \mu_2\tilde{\mathbf{M}}\tilde{\mathbf{c}}_2 - \mu_1\tilde{\mathbf{M}}\tilde{\mathbf{c}}_1 = \tilde{\mathbf{f}}_2, \tag{7.22}$$

where $\tilde{\mathbf{c}}_i$ is the concentration vector at all the nodes on the domain grid including the 1st-type (Dirichlet) boundary nodes, the matrices $\mathbf{M}$ and $\mathbf{K}$ in equations (7.8) and (7.9) are submatrices of $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$ obtained by partitioning out several rows and columns from $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$. These rows and columns correspond to the 1st-type boundary nodes. The $\tilde{\mathbf{f}}_i$ are zero vectors in this case, because the 2nd-type boundary condition is zero and there are no other source/sinks. The right hand side vectors in (7.8) and (7.9), $\mathbf{f}_i$, may be obtained by transferring the information from the 1st-type boundary nodes to the nodes within the same element see

[52, 69].

For example, suppose in (7.22) that node k is not a 1st-type boundary node but with the 1st-type boundary node j within a same element. Denote that $(\Gamma_1)_k^j$ to be the subset of such nodes j and $(f_2)_k$, $(\tilde{f}_2)_k$ to be the k'th entry of $\mathbf{f}_2$ and $\tilde{\mathbf{f}}_2$, respectively. According to the finite element process, row j and column k are partitioned from $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{K}}$ to form matrices $\mathbf{M}$ and $\mathbf{K}$, and the $(\tilde{f}_2)_j$ is also partitioned from $\mathbf{f}_2$. The entry $(f_2)_k$ is constructed by

$$(\mathbf{f}_2)_k = \sum_{j \in (\Gamma_1)_k^j} (\mathbf{f}_2)_k^j, \tag{7.23}$$

where $(\mathbf{f}_2)_k^j$ is formed by using the following equalities, from equation (7.9),

$$\begin{aligned}
(\mathbf{f}_2)_k^j &= -\tilde{M}_{kj}(\dot{\tilde{c}}_2)_j - \tilde{K}_{kj}(\tilde{c}_2)_j - \mu_2 \tilde{M}_{kj}(\tilde{c}_2)_j + \mu_1 \tilde{M}_{kj}(\tilde{c}_1)_j \\
&= -\tilde{K}_{kj}(\tilde{c}_2)_j - \tilde{M}_{kj}\left((\dot{\tilde{c}}_2)_j + \mu_2(\tilde{c}_2)_j - \mu_1(\tilde{c}_1)_j\right) \\
&= -\tilde{K}_{kj}(\tilde{c}_2)_j.
\end{aligned}$$

The last equality holds because $(\tilde{c}_1)_j$ and $(\tilde{c}_2)_j$ are 1st-type boundary nodes and they satisfy equation (7.5).

Substituting the above result into equation (7.23) and taking account of (7.7),

we can express $(\mathbf{f}_2)_k$ explicitly as

$$(\mathbf{f}_2)_k = \begin{cases} \dfrac{C_1^0\mu_1}{\mu_2-\mu_1}\left(-\displaystyle\sum_{j\in(\Gamma_1)_k^j}\tilde{K}_{kj}\right)e^{-\mu_1 t} \\ \qquad\qquad +\left(C_2^0+\dfrac{C_1^0\mu_1}{\mu_1-\mu_2}\right)\left(-\displaystyle\sum_{j\in(\Gamma_1)_k^j}\tilde{K}_{kj}\right)e^{-\mu_2 t}, & \mu_1\neq\mu_2 \\ (C_2^0+C_1^0\mu_1 t)\left(-\sum_{j\in(\Gamma_1)_k^j}\tilde{K}_{kj}\right)e^{-\mu_1 t} & \mu_1=\mu_2. \end{cases}$$

For the nodes which are not related to the 1st-type boundary nodes, the corresponding rows and columns of $\tilde{\mathbf{M}}$, $\tilde{\mathbf{K}}$ and the corresponding entries of $\tilde{\mathbf{f}}_2$ remain unchanged.

Similar results for the right hand side vector of (7.8), $\mathbf{f}_1$, can be obtained. In summary, denote ℓ as a subset of nodes in which each node is related to the 1st-type boundary nodes such as node k described above. If $\mu_1\neq\mu_2$, then $\mathbf{f}_1$ and $\mathbf{f}_2$ take the forms

$$\mathbf{f}_1 = \mathbf{b}_1 e^{-\mu_1 t}, \qquad \text{and} \qquad \mathbf{f}_2 = \mathbf{b}_2^1 e^{-\mu_1 t} + \mathbf{b}_2^2 e^{-\mu_2 t},$$

where $\mathbf{b}_1$, $\mathbf{b}_2^1$ and $\mathbf{b}_2^2$ satisfy

$$(b_1)_k = \begin{cases} 0 & k\notin\ell \\ -\displaystyle\sum_{j\in(\Gamma_1)_k^j}\tilde{K}_{kj}C_1^0 & k\in\ell, \end{cases}$$

$$(b_2^1)_k = \begin{cases} 0 & k\notin\ell \\ -\displaystyle\sum_{j\in(\Gamma_1)_k^j}\tilde{K}_{kj}\frac{C_1^0\mu_1}{\mu_2-\mu_1} & k\in\ell, \end{cases}$$

$$(b_2^2)_k = \begin{cases} 0 & k \notin \ell \\ -\sum_{j \in (\Gamma_1)_k^j} \tilde{K}_{kj} \left(C_2^0 + \dfrac{C_1^0 \mu_1}{\mu_1 - \mu_2} \right) & k \in \ell, \end{cases}$$

Therefore equations (7.8) and (7.9) can be rewritten as

$$\mathbf{M}\dot{\mathbf{c}}_1 + \mathbf{K}\mathbf{c}_1 + \mathbf{M}\mathbf{c}_1 = \xi_1 \mathbf{b} e^{-\mu_1 t}, \tag{7.24}$$

$$\mathbf{M}\dot{\mathbf{c}}_2 + \mathbf{K}\mathbf{c}_2 + \mu_2 \mathbf{M}\mathbf{c}_2 - \mu_1 \mathbf{M}\mathbf{c}_1 = \xi_2^1 \mathbf{b} e^{-\mu_1 t} + \xi_2^2 \mathbf{b} e^{-\mu_2 t}, \tag{7.25}$$

where

$$\xi_1 = C_1^0, \quad \xi_2^1 = \frac{C_1^0 \mu_1}{\mu_2 - \mu_1}, \quad \xi_2^2 = C_2^0 + \frac{C_1^0 \mu_1}{\mu_1 - \mu_2},$$

and $\mathbf{b}$ is the vector with entries

$$b_k = \begin{cases} 0, & k \notin \ell, \\ -\sum_{j \in (\Gamma_1)_k^j} \tilde{K}_{kj}, & k \in \ell. \end{cases} \tag{7.26}$$

Similarily, if $\mu_1 = \mu_2$, equation (7.8) and (7.9) can be rewritten as

$$\mathbf{M}\dot{\mathbf{c}}_1 + \mathbf{K}\mathbf{c}_1 + \mathbf{M}\mathbf{c}_1 = C_1 \mathbf{b} e^{-\mu_1 t}, \tag{7.27}$$

$$\mathbf{M}\dot{\mathbf{c}}_2 + \mathbf{K}\mathbf{c}_2 + \mu_2 \mathbf{M}\mathbf{c}_2 - \mu_1 \mathbf{M}\mathbf{c}_1 = (C_2^0 + C_1^0 \mu_1 t) \mathbf{b} e^{-\mu_2 t}, \tag{7.28}$$

where $\mathbf{b}$ is the form of (7.26).

Equations (7.24), (7.25) and (7.27), (7.28) show that $\mathbf{f}_1$ and $\mathbf{f}_2$ are all parallel to the vector $\mathbf{b}$. Thus $\mathbf{K}^{-1}\mathbf{b}$ is the right choice for a starting vector. Therefore the analysis of the residual error of the reduced system can proceed as before. For example, if $\mu_1 \neq \mu_2$, the reduced system after the ULR method, (7.13) and

(7.14), becomes

$$\text{(7.29)}\quad \mathbf{T}_m\dot{\mathbf{w}}_1 + \mathbf{w}_1 + \mu_1\mathbf{T}_m\mathbf{w}_1 = \xi_1\beta_1\mathbf{e}_1 e^{-\mu_1 t},$$

$$\text{(7.30)}\quad \mathbf{T}_m\dot{\mathbf{w}}_2 + \mathbf{w}_2 + \mu_2\mathbf{T}_m\mathbf{w}_2 = \xi_2^1\beta_1\mathbf{e}_1 e^{-\mu_1 t} + \xi_2^2\beta_1\mathbf{e}_1 e^{-\mu_2 t} + \mu_1\mathbf{T}_m\mathbf{w}_1,$$

where $\beta_1 = \|\mathbf{K}^{-1}\mathbf{b}\|$. The corresponding residual errors are (7.17) and (7.18). It is not difficult to find that the sums of the first two terms in the right hand side of equations (7.17) and (7.18) vanish. In a manner analogous to the derivation of (7.20), they are

$$\begin{aligned}
\mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{f}_1 - \mathbf{K}^{-1}\mathbf{f}_1 &= \xi_1(\mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{b} - \mathbf{K}^{-1}\mathbf{b})e^{-\mu_1 t} = \mathbf{0},\\
\mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{f}_2 - \mathbf{K}^{-1}\mathbf{f}_2 &= \xi_2^1(\mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{b} - \mathbf{K}^{-1}\mathbf{b})e^{-\mu_1 t}\\
&+ \xi_2^2(\mathbf{Q}_m\mathbf{P}_m^*\mathbf{M}\mathbf{K}^{-1}\mathbf{b} - \mathbf{K}^{-1}\mathbf{b})e^{-\mu_2 t} = \mathbf{0}.
\end{aligned}$$

Hence equations (7.17) and (7.18) become

$$\text{(7.31)}\quad \theta_1^m(t) = \mathbf{r}_m\mathbf{e}_m^*(\dot{\mathbf{w}}_1 + \mu_1\mathbf{w}_1),$$

$$\text{(7.32)}\quad \theta_2^m(t) = \mathbf{r}_m\mathbf{e}_m^*(\dot{\mathbf{w}}_2 + \mu_2\mathbf{w}_2 - \mu_1\mathbf{w}_1).$$

These equations show that the residual error in the ULR application to the decay chain problem is stable as time increases. This issue will be discussed further in the next section.

7.4 Stability and termination of the process

7.4.1 Analysis of stability

The requirement of stability for equations (7.29) and (7.30) is that the real part of all eigenvalues of matrix $-\mathbf{T}_m^{-1}(\mathbf{I}+\mu_i\mathbf{T}_m)$, $i = 1, 2$, are negative, or that of $\mathbf{T}_m^{-1} + \mu_i\mathbf{I}$ are positive. Denoting $\lambda(\mathbf{A})$ to be an any eigenvalues of matrix $\mathbf{A}$and $Re_\lambda(\mathbf{A})$ to be the real part of the corresponding eigenvalue λ, it follows that

$$\lambda(\mathbf{T}_m^{-1} + \mu_i\mathbf{I}) = \lambda(\mathbf{T}_m^{-1}) + \mu_i.$$

Therefore, $Re_\lambda(\mathbf{T}_m^{-1} + \mu_i\mathbf{I}) > 0$ if and only if $Re_\lambda(\mathbf{T}_m^{-1}) > -\mu_i$. Since

$$\lambda(\mathbf{T}_m^{-1}) = \frac{1}{\lambda(\mathbf{T}_m)}, \quad \text{and} \quad Re_\lambda(\mathbf{T}_m^{-1}) = \frac{Re_\lambda(\mathbf{T}_m)}{|\lambda(\mathbf{T}_m)|^2},$$

the condition for the stability of equation (7.29) and (7.30) is

$$Re_\lambda(\mathbf{T}_m) > -\mu_i|\lambda(\mathbf{T}_m)|^2$$

for $i = 1, 2$, or

$$Re_\lambda(\mathbf{T}_m) > \max\{-\mu_1|\lambda(\mathbf{T}_m)|^2, -\mu_2|\lambda(\mathbf{T}_m)|^2\}. \tag{7.33}$$

The ET technique can be applied to equations (7.29) and (7.30) if this requirement is not satisfied. Recall that all details of the process have been discussed in Chapter 4. After applying the ET technique, we obtain the translated system,

for example,

$$\begin{aligned}\bar{\mathbf{T}}_m\dot{\mathbf{w}}_1 + \mathbf{w}_1 + \mu_1\bar{\mathbf{T}}_m\mathbf{w}_1 &= \xi_1\beta_1\mathbf{e}_1e^{-\mu_1 t},\\ \bar{\mathbf{T}}_m\dot{\mathbf{w}}_2 + \mathbf{w}_2 + \mu_2\bar{\mathbf{T}}_m\mathbf{w}_2 &= \xi_2\beta_1\mathbf{e}_1e^{-\mu_1 t} + \xi_3\beta_1\mathbf{e}_1e^{-\mu_1 t} + \mu_1\bar{\mathbf{T}}_m\mathbf{w}_1,\end{aligned}$$

where

$$\bar{\mathbf{T}}_m = \mathbf{T}_m[\mathbf{I} + \mathbf{X}_U(\mathbf{D}_U^{-1}\tilde{\mathbf{D}}_U - \mathbf{I}_\ell)\mathbf{Y}_U^*].$$

When $\mu_1 = \mu_2$, a similar result can be obtained.

7.4.2 Monitoring and terminating the process

The analysis and algorithm are almost the same as described in §3.4. At each step j during the ULR process, the relative residual errors for equations (7.24) and (7.25) are defined as

$$\delta_1^j(t) = \frac{\|\theta_1^j(t)\|}{\|\xi_1\mathbf{K}^{-1}\beta^{-\mu_1 t}\|}, \tag{7.34}$$

and

$$\delta_2^j(t) = \frac{\|\theta_2^j(t)\|}{\|\xi_2^1\mathbf{K}^{-1}\beta^{-\mu_1 t} + \xi_2^2\mathbf{K}^{-1}\beta^{-\mu_2 t}\|}, \tag{7.35}$$

where $\theta_i^j(t)$ is the relative residual error (7.15) and (7.16) at step j. The bounds of $\delta_1^j(0)$ and $\delta_2^j(0)$ are computed to monitor the process. Because the initial conditions state that $w_1(0) = w_2(0) = 0$, then from (7.29) and (7.30), it follows that

$$\dot{\mathbf{w}}_1(0) = \beta_1\xi_1\mathbf{T}_m^{-1}\mathbf{e}_1, \quad \text{and} \quad \dot{\mathbf{w}}_2(0) = \beta_1(\xi_2^1 + \xi_2^2)\mathbf{T}_m^{-1}\mathbf{e}_1. \tag{7.36}$$

Substituting equations (7.31) and (7.32) into (7.34) and (7.35), and noting (7.36), bounds for the relative residual error at $t = 0$ and at step j are obtained, i.e.,

$$\delta_1^j(0) \leq \frac{\|\mathbf{r}_j\|\|\beta_1\|\|\xi_1\|\|\mathbf{T}_j^{-1}\mathbf{e}_1\|}{\|\xi_1\mathbf{K}^{-1}\mathbf{b}\|} = |\beta_{j+1}|\|\mathbf{T}_j^{-1}\mathbf{e}_1\|,$$

and

$$\delta_2^j(0) \leq \frac{\|\mathbf{r}_j\|\|\beta_1\|(|\xi_2^1| + |\xi_2^2|)\|\mathbf{T}_j^{-1}\mathbf{e}_1\|}{(|\xi_2^1| + |\xi_2^2|)\|\mathbf{K}^{-1}\mathbf{b}\|} = |\beta_{j+1}|\|\mathbf{T}_j^{-1}\mathbf{e}_1\|.$$

These expressions are identical, therefore the relative residual bound can be defined as

$$\delta_j = |\beta_{j+1}|\|\mathbf{T}_j^{-1}\mathbf{e}_1\|,$$

which can be used to monitor the process by calculating δ_j at every step j. When a prefixed criterion is satisfied, then the process can be terminated. The calculation of δ_j follows Algorithm 3.3 in §3.4.

7.5 Numerical examples

An example problem is chosen to demonstrate the procedure developed above and is also based on a conceptual model similar to the Whiteshell Research Area (WRA) (Figure 6.1). The species transformation from ^{240}Pu to ^{236}U is selected from the first segment of an actinide decay chain $^{240}\text{Pu} \rightarrow {}^{236}\text{U} \rightarrow {}^{230}\text{Th} \rightarrow {}^{228}\text{Ra} \rightarrow {}^{226}\text{Th}$, see [22]. The half-lives of ^{240}Pu and ^{236}U are 6.54×10^3 and 2.34×10^7 years, and their corresponding decay constants, (7.3), are 0.46×10^{-4} and 0.13×10^{-7} (1/year) respectively.

The geological and hydrological properties, and the finite element discretization of the domain in this hypothetical simulation are the same as those of discussed in §6.1. The boundary conditions for the flow and transport equations are also the same as imposed for case 1, subcase 1, i.e., only the bottom boundary is considered to be impermeable and the disposal site is in the depth range of 460 $m < z <$ 480 m.

Two cases are simulated. The first case assumes that both species ^{240}Pu and ^{236}U are released from the vault room and the initial concentration of each species is normalized to one. The second case assumes that only ^{240}Pu is released from the vault room. Its initial concentration is also set at one and that of ^{236}U is zero. The parameters for the ULR method are exactly the same as listed in Table 6.2.

Migration of these two species for case 1 is depicted in Figure 7.1 and 7.2. They are similar to Figure 6.7 in their travel paths and the spreading patterns of the leading edge of the plume, but are different in the concentration of the various species. For ^{240}Pu, the contaminant source decays from 1 to 0.74 after 3000 years. For ^{236}U, during the same period of time the source increases from 1 to 1.13. This behavior results from the decay constant of ^{240}Pu being much higher than that of ^{236}U, and consequently, the speed of transformation from ^{240}Pu to ^{236}U is much faster than that from ^{236}U to the next daughter. This phenomenon can also be observed from Figure 7.3 where only ^{240}Pu is released. The concentration of ^{236}U at the disposal site has grown from zero to 0.26 in 3000 years.

The behavior of the ULR algorithm is similar to that shown in the first column of Table 6.6. The size of the reduced system is 100. The relative residual error at the termination step is 0.95×10^{-1}. The first breakdown occurs at step 48. Two

Time	Figure 7.1 (^{240}Pu)	Figure 7.2 (^{236}U)	Figure 7.3 (^{236}U)
	RMS error		
100 years	0.465D-03	0.474D-03	0.467D-05
500 years	0.164D-04	0.181D-04	0.843D-06
1000 years	0.446D-05	0.540D-05	0.469D-06
3000 years	0.244D-06	0.415D-06	0.855D-07
	Maximum error		
100 years	0.947D-02	0.966D-02	0.952D-04
500 years	0.206D-03	0.227D-03	0.105D-04
1000 years	0.780D-04	0.944D-04	0.821D-05
3000 years	0.454D-05	0.772D-05	0.159D-05
	Execution time (hour: minute: second)		
ULR	00:22:13		00:26:00
Classic CN	13:51:13		14:03:31

Table 7.1: RMS error, Maximum error and execution time comparisons with respect to the classic Crank-Nicolson solver for decay chain problems.

eigenvalues of the reduced system have negative real parts and the RMS errors and maximum errors can been seen in Table 7.1. The highest RMS error is of the order of 10^{-4}, while the lowest is of the order of 10^{-8}. The maximum error is of the order of 10^{-3}, while the lowest is the order of 10^{-6}. Note the large reduction in execution time. The ULR method only needs about 3% execution time of the classic Crank-Nicolson solver.

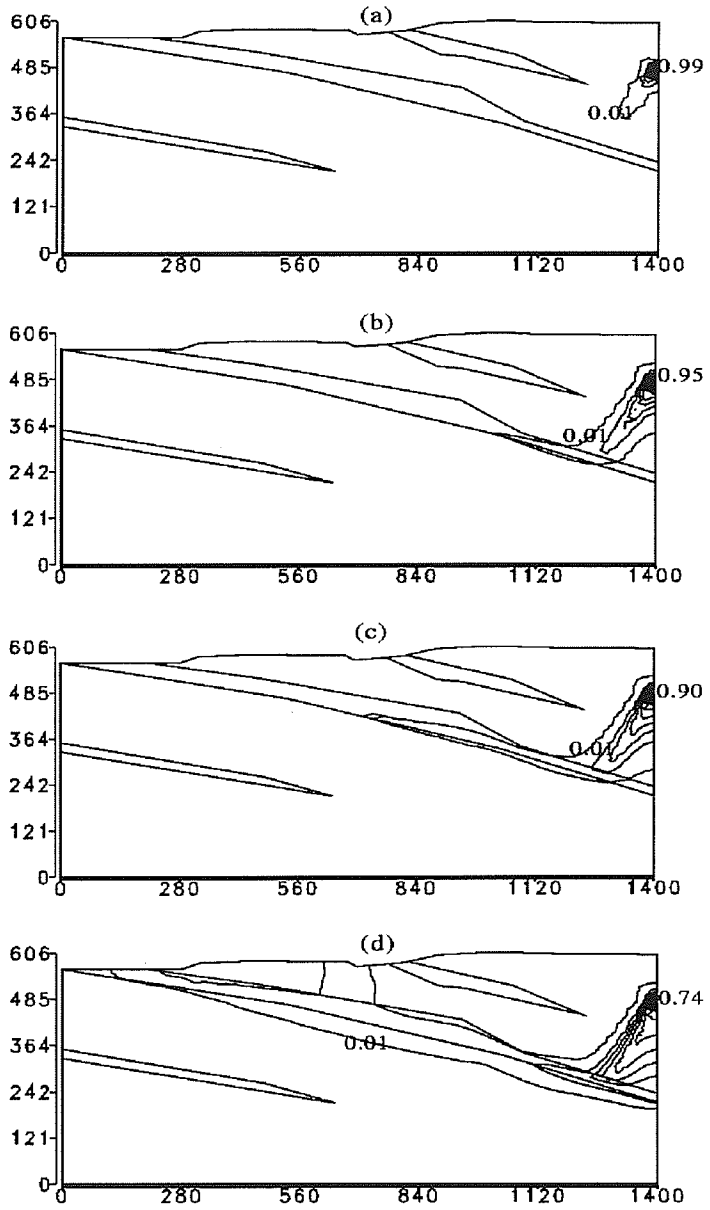

Figure 7.1: Radionuclide transport plumes for ^{240}Pu, (a) t=100 years , contour from 0.01 to 0.99 by 0.1. (b) t=500 years, contour from 0.01 to 0.95 by 0.1. (c) t=1000 years, contour from 0.01 to 0.9 by 0.09. (d) t=3000 years, contour from 0.01 to 0.74 by 0.07.

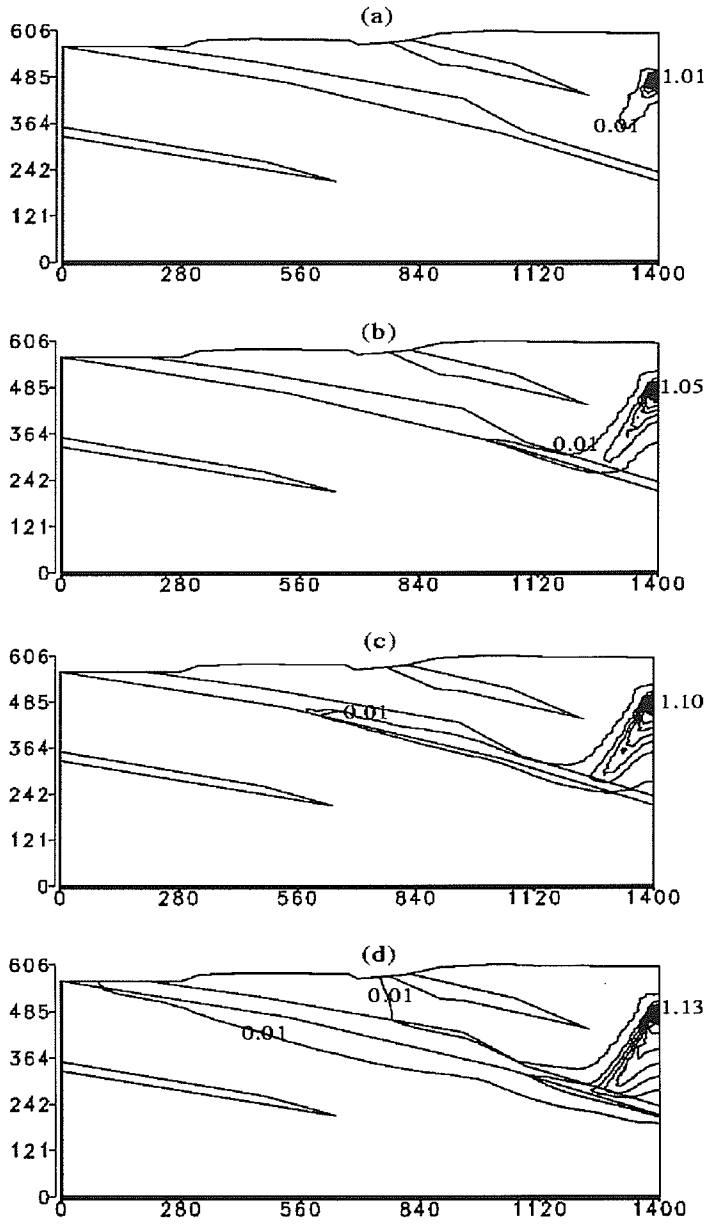

Figure 7.2: Radionuclide transport plumes for ^{236}U, (a) t=100 years , contour from 0.01 to 1.01 by 0.1. (b) t=500 years, contour from 0.01 to 1.05 by 0.1. (c) t=1000 years, contour from 0.01 to 1.10 by 0.11. (d) t=3000 years, contour from 0.01 to 1.26 by 0.13.

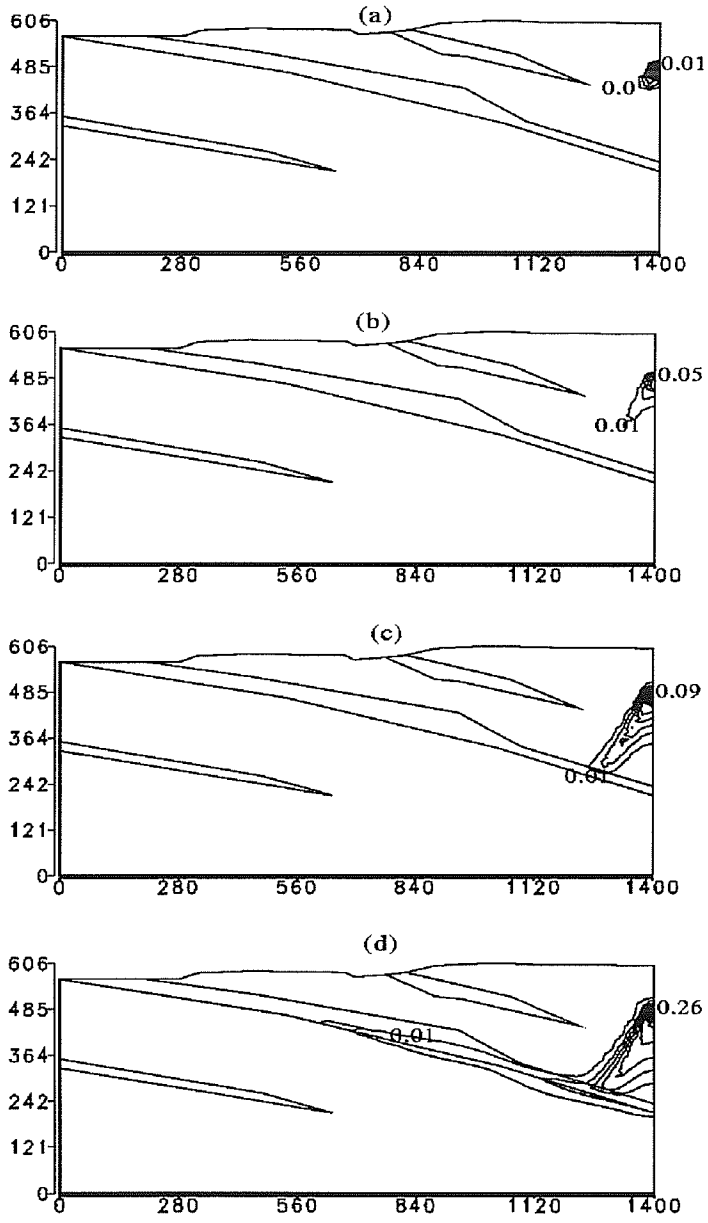

Figure 7.3: Radionuclide transport plumes for ^{236}U of case 2, where ^{236}U is not released from disposal room but is transformed from ^{240}Pu. (a) t=100 years, contour from 0.00 to 0.01. (b) t=500 years, contour from 0.01 to 0.05 by 0.01. (c) t=1000 years, contour from 0.01 to 0.09 by 0.01. (d) t=3000 years, contour from 0.01 to 0.26 by 0.03.

7.6 Conclusions and suggested future work

By means of analysis and illustrative examples, the advantages of applying of the ULR method to the simple two species decay chain problems have been demonstrated; specifically computation and storage savings. For long period predictions, the method is extremely encouraging. Recall that in §5.4, where the single transport problem is analyzed, the limitation of storage is a problem for the application of the ULR method. Here, on the contrary, storage saving is an advantage.

Further work on general multi-decay chain problems can be developed as outlined below. If the general decay chain problems involve non-zero 2nd-type boundary conditions and other sink/sources, then the $\mathbf{f}_i$ in (7.8) and (7.9) are not be parallel to each other. They may have the forms

$$\mathbf{f}_1 = \bar{\mathbf{f}}_1 + \xi_1 \mathbf{b} e^{-\mu_1 t}, \tag{7.37}$$

$$\mathbf{f}_2 = \bar{\mathbf{f}}_2 + \xi_2 \mathbf{b} e^{-\mu_1 t} + \xi_3 \mathbf{b} e^{-\mu_2 t}, \tag{7.38}$$

where $\bar{\mathbf{f}}_i$ is a constant vector constructed from the non-zero 2nd-type boundary conditions and the other sink/sources. It is reasonable to further assume that $\bar{\mathbf{f}}_1 = \bar{\mathbf{f}}_2 = \bar{\mathbf{f}}$, because the 2nd-type boundary conditions and the other source/sinks are the same for two species. Due to the homogeneous initial condition (the initial concentrations on all unknowns are zero), equations (7.8) and (7.9) can be split into several systems of equations. For equation (7.8), this is equivalent to the

following two systems of equations,

$$\mathbf{M}\dot{\mathbf{c}}_1' + \mathbf{K}\mathbf{c}_1' + \mu_1\mathbf{M}\mathbf{c}_1' = \bar{\mathbf{f}}, \tag{7.39}$$

$$\mathbf{M}\dot{\mathbf{c}}_1'' + \mathbf{K}\mathbf{c}_1'' + \mu_1\mathbf{M}\mathbf{c}_1'' = \xi_1\mathbf{b}e^{-\mu_1 t}, \tag{7.40}$$

where $\mathbf{c}_1 = \mathbf{c}_1' + \mathbf{c}_1''$. The ULR method generates Krylov subspaces by using two starting vectors $\mathbf{K}^{-1}\bar{\mathbf{f}}$ and $\mathbf{K}^{-1}\mathbf{b}$, respectively. That is, $\mathbf{P}_m'$, $\mathbf{Q}_m'$ and $\mathbf{T}_m'$ are produced by starting vector $\mathbf{K}^{-1}\bar{\mathbf{f}}$, and $\mathbf{P}_m''$, $\mathbf{Q}_m''$ and $\mathbf{T}_m''$ are produced by starting vector $\mathbf{K}^{-1}\mathbf{b}$. Then in terms of Rayleigh-Ritz process, the above equations may be reduced to

$$\mathbf{T}_m'\dot{\mathbf{w}}_1' + \mathbf{w}_1' + \mu_1\mathbf{T}_m'\mathbf{w}_1' = \beta_1'\mathbf{e}_1, \tag{7.41}$$

$$\mathbf{T}_m''\dot{\mathbf{w}}_1'' + \mathbf{w}_1'' + \mu_1\mathbf{T}_m''\mathbf{w}_1'' = \xi_1\beta_1''\mathbf{e}_1 e^{-\mu_1 t}, \tag{7.42}$$

where $\beta_1' = \|\mathbf{K}^{-1}\bar{\mathbf{f}}\|$ and $\beta_1'' = \|\mathbf{K}^{-1}\mathbf{b}\|$, $\mathbf{c}_1' = \mathbf{Q}_m'\mathbf{w}_1'$ and $\mathbf{c}_1'' = \mathbf{Q}_m''\mathbf{w}_1''$

Similar reduction process can be carried out for equation (7.9) except for the term $\mu_1\mathbf{M}\mathbf{c}_1$. The principle of the split is to make the each equation related to only one starting vector. So, after substituting $\mathbf{c}_1 = \mathbf{c}_1' + \mathbf{c}_1''$ and $\mathbf{c}_2 = \mathbf{c}_2' + \mathbf{c}_2''$, equation (7.9) is equivalent to equations

$$\mathbf{M}\dot{\mathbf{c}}_2' + \mathbf{K}\mathbf{c}_2' + \mu_2\mathbf{M}\mathbf{c}_2' = \bar{\mathbf{f}} + \mu_1\mathbf{M}\mathbf{c}_1', \tag{7.43}$$

$$\mathbf{M}\dot{\mathbf{c}}_2'' + \mathbf{K}\mathbf{c}_2'' + \mu_2\mathbf{M}\mathbf{c}_2'' = \xi_2\mathbf{b}e^{-\mu_1 t} + \xi_3\mathbf{b}e^{-\mu_2 t} + \mu_1\mathbf{M}\mathbf{c}_1''. \tag{7.44}$$

Thus, using the same Krylov subspaces generated by the starting vectors $\mathbf{K}^{-1}\bar{\mathbf{f}}$

and $\mathbf{K}^{-1}\mathbf{b}$, the above equations can be reduced to

$$(7.45)\quad \mathbf{T}'_m\dot{\mathbf{w}}'_2 + \mathbf{w}'_2 + \mu_2\mathbf{T}'_m\mathbf{w}'_2 = \beta'_1\mathbf{e}_1 + \mu_1\mathbf{T}'_m w'_1,$$

$$(7.46)\quad \mathbf{T}''_m\dot{\mathbf{w}}''_2 + \mathbf{w}''_2 + \mu_2\mathbf{T}''_m\mathbf{w}''_2 = \xi_2\beta''_1\mathbf{e}_1 e^{-\mu_1 t} + \xi_3\beta''_1\mathbf{e}_1 e^{-\mu_2 t} + \mu_1\mathbf{T}'_m\mathbf{w}''_1,$$

where $\mathbf{c}'_2 = \mathbf{Q}'_m\mathbf{w}'_2$, $\mathbf{c}''_2 = \mathbf{Q}''_m\mathbf{w}''_2$.

In this approach, two ULR processes are performed, but only one LU factorization of $\mathbf{K}$ needs to be computed. It is expected that same advantages as described in §7.2 can be realized.

Another approach to the solution of the general equations is to construct equations (7.8) and (7.9) in block form, i.e.,

$$(7.47)\qquad \tilde{\mathbf{M}}\dot{\mathbf{c}} + \tilde{\mathbf{K}}\mathbf{c} = \tilde{\mathbf{f}} = \tilde{\mathbf{f}}_1 + \tilde{\mathbf{f}}_2 e^{-\mu_1 t} + \tilde{\mathbf{f}}_3 e^{-\mu_2 t},$$

where

$$(7.48)\qquad \tilde{\mathbf{M}} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K} + \mu_1\mathbf{M} & \mathbf{0} \\ -\mu_1\mathbf{M} & \mathbf{K} + \mu_2\mathbf{M} \end{bmatrix},$$

and

$$(7.49)\quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}, \quad \tilde{\mathbf{f}}_1 = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \quad \tilde{\mathbf{f}}_2 = \begin{bmatrix} \xi_1\mathbf{b} \\ \xi_2\mathbf{b} \end{bmatrix}, \quad \text{and} \quad \tilde{\mathbf{f}}_3 = \begin{bmatrix} \mathbf{0} \\ \xi_3\mathbf{b} \end{bmatrix}.$$

Equation (7.47) can then be split into three equations and reduced into three small equations with three different starting vectors according to the right hand side of equation (7.47). The size of the system becomes twice as large as the original size and the lower bandwidth of $\bar{\mathbf{K}}$ is significantly increased. Hence, the time involved

in computing the LU factorization of $\bar{\mathbf{K}}$ will be considerably increased. However, sparse storage and iterative solvers may be used in overcoming this problem. A detailed analysis and comparison of the above two approaches should be carried out in the future.

Part IV

Conclusions

Chapter 8

Concluding Remarks

8.1 Concluding remarks

The present thesis addresses the development of the ULR method to solve the advection dispersion equation, with particular emphasis on methods dealing with the breakdown and instability problems which occasionally occur during the process. Robust algorithms including the MPNSV algorithm, switching algorithm, termination algorithm for the process, and the ET algorithm are presented. Applications to contaminant transport problems, including field problems and radionuclide decay chain problems are also carried out. The numerical results have shown that this method can result in large computational savings for long term period prediction. This conclusion is especially true in the absence of breakdown and instability of the reduced system. The application to the radionuclide decay chain problems also shows that the computation time saving of the ULR method is large. In addition, the storage saving in this application is also promising.

The following is a summary of the results given in this thesis. Some open questions that are important topics for further research will also be presented.

- As a modal reduction method, the ULR method reduces a semi-discretized first order differential equation to a much smaller size system. In the absence of breakdown, the reduced system is tridiagonal, or a special upper Hessenberg with a lower band width of one and irregular upper band width. Because the reduced system is very small in comparison to the original size, great computational cost savings can then be obtained during the time-stepping process. Experiments showed that the method can save more than 95% execution time comparing to the classic Crank-Nicolson method.

- The Maximum-Pivot New-Start Vector (MPNSV) method has been shown to be effective in dealing with the problem of breakdown. The new pivot produced by the MPNSV method is the maximum, so one can quickly check whether the breakdown is pathological, and therefore determine whether to apply the Switch method or not.

- The analysis of the residual error bound shows that as time t increases, the bound goes to zero since the ULR method together with the ET technique ensures that the solution is stable. This means that the ULR method can be used efficiently to obtain more accurate approximate solutions for predictions over long time period.

- Numerical experiments have shown that increasing the pivot tolerance (often forcing earlier breakdowns) produces highly accurate approximations, while decreasing the pivot tolerance can postpone or avoid the breakdown but the method will lose accuracy. The choice of an appropriate pivot tolerance is therefore a balance between the need for accuracy and the need to avoid

early breakdowns with their associated extra computation costs.

- Relative residual error bounds can be computed recursively during the ULR process. In addition, numerical experiments have shown the RMS error and the maximum error (with respect to the classic Crank-Nicolson method) are much smaller than the relative residual bounds. Therefore it is appropriate to use the relative residual errors to monitor and terminate the recursive procedure.

- The condition number of the semi-discretized system is important for the application of the ULR method. If the condition number is high, roundoff error will cause Cholesky factorization, which is used to construct a new start vector in the MPNSV method, to breakdown and lose accuracy of LU factorization of $\mathbf{K}$. Experiments have shown that a coarse grid may improve the condition number, but will increase the grid Peclet number, and consequently the semi-discretized system may become unstable. Thus an appropriate grid size is important in the application of the ULR method. Theoretical substantiation is necessary for the observation of this phenomenon.

- The ULR method or the other reduction methods (for instance, SLR and AR methods) can be used to solve the multi-species decay chain problems. Because the governing equations of a multi-species decay chain problem have the same structural forms due to the same hydrogeological and hydraulic properties, the application of the ULR method to the problem has demonstrated great advantages over the classic approach.

- For the ULR applications to the multi-species decay chain problem, accord-

ing to the analyses of the residual errors and the right hand side vectors in the semi-discretized equations, it has been provn that one can find a common starting vector to generate a single pair Krylov subspaces for all equations. For the problems with complex boundary conditions, similar results can be obtained.

- Experiments also shown that the ET method is effective in overcoming the instability problem. Further insight into the comparison with the Implicitly Restart Lanczos method [23] is desirable.

Although much research still remains to be carried out to give further insights into the behavior of the ULR method, the analyses in this thesis will hopefully promote its use as an option for solving contaminant transport in porous media.

Bibliography

[1] E. Anderson, Z. Bai, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *Users' Guide*, SIAM, 1992.

[2] W. E. Arnoldi, , *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math. 9, pp. 17–29, 1951.

[3] O. Aselsson, *A Generalised Conjugate Direction Method and Its Application on a Singular Perturbation Problem*, Lecture Notes in Mathematics 773, Springer-Veriag, Berlin, Heidelberg, New York, 1980.

[4] O. Aselsson, *Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations*, Linear Algebra Appl., 29, pp. 1–16, 1980.

[5] H. Bateman, *The solution of a system of differential equations occurring in the theory of radioactive transformation*, Proc. Cambridge Philos.Soc., 15, 1910.

[6] J. Bear, *Hydraulics of Groundwater*, McGraw-Hill, New York.

[7] J. Bear, *Dynamics of fluids in porous media*, New York, Dover, 1988.

[8] A, BJORCK, *Solving linear least squares problems by Gram-Schmidt orthogonalization*, BIT 7 pp. 1-21, 1967.

[9] , J. BRAMBLE, J. PASCIAK AND A. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring.I*, Math. of Comput., 175, pp. 103–134, 1986

[10] G. F. CAREY AND J. T. ODEN, *Finite Element: Fluid Mechanics, Volume VI*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[11] G. F. CAREY AND K. SEPEHRNOORI, *Gershgorin theory for stiffness and stability of evolution systems and convection-diffusion*, Computer Methods in Applied Mechanics and Engineering, 22, p. 23–48, 1980.

[12] P. CONCUS AND G. H. GOLUB, *A Generalized Conjugate Gradient Method for Unsymmetric Systems of Linear Equations*, Lecture Notes in Economics and Mathematics Systems 139, Springer-Verlag, Berlin, Heidelberg, New York, 1976.

[13] C. C. DAVISON, A. BROWN, M. GASCOYNE, D. C. KAMINENI, G. S. LODHA, T. W. MELBYK, B. W. NAKKA, P. A. O'CONNOR, D. U. OPHORI, N. W. SCHEIER, N. M. SOONAWALA, F. W. STANCHELL, D. R. STEVENSON, G. A. THORNE, T. T. VANDERGRAAF, P. VILKS, S. H. WHITAKER, *The Disposal of Canada's Nuclear Fuel Waste: The Geosphere Model for Postclosure Assessment*, AECL-10719, COG-93-9, Whiteshell Laboratories, Pivawa, Manitoba, R0E IL0, 1994.

[14] P. A. DOMENICO AND F. W. SCHWARTZ, Physical and Chemical Hydrogeology, John Wiley and Sons Inc., 1990.

[15] W. S. DUNBAR AND A. D. WOODBURY, *Application of the Lanczos Algorithm to the solution of the groundwater flow equation*, Water Resources Research, No.25, pp. 551–558, 1989.

[16] W. S. DUNBAR, A. D. WOODBURY AND B. NOUR-OMID, *Comment on "On time integration of groundwater flow equations by spectral methods" by G. Gambolati*, Water Resources Research, Vol. 30, No. 7, P. 2347–2352, July 1994.

[17] H. C. ELMAN, Y. SAAD AND P. E. SAYLOR, *A hybrid Chebyshev Krylov subspace algorithm for solving nonsymmetric systems of linear equations.* Tech. Report 301, Department of Computer Science, Yale University, New Haven. CT. 1984; SIAM J. Sci. Statist. Comput., 7, pp. 840–855, 1986.

[18] R. FLETCHER, *Conjugate Gradient Methods for Indefinite Systems*, Lecture Notes in Mathematics 506, springer-Verlag, Berlin, Heidelgerg, New York, pp. 73–89, 1976.

[19] R. W. FREUND, M. H. GUTKNECHT AND N. M. NACHTIGAL, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., Vol. 14, No. 1, pp. 137–158, Jan. 1993.

[20] E. GALLOPOULOS AND Y. SAAD, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Stat. Comput., Vol.13, No.5, pp. 1236–1264, 1992.

[21] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd edition, The Johns Hopkins University Press, Baltimore and London, 1989.

[22] B. W. GOODWIN, D. B. MCCONNELL, T. H. ANDRES, W. C. HAJAS, D. M. LENEVEU, T. W. MELNYK, G. R. SHERMAN, M. E. STEPHENS, J. G. SZEKELY, P. C. BERA, C. M. COSGROVE. K. D. DOUGAN, S. B. KEELING, C. I. KITSON, B. C. KUMMEN, S. E. OLIVER, K. WITZKE, L. WOJCIECHOWSKI, A. G. WIKJORD, *The Disposal of Canada's Nuclear Fuel Waste: Postclosure Assessment of a Reference System*, AECL-10717, COG-93-7, Whiteshell Laboratories, Pivawa, Manitoba, R0E IL0, 1994.

[23] E. GRIMME, D. SORENSEN AND P. V. DOOREN, *Stable Partial Realizations via an Implicitly Restarted Lanczos Method*, Proceedings of the American Control Conference, V.3, pp. 2814–2818, 1994.

[24] J. CULLUM AND R. A. WILLOUGHBY, *A practical procedure for computing eigenvalues of large sparse nonsymmetric matrices*, in J. CULLUM AND R. A. WILLOUGHBY (eds.), *Large Scale Eigenvalue Problems*, North-Holland, Amsterdam, 1986.

[25] K. K. GUPTA AND C. L. LAWSON, *Development of a block Lanczos algorithm for free vibration analysis of spinning structures*, Int. j. numer. methods eng., 26, pp. 1029–1037, 1988.

[26] M. GUTKNECHT, *A completed theory of the unsymmetric Lanczos Process and related algorithms, part I*, SIAM J. Matrix Anal., 13, pp. 594–639, 1992.

[27] M. GUTKNECHT, *A completed theory of the unsymmetric Lanczos Process and related algorithms, part II*, SIAM J. Matrix Anal., 15, pp. 15—58, 1994.

[28] M. HARADA, P. L. CHAMBRE, M. FOGLIA, K. HIGASHI, F. IWAMOTO, D. LEUNG, T. H. PIGFORD AND D. TING, Migration of radionuclides through sorbing media: analytical solutions-I, Battelle Office of Nuclear Waste Isolation, Technical Report ONWI-359.

[29] W. HOFFMANN, *Iterative algorithms for Gram-Schmidt orthogonalization*, Computing 41, pp. 335–348, 1989

[30] P. S. HUYAKORN AND G. F. PINDER, *Computational Methods in Subsurface Flow*, Academic Press, New York, 1983.

[31] W. D. JOUBERT, *Lanczos methods for the solution of nonsymmetric systems of linear equations*, SIAM J. Matrix Anal. Appl., Vol.13, No.3, pp. 926–943, July 1992.

[32] W. D. JOUBERT, *Generalized conjugate Gradient and Lanczos methods for the solution of nonsymmetric systems of linear equations* Ph. D. Dissertation (1990), the University of Texas at Austin, UMI Dissertation Services, Bell & Howell company, Michigan, 1994.

[33] S. A. KHARCHENKO AND A. YU. YEREMIN, *Eigenvalue translation based preconditioners for the* GMRES(k) *method*, Technical Report EM-RR 2/92, Elegant Mathematics, Inc., Feb. 1992.

[34] H. M. KIM AND R. R. CRAIG, JR, *Structural dynamic analysis using an unsymmetric block Lanczos algorithm*, Int. j. numer. methods eng., 26, pp. 2305–2318, 1988.

[35] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards 45, pp. 255–282, 1950.

[36] C. LANCZOS, *Solution of systems of linear equations by minimized iterations*, Ibid., 49, pp. 33–53, 1952.

[37] J. W. H. LIU AND A. GEORGE, *Computer solution of large sparse positive definite systems*, Englewood Cliffs, N. Y., Prentive–Hall, 1981.

[38] T. A. MANTEUFFEL, *Adaptive procedure for extimating parameters for the nonsymmetric Tchebychev iteration*, Numer., Math., 31, pp. 183–208. 1978.

[39] T. A. MANTEUFFEL, *Tchebychev iteration for nonsymmetric linear systems*, Numer., Math., 28, pp. 307-327, 1977.

[40] J. A. MEIJERINK AND H. A. VAN DER VORST, *Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems*, J. Comp. Phys., 44, pp. 134–155, 1981.

[41] A. NAUTS AND R. E. WYATT, *Theory of laser-module interaction: The recursive-residue-generation method*, Phys. Rev., 30, pp. 872–883, 1984.

[42] B. NOUR-OMID AND R. W. CLOUGH, *Dynamic Analysis of Structures Using Lanczos Coordinates*, Earthquake Engineering Structural Dynamic, No.12, pp. 565–577, 1984.

[43] B. NOUR-OMID, *Lanczos method for heat conduction analysis*, International Journal for Numerical Methods in Engineering, Vol. 24, pp. 251–262, 1987.

[44] B. NOUR-OMID, *Application of the Lanczos method*, Computer Physics Communications, 53, pp. 157–168, 1989.

[45] B. NOUR-OMID, W. S. DUNBAR AND A. D. WOODBURY, *Lanczos and Arnoldi Methods for the Solution of Convection-Diffusion Equations*, Computer Methods in Applied Mechanics and Engineering, 88, pp. 25–95, 1991.

[46] B. NOUR-OMID AND M. E. REGELBRUGGE, *Lanczos method for dynamic analysis of damped structural systems*, Earthquake Engineering Structural Dynamic, No.18, pp. 1091–1104, 1989.

[47] B. NOUR-OMID, W. S. DUNBAR AND A. D. WOODBURY, *Ordered modified Gram-Schmidt orthogonalization*, SIAM Journal on Matrix Analysis and Applications (to appear).

[48] O. OSTERBY AND Z. ZLATEV, *Direct methods for sparse matrices*, Berlin, N. Y., Springer–Verlag, 1983.

[49] C. C. PAIGE AND M. A. SAUNDERS, *LSQR: An algorithm for sparse linear equations and aparse least squares*, ACM Trans. Math. Software, 8, pp. 43–71, 1982.

[50] T. J. PARK AND J. C. LIGHT, *Unitary quantum time evolution by iterative Lanczos reduction*, J. Chem. Phys., 85, pp. 5870–5876, 1986.

[51] B. N. PARLETT, D. R. TAYLOR AND Z. A. LIU, *A Look-Ahead Lanczos algorithm for Unsymmetric matrices*, Mathematics of Computation, Vol.44, No.169, pp. 105–124, Jan.1985.

[52] G. F. PINDER AND W. G. GRAY, *Finite element simulation in surface and subsurface hydrology*, Academic Press, New York, 1977.

[53] C. RAJAKUMAR AND C. R. ROGERS, *The Lanczos algorithm applied to unsymmetric generalized eigenvalue problem*, International Journal for Numerical Methods in Engineering, Vol. 32, pp. 1009-1026, 1991.

[54] Y. SAAD, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal. Vol.29, No.1, pp. 209-228, Feb. 1992.

[55] Y. SAAD, *Projection and Deflation Methods for Partial Pole Assignment in Linear State Feedback*, IEEE Transactions on Automatic Control, Vol.33, No.3, March 1988.

[56] P. SONNEVELD, CGS, *a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM Sci. Stat. Comput. Vol 10, No. 1, pp. 36-52, Jan. 1989.

[57] E. A. SUDICKY, *The Laplace Transform Galerkin Technique: A Time–Continuous Finite Element Theory and Application to Mass Transport in Groundwater*, Water Resources Research, Vol. 25, No. 8, P. 1833–1846, Aug. 1989.

[58] L. R. TOWNLEY, J. L. WILSON, *Description of and User's Manual for a Finite Element Aquifer Flow Model* AQUIFEM-1, Technology Adaptation Program Report No. 79-3, MIT, Cambridge, Massachusetts, 1980.

[59] P. W. VINSOME, *Orthomin, an iterative method for solving sparse sets of simultaneous linear equations*, paper SPE 5729, Society of Petroleum Engineers of AIME, 1976.

[60] H. F. WALKER, *Implementation of the* GMRES *method using Householder transformations*, SIAM Sci. Stat. Comput., Vol. 9, No. 1, January 1988.

[61] P.WESSELING AND P. SONNEVELD, *Numerical Experiments with a Multiple Grid and a Preconditioned Lanczos type Method*, Lecture Notes in Mathematics 771, Springer-Verlag, Berlin, Heidelberg, New York, pp. 543–562, 1980.

[62] O. WIDLUND, *A Lanczos method for a class of nonsymmetric systems of linear equations*, SIAM Numer. Anal. Vol. 15. No. 4, August 1978.

[63] J. H. WILKINSON, *The algebraic eigenvalue problem*, Claredon Press, Oxford, 1965.

[64] J. H. WILKINSON, *A priori error analysis of algebraic processes*, Proc. International Congress Math., (Moscow: Izdat. Mir., 1968), pp. 629—639.

[65] A. D. WOODBURY, W. S. DUNBAR AND B. NOUR-OMID, *Application of the Arnoldi Algorithm to the Solution of the Advection-Dispersion Equation*", Water Resources Research, Vol.26, No.10, pp. 2579–2590, 1990.

[66] Q. YE, *A Breakdown-Free Variation of the Nonsymmetric Lanczos Algorithms*, Mathematics of Computation, 60, No.205, pp. 179–207, 1994.

[67] D. M. YOUNG, *Iterative solution of large linear system*, New York, Academic Press, 1971.

[68] D. M. YOUNG AND K. C. JEA, *Generalized conjugate gradient acceleration of nonsymmetrizable iterative method*, Linear Algebra Appl., 34, pp. 159–194, 1980.

[69] O. C. ZIENKIEWICZ AND R. L. TAYLOR, *Finite element method*, 4th edition, Vol. 1 and Vol. 2, McGraw-Hall Book Company, 1991.

[70] Z. ZLATEV, *Computational methods for general sparse matrices*, Dardrecht, Boston, Kluwer Academic, 1991.