NOVEL OPTIMISATION METHODS FOR NUMERICAL INVERSE PROBLEMS

Na Tian

A thesis submitted in partial fulfilment of the requirements of

the University of Greenwich

for the Degree of Doctor of Philosophy

Department of Mathematical Sciences

School of Computing and Mathematical Sciences

University of Greenwich

Park Row, London SE10 9LS

2011

DECLARATION

I certify that this work has not been accepted in substance for any degree, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigation except where otherwise identified by references and that I have not plagiarised the work of others.

Student:

Na Tian

Supervisor:

Professor Choi-Hong Lai

Second Supervisor:

Professor Koulis Pericleous

ACKNOWLEDGEMENTS

The author wishes to express her gratitude to the University of Greenwich for the sponsorship of this research.

I would like to thank my supervisors Professor Choi-Hong Lai and Professor Koulis Pericleous for their concern and guidance during the various stages of the research process.

I would like to thank Dr. Jun Sun, Jiangnan University, China, for the inspiring discussions on the improvements of the quantum-behaved particle swarm optimisation algorithms.

Last but not least, I would like to thank my parents and my husband for their full support. To them all, I dedicate this thesis.

Na Tian

London, UK

2011

PUBLICATIONS

1. N. Tian, W.B. Xu, J. Sun and C.-H. Lai, Estimation of heat flux in inverse heat conduction problems using quantum-behaved swarm optimization, Journal of Algorithms and Computational Technology, 2010, 4 (1): 25-46.

   ---refer to section 4.6 and section 5.2.

2. N. Tian, J. Sun, W.B. Xu and C.-H. Lai, An improved quantum-behaved particle swarm optimization with perturbation operator and its application in estimating groundwater contaminant source, Inverse Problems in Science and Engineering, 2011,19 (2): 181-202.

   ---refer to section 4.4.1 and section 5.4.

3. N. Tian, J. Sun,W.B. Xu and C.-H Lai, Estimation of unknown heat source function in inverse heat conduction problem using quantum-behaved particle swarm optimization, International Journal of Heat and Mass Transfer, 2011, 54(17-18): 4110-4116.

   ---refer to section 5.3.

4. N. Tian, J. Sun, W.B. Xu and C.-H. Lai, Quantum-behaved particle swarm optimization with ring topology and its application in estimating temperature-dependent thermal conductivity, Numerical Heat Transfer, Part B: Fundamentals, 2011, 60 (2): 73-95.

   ---refer to section 4.4.2 and section 5.6.

5. N. Tian, C.-H. Lai, K. Pericleous, J. Sun and W.B. Xu, Contraction-expansion coefficient learning in quantum-behaved particle swarm optimization, International Symposium on Distributed Computing and Applications to Business Engineering and Science 2011.

   ---refer to section 4.4.4.

ABBREVIATIONS

GA:	genetic algorithm
PSO:	particle swarm optimisation
QPSO:	quantum-behaved particle swarm optimisation
QPSO-PER:	QPSO with perturbation operation
SQPSO:	QPSO with ring topology
QPSO1:	QPSO with linearly decreasing parameter
QPSO2:	QPSO with constant parameter
SQPSO1:	QPSO with ring topology and linearly decreasing parameter
SQPSO2:	QPSO with ring topology and constant parameter
GQPSO:	QPSO with Gaussian mutation
CQPSO:	QPSO with Cauchy mutation
MGQPSO:	QPSO with Gaussian mutation on the mean best position
GGQPSO:	QPSO with Gaussian mutation on the global best position
MCQPSO:	QPSO with Cauchy mutation on the mean best position
GCQPSO:	QPSO with Cauchy mutation on the global best position
QPSO-CON:	QPSO with constant parameter
QPSO-LIN:	QPSO with linearly decreasing parameter
QPSO-COS:	QPSO with consine function parameter
QPSO-ANN:	QPSO with annealing parameter
BEM:	boundary element method
FDM:	finite difference method
IHCP:	inverse heat conduction problem
MRE:	minimum relative entropy
CGM:	conjugate gradient method
SDM:	steepest descent method
HM1:	hybrid method with polynomial approximation
HM2:	hybrid method with interpolation

ABSTRACT

Inverse problems involve the determination of one or more unknown quantities usually appearing in the mathematical formulation of a physical problem. These unknown quantities may be boundary heat flux, various source terms, thermal and material properties, boundary shape and size. Solving inverse problems requires additional information through in-situ data measurements of the field variables of the physical problems. These problems are also ill-posed because the solution itself is sensitive to random errors in the measured input data. Regularisation techniques are usually used in order to deal with the instability of the solution. In the past decades, many methods based on the nonlinear least squares model, both deterministic (CGM) and stochastic (GA, PSO), have been investigated for numerical inverse problems.

The goal of this thesis is to examine and explore new techniques for numerical inverse problems. The background theory of population-based heuristic algorithm known as quantum-behaved particle swarm optimisation (QPSO) is re-visited and examined. To enhance the global search ability of QPSO for complex multi-modal problems, several modifications to QPSO are proposed. These include perturbation operation, Gaussian mutation and ring topology model. Several parameter selection methods for these algorithms are proposed. Benchmark functions were used to test the performance of the modified algorithms. To address the high computational cost of complex engineering optimisation problems, two parallel models of the QPSO (master-slave, static subpopulation) were developed for different distributed systems. A hybrid method, which makes use of deterministic (CGM) and stochastic (QPSO) methods, is proposed to improve the estimated solution and the performance of the algorithms for solving the inverse problems.

Finally, the proposed methods are used to solve typical problems as appeared in many research papers. The numerical results demonstrate the feasibility and efficiency of QPSO and the global search ability and stability of the modified versions of QPSO. Two novel methods of providing initial guess to CGM with approximated data from QPSO are also proposed for use in the hybrid method and were applied to estimate heat fluxes and boundary shapes. The simultaneous estimation of temperature dependent thermal conductivity and heat capacity was addressed by using QPSO with Gaussian mutation. This combination provides a stable algorithm even with noisy measurements. Comparison of the performance between different methods for solving inverse problems is also presented in this thesis.

Contents

FIGURES

TABLES

Chapter 1 **INTRODUCTION**

This chapter gives the fundamentals of inverse problems. For simplicity and easy explanation, heat conduction problems are often used in this thesis for various illustration and discussion. Numerical methods used to solve inverse problems are discussed along with the objectives and the outline of this thesis.

1.1 Inverse Problems

Inverse problems arise in many branches of science and mathematics, including environmental science, water pollution, medical analysis, etc. They may be described as problems where results, or consequences are known, but not the cause. Solutions of an inverse problem involve determining unknown causes based on observations of their effects. This is in contrast to the corresponding direct problem, to which solutions involve finding effects based on a complete description of their causes.

The inverse problem has a wide range of applications, such as seismic surveys of locating ground water, oil and gas resources; medical tomography of reconstructing the internal structure of an organ; non-destructive evaluation of materials; electromagnetic remote sensing; determination of the Earth's interior structure, etc.

In this thesis, the research focuses on the inverse heat conduction problems (IHCP) and contaminant propagation problems, which arise in the modelling and control of many processes with heat propagation in thermophysics and flow in continuous media. An excellent introductory work of IHCP can be found in [1]. Recently in [2], Taler and Duda discussed the theoretical basis, analytical and numerical methods of the IHCP. The IHCP

finds wide applications in quenching and many other thermal-related industries, one of which is the determination of the surface heat flux histories of reentering heat shields in aerospace industry.

A direct heat conduction problem is fully defined by the following: the governing partial differential equation (elliptic for steady heat conduction or parabolic for unsteady heat conduction), the thermal properties of the material (the coefficients in the governing equation), the initial conditions and the boundary conditions (Dirichlet, Neumann or Robin type), the shape and size of the domain, and the internal heat source distribution. In contrast, an IHCP is defined as the estimation of any unavailable information of the above from one or more measured temperatures within the heat conducting body. Here 'estimation' is used because in measuring the temperatures, errors are always present to some extent and they affect the accuracy of the calculation. This can happen in a number of practical situations as described below.

For example, it is often difficult or even impossible to use sensors to measure temperatures and heat fluxes on certain boundaries such as those of combustion chambers. The placement of thermal sensors may also be impossible because of the prohibitively small size of the domain, as is the case of a computer chip or in the coolant flow passage of a turbine blade. Thus, in many cases, solutions of an ill-posed boundary condition problem, where the size and the shape of the domain are known, while thermal boundary conditions are unavailable on parts of the boundary and over-specified on the rest of the boundary, are required.

A similar type of problem arises when one requires the unknown heat source in the domain. Using sensors in a highly volatile environment, such as in the case of a buried toxic waste site, temperature measurements are impossible to obtain. Thus, measurements of both temperature and heat flux data on certain part of the boundary may be required in order to solve this inverse heat source problem.

Another class of inverse problems arises when the size and shape of some parts of the domain are unknown. In order to determine the unknown boundaries of the domain, additional boundary conditions must be provided in the form of independently specified Dirichlet and Neumann boundary conditions at the same points of the known boundary. Thus, when the thermal boundary conditions are over-specified on a part of the boundary and the

remaining boundary is not known, the problem is referred to as an ill-posed inverse shape design problem.

The unsteady (transient) inverse heat conduction problems (UIHCP) represent a subclass of ill-posed problems which have been extensively investigated. The UIHCP involves an estimation of the initial conditions (temperatures or heat fluxes) or an estimation of unsteady boundary conditions utilizing measured interior temperature histories. The major concern when attempting to solve the UIHCP computationally has been with the automatic filtering of noisy data in the discrete thermocouple measurements. The measurement data errors, as well as round-off errors, are amplified by the typical UIHCP algorithms.

Heat transfer across materials depends on thermo-physical properties such as the thermal conductivity and heat capacity per unit volume. These properties have a significant influence on the temperature distribution and heat flow rate when the material is heated and also in the analysis of thermal instability problems. Direct measurement of the thermal properties is always impractical, since they are often temperature dependent. An efficient and economical method for estimating the thermal properties is required especially in the material design industry.

The heat transfer coefficient, in thermodynamics, mechanical and chemical engineering is used in calculating the heat transfer, typically by convection or phase change between a fluid and a solid. The accurate knowledge of the heat transfer coefficient at the surface of the plate is important in many engineering applications, including the cooling of continuously cast slabs and electronic chips. Many more fields of science and technology, such as astronomy, chemistry and medicine require solutions to inverse problems. All the above applications require the development of accurate, fast, efficient and stable algorithms to solve the relevant inverse problems.

Environmental contamination is a widespread problem that may affect the use of environmental resources such as a groundwater aquifer or a surface water body. Identifying contaminant sources in groundwater is important for developing effective remediation strategies and finding responsible parties in a contamination incident. Groundwater contamination broadly defines any constituent that reduces the quality of groundwater. Contamination can be chemical, physical or biological. Chemical contamination can be broken down further into soluble components and non-aqueous phase liquid components. Soluble components are dissolved in the groundwater and are transported with the

groundwater as it moves. Non-aqueous phase liquids are bodies of liquid that are separate from the water and are generally not transported with bulk groundwater movement. If the initial and boundary conditions, model parameters and contaminant release history are known, the advection-dispersion equation can be solved directly using analytical techniques or numerical simulations to obtain the distribution of contaminant concentration. This process is called a forward advection-dispersion problem which has a unique solution if it is well-posed. In contrast, the inverse advection-dispersion problem for groundwater models may involve the determination of the unknown time-dependent contaminant release history from the knowledge of concentration measurements taken within the medium.

1.2 Methods for Solving Inverse Problems

The aim of this thesis is to examine fast and reliable numerical approaches of either iterative or non-iterative type for various inverse problems arising in heat conduction or contaminant flow formulated as optimisation problems.

Solving inverse problems is complicated due to their ill-posedness. A problem is well-posed if the solution (1) exists, (2) is unique and (3) is stable. If one of these three conditions is not satisfied, the problem is said to be ill-posed. Analysis from the perspective of a partial differential equation, an integral equation and a set of linear algebraic equations can be found in [1]. In many inverse problems, including those examples in this thesis, the existence and the uniqueness of the solution are well-established according to the a priori knowledge of the engineering problems. However, solutions to the inverse problems are extremely sensitive to measurement errors [1]. In other words, an arbitrarily small perturbation of the measured data may produce a large difference in the output solution. Therefore, any algorithm developed for inverse problems should satisfy the stability condition. There are a number of procedures that have been developed for the solution of ill-posed problems in general. One of these procedures is known as the regularisation technique developed by Tikhonov and Arsenin [4] and is used to reduce the sensitivity of ill-posed problems to measurement errors.

In general, inverse problems can be solved as a parameter estimation problem or as a function estimation problem [1]-[5]. If information is available on the functional form of the unknown quantity, the inverse problem can be reduced to the estimation of a few unknown parameters [5]. On the other hand, if no prior information is available on the functional form of the unknowns, the inverse problem becomes a function estimation problem in an infinite

dimensional space of functions. In real problem solving, the functional form is usually unknown causing difficulties in the actual inverse solution process. This thesis emphasises the numerical aspects of function estimation rather than parameter estimation.

In [1], Beck et al. gave the classification of methods for solving inverse heat conduction problems (IHCPs). One classification relates to the ability of a method to treat nonlinear as well as linear IHCPs. This thesis pays particular attention to algorithms that can be employed for both linear and nonlinear problems. Some methods are inherently linear such as those based on the Laplace transform, which are not considered because the nonlinear case is more important for industrial applications; Methods of solving direct heat conduction problems include Duhamel's theorem, finite differences, finite elements, finite volumes and boundary elements. The use of Duhamel's theorem restricts IHCPs to the linear case, whereas the other procedures can treat the nonlinear problems; the time domain utilized in IHCPs can also be used to classify the methods. Three time domains have been proposed: (1) restricted to the current time only, (2) using the current time and a few future time steps, and (3) the whole time domain. The use of measurements only at the current time with a single sensor allows the calculated temperature to match the corresponding measured temperature in an exact manner. Such exact match is intuitively appealing but the algorithms based on it are extremely sensitive to measurement errors. In the second method, a few future temperatures are used. The associated algorithms are called 'sequential'. Great improvements are obtained compared with exact matching in reduced sensitivity to measurement errors and in the much smaller time steps that are possible. The whole domain estimation procedure is also very powerful because very small time steps can be taken; The last classification is relative to the dimensionality of the IHCP. In the use of Duhamel's theorem, the physical dimension of the problem is not of concern. When finite difference or other methods are employed, the dimensionality of the problem depends on the number of space coordinates needed to describe the heat-conducting body.

Several criteria were proposed to evaluate the methods for solving IHCP in [6]. However these criteria could well be generalised to other types of inverse problems. These critera are listed as follows:

(1) The estimated solution should be accurate if the measured data are of high accuracy.

(2) The method should be insensitive to measurement errors.

(3) The method should be stable for small time steps or intervals.

(4) Field variable measurements from one or more sensors should be permitted.

(5) The method should not be restricted to any fixed number of observations.

(6) Field variable dependent properties should be permitted.

(7) The method should be easy to program.

(8) The computer cost should be moderate.

(9) The method should permit extension to more than one unknown quantities.

Most of the IHCPs that have been investigated so far are concerned with the estimation of boundary heat fluxes. The sequential function specification method proposed by Beck et al. in [1] is a computationally efficient method to address the diffusive nature of transient heat conduction. Conjugate gradient method (CGM), as a whole domain iterative regularisation function estimation method, was used to solve various inverse problems of estimating time-varying heat flux as well as other unknown quantities [59]-[61].

Determination of time varying heat source in IHCPs was also addressed in many papers. In [90], temperature measurements in the whole domain are required to get the solution of the inverse heat source problem, which is usually impractical in engineering. In [91]-[94], GA was used to address the inverse parameter and function estimation problems, in which every chromosome represents a candidate solution of the unknown quantity in hand. Then during the process of evolution, selection, crossover, mutation operators are applied to the chromosomes, until the optimum is achieved. The advantages of the stochastic search methods are that they do not require the gradient computation and the choice of initial guess.

Estimation of temperature dependent thermal properties is difficult because of the nonlinearity of the inverse problem. Various methods were proposed to address this problem. CGM [64]-[68] is the most commonly used method, in which, the temperature-dependent properties are treated as function of space and time. In [71], Terrola formulated the inverse problem as an optimisation problem and applied the Davidon-Fletcher-Powell method to solve the optimisation problem to determine the temperature dependent thermal conductivity. Kim et al. [72] formulated the problem to find the solution through the direct integral method, which requires the material to be homogeneous. Yeung and Lam [73], Chang and Chang [74]-[75] estimated the thermal conductivity by using non-iterative methods, in which the governing equation of heat conduction is discretised into a system of linear equations using the temperature measurements at the discrete grid points, and then the unknown thermal

conductivities can be obtained by solving the system of equations directly. But actually, temperature measurements of the whole domain are usually difficult to obtain. In [76], a sensitivity equation was used to estimate the parameters in the known functional form of thermal conductivity. In [77]-[82], genetic algorithms (GA) were applied to solve the IHCPs, but the functional forms of the unknown quantities are all required to be known a priori. In [83]-[84], the particle swarm optimisation (PSO) was used to solve the inverse heat transfer problems of determining the heat source and unknown variables. However PSO has not been applied to estimate the temperature-dependent thermal properties.

Simultaneous determination of two unknown quantities is even more difficult, e.g. temperature dependent thermal conductivity and heat capacity. CGM was used to simultaneously estimate temperature dependent thermal conductivity and heat capacity, but the most difficult part is the choice of the initial guess of the two unknown quantities [67]. In [85]-[86], the direct integration approach was applied to estimate the thermal conductivity and heat capacity, which vary linearly with respect to temperature. Unfortunately for many other functional forms, the direct integration approach does not work. Flach and Ozisik [87] employed the least-squares method to estimate spatially varying thermal conductivity and heat capacity. In [88], a hybrid numerical algorithm of Laplace transform technique and the control-volume method are proposed to estimate the temperature dependent thermal conductivity and heat capacity. Genetic algorithm (GA) was proposed to identify the temperature dependent thermal properties [79], [89], in which, the functional form is assumed to be known. These problems are essentially parameter estimation problems instead of function estimation problems.

In the inverse problems of identifying boundary shapes, the boundary element method (BEM) is used to solve the direct problems, which may require much computational CPU time if the number of boundary elements becomes large. In [95], Nachaoui estimated the boundary shape using CGM, but the two end points are always difficult to find. Mera et al. used GA to solve the boundary detection problem in [96], which requires the information of the functional form of the boundary shape. In [97]-[98], the inclusion detection problems are investigated by using PSO.

The heat transfer coefficient at the surface of a plate, where heat lost is due to convection to surrounding cooling fluid, was estimated by three versions of CGM in [99]. Chen and Wu [100] applied a hybrid scheme of Laplace transform, finite difference and least-squares

method in conjunction with a sequential-in-time concept, cubic spline and temperature measurements to predict the heat transfer coefficient distribution on a boundary surface. Slodicka et al. [101] used the BEM and Tikhonov regularisation to construct the time-dependent heat transfer coefficient. Chantasiriwan [102] used the sequential function specification method with the linear basis function and an assumption of linearly varying future boundary heat flux or temperature components to estimate the time-dependent Biot number.

During the last decade, inverse problems of groundwater contaminant transport have received wide attention. A concise review of the most relevant work is given in [103]. Gorelick et al. [104] used the least-squares and linear programming to determine the location and strength of the source pollutant in the field. Their numerical model was tested on two sets of hypothetical data representing a steady-state and a transient case. The model assumed other transport parameters are previously known. Wagner [105] estimated the transport parameters and contaminant source simultaneously. Zou and Parr [106] developed an analytical solution to determine the longitudinal and transverse dispersivities. Skaggs and Kabala [107]-[109] solved the inverse source problem with Tikhonov regularisation and the method of quasi-reversibility. Woodbury and Ulrych [110]-[111] solved the problem using minimum relative entropy (MRE) inversion. Snodgrass and Kitanidis [112] used a geostatistical approach to solve the same problem. In [113]-[115], some optimisation methods, such as CGM, are used to solve various inverse problems, they converge fast but strongly depend on initial guess and can't guarantee the global optimum. The Tikhonov regularisation method is more robust in solving the inverse problem with noisy sampling. However, it can not be used to reconstruct the non-smooth source history efficiently. The MRE method is a gradient-based approach, which is more efficient in dealing with the source history with many peaks. On the other hand, it is not effective for problems that contain measurement error of unknown magnitude. Furthermore, the gradient computation of the objective function is very complicated, and the gradient may not even exist for some objective functions. The determination of the Lagrange multiplier is also a key difficulty in the MRE method. For these types of problems, heuristic global search approaches such as particle swarm optimisation (PSO) are more effective. Bharat et al. first use PSO [116] to solve the inverse source problem in groundwater contaminant, but PSO does not seem to stabilise the inverse solution.

Most of the above methods for inverse problems are based on the nonlinear least-squares method [57] which minimises the difference between the experimental measurements and the calculated responses of the system. Such inverse problems can be treated as an optimisation problem [33]. Despite their similarities, inverse and optimisation problems are conceptually different. Inverse problems concern with the identification of unknown quantities appearing in the mathematical model of physical problems, by using measurements of the system response. On the other hand, optimisation problems generally deal with the minimization or maximization of a certain objective function, in order to find the best design variables that will result in desired state variables. For example, the solution technique for an inverse problem is required to cope with instabilities resulting from the noisy measured input data, while for an optimisation problem the input data is given by the desired response of the system. In contrast to the inverse problems, the solution uniqueness may not be an important issue for optimisation problems.

This thesis addresses solution methodologies for numerical inverse problems being treated as single-objective optimisation problems based on minimisation techniques. Several gradient-based and stochastic techniques are re-visited, together with their basic implementation steps and algorithm procedures. Two deterministic methods, Steepest Descent Method (SDM) and Conjugate Gradient Method (CGM) are presented. In addition, the thesis gives some of the most promising stochastic approaches, such as Genetic Algorithm (GA) [29]-[31], Particle Swarm Optimisation (PSO) [13] and Quantum-behaved Particle Swarm Optimisation (QPSO) [23].

QPSO, a novel variant of PSO first introduced by Sun in 2004 [23]-[25], makes the assumption that all particles have a quantum behaviour instead of the classical Newtonian dynamics that is assumed in original PSO. In QPSO system, the wave-function is used to describe the state of the particles instead of the position and velocity. By employing the Monte Carlo method, the iterative equation of particles position is derived from the quantum probability density function. Comparing with the PSO algorithm, it only requires to update the position of the particles without velocity and has fewer algorithmic parameters to control. When the QPSO was tested on a set of benchmarking functions, it demonstrated superior performance as compared to the PSO [23]. Moreover, it does not require gradient information of the objective function, but only its values, and it uses only elementary mathematical operators. The QPSO method has been tested to be an efficient method for many optimisation problems. Compared with the traditional gradient methods which go from one initial

approximation in the search domain to another approximation at every iteration, the QPSO method requires to search for as many solutions as possible simultaneously and thus has the potential to give unbiased estimation. This provides a better avenue of finding the global optimum in the search space. In this thesis, the author gives a new insight into the background reasoning of the quantum behaviour of particles and the use of delta-well in the deviation of the method.

Deterministic methods are in general computationally faster that stochastic methods, but they might converge to local optima instead of the global optima. On the other hand, stochastic algorithms are able to converge to global optima, although they are computationally slower than the deterministic algorithms. Indeed, the stochastic algorithms may require thousands of evaluations of the objective function and, in some complicated cases, become non-practical. In order to overcome these difficulties, a hybrid method, which takes advantages of the robustness of the stochastic methods (e.g. QPSO) and of the fast convergence of the deterministic methods (e.g. CGM), is proposed. Since the particles are relatively independent in the searching process and the only shared information is the global best position, the QPSO can be parallelized and dispatch parallel processing for the time-consuming fitness evaluations. In this thesis, two parallel models of QPSO, master-slave parallelization (synchronous and asynchronous) and static subpopulation parallelization, are considered.

In QPSO algorithm, like most of the population-based evolutionary algorithms, the loss of diversity in the population is also inevitable due to the collectiveness. During the latter search period, the particles are investigated to cluster together and its search area is so limited that the whole swarm can easily get trapped into a local minimum. In order to help the particles avoid premature convergence and increase the diversity of the population, several improvements to the QPSO algorithm in different aspects are considered in this thesis.

1.3 Objectives

The main objectives of this thesis are as follows:

(1) Examine properties of gradient-based methods. Investigate the advantages and disadvantages of CGM.

(2) Search for algorithms that do not depend on initial guess, even allow random initial approximations.

(3) Enhance the global search ability of the QPSO method.

(4) Develop parallel methods for QPSO to address the high computational cost.

(5) Examine possibilities of a hybrid method combining advantages of different methods.

(6) Applications to various industrial related problems.

1.4 Outline of the Thesis

The remaining part of the thesis is organized as follows. In chapter 2, the mathematical modelling of the inverse problem and the regularisation techniques are introduced. The chapter begins with a brief review of the partial differential equations and the corresponding analytical and numerical methods. The typical heat conduction equation is considered as an example model to illustrate the methods. The Tikhonov regularisation method is presented to deal with the ill-posed problem, e.g. instability, following with three methods for choosing the regularisation parameter, in which, L-curve method is used in this thesis.

In chapter 3, two gradient-based deterministic methods, steepest descent method (SDM) and CGM, are reviewed. Only the details of the CGM for solving the inverse problem of estimating an unknown heat flux in the example model are given because the method reduces to the SDM when the conjugate coefficient is zero.

In chapter 4, three stochastic algorithms, namely GA, PSO and QPSO are introduced. The QPSO shows superiority over GA and PSO in terms of simplicity and global search ability. The reasons behind the use of quantum theory are explored in order to provide a better background supporting the concept. Several modified QPSO methods are proposed, and the comparison of the proposed algorithms with other methods on benchmark functions is also presented. Two parallel models of QPSO are developed to reduce the computational time and increase the efficiency of the algorithm. Finally a hybrid method is developed, in which QPSO provides CGM with initial guess values.

Chapter 5 discusses the application in several industrial related inverse problems. The methods developed in chapters 3 and 4 were used to solve various inverse problems, including estimation of heat flux, heat source, temperature-dependent thermal properties and

heat transfer coefficient in heat conduction problems, and identification of contaminant source in an advection-dispersion problem. Simultaneous estimation of heat capacity and thermal conductivity and two-dimensional problems are also presented in this chapter.

Finally, chapter 6 gives the conclusions which summarize the main work and contributions of this thesis. Suggestions of future work are also given.

Chapter 2 MATHEMATICAL PRELIMINARIES

This chapter gives a brief introduction of the partial differential equations, analytical and numerical methods for the solution of the direct problems. A brief description is given of ill-posed problems, the Tikhonov regularisation technique, and methods of choosing the regularisation parameter. For the convenience of illustration, an example of a heat conduction problem is used here.

2.1 Partial Differential Equations and Direct Problems

2.1.1 Partial differential equations

Partial differential equations (PDEs) are fundamental in the modelling of many natural phenomena, such as the propagation of sound or heat, electrostatics, electrodynamics, fluid flow and elasticity. Common examples of PDEs include Laplace's equation

$$\nabla^2 u = 0 \tag{2.1}$$

the heat equation

$$\frac{\partial u}{\partial t} = \nabla^2 u \tag{2.2}$$

and the wave equation

$$\frac{\partial^2 u}{\partial t^2} = \nabla^2 u\,. \tag{2.3}$$

In this thesis, the main focus is the heat equation in transient and steady state forms. As the name suggests, Equation (2.2) describes the conduction of heat (with the dependent variable

u usually interpreted as temperature), but it is also applied to a range of physical phenomena involving diffusion.

Suppose the above PDE is defined in Ω, a bounded region in $\Re^n$, where $n = 1, 2, 3$, with boundary $\partial\Omega$. Since the heat equation is first order in time, an initial condition on the solution is required, such as

$$u(\vec{x}, 0) = u_0(\vec{x}), \tag{2.4}$$

where $\vec{x} \in \Omega$.

There are three common types of boundary conditions on $\partial\Omega$:

(i) Dirichlet condition: A fixed u which takes on prescribed values $u(\vec{x}, t) = d(\vec{x}, t)$ for $\vec{x} \in \partial\Omega$ and $t \in (0, \infty)$ is required.

(ii) Neumann condition: A fixed normal derivative of u, $\frac{\partial u}{\partial n}(\vec{x}, t) = s(\vec{x}, t)$ for $\vec{x} \in \partial\Omega$ and $t \in (0, \infty)$ is required, where $\vec{n}$ is the unit outward normal to $\partial\Omega$. A simplified version of Fourier's law of heat conduction says that the heat flux vector q at a point $\vec{x}$ at time t is given by

$$q(\vec{x}, t) = -K \frac{\partial u}{\partial n}(\vec{x}, t), \tag{2.5}$$

where K is a positive constant known as the thermal conductivity.

(iii) Robin condition: A mixture of Dirichlet and Neumann conditions $\alpha u(\vec{x}, t) + \beta \frac{\partial u}{\partial n}(\vec{x}, t) = f(\vec{x}, t)$ for $\vec{x} \in \partial\Omega$ and $t \in (0, \infty)$, where α and β are positive constants.

2.1.2 Direct problems

The problems described in the previous section are non-dimensionlised equations where model parameters, such as density, thermal conductivity, heat capacity, etc., are incorporated into the dependent variable u and the independent variable $(\vec{x}, t)$ by using suitable dimensionlisations. In mathematical physics, a direct problem is usually a problem of modelling some physical fields, processes, or phenomena (electromagnetic, acoustic, heat,

etc.). The purpose of solving a direct problem is to predict the measurable data for given values of the model parameters. The description of a direct problem includes:

(i) the equation governing the physical process;

(ii) the domain in which the process is studied;

(iii) the initial conditions (if the process is transient), and

(iv) the conditions on the boundary of the domain.

For example, the direct initial-boundary value problem for a one-dimensional heat conduction problem can be described as

$$\begin{cases} \rho C \dfrac{\partial u}{\partial t}(x,t) = K \dfrac{\partial^2 u}{\partial x^2}(x,t), & t > 0, 0 < x < 1 \\ u\big|_{t=0} = u_0, & 0 \le x \le 1 \\ -K \dfrac{\partial u}{\partial x}\Big|_{x=0} = q(t), & t > 0 \\ -K \dfrac{\partial u}{\partial x}\Big|_{x=1} = 0, & t > 0 \end{cases} \tag{2.6}$$

with density ρ, heat capacity C and thermal conductivity K, initial condition u_0 and Neumann boundary condition $q(t)$. Like most direct problems of mathematical physics, the problem is well-posed, i.e., there exists a unique solution which depends continuously on the model data.

The methods described in the following section refer to the direct problem given by Equation (2.6).

2.1.3 Methods of solutions for direct problems

Direct problems can be solved by using a variety of methods including analytical and numerical methods. Analytical methods include the classical methods of separation of variables and Laplace transforms. Numerical methods include the finite difference method, finite volume method, finite element method and boundary element method.

(1) Separation of variables

The method of separation of variables is a powerful approach designed to obtain solutions of initial and boundary value problems for some linear PDEs. Assuming it is possible to

separate the function of the independent variables into separate functions that each involves only one independent variable, the solution to $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ takes the form

$$u(x,t) = X(x)T(t), \tag{2.7}$$

where T depends only on t and X depends only on x. Substituting Equation (2.7) into the heat equation Equation (2.6) leads to

$$X\frac{dT}{dt} = T\frac{d^2X}{dx^2}, \tag{2.8}$$

which can be rearranged to give the following

$$\frac{1}{T}\frac{dT}{dt} = \frac{1}{X}\frac{d^2X}{dx^2}. \tag{2.9}$$

The left-hand side of Equation (2.9) is a function of t only, and the right hand side of Equation (2.9) is a function of x only. Since Equation (2.9) must be satisfied for all values of x and t, each side of Equation (2.9) must be equal to the same constant value A

$$\frac{1}{T}\frac{dT}{dt} = \frac{1}{X}\frac{d^2X}{dx^2} = A, \tag{2.10}$$

which leads to two separate ordinary differential equations (ODEs) given by

$$\frac{1}{T}\frac{dT}{dt} = A \tag{2.11}$$

$$\frac{1}{X}\frac{d^2X}{dx^2} = A \tag{2.12}$$

These ODEs are easy to solve with the given initial and boundary conditions.

(2) Laplace transform

In mathematics, the Laplace transform is a widely used integral transform. It is a linear operator of a function $f(t)$ with a real argument t $(t \geq 0)$ that transforms it to a function $F(s)$ with a complex argument s [142]

$$F(s) = L\{f(t)\} = \int_0^\infty f(t)e^{-st}dt, \tag{2.13}$$

$$f(t) = L^{-1}\{F(s)\} = \frac{1}{2\pi i}\int_{\mu-i\infty}^{\mu+i\infty} F(s)e^{st}ds, \tag{2.14}$$

where μ is a real number.

Laplace transform is a powerful tool for transforming PDEs into ODEs and can be used to transform initial-value problems for ODE into algebraic equations.

Consider the intial boundary-value problem in Equation (2.6), the transformations for partial derivatives are

$$L\{u_t\} = \int_0^\infty u_t(x,t)e^{-st}dt = sU(x,s) - u(x,0) \tag{2.15}$$

$$L\{u_x\} = \int_0^\infty u_x(x,t)e^{-st}dt = \frac{\partial U}{\partial x}(x,s) \tag{2.16}$$

$$L\{u_{xx}\} = \int_0^\infty u_{xx}(x,t)e^{-st}dt = \frac{\partial^2 U}{\partial x^2}(x,s) \tag{2.17}$$

where $U(x,s) = L\{u(x,t)\}$. Then Equation (2.6) becomes an ODE in x as shown below

$$\begin{cases} sU(x) - u_0 = \dfrac{d^2U}{dx^2}(x), \quad 0 < x < 1 \\ \dfrac{dU}{dx}(1) = 0 \\ \dfrac{dU}{dx}(0) = -\int_0^\infty q(t)e^{-st}dt \end{cases} \tag{2.18}$$

After the solution $U(x)$ of Equation (2.18) is obtained, the solution $u(x,t)$ of Equation (2.6) can be computed from the inverse Laplace transform Equation (2.14) as

$$u(x,t) = L^{-1}\{U(x,s)\}. \tag{2.19}$$

(3) Finite difference method

The finite difference method [34] is one of the earliest numerical methods that may be used to determine temperature values at discrete spatial nodes and temporal points. It is based on the approximation of the differential equations by finite difference equations. The resulting set of finite difference equations allows solutions to be calculated at the grid points such as those depicted in Figure 2.1.

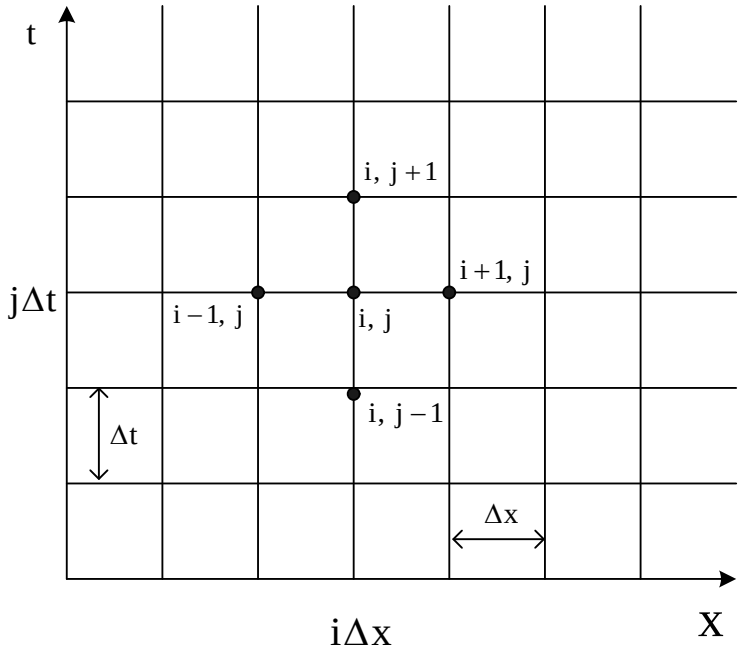


Figure 2.1: Finite difference mesh for two independent variable x and t.

A finite difference solution procedure basically involves three steps:

(i) Dividing the physical domain into grids of nodes.

(ii) Approximating the given differential equation by a finite difference equivalence that relates the approximated solutions at the grid points.

(iii) Solving the set of difference equations subject to the prescribed boundary conditions and initial conditions.

Consider the function $u(x,t)$ in Equation (2.6), its first-order spatial derivative at the grid point (x_i,t) can be approximated by the forward difference formula

$$\frac{\partial u(x_i,t)}{\partial x} \cong \frac{u(x_{i+1},t)-u(x_i,t)}{\Delta x}, \tag{2.20}$$

backward difference formula

$$\frac{\partial u(x_i,t)}{\partial x} \cong \frac{u(x_i,t)-u(x_{i-1},t)}{\Delta x}, \tag{2.21}$$

or central difference formula

$$\frac{\partial u(x_i,t)}{\partial x} \cong \frac{u(x_{i+1},t)-u(x_{i-1},t)}{2\Delta x}. \tag{2.22}$$

The second derivative of $u(x_i,t)$ can be approximated by

$$\frac{\partial^2 u(x_i,t)}{\partial x^2} \cong \frac{u(x_{i+1},t)-2u(x_i,t)+u(x_{i-1},t)}{(\Delta x)^2}. \tag{2.23}$$

Using forward difference in temporal derivative and central difference in spatial derivative, a finite difference approximation of Equation (2.6) is

$$\rho C \frac{u_i^{j+1}-u_i^j}{\Delta t} = K \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2}\ , \tag{2.24}$$

where $x_i = i\Delta x$, $i = 1,2,...,n$, $t_j = j\Delta t$, $j = 1,2,...$, u_i^j is the discretised form of $u(x_i,t_j)$.

Letting $s = \frac{K\Delta t}{\rho C(\Delta x)^2}$, Equation (2.24) can be rewritten as

$$u_i^{j+1} = su_{i+1}^j + (1-2s)u_i^j + su_{i-1}^j. \tag{2.25}$$

This is an explicit formula which can be used to compute $u(x,t+\Delta t)$ from $u(x,t)$.The major advantage of an explicit finite difference scheme is that it is relatively simple and computationally fast. However, the main drawback is that stable solutions are obtained only with the condition

$$0 < s < 0.5. \tag{2.26}$$

If this condition is not satisfied, the solution becomes unstable and oscillating.

Unlike the explicit scheme, an implicit finite difference scheme is unconditionally stable and the discretisation of Equation (2.6) becomes

$$\rho C \frac{u_i^{j+1}-u_i^j}{\Delta t} = K \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{\Delta x^2}. \tag{2.27}$$

Rearrange the above equation so that unknown terms are kept on the left and known terms are kept on the right, one obtains

$$-su_{i+1}^{j+1} + (1+2s)u_i^{j+1} - su_{i-1}^{j+1} = u_i^j. \tag{2.28}$$

A linear system of equations needs to be solved. The main advantage [35] of an implicit finite difference scheme is that there are no restrictions on the time step. The implicit method is second order accurate in space but only first order accurate in time (i.e. $O(\Delta t, \Delta x^2)$). The Crank-Nicolson scheme, which is second-order accurate both in time and in space (i.e.

$O(\Delta t^2, \Delta x^2)$), is unconditionally stable. In this scheme, the term $\frac{\partial^2 u}{\partial x^2}$ in Equation (2.6) is discretised by the average of the central difference formulae on the j^{th} and $(j+1)^{th}$ time steps, i.e.,

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{K}{2}\left(\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{\Delta x^2} + \frac{u_{i+1}^{j} - 2u_i^{j} + u_{i-1}^{j}}{\Delta x^2} \right) \tag{2.29}$$

which may be rewritten as

$$-su_{i+1}^{j+1} + 2(1+s)u_i^{j+1} - su_{i-1}^{j+1} = su_{i+1}^{j} + 2(1-s)u_i^{j} + su_{i-1}^{j} \tag{2.30}$$

Note that finite difference methods are efficient and simple to use, especially for rectangular computational domains.

(iv) Boundary element methods

The boundary element method (BEM) for the numerical solution of a linear PDE is based on an integral formulation. The two-dimensional Laplace equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{2.31}$$

is used to illustrate this technique.

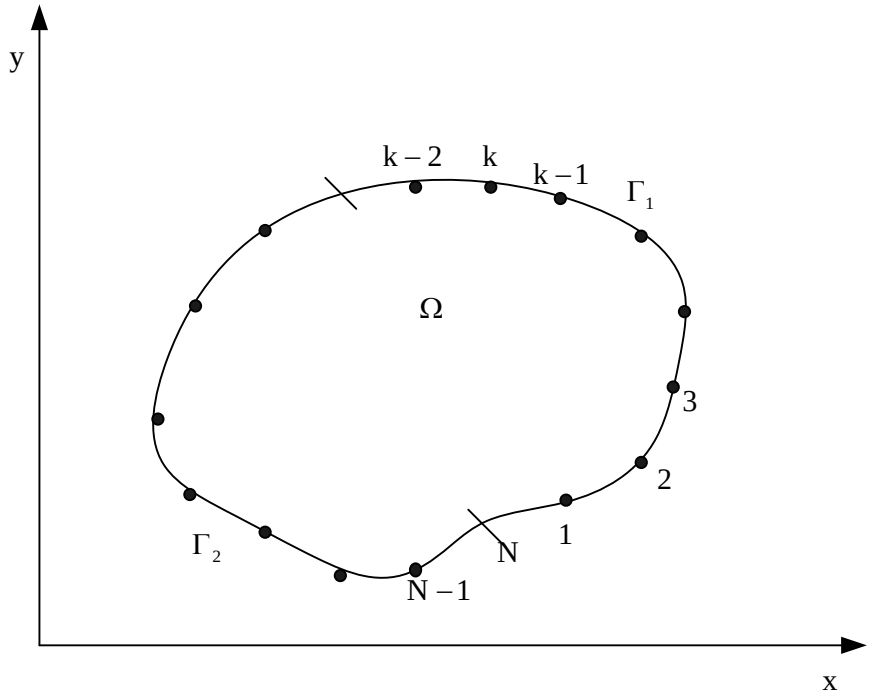


Figure 2.2: Boundary elements for a two dimensional domain

The boundary $\Gamma = \Gamma_1 \cup \Gamma_2$ in Figure 2.2 is divided into N elements. A fundamental solution of Laplace's equation in two dimensions is given by

$$u^* = \frac{1}{2\pi} \ln\left(\frac{1}{\sqrt{(x-\xi)^2 + (y-\eta)^2}} \right), \tag{2.32}$$

which satisfies Laplace's equation (Equation (2.31)) everywhere in the domain Ω except at the point (ξ, η) where it has a singularity. Applying Green's second identity, the boundary integral equation can be derived as [40]

$$\alpha u(\xi,\eta) = \int_\Gamma \left(u \frac{\partial u^*}{\partial n} - \frac{\partial u}{\partial n} u^* \right) d\Gamma \tag{2.33}$$

where
$$\alpha = \begin{cases} 0, & \text{if } (\xi,\eta) \notin \Omega \cup \Gamma \\ \frac{1}{2}, & \text{if } (\xi,\eta) \text{ lies on a smooth part of } \Gamma \\ 1, & \text{if } (\xi,\eta) \in \Omega \end{cases} \tag{2.34}$$

If everything on the right-hand side is known, we can in principle find the values of $u(\xi,\eta)$ for all points (ξ,η) inside the domain Ω. Unfortunately, not all the information required is in hand and one has to find approximations to the unknowns on the boundary. If (ξ,η) lies on the Γ, an approximation to Equation (2.33) can be rewritten as

$$\frac{1}{2} u(\xi,\eta) = \sum_{k=1}^{N} \left(u_k G_k(\xi,\eta) - \frac{\partial u_k}{\partial n} F_k(\xi,\eta) \right), \tag{2.35}$$

where u_k is $u(x_k, y_k)$, N is the number of elements on the boundary,

$$F_k(\xi,\eta) = \int_{C_k} u^*(\xi,\eta) ds, \tag{2.36}$$

$$G_k(\xi,\eta) = \int_{C_k} \frac{\partial u^*(\xi,\eta)}{\partial n} ds. \tag{2.37}$$

Equation (2.35) can be written in matrix form as $AX = b$, where column vector X contains the values of unknown temperatures and heat fluxes at the boundary nodes. In other words, for each boundary element $m = 1, 2, ..., N$, one needs to construct $\sum_{k=1}^{N} A_{mk} X_k = \sum_{k=1}^{N} b_{mk}$, where A_{mk} and b_{mk} are defined respectively as

$$A_{mk} = \begin{cases} -F_k(x_m, y_m) & \text{if u given over } C_k \\ G_k(x_m, y_m) & \text{if q given over } C_k \text{ and } k \neq m \\ G_k(x_m, y_m) - \dfrac{1}{2} & \text{if q given over } C_k \text{ and } k = m \end{cases} \tag{2.38}$$

$$b_{mk} = \begin{cases} q_k F_k(x_m, y_m) & \text{if q given over } C_k \\ -u_k G_k(x_m, y_m) & \text{if u given over } C_k \text{and } k \neq m \\ -u_k \left(G_k(x_m, y_m) + \dfrac{1}{2} \right) & \text{if u given over } C_k \text{and } k = m \end{cases} \tag{2.39}$$

where $q = \dfrac{\partial u}{\partial n}$. When all the values on the boundary are known, Equation (2.36) can be used to obtain values at any interior point in the domain Ω.

The advantages of the BEM consist of the fact that only the boundary of the domain requires to be discretised, while in finite difference method the whole domain requires discretisation. Thus the dimension of the problem using the BEM is reduced by one.One disadvantage of the BEM is that it requires large computational time to obtain the numerical solution, especially when the number of elements on the boundary is large, since A is a full matrix. Another disadvantage of BEM is that the fundamental solutions do not exist for all PDEs.

2.2 Inverse and Ill-posed Problems

In contrast to well-posed problems, an ill-posed problem is a problem that has no solution, many solutions, or unstable solutions (i.e. arbitrary small errors in the input data may lead to indefinitely large errors in the solutions).

Inverse problems concern the determination of the model parameters from the knowledge of the measured data. Solving inverse problems can also be used to determine the location, shape, and structure of intrusions, defects, sources of heat, waves, potential difference, pollution, etc. As an example, inverse problems dealing with heat conduction may be associated with the estimation of an unknown boundary heat flux by using temperature measurements taken below the boundary surface. Therefore, while in the classical direct heat conduction problem, the cause (boundary heat flux) is given and the effect (temperature field)

is determined, the inverse problem involves the estimation of the cause by utilizing the knowledge of the effect.

Inverse problems are mathematically classified as ill-posed [142]. The existence of a solution for an inverse problem may be assumed by physical reasoning. On the other hand, the uniqueness of the solution of inverse problems can be mathematically proved only for some special cases. Inverse problems are very sensitive to random errors in the measured input data. Special techniques are required in order for their solution to satisfy the stability condition. Successful solution of an inverse problem generally involves its reformulation as an approximate well-posed problem through the use of regularisation techniques, as introduced in section 2.3.

As an example, the inverse problem of Equation (2.6) is to find the unknown quantities, including parameters (e.g. K) and functions (e.g. $q(t)$) from some additional information usually obtained as measured data, such as

$$u^{mea}(x_i, t) = Y_i(t), \ i = 1, 2, ..., N, \tag{2.40}$$

where N is the number of sensors involved in obtaining the measured data.

Let v represent all the unknowns to be determined in the inverse problem with which the computed data is equal to the measured data, i.e.

$$u(v; x_i, t) = Y_i(t), \ i = 1, 2, ..., N. \tag{2.41}$$

A minimisation technique could be used to minimise the difference between measured and computed data, which is defined as the objective function by

$$J[v] = \sum_{i=1}^{N} \int_{t=0}^{t_f} \left(u(v; x_i, t) - Y(x_i, t) \right)^2 dt. \tag{2.42}$$

Inverse problems involve either parameter estimation or function estimation. If information is available of the functional form of the unknown quantity, the inverse problem can be reduced to the estimation of a few unknown parameters. On the other hand, if no prior information is available on the functional form of the unknown, the inverse problems need to be treated as a function estimation approach in an infinite dimensional space of functions.

2.3 Tikhonov Regularisation Technique

For the convenience of description, the nonlinear function

$$F(v) = y \tag{2.43}$$

is used to represent Equation (2.41), where $v \in X$ is the unkown quantity, $y \in Y$ are the measured data, $F : D(F) \subset X \to Y$ is a nonlinear, weakly closed and continuous operator between Hilbert spaces X and Y. As the notion of a solution of the equation (2.43), a v^*-minimum-norm-solution v_0 (v^*-M.N.S.) is chosen as the solution, i.e.

$$F(v_0) = y \tag{2.44}$$

and

$$\|v_0 - v^*\| = \min_{v \in D(F)} \left\{ \|v - v^*\| : F(v) = y \right\}. \tag{2.45}$$

In the following, we always assume the existence of an v^*-M.N.S. for exact data y. Note that due to the nonlinearity of F, this solution is not required to be unique. The element $v^* \in X$ in Equation (2.45) plays the role of a selection criterion [7].

If Equation (2.43) is ill-posed in the sense of lack of continuity of its solution with respect to the data, regularisation techniques are required. Tikhonov regularisation has been investigated in [7]-[9] to solve nonlinear ill-posed problems in a stable manner. In Tikhonov regularisation, a solution of the problem in Equation (2.43) is approximated by a solution of the minimization problem

$$\min_{v \in D(F)} \left\{ \|F(v) - y_\delta\|^2 + \lambda \|L(v - v^*)\|^2 \right\} \tag{2.46}$$

where $\lambda > 0$ is known as the regularisation parameter, L is a suitably chosen operator. The size of the regularized solution is measured by the norm $\|L(v - v^*)\|$, while the residual is measured by $\|F(v) - y_\delta\|^2$. v^* is an a priori estimate of v which is set to zero when no a priori information is available. $y_\delta \in Y$ is the available noisy data and there is additional information as

$$\|y_\delta - y\| \le \delta \tag{2.47}$$

where δ is the measurement error.

Convergence results for this method can be found in [7] and [8]. For analogous results in the case that is solvable only in the least-squares sense, see [10].

The regularisation parameter is an important quantity that should be carefully chosen for a good regularized output. There is often a trade-off between the regularized output and the original sets of data. In order to obtain a balance or minimize the trade-off, the optimal selection of the regularisation parameter λ becomes important. There are various methods in selecting regularisation parameters, e.g. Morozov's discrepancy principle [36], Generalized Cross Validation [12], L-curve [37], [38]. The discrepancy principle requires to know δ, while the other two do not depend on a prior knowledge of δ.

2.3.1 Morozov's discrepancy principle

To obtain convergence rates for Tikhonov regularisation, one has to assume a smoothness condition $v^* - v_0 = F'(v^*)\omega$ with sufficiently small $\|\omega\|$. With an a priori parameter choice $\lambda = c\delta$, $c > 0$, a convergence rate

$$\left\| v_\lambda^\delta - v^* \right\| \le k(c)\sqrt{\delta} \tag{2.48}$$

can be obtained [7]. An examination of the convergence proof shows that $k(c)$ is minimized by the optimal parameter choice $\lambda = c_{opt}\delta$, $c_{opt} = \|\omega\|^{-1}$, and

$$\left\| v_\lambda^\delta - v^* \right\| \le \frac{2\|\omega\|^{1/2}}{(1 - L\|\omega\|)^{1/2}} \sqrt{\delta} \tag{2.49}$$

(L denotes the Lipschitz-constant for the Frechet derivative). In general, the value of $\|\omega\|$ is not available, and so is c_{opt}. As a consequence, one can never obtain the optimal constant $k(c)$ for an a priori parameter choice.

An alternative is a posterior parameter strategy. A well studied method is Morozov's discrepancy principle, where the regularisation parameter λ satisfies

$$\left\| F(v_\lambda^\delta) - y_\delta \right\| = c\delta, \tag{2.50}$$

where $c \ge 1$. An advantage of Morozov's principle is that even without knowing $\|\omega\|$, one can always obtain the estimate

$$\left\| v_\lambda^\delta - v^* \right\| \le \frac{(2(1+c)\|\omega\|)^{1/2}}{(1-L\|\omega\|)^{1/2}} \sqrt{\delta} \ . \tag{2.51}$$

For $c = 1$, the optimal error bound Equation (2.49) is obtained. For $c > 1$, this bound is multiplied by $\sqrt{(1+c)/2}$. A drawback of the discrepancy principle is that a regularisation parameter satisfying Equation (2.50) might not exist for a general operator F. Moreover, even if such a parameter exists it requires an additional optimisation process to find it numerically. Another disadvantage with the discrepancy principle is that the measurement error usually cannot be predicted precisely. Hence the regularisation parameter λ cannot be estimated accurately.

2.3.2 Generalized cross validation

Generalized cross validation (GCV) [12] is a popular method for practical problems with discrete data and stochastic noise. It originates from the ordinary cross-validation. The rationale is to consider all the 'leave-one-out' regularized solutions and choose the parameter that minimizes the average of the squared prediction errors in using each solution to predict the missing data value.

The main idea of GCV is that a good model could be used to predict new data points. It is impractical to go to the field and measure a new data value each time we try a new regularisation parameter to verify our solution. Therefore, the experiment is simulated by eliminating one value from the data set. A good solution of the reduced data set, should predict this data fairly well, even if it was not used when calculating the model. This idea is repeated for each datum and therefore, the model obtained in this way is the model which can predict most of the data points even if these data points are not used.

In mathematical language, this is done by introducing the following notation. Let $v_\lambda^\delta(k)$ minimize

$$\min_{v \in D(F)} \left\{ \left\| F(v) - y_\delta \right\|^2 - \left(F(v_\lambda^\delta(k)) - y_\delta(k) \right)^2 + \lambda \|v\|^2 \right\}, \tag{2.52}$$

where $v_\lambda^\delta(k)$ is the k^{th} component of v and $y_\delta(k)$ is the k^{th} component of y. Notice that Equation (2.52) is the same as Equation (2.46), while with the k^{th} data point missing. We want to know how well is the k^{th} data predicted when it's not used. This can be measured by

$$\left(\mathrm{F}(v_\lambda^\delta(\mathrm{k})) - y_\delta(\mathrm{k})\right)^2. \tag{2.53}$$

The Cross Validation function is defined as the sum of the squares of these differences between a predicted data and the actual data, for all data points, namely

$$\mathrm{OCV}(\lambda) = \frac{1}{\mathrm{N}} \sum_{\mathrm{k}=1}^{\mathrm{N}} \left(\mathrm{F}(v_\lambda^\delta(\mathrm{k})) - y_\delta(\mathrm{k})\right)^2. \tag{2.54}$$

For a series of values of the regularisation parameter, $\mathrm{OCV}(\lambda)$ is calculated and the value of λ that corresponds to the minimum of $\mathrm{OCV}(\lambda)$ is identified as the optimal regularisation parameter

$$\bar{\lambda} = \arg\min_{\lambda \in \mathbb{R}^+} \mathrm{OCV}(\lambda). \tag{2.55}$$

But the above definition is not very practical to compute because we need to solve a nonlinear system for each different regularisation parameter. A short cut was found in [12]. There exists an $\mathrm{N} \times \mathrm{N}$ influence matrix $A(\lambda)$, with the property

$$\begin{bmatrix} \mathrm{F}(v_\lambda^\delta(1)) \\ \mathrm{F}(v_\lambda^\delta(2)) \\ \\ \mathrm{F}(v_\lambda^\delta(\mathrm{N})) \end{bmatrix} = A(\lambda)\, y_\delta. \tag{2.56}$$

Therefore, OCV can be rewritten as

$$\mathrm{OCV}(\lambda) = \frac{1}{\mathrm{N}} \sum_{\mathrm{k}=1}^{\mathrm{N}} \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} (a_{\mathrm{ki}}\, y_\delta(\mathrm{i}) - y_\delta(\mathrm{k}))^2}{(1 - a_{\mathrm{kk}})^2}, \tag{2.57}$$

where a_{ki}, $\mathrm{k,i} \in 1, 2, ..., \mathrm{N}$ is element $(\mathrm{k,i})$ of $A(\lambda)$.

In the definition of GCV, we let $A(\lambda)$ be the influence matrix defined above, then the GCV function is defined as

$$\mathrm{GCV}(\lambda) = \frac{\frac{1}{\mathrm{N}} \left\| \mathrm{I} - A(\lambda) \right\|^2}{\left(\frac{1}{\mathrm{N}} \mathrm{tr}(\mathrm{I} - A(\lambda)) \right)^2}. \tag{2.58}$$

So the GCV estimate of the regularisation λ is

$$\bar{\lambda} = \arg\min_{\lambda \in \mathbb{R}^+} \mathrm{GCV}(\lambda). \tag{2.59}$$

Equation (2.58) can be rewritten as

$$\mathrm{GCV}(\lambda) = \frac{1}{N}\sum_{k=1}^{N}\left(F(v_\lambda^\delta(k)) - y_\delta(k)\right)^2 w_k(\lambda), \tag{2.60}$$

where $w_k(\lambda)$ is given by

$$w_k(\lambda) = \left(\frac{1 - a_{kk}(\lambda)}{\frac{1}{N}\mathrm{tr}(I - A(\lambda))}\right)^2. \tag{2.61}$$

2.3.3 L-curve method

In recent years, L-curve has gained attention for computing the selection of regularisation parameters. It's a log-log plot of the regularized solution against the squared norm of the regularized residual for a range of values of regularisation parameters. The numerical computation and limitation of the L-curve are explained in [38] and [39].

The L-curve method is a parametric plot of $(\rho(\lambda), \eta(\lambda))$, where $\eta(\lambda)$ and $\rho(\lambda)$ measure the size of regularized solution $\left\|v_\lambda^\delta - v^*\right\|^2$ and the corresponding residual $\left\|F(v_\lambda^\delta) - y_\delta\right\|^2$. The L-curve has a distinct L-shaped corner located exactly where the solution x_λ changes in nature from being dominated by regularisation errors (i.e., by over-smoothing) to being dominated by the errors in the right-hand side. Hence the corner of the L-curve corresponds to a good balance between minimization of the sizes and the corresponding regularisation parameter λ being a good one. The idea of using the corner of the L-curve as a means for computing a good regularisation parameter was originally proposed in [11], where it was also demonstrated that under certain assumptions this criterion is indeed similar to both GCV and the discrepancy principle. Experiments confirm that whenever CGV finds a good regularisation parameter, the corresponding solution is located at the corner of the L-curve. The L-curve method for choosing the regularisation parameter has advantages over GCV: computation of the corner is a well-defined numerical problem, and the method is rarely 'fooled' by correlated errors. Even highly correlated errors will make the size of the solution grow once the regularisation parameter λ becomes too small, thus producing a corner on the L-curve.

The L-curve for Tikhonov regularisation has two characteristic part [143], namely, a 'horizontal' part where the regularisation parameter is too large and the regularized solution v_λ is dominated by the bias errors and an almost 'vertical' part where the regularisation parameter is too small and v_λ is dominated by the noisy errors.

The idea of the L-curve criterion is to choose a point on this curve that is at the 'corner' of the vertical piece. The following are two ways of viewing the problem of corner location:

(i) We could seek the point on the curve closest to the origin. The definition of 'closest' can vary from method to method. For example, Tikhonov regularisation measures distance as $\rho + \lambda^2 \eta$.

(ii) We could choose the point on the L-curve where the curvature is maximal [34]. The curvature is a purely geometrical quantity that is independent of transformations of the regularisation parameter.

Here, a convenient expression for the curvature is given by letting $\mathrm{x}^* = 0$

$$\eta(\lambda) = \left\| \mathrm{v}_\lambda^\delta \right\|^2, \ \rho(\lambda) = \left\| \mathrm{F}(\mathrm{v}_\lambda^\delta) - \mathrm{y}_\delta \right\|^2, \tag{2.62}$$

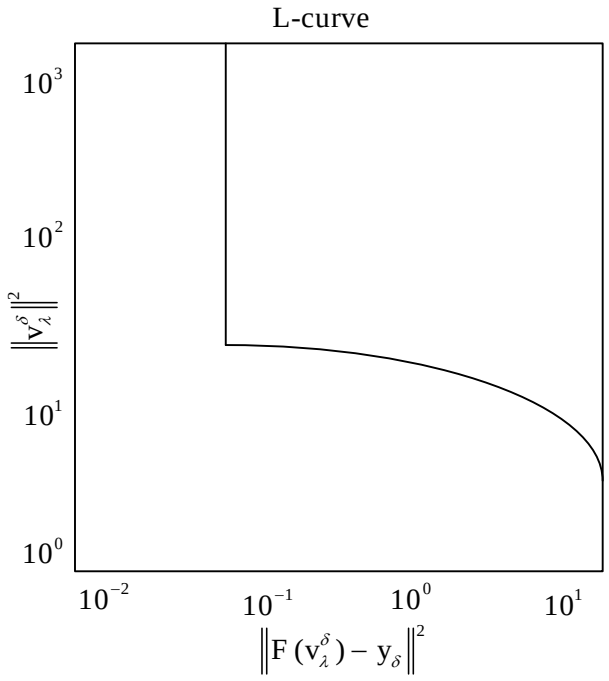
and

$$\eta(\lambda) = \log \eta, \ \rho(\lambda) = \log \rho. \tag{2.63}$$

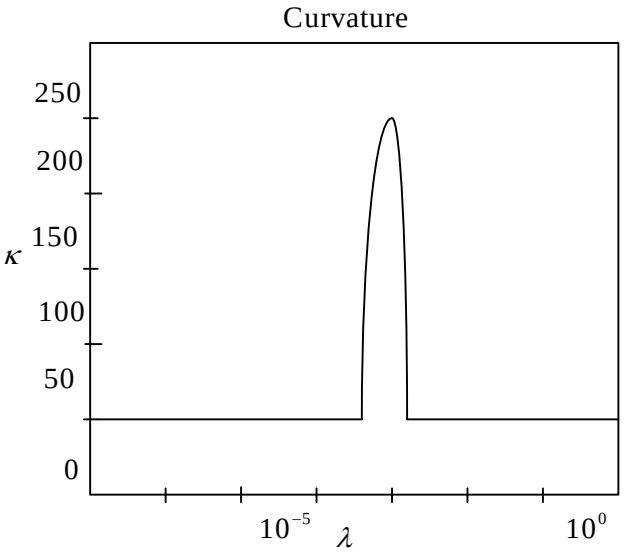
Let η', ρ', η'' and ρ'' denote the first and second derivatives of η and ρ with respect to λ. Then the curvature κ of the L-curve, as a function of λ, is given by

$$\kappa = 2 \frac{\rho' \eta'' - \rho'' \eta'}{\left(\left(\rho' \right)^2 + \left(\eta' \right)^2 \right)^{3/2}}. \tag{2.64}$$

Figure 2.3 illustrates the L-curve criterion: (a) shows the L-curve where the corner is clearly visible and (b) shows the curvature κ of the L-curve as a function of λ. The sharp peak in the κ-curve corresponds to the sharp corner on the L-curve.



(a)



(b)

Figure 2.3: A typical L-curve (a) and a plot (b) of the curvature κ versus the regularisation parameter

Experimental comparisons of the L-curve criterion with other methods for computing λ, most notably the method of GCV developed in [12], [40], are presented in [37]. The conclusion from these experiments is that the L-curve criterion for Tikhonov regularisation gives a very robust estimation of the regularisation parameter, while the GCV method

occasionally fails to do so. On the other hand, when GCV works it usually gives a very good estimate of the optimal regularisation parameter, while the L-curve criterion tends to produce a regularisation parameter that is slightly over-smooth, i.e., it is slightly too large.

Further experiments with correlated noise in [37] show that the L-curve criterion in this situation is superior to the GCV method which consistently produces severe under-smoothing.

2.4 Closure

This chapter provides an overview of various analytical and numerical methods for direct problems and a brief introduction to inverse problems. A typical inverse problem example of heat conduction is used in the presentation. The solution of direct problems is of importance in the solution process of an inverse problem. The finite difference method, which is mostly used in the rest of the thesis, is briefly introduced. The nonlinear least-squares method is used to model the inverse problems. The concept of ill-posedness is also described. Tikhonov regularisation technique is introduced to handle the ill-posedness leading to a stable solution. Three regularisation parameter selection methods are described, only the L-curve method will be used in the thesis.

Chapter 3 DETERMINISTIC METHODS FOR INVERSE PROBLEMS

This chapter gives a brief overview of two gradient-based deterministic methods, the namely steepest descent method (SDM) and conjugate gradient method (CGM), for the solutions of inverse problems. As the SDM is a special form of the CGM with the conjugate coefficient equal to zero, only the CGM is described in detail. Throughout this chapter, the heat conduction problem described in chapter 2 is used in the numerical illustration.

3.1 Steepest Descent Method

The SDM is the simplest of the gradient based method. It is basically an optimisation algorithm of finding the local minimum of a function. Consider Equation (2.42), in which $J(v)$ is assumed to be differentiable within a given region. The direction in which the function value decreases fastest would be the negative gradient of $J(v)$. The SDM follows a zig-zag like path from an arbitrary starting point v_0 and gradually slides along the gradient, until it converges to the actual point of minimum (Figure 3.1).

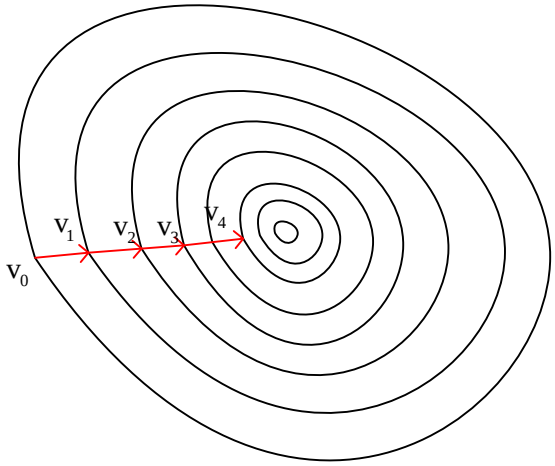


Figure 3.1: Convergence path of the SDM starting with an arbitrary starting point v_0.

Let v^k denote the k^{th} iterative approximation of the minimisation problem, the iterative equation of the SDM can be written as

$$v^{k+1} = v^k - \beta^k \nabla J(v^k) \tag{3.1}$$

where $k = 0,1,2,...$ is the iteration number, $\nabla J(v^k)$ is the gradient at v^k, β^k is the step size. It is obvious that in order to find the point where $J(v)$ is a minimum, the directional derivative at that point should be zero, i.e.

$$\frac{d}{d\beta^k} J(v^{k+1}) = \nabla J(v^{k+1})^T \frac{d}{d\beta^k} v^{k+1} = -\nabla J(v^{k+1})^T \nabla J(v^k) = 0. \tag{3.2}$$

It is clear that β^k should be chosen so that $\nabla J(v^{k+1})$ and $\nabla J(v^k)$ are orthogonal. The following iterative step is then taken in the direction of the negative gradient at this current position leading to a zig-zag pattern as illustrated in Figure 3.2. This iteration continues until the local minimum is determined within a chosen accuracy ε.

Procedure of the steepest descent method:

Initialize: $k = 0$, $g^0 = \nabla J(v^0)$, $d^0 = -g^0$;

Do while ($\|g^k\| > \varepsilon$)

Determine the step size β^k: $\min_{\beta^k > 0} J(v^k + \beta^k d^k)$;

Calculate the new point: $v^{k+1} = v^k + \beta^k d^k$;

Calculate the gradient: $g^{k+1} = \nabla J(v^{k+1})$;

Set direction of search: $d^{k+1} = -g^{k+1}$;

$k = k + 1;$

End do

The method is easy to apply. Each iteration does not involve much computational work. It is also very stable and guarantes to locate the minimum point as long as it exists [141]. However, the method has significant drawbacks in that it generally has slow convergence when used on a badly scaled system, and is dependent on the choice of the starting point.

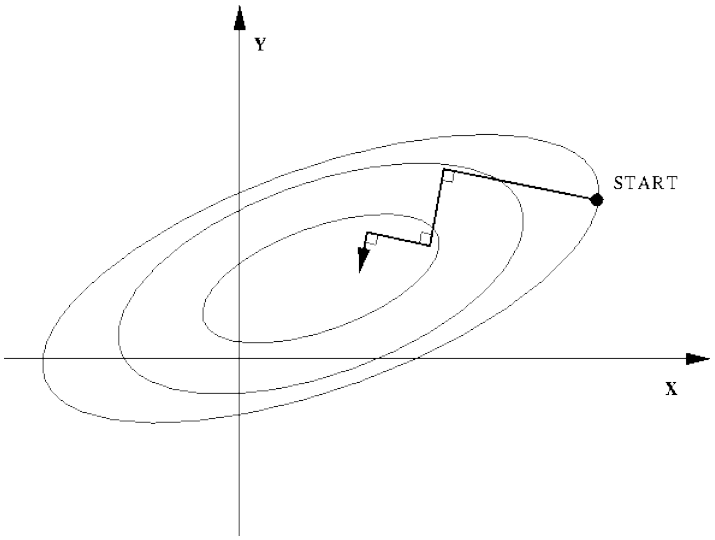


Figure 3.2: SDM approaches the minimum in a zig- zag manner.

It is suggested that [144] the method should be used when one has certain knowledge of where the minimum is. But it is generally considered to be a poor choice for any optimisation problem. It may be used in conjunction with other optimizing methods to achieve better convergence.

3.2 Conjugate Gradient Method

As seen in the previous subsection, the reason why the SDM converges slowly is that it has to take a right angle turn after each step, and consequently search in the same direction as earlier steps (Figure 3.2). The CGM attempts to remove this problem by 'learning' from experience by selecting the successive direction vectors as a conjugate of the successive gradients obtained as the iteration proceeds. Thus, the directions are not specified beforehand, but rather are determined sequentially at each step of the iteration. At step k one evaluates the

current negative gradient vector and adds to it a linear combination of the previous direction vectors to obtain a new conjugate direction vector along which the approximation moves.

The iterative equations of the CGM are as given as below

$$v^{k+1} = v^k - \beta^k d^{k+1}, \tag{3.3}$$

$$d^{k+1} = -\nabla J(v^k) + \gamma^k d^k, \tag{3.4}$$

where γ^k is the conjugate coefficient, the step size β^k is determined in the same way as that in the SDM.

There are different versions of conjugate gradient and they are distinguished by the way in which the conjugate coefficient γ^k is determined [99]. For the Fletcher-Reeves formula, the constant γ^k is equal to the ratio of the squared norm of the current gradient to the squared norm of the previous gradient

$$\gamma^k = \frac{(g^{k+1})^T g^{k+1}}{(g^k)^T g^k}. \tag{3.5}$$

For the Polak-Ribiere formula, the constant γ^k is determined by taking the inner product of the previous change in the gradient with the current gradient divided by the squared norm of the previous gradient

$$\gamma^k = \frac{(g^{k+1})^T (g^{k+1} - g^k)}{(g^k)^T g^k}. \tag{3.6}$$

It is difficult to predict which version performs better on a given problem. The storage requirements for Polak-Ribiere (four vectors) are slightly larger than for that Fletcher-Reeves (three vectors).

Procedure of the conjugate gradient method:

Initialize: $k = 0$, $g^0 = \nabla J(v^0)$, $d^0 = -g^0$;

Do while ($\|g^k\| > \varepsilon$)

Determine the step size β^k: $\min_{\beta^k > 0} f(v^k + \beta^k d^k)$;

Calculate the new point: $v^{k+1} = v^k + \beta^k d^k$;

Calculate the gradient: $g^{k+1} = \nabla J(v^{k+1})$;

Calculate the conjugate coefficient: γ_k according to Equation (3.5) or Equation (3.6);

Determine the direction of search: $d^{k+1} = -g^{k+1} + \gamma^k d^k$;

$k = k+1$;

End do

There are three primary advantages of choosing such direction in this method:

(1) Unless the solution is attained in less than n steps, the gradient is always nonzero and linearly independent of all previous direction vectors.

(2) The new direction vector may be computed by using a simple formula shown in Equation (3.4), which does not increase the computational complexity very much.

(3) The gradual change of the direction during the iterative process ensures a uniform convergence towards the solution.

3.3 Solving Inverse Problems by Conjugate Gradient Method

The CGM is also known as an iterative regularisation method, which means the regularisation procedure is performed during the iterative processes and the regularisation parameter is the iteration number. The CGM derives from the perturbation principle which transforms an inverse problem to the solution of three problems, namely, the direct, sensitivity and the adjoint problem [42].

Consider the one-dimensional heat conduction problem (Equations (2.6)) of estimating the unknown heat flux $q(t)$ by minimizing the objective funtion

$$\begin{aligned} J[q(t)] &= \sum_{i=1}^{N} \int_{t=0}^{t_f} \left(u(q; x_i, t) - Y(x_i, t)\right)^2 dt \\ &= \sum_{i=1}^{N} \int_{t=0}^{t_f} \left(u_i(q;t) - Y_i(t)\right)^2 dt \end{aligned} \tag{3.7}$$

where $u_i(q;t)$ is the computed temperature at the measurement locations x_i at time t which is determined from the solution of the direct problem with given $q(t)$.

The CGM iterations involve

$$q^{k+1} = q^k - \beta^k d^k, \tag{3.8}$$

$$d^{k+1} = -\nabla J(q^{k+1}) + \gamma^k d^k, \tag{3.9}$$

where the step size β^k , the conjugate coefficient γ^k and the gradient $\nabla J(q^k)$ are required. These quantities may be determined by using the sensitivity problem and the adjoint problem as described below.

3.3.1 Sensitivity problem and search step size

The object function $J[q^{k+1}]$ for iteration $k+1$ is obtained by rewriting Equation (3.7) as

$$J[q^{k+1}] = \sum_{i=1}^{N} \int_{t=0}^{t_f} \left(u_i(q^k - \beta^k d^k; t) - Y_i(t) \right)^2 dt , \tag{3.10}$$

where q^{k+1} is replaced by the expression given by Equation (3.8). If temperature $u_i(q^k - \beta^k d^k; t)$ is linearized by a Taylor expansion, Equation (3.10) takes the form

$$J[q^{k+1}] = \sum_{i=1}^{N} \int_{t=0}^{t_f} \left(u_i(q^k; t) - \beta^k \Delta u_i(d^k; t) - Y_i(t) \right)^2 dt . \tag{3.11}$$

The search step size β^k is determined by minimizing Equation (3.11) with respect to β^k

$$\frac{\partial J(q^{k+1})}{\partial \beta^k} = 2 \int_{t=0}^{t_f} \sum_{i=1}^{N} \left(-\Delta u_i(d_i) \right) \left(u_i(q^k; t) - \beta^k \Delta u_i(d_i; t) - Y_i(t) \right) dt = 0 , \tag{3.12}$$

$$\beta^k = \frac{\int_{t=0}^{t_f} \sum_{i=1}^{N} [u_i(t) - Y_i(t)] \Delta u_i(d^k; t) dt}{\int_{t=0}^{t_f} \sum_{i=1}^{N} [\Delta u_i(d^k; t)]^2 dt} , \tag{3.13}$$

where $u_i(q^k; t)$ is the solution of the direct problem by using estimate q^k at x_i and time t. $\Delta u_i(d^k; t)$ is the sensitivity function, which is taken as the solution of the sensitivity problem. The sensitivity problem is obtained from the original direct problem defined by Equation (2.6) in the following way: It is assumed that when q(t) undergoes a variation $\Delta q(t)$, u is perturbed by Δu . By replacing q by $q + \Delta q$ and u by $u + \Delta u$ in the direct problem, then subtracting from Equation (2.6), neglecting the second order terms, the sensitivity equation can be obtained as below

$$\begin{cases} \rho C \dfrac{\partial \Delta u}{\partial t}(x,t) = K \dfrac{\partial^2 \Delta u}{\partial x^2}(x,t), \ t > 0, 0 < x < 1 \\ \Delta u\big|_{t=0} = 0, \qquad 0 \le x \le 1 \\ -K \dfrac{\partial \Delta u}{\partial x}\bigg|_{x=0} = \Delta q(t), \qquad t > 0 \\ -K \dfrac{\partial \Delta u}{\partial x}\bigg|_{x=1} = 0, \qquad t > 0 \end{cases} \tag{3.14}$$

$\Delta u_i(d^k;t)$ can be obtained by letting $\Delta q = d^k$.

3.3.2 Adjoint problem and gradient equation

Either in Equation (3.5) or Equation (3.6), the conjugate coefficient depends on the gradient value. To obtain the gradient value, the adjoint problem is constructed in the following way: Equation (3.14) is multiplied by the Lagrange multiplier (or adjoint function) $\lambda(x,t)$ and the resulting expression is integrated over the corresponding space and time domain, and added to Equation (3.7) to yield the expression as below

$$\begin{aligned} J[q(t)] = & \int_{t=0}^{t_f} \int_{x=0}^{1} (u(q;t) - Y(t))^2 \delta(x - x_i) dx dt \\ & + \int_{t=0}^{t_f} \int_{x=0}^{1} \lambda \left\{ K \frac{\partial^2 u}{\partial x^2}(x,t) - \rho C \frac{\partial u}{\partial t}(x,t) \right\} dx dt \end{aligned} . \tag{3.15}$$

The variation ΔJ is obtained by perturbing q by Δq and u by Δu in Equation (3.15), subtracting from the resulting expression the original Equation (3.15) and neglecting the second-order terms to find

$$\begin{aligned} \Delta J = & \int_{t=0}^{t_f} \int_{x=0}^{1} 2(u - Y) \Delta u \delta(x - x_i) dx \\ & + \int_{t=0}^{t_f} \int_{x=0}^{1} \lambda \left[K \frac{\partial^2 \Delta u}{\partial x^2}(x,t) - \rho C \frac{\partial \Delta u}{\partial t}(x,t) \right] \end{aligned} . \tag{3.16}$$

In which, the second double integral term is reformulated based on the Green's second identity. The boundary conditions of the sensitivity problem given in Equation (3.14) are utilized and then ΔJ is allowed to be zero. The vanishing of the integrands containing Δu leads to the following adjoint problem for the determination of $\lambda(x,t)$:

$$\begin{cases} K\dfrac{\partial^2\lambda}{\partial x^2}+\rho C\dfrac{\partial\lambda}{\partial t}+2(u-Y)\delta(x-x_i)=0, & 0<x<1,\ t>0 \\ \lambda\big|_{t=t_f}=0, & 0\le x\le 1 \\ K\dfrac{\partial\lambda}{\partial x}=0, & t>0 \\ K\dfrac{\partial\lambda}{\partial x}=0, & t>0 \end{cases} \tag{3.17}$$

The adjoint problem is different from the standard initial value problems as in the direct and sensitivity problems, in that the final time condition at time $t=t_f$ is specified instead of the customary initial condition at $t=0$. However, this problem can be transformed to an initial value problem by the transformation of the time variables as $\tau=t_f-t$.

Finally, the following integral term is left as

$$\Delta J=\int_{t=0}^{t_f}\lambda(x=0,t)\Delta q(t)dt. \tag{3.18}$$

From definition in [41], the functional increment can be presented as

$$\Delta J=\int_{t=0}^{t_f}\nabla J[q(t)]\Delta q(t)dt. \tag{3.19}$$

A comparison of Equation (3.18) and (3.19) leads to the following expression for the gradient $J'[q(t)]$ of functional $J[q(t)]$

$$\nabla J[q(t)]=\lambda(x=0,t). \tag{3.20}$$

3.3.3 Stopping criterion

If the problem contains no measurement errors, the traditional check condition is specified as

$$J[q^{k+1}(t)]<\varepsilon, \tag{3.21}$$

where ε is a small specified number. However, the observed data may contain measurement errors. Therefore, Equation (3.7) is not expected to be equal to zero at the final iteration step. Following the experience in [42], the discrepancy principle is used as the stopping criterion:

$$u_i(t)-Y_i(t)\approx\sigma, \tag{3.22}$$

where σ is the standard deviation of the measurements, which is assumed to be a constant. Substituting Equation (3.22) into Equation (3.7), the following expression is obtained for stopping criterion ε

$$\varepsilon = N\sigma^2 t_f . \tag{3.23}$$

Then the stopping criterion is given by Equation (3.21) with ε determined from Equation (3.23).

3.3.4 Computational procedure

The computational procedure for the solution of the inverse problem using CGM may be summarized as follows:

Initialize: $k = 0$, q^0, $\gamma^0 = 0$;

Solve the direct problem given by Equation (2.6) for $u(x,t)$;

Do while (stopping criterion given by Equation (3.21) is not satisfied)

Solve the adjoint problem given by Equation (3.17) for $\lambda(x,t)$;

Compute the gradient ∇J from Equation (3.20);

Compute the conjugate coefficient γ^k from Equation (3.5) or (3.6);

Compute the direction of descent d^k from Equation (3.9);

Set $\Delta q = d^k$, and solve the sensitivity problem given by Equation (3.14) for $\Delta u(x,t)$;

Compute the search step size β^k from Equation (3.13);

Compute the new estimation for q^{k+1} from Equation (3.8);

Solve the direct problem given by Equation (2.6) for $u(x,t)$;

$k = k + 1$;

End do

3.3.5 Disadvantages of conjugate gradient method

The CGM is called an iterative regularisation method because the regularisation procedure is performed during the iterative processes. Therefore, the choice of a suitable quadratic norm, determination of a proper regularisation order and the determination of an optimal regularisation coefficient in the conventional regularisation methods are not needed.

CGM is computationally efficient with a carefully chosen initial value, otherwise, the convergence and the quality of the solution will be degraded.

Furthermore, it can be noted in section 3.3 that the gradient ∇J at the final time $t = t_f$ is always equal to zero since $\lambda(x, t = t_f) = 0$. If the initial guess values q^0 cannot be predicted correctly before the inverse calculation, the estimated values of heat flux q will deviate from exact values near the final time conditions. As in [42], we let

$$\nabla J(t_f) = \lambda(0, t_f) = \lambda(0, t_f - \Delta t), \tag{3.24}$$

where Δt denotes the time increment. By replacing the artificial gradient Equation (3.24) to the gradient Equation (3.20), the singularity at the final time $t = t_f$ can be avoided.

3.4 Closure

This chapter provides an overview of the two deterministic methods, SDM and CGM. As the SDM is a special form of the CGM with conjugate coefficient equal to zero, only the details of using the CGM for inverse problems are given. Applications of the method for various inverse problems will be shown in chapter 5. The advantage of the gradient-based methods is fast convergence, while the inherent difficulties with the gradient-based methods are the dependence on the initial guess and the complicated gradient computation. These inherent difficulties are addressed in chapter 4.

Chapter 4 STOCHASTIC METHODS FOR INVERSE PROBLEMS

This chapter gives the description of two classes of popular stochastic methods: the first one is based on natural evolution: survival of the fittest, e.g. genetic algorithm (GA); the second one is based on the collaboration between individuals in an ecosystem, e.g. particle swarm optimisation (PSO) and quantum-behaved particle swarm optimisation (QPSO). Several modifications to the QPSO are proposed, including Gaussian mutation, ring topology. Two models of parallel QPSO are also given in this chapter, i.e. master-slave parallelisation, static subpopulation parallelisation. Finally, the hybrid method combining QPSO and CGM is described.

Usually a fitness function is used in these algorithms to evaluate the fitness of individuals. In this chapter the fitness function can be the least-squares function of an inverse problem such as the one in Equation (2.42). For simplicity and convenience, $f(X)$ is used as the objective fitness function to represent $J(v)$, which is a minimisation problem, defined in a feasible search domain $X \in \Omega \subseteq \Re^D$.

4.1 Genetic Algorithms

GA, originally proposed by Holland [29]-[30], is a search heuristic which mimics the process of natural evolution: survival of the fittest. Only a brief overview of the GA is presented here and in-depth details may be found in [31].

GA generates solutions to an optimisation problem using operations such as selection (reproduction), crossover (recombination) and mutation, with each individual or a candidate

solution in the population represented by a binary string of 0s and 1s or by other forms of encodings. The evolution (or the search process) starts from a population of chromosomes generated randomly in the search space and evolves according to rules every generation. In each generation, the fitness of every individual is evaluated, and multiple individuals are randomly selected from the current population based on their fitness and modified by recombination and mutation operation to form a new population, which is then used in the next generation of the evolution. Generally, the search process terminates when either a maximum number of generations has been produced or a satisfactory fitness level has been reached for the population.

Consider a GA system with M chromosomes, $X(k) = \{X_1(k), X_2(k), ..., X_M(k)\}$, such that each chromosome $X_i(k)$ is in the D-dimensional space $X_i(k) = \{X_{i,1}(k), X_{i,2}(k), ..., X_{i,D}(k)\}$. For the convenience of description, $X_a(k)$ and $X_b(k)$ are assumed to be two chromosomes in the k^{th} generation. In GA, a maximisation problem is used to evaluate the fitness of the chromosomes, therefore $-f(X)$ is used as below

$$\max_X \left(-f(X)\right). \tag{4.1}$$

Encoding

The most important issue in GA is the encoding of chromosomes, which depends on the problem itself. The canonical GA makes use of the binary format to represent the genotypes, in which a chromosome X_a is treated as a string of bits in 0 or 1, e.g.

Chromosome X_a	101100101100101011100101

Figure 4.1: Binary encoding.

In an improved version of GA [32], ordinary real-valued numbers are used to represent the population members in whch X_a is a string of real values, e.g.

Chromosome X_a	1.2324 5.3243 0.4556 2.3293 2.4545

Figure 4.2: Real-valued encoding

Selection

According to Darwin's evolution theory, the best ones should survive and create new offspring. There are many methods to select the best chromosomes, for example, roulette wheel selection, Boltzman selection, tournament selection, rank selection, and steady state selection.

The most common selection method is the roulette wheel selection. Here, parents are selected according to their fitness and the fittest chromosomes have a greater chance of survival than weaker ones. This could be imagined as similar to a roulette wheel in a casino. This can be simulated by the algorithm below

(i) Calculate the sum of all chromosome fitness in the population: $S = \sum_{i=1}^{M} f(X_i)$.

(ii) Calculate the normalized fitness value of each chromosome: $p_i = \frac{f(X_i)}{S}$.

(iii) The population is sorted in descending value of the normalized fitness.

(iv) Calculate the accumulated normalized fitness values.

(v) Generate a random number r from the interval (0,1). The selected chromosome is the first one with accumulated normalized fitness value greater than r. This step is repeated until there are enough selected chromosomes.

Crossover

There are two popular crossover operators which may be applied to the binary-coded representations, i.e. the one-point and two-point crossover operators. Two parents (e.g. $X_a(k)$, $X_b(k)$) are selected for recombination, and segments of their bit strings are exchanged between the two parents to form the two offsprings.

The one-point crossover proceeds by picking a point randomly in the bit string and exchanging all the bits after that point. An example shown in Figure 4.3 exchanges all the bits in the third columes.

Chromosome $X_a(k)$	1011001011001	01011100101
Chromosome $X_b(k)$	1111111000001	10000011111

Chromosome $X_a(k+1)$	1011001011001	10000011111
Chromosome $X_b(k+1)$	1111111000001	01011100101

Figure 4.3: One-point crossover.

Two-point crossover calls for two points to be selected on the parent strings (e.g. $X_a(k)$, $X_b(k)$), everything between the two points is swapped between the parent strings, rendering two child strings (Figure 4.4).

Chromosome $X_a(k)$	101100	1011001	01011100101
Chromosome $X_b(k)$	111111	1000001	10000011111

Chromosome $X_a(k+1)$	101100	1000001	01011100101
Chromosome $X_b(k+1)$	111111	1011001	10000011111

Figure 4.4: Two-point crossover.

All crossover operators from binary encoding can be used for real-valued encoding. Another popular crossover operator in real-valued GA is called the arithmetic crossover. The two offsprings can be obtained by using the two updates below

$$X_a(k+1) = rX_a(k) + (1-r)X_b(k), \tag{4.2}$$

$$X_b(k+1) = rX_b(k) + (1-r)X_a(k), \tag{4.3}$$

where r is a uniform random variable .

Mutation

The aim of the mutation operator is to introduce some diversity into the population, thereby extending the effective area of the search space that the algorithm considers. A high mutation rate may destabilise the population by disrupting the existing good solutions. Since GAs usually rely on their recombination operators, the mutation rate is usually set quite low. For the binary encoding, the mutation operator is simply to change the value of each bit (from either 0 to 1 or 1 to 0) according to the mutation rate (Figure 4.5).

Chromosome X_a	101100101100101011100101

Chromosome X_a	101100001101101011000101

Figure 4.5: Binary mutation.

For the real-valued encoding, the mutation operator is to add a small number to the selected values

$$X_a(k+1) = X_a(k) + r\,. \tag{4.4}$$

The example shown in Figure 4.6 has the third gene of the chromosome being mutated by adding a value 0.11.

Chromosome X_a	1.2324 5.3243 0.4556 2.3293 2.4545

Chromosome X_a	1.2324 5.3243 0.5656 2.3293 2.4545

Figure 4.6: Real-valued mutation.

The framework of GAs is described as below:

Choose the initial population of individuals;

Evaluate the fitness of each individual in the population;

Do while (stopping criterion is not satisfied)

Select the best-fit individuals for reproduction;

Breed new individuals through crossover and mutation operations to give birth to offsprings;

Evaluate the individual fitness of new individuals;

Replace least-fit population with new individuals;

End do

4.2 Particle Swarm Optimisation

4.2.1 The original particle swarm optimisation

Particle swarm optimisation (PSO) is a population based stochastic optimisation technique developed by Eberhart and Kennedy in 1995 [13]. The concept of the method was inspired by social behaviour of bird flocking or fish schooling. The system maintains a population of particles, in which each particle represents a potential solution of an optimisation problem. The position of each particle is evaluated as according to an objective fitness function. The particles in a local neighbourhood share memories of their best previous positions (the position gives the best fitness value). These memories are used to adjust the particles' own velocities and their subsequent positions.

Consider a PSO system with M particles $X = \{X_1, X_2, ..., X_M\}$, where each particle is treated as a volume-less body in the D-dimensional space $X \in \Omega \subseteq \Re^D$. Let the D-dimensioanl position vector and velocity vector of particle i at the time k be denoted as $X_i(k) = (X_{i,1}(k), X_{i,2}(k), \cdots, X_{i,D}(k))$ and $V_i(k) = (V_{i,1}(k), V_{i,2}(k), \cdots, V_{i,D}(k))$, ($i = 1, 2, ..., M$). Consider the minimization problems

$$\min_X f(X) \ . \tag{4.5}$$

The personal best position associated with particle i, $P_i(k) = (P_{i1}(k), P_{i2}(k), ..., P_{iD}(k))$, is the best previous position, which yields the best objective function value.

$P_g(k) = (P_{g1}(k), P_{g2}(k), \cdots, P_{gD}(k))$ is the global best position of all the particles, $g \in \{1, 2, ..., M\}$. The personal best position of particle i, $P_i(k)$ may be determined as

$$P_i(k) = \begin{cases} X_i(k) & \text{if } f(X_i(k)) < f(P_i(k-1)) \\ P_i(k-1) & \text{if } f(X_i(k)) \geq f(P_i(k-1)) \end{cases} \tag{4.6}$$

The global best position of the swarm can be found by solving

$$g = \arg \min_{1 \leq i \leq M} (f(P_i(k))). \tag{4.7}$$

From the above definition, the velocity of a particle is updated according to the equation

$$V_{i,j}(k+1) = V_{i,j}(k) + c_1 r_1 (P_{i,j}(k) - X_{i,j}(k)) + c_2 r_2 (P_{g,j}(k) - X_{i,j}(k)), \tag{4.8}$$

where $i = 1, 2, ..., M$, $j = 1, 2, ..., D$. c_1 and c_2 are two constants known as acceleration coefficient. c_1 influences the maximum step size towards the personal best position, and c_2 influences the maximum step size towards the global best position. r_1 and r_2 are random numbers distributed uniformly in (0,1). From the above equation, the velocity of a particle is determined by three factors: (1) the current velocity - a momentum term used to prevent excessive oscillations in the search direction; (2) the cognitive component - the distance from the current position to the personal best position representing the natural tendency of individuals to return to environments where they experienced their best performance; (3) the social component - the distance from the current position to the global best position representing the tendency of individuals to follow the success of other individuals.

In general, the value of $V_{i,j}$ is restricted in the range $[-V_{max}, V_{max}]$ in order to reduce the probability that the particle might leave the search space. If the search space is defined by the bounds $[-X_{max}, X_{max}]$, then the value of V_{max} is set as $V_{max} = aX_{max}$, where $0.1 \leq a \leq 1.0$ [14]. Then the position of each particle is updated using the new velocity

$$X_{i,j}(k+1) = X_{i,j}(k) + V_{i,j}(k+1). \tag{4.9}$$

The initialisation process of the PSO algorithm is described by

(1) Initialise each position component $X_{i,j}$ with a value drawn from the uniform random distribution on the interval $[-X_{max}, X_{max}]$, for all $i = 1, 2, ..., M$ and $j = 1, 2, ..., D$. This distributes the initial positions of the particles throughout the search space.

(2) Initialize each velocity $V_{i,j}$ with a value drawn from the uniform random distribution on the interval $[-V_{max}, V_{max}]$, for all $i = 1, 2, ..., M$ and $j = 1, 2, ..., D$.

(3) Set $P_i = X_i$, for all $i = 1, 2, ..., M$.

The procedure of the original PSO is described in the following pseudo code.

Algorithm of the Original PSO

Create and initialize the positions and velocities of the population according to the above initialization process.

Do while ($k < k_{max}$)

For each particle $i \in 1, 2, ..., M$:

Evaluate the fitness $f(X_i(k))$;

If $f(X_i(k)) < f(P_i(k))$ then

$P_i(k) = X_i(k)$;

If $f(P_i(k)) < f(P_g(k))$ then

$P_g(k) = P_i(k)$;

End for

For each particle $i \in 1, 2, ..., M$:

Update the velocity $V_i(k+1)$ according to Equation (4.8);

Update the position $X_i(k+1)$ according to Equation (4.9);

End for

End do

There are different versions of modification of the PSO algorithm proposed by various researchers to improve the performance of the algorithm. The most important improvements are the version with an inertia weight w [15] and the one with a constriction factor χ [16].

4.2.2 Inertia weight particle swarm optimisation

The inertia weight controls the momentum of the particle by the weighted contribution of the previous velocity. Essentially it controls the amount of memory of the previous flying direction that would influence the new velocity. The velocity is updated by

$$V_{i,j}(k+1) = wV_{i,j}(k) + c_1 r_1 (P_{i,j}(k) - X_{i,j}(k)) + c_2 r_2 (P_{g,j}(k) - X_{i,j}(k)), \quad (4.10)$$

where w is extremely important to ensure convergent behaviour [17]-[19]. For $w > 1$, velocities increase over time causing divergent behaviour. Particles fail to change direction in order to move back towards promising areas. For $0 < w \leq 1$, particles decelerate until their velocities reach zero. Shi and Eberhart [19] investigated the effect of w values in the range $[0,1.4]$, as well as varying w over time. Their results indicate that choosing $w \in [0.8,1.2]$ results in faster convergence. Further empirical experiments [20] were performed with an inertia weight set to decrease linearly from 0.9 to 0.4 during the course of simulation. This choice of the inertia weight allows the PSO to explore a large area at the start with a large inertia weight, and to refine the search later by using a smaller inertia weight

$$w(k) = 0.9 - \frac{k}{k_{max}}(0.9 - 0.4), \quad (4.11)$$

where k_{max} is the maximal iteration number.

4.2.3 Constriction particle swarm optimisation

Clerc [16] indicated that a constriction factor may help ensure convergence. The constriction factor model describes a way of choosing the values of w, c_1 and c_2 so that convergence is ensured. The modified velocity update equation with constriction factor is given by

$$V_{i,j}(k+1) = \chi[V_{i,j}(k) + c_1 r_2 (P_{i,j}(k) - X_{i,j}(k)) + c_2 r_1 (P_{g,j}(k) - X_{i,j}(k))], \quad (4.12)$$

where

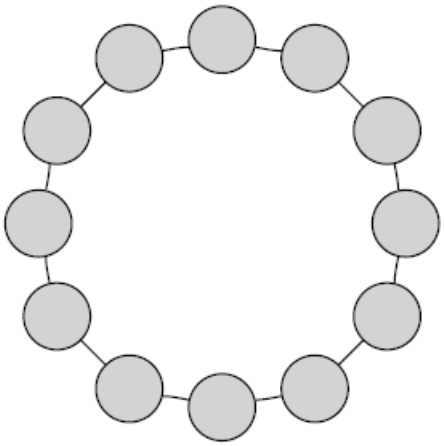
$$\chi = \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|}, \ \phi = c_1 + c_2, \ \phi > 4. \quad (4.13)$$

Let $c_1 = c_2 = 2.05$. Substituting $\phi = c_1 + c_2 = 4.1$ into Equation (4.13) yields $\chi = 0.7298$.

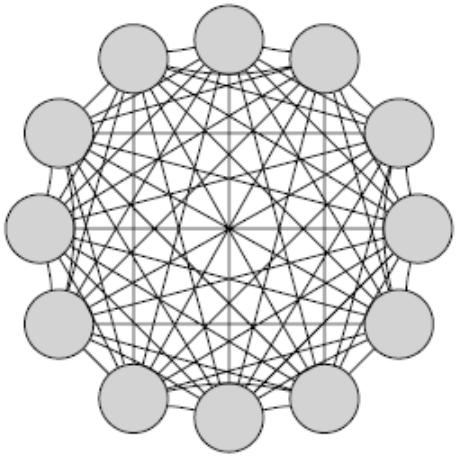
Eberhart and Shi [17] compared the performance of a swarm using the inertia weight and that using the constriction factor. The results indicate that using the constriction factor usually gives a better rate of convergence. Note that the two versions are equivalent when $c_1 = c_2 = 1.4926$ and $w = 0.7298$ are used in PSO with inertia weight.

4.2.4 Standard particle swarm optimisation

In 2007, Bratton and Kennedy [22] defined a standard for PSO algorithm (SPSO). It involves the local topology model (i.e. lbest swarm model) as shown in Figure 4.7a. While in the original PSO the global topology model (i.e. gbest swarm model) as shown in Figure 4.7b is used. In the local ring topology swarm model, each particle only connects to two other particles in the swarm. This is in contrast to the gbest model where every particle is able to obtain information from the very best particle in the entire swarm population. Therefore, every particle has its own local best $LBEST_i$ $(i = 1, ..., M)$ to replace P_g in Equation (4.8). The advantage of the lbest model appears to lie in its slower convergence rate relative to the gbest model, especially for the complex multimodal problems. Despite that, the faster convergence rate of a global best topology would usually result to better performance on simple unimodal problems than that of local topology due to the lack of any danger of convergence to a local minimum.



(a) ring topology- lbest



(b) global topology- gbest

Figure 4.7: Particle swarm topologies.

As in [17], a comparison study of the inertia weights and constriction factors in PSO demonstrated that the PSO algorithm with constriction is in fact a special case of the algorithm with inertia weight in which the values of the parameters may be determined analytically. When the constriction factor is used in the standard PSO algorithm the velocity update equation becomes

$$V_{i,j}(k+1) = \chi[V_{i,j}(k) + c_1 r_2 (P_{i,j}(k) - X_{i,j}(k)) + c_2 r_1 (LBEST_{i,j}(k) - X_{i,j}(k))]. \quad (4.14)$$

When the global optimum of the problem being solved is located at or near the centre of the feasible region, the population may be initialized within a subspace of the entire feasible search space that does not contain the global optimum. This is referred to as region scaling or non-uniform swarm initialization.

The definition of the standard PSO algorithm includes [22]

(1) a local ring topology as shown in Figure 4.7a;

(2) the constricted velocity update equation as in Equation (4.14);

(3) 50 particles;

(4) non-uniform swarm initialization in a quarter of the feasible search space;

(5) and boundary conditions wherein a particle is not evaluated when it exits the feasible search space.

4.3 Quantum-Behaved Particle Swarm Optimisation

In classical mechanics, the state of a particle is defined by the position vector and velocity vector which determine the trajectory of the particle according to Newtonian law of motion. The particle moves along a determined trajectory in the case of Newtonian mechanics. It is not the same case in quantum mechanics in which the term 'trajectory' is meaningless. This is because the position and velocity of a particle cannot be determined simultaneously according to the uncertainty principle. Therefore, if particles in a PSO system behave in quantum form, the PSO algorithm is bound to work in a different way.

Trajectory analysis in [26] demonstrates the fact that the convergence of the PSO algorithm may be achieved if each particle converges to its local attractor, $p_i = (p_{i1}, p_{i2}, \cdots p_{iD})$ with its components defined by

$$p_{i,j}(k) = \frac{\varphi_1 P_{i,j}(k) + \varphi_2 P_{g,j}(k)}{\varphi_1 + \varphi_2}, \tag{4.15}$$

where P_i is personal best position of particle i, P_g is the global best position of all particles, $\varphi_1 = c_1 r_1$, $\varphi_2 = c_2 r_2$. Equation (4.8) can be simplified as

$$V_{i,j}(k+1) = V_{i,j}(k) + \varphi(p_{i,j}(k) - X_{i,j}(k)), \tag{4.16}$$

where $\varphi = \varphi_1 + \varphi_2$. The effect of φ on the system is very important since the particle's trajectory is dependent on the value of the control parameter.

It can be seen that p_i is a stochastic attractor of particle i that lies in a hyper-rectangle with P_i and P_g being two ends of its diagonal and moves by following P_i and P_g. In fact, when the particles are converging to their own local attractors, their personal best positions P_i, local attractors p_i and the global best positions P_g will all converge to one point, leading the PSO algorithm to converge.

In quantum mechanics, the governing equation is the general time-dependent Schrodinger equation

$$i\hbar \frac{\partial}{\partial t}\psi = H\psi, \tag{4.17}$$

where $\psi(y,t)$ is the wave function, which is used to describe the state of a particle instead of position X and velocity V in classical mechanics. $|\psi(y,t)|^2$ measures the chance of finding the particle at a certain time and position, which satisfies

$$\int_{-\infty}^{\infty} |\psi(y,t)|^2 \, dy = 1, \tag{4.18}$$

H is a time-independent Hamiltonian operator given by

$$H = -\frac{\hbar^2}{2m}\nabla^2 + V(y) \tag{4.19}$$

and $\hbar$ is Planck's constant, m is the mass of the particle, and $V(y)$ is the potential energy distribution.

The particle moves around and careers toward point p_i with its kinetic energy (velocity) declining to zero, like a returning satellite orbiting the earth. From the point view of dynamics, to avoid explosion and to guarantee convergence, the particle must be in a bound state, moving in an attraction potential field with centre in point p_i. In classical dynamics, the particle in the original PSO system flies in an attraction potential field (gravitational potential) which can be determined using Newton's law of universal gravitation. On the other hand, particles in PSO with quantum behaviour have to move in an attractive potential field that can ensure a bound state. In QPSO, the simplest potential well, delta-well, is employed to bind the particle

$$V(y) = -\gamma\delta(y) \tag{4.20}$$

where $y = X_{i,j} - p_{i,j}$, p is the attractor, γ is a positive number proportional to the depth of the potential well, the depth is infinite at the origin and zero elsewhere. Thus, the delta potential well is an idealized realization of an infinitely strong attractive potential field that works at a single location.

Assuming the principle of separation of variables, the time-dependent wave-function is separated from the spatial dependence [142]. Then $\psi(y)$ can be found by solving the following time-independent Schrodinger equation

$$E\psi = [-\frac{\hbar^2}{2m}\nabla^2 + V(y)]\psi \tag{4.21}$$

Such solutions are called stationary states. Sun et al. [23] give the details of solving Equation (4.21). The wave-function at iteration $k+1$ is obtained as

$$\psi(y_{i,j}(k+1)) = \frac{1}{\sqrt{L_{i,j}(k)}} \exp\left(-\frac{\left| X_{i,j}(k+1) - p_{i,j}(k) \right|}{L_{i,j}(k)} \right). \tag{4.22}$$

Hence, the probability density function Q is a double exponential distribution as shown below

$$Q(y_{i,j}) = \left| \psi(y_{i,j}) \right|^2 = \frac{1}{L_{i,j}(k)} \exp\left(-2\frac{\left| X_{i,j}(k+1) - p_{i,j}(k) \right|}{L_{i,j}(k)} \right), \tag{4.23}$$

where $L_{i,j}(k)$ is the characteristic length of the potential well.

To 'measure' the location of the particle, one needs to collapse the wave function of a moving particle into the localized space of the measurement. The localization process can be easily accomplished through the Monte Carlo procedure as: (1) generate a random variable uniformly distributed in the local space $s = \frac{1}{L}u$, u is a random number uniformly distributed in $(0,1)$; (2) equate the uniform distribution to the true probability distribution (Equation (4.23)) to obtain

$$u = \exp\left(-2\frac{\left| X_{i,j}(k+1) - p_{i,j}(k) \right|}{L_{i,j}(k)} \right); \tag{4.24}$$

(3) solve for the position $X_{i,j}$ in terms of the random variable as

$$X_{i,j}(k+1) = p_{i,j}(k) \pm \frac{L_{i,j}(k)}{2} \ln\left(\frac{1}{u} \right). \tag{4.25}$$

where $L_{i,j}(k)$ is evaluated in [23] as

$$L_{i,j}(k) = 2\alpha \left| p_{i,j}(k) - X_{i,j}(k) \right| \tag{4.26}$$

with which, the update equation for the position of particles becomes

$$X_{i,j}(k+1) = p_{i,j}(k) \pm \alpha \left| p_{i,j}(k) - X_{i,j}(k) \right| \ln\left(\frac{1}{u} \right). \tag{4.27}$$

Here α, named as the contraction-expansion coefficient, is the only parameter in QPSO to control the rate of convergence. The most commonly used control strategy of α is decreasing linearly from 1.0 to 0.5. That is

$$\alpha(k) = 1.0 - 0.5\frac{k}{k_{max}}. \tag{4.28}$$

$L_{i,j}(k)$ can be also determined from [24] as

$$L_{i,j}(k) = 2\alpha\left|C_j(k) - X_{i,j}(k)\right|, \tag{4.29}$$

where C is the mean best position which is defined as the mean value of the personal best positions of all particles

$$C(k) = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}(k), \quad \frac{1}{M}\sum_{i=1}^{M} P_{i2}(k), \quad \ldots \quad , \quad \frac{1}{M}\sum_{i=1}^{M} P_{iD}(k)\right) \tag{4.30}$$

Hence, the position of the particle updates according to the equation

$$X_{i,j}(k+1) = p_{i,j}(k) \pm \alpha\left|C_{i,j}(k) - X_{i,j}(k)\right|\ln\left(\frac{1}{u}\right). \tag{4.31}$$

Equation (4.31) is known as the quantum-behaved particle swarm optimization (QPSO).

Some benchmark function tests shown in [24] demonstrated the superiority of Equation (4.31) over Equation (4.27). The QPSO procedure can be described in the following pseudo code.

Algorithm of QPSO

Initialize the positions of the population $X(0)$, personal best positions $P(0)$, $\alpha = 1.0$;

Do while ($k < k_{max}$)

For each particle $i = 1, 2, ..., M$

Evaluate the fitness $f(X_i(k))$;

If $f(X_i(k)) < f(P_i(k))$ then

$P_i(k) = X_i(k)$;

If $f(P_i(k)) < f(P_g(k))$ then

$P_g(k) = P_i(k)$;

End for

Compute the mean best position $C(k)$ according to Equation (4.30);

For each particle $i = 1, 2, ..., M$

Compute $p_i(k)$ according to Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

End for

Update α according to Equation (4.28);

End do

The QPSO algorithm is different from the original PSO algorithm. The update equation of the former method given by Equation (4.31) ensures particles appear in the entire D-dimensional search space during each of the iteration. The particles in the latter method can only move in a bounded space. Using the global convergence criterion in [14], one can conclude that the QPSO is a global convergent algorithm whereas the original PSO is not. Moreover, unlike the original PSO method, the QPSO method does not require velocity vectors for the particles at all and has fewer parameters to control, making the method easier to implement. Experimental results performed on some well-known benchmark functions show that the QPSO method has better performance than the original PSO method [23]-[25].

4.4 Modified Quantum-Behaved Particle Swarm Optimisation

The performance of an evolutionary algorithm depends on the global search ability, convergence rate, solution precision, robustness, etc. In QPSO algorithm, the loss of diversity and prematurity is inevitable, as other population-based evolutionary algorithms. The particles cluster together gradually in the latter search period. As a result the swarm is likely to be trapped into local optima. In order to improve the performance for the complex multi-modal problems and to avoid the premature convergence problem of QPSO, various strategies have been proposed from different aspects.

Experiments were performed on a set of benchmark functions to demonstrate the efficiency of the modifications before being applied to real applications. Four benchmark functions are used to evaluate the performance of the modified methods, both in terms of the

optimum solution and the rate of convergence to the optimum solution. These benchmark functions were widely used in evaluating performance of evolutionary methods [13]-[28]. All the functions are all minimization problems and have the global minimum at the origin or very close to the origin, which are given in Table 4.1. Functions f_1 and f_2 are simple unimodal problems, f_3 and f_4 are highly complex multimodal problems with several local minima. All the particles are randomly initialized in an area equal to one quarter of the feasible search domain that is guaranteed not to contain in the optimal solution.

Table 4.1: Benchmark functions and parameter settings.

Functions	Mathematical Expression	Range	Initialization
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$(-100, 100)^D$	$(50, 100)^D$
Rosenbrock	$f_2(x) = \sum_{i=1}^{D} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$	$(-30, 30)^D$	$(15, 30)^D$
Rastrigrin	$f_3(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$(-5.12, 5.12)^D$	$(2.56, 5.12)^D$
Griewank	$f_4(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$(-600, 600)^D$	$(300, 600)^D$

4.4.1 Quantum-behaved particle swarm optimisation with perturbation operator

The global convergence of QPSO or other random search algorithm means that the algorithm will only hit the global optimal solution in an infinite number of iterations. However, when the algorithm is applied to real world problems, only a finite number of iterations is allowed so that premature convergence is inevitable. There is much room for improvement in QPSO, particularly when the algorithm is used to solve the present complex ill-posed problem. So far many improved strategies have been proposed to enhance the search ability of the algorithm [43]-[45]. In this work, a novel perturbation operation [145] is incorporated into the algorithm to enhance the efficiency of QPSO in finding the global optimal solutions on complex functional terrains [46]. In this modified version of QPSO, the diversity of the swarm can be enhanced by exerting the random perturbation on each particle as

$$X_i^{per}(k) = X_i(k) + pert_{coff}\, X_i(k)(r_1 - r_2), \qquad (4.32)$$

where $pert_{coff} = a \exp\left\{2.3025851\frac{(k_{max}-k)}{(k_{max}-1)}\right\}$ is a nonlinear perturbation coefficient varying from 10a to a (a is a scale parameter which can be adjusted according to the specific problems) depicted in Figure 4.8. Here r_1, r_2 are uniformly distributed random numbers in $(0,1)$. This diversification strategy indeed can improve the global search ability of the swarm, particularly at the later stage of the search process when the diversity is at such a low level that further global search may be impossible for the particles leading to premature convergence. The procedure of the perturbed QPSO is given in the following pseudo code.

Algorithm of QPSO-PER

Initialize the positions of the population $X(0)$, personal best positions $P(0)$, $\alpha = 1.0$;

Do while ($k < k_{max}$)

For each particle $i = 1, 2, ..., M$

Evaluate the fitness $f(X_i(k))$;

If $f(X_i(k)) < f(P_i(k))$ then

$P_i(k) = X_i(k)$;

If $f(P_i(k)) < f(P_g(k))$ then

$P_g(k) = P_i(k)$;

End for

Compute the mean best position $C(k)$ according to Equation (4.30);

For each particle $i = 1, 2, ..., M$

Compute $p_i(k)$ according to Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

Do perturbations to position $X_i(k+1)$ according to Equation (4.32);

End for

Update α according to Equation (4.28);

End do

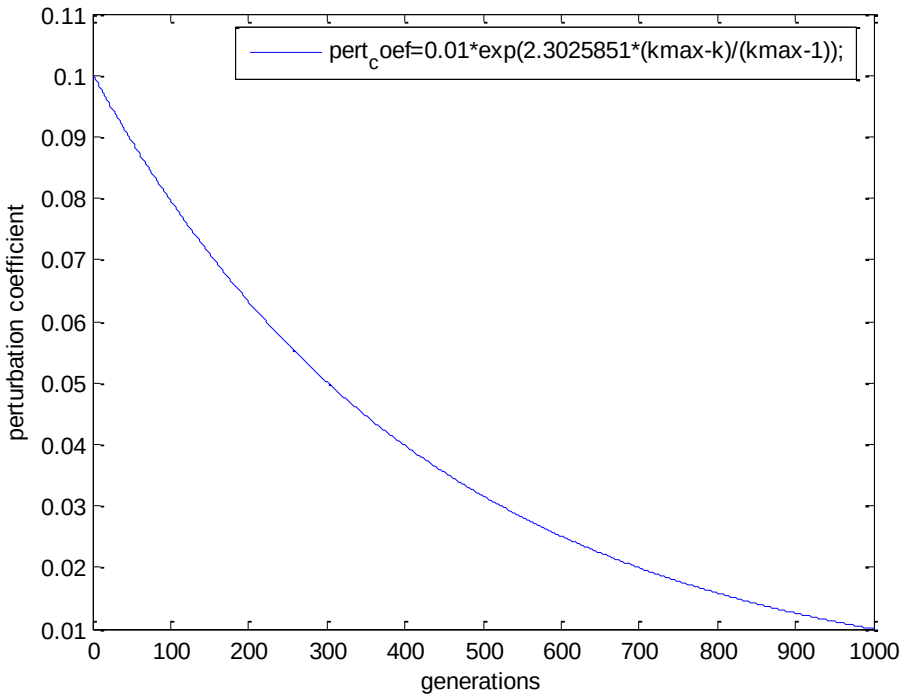


Figure 4.8: Perturbation coefficient decreasing with generation.

Simulations were carried out to observe the rate of convergence and the quality of the optimum solution of the proposed method introduced in this investigation in comparison with the original PSO and QPSO. The neighbourhood of a particle is all the population, which is named as global best model. All benchmark functions in Table 4.1 were tested with dimensions 10, 20 and 30. For each function, 50 trials were carried out and the average optimal value and the standard deviation are presented. Different numbers of maximal generations (k_{max}) are used according to the dimensionality of the problem under consideration. In this section, all empirical experiments were carried out with a population size of 30.

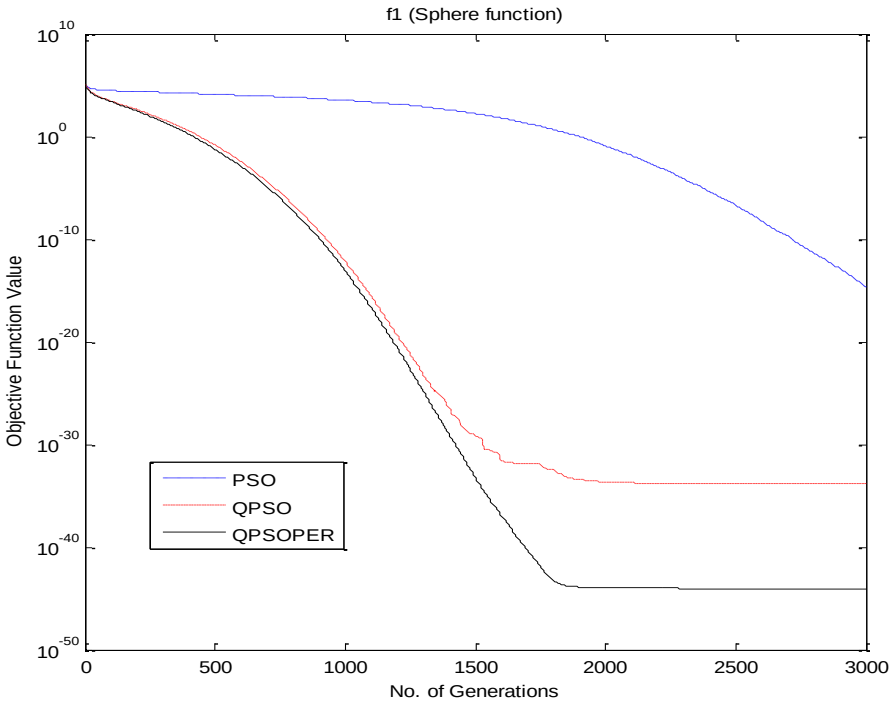
The mean best values and standard deviation for 50 trials of each algorithm on each of the nine benchmark functions are listed in Table 4.2. The numerical results show that the modified QPSO with a perturbation operator (QPSO-PER) performed better on all the tested functions than the original PSO and QPSO.

Table 4.2: Results of the benchmark functions.

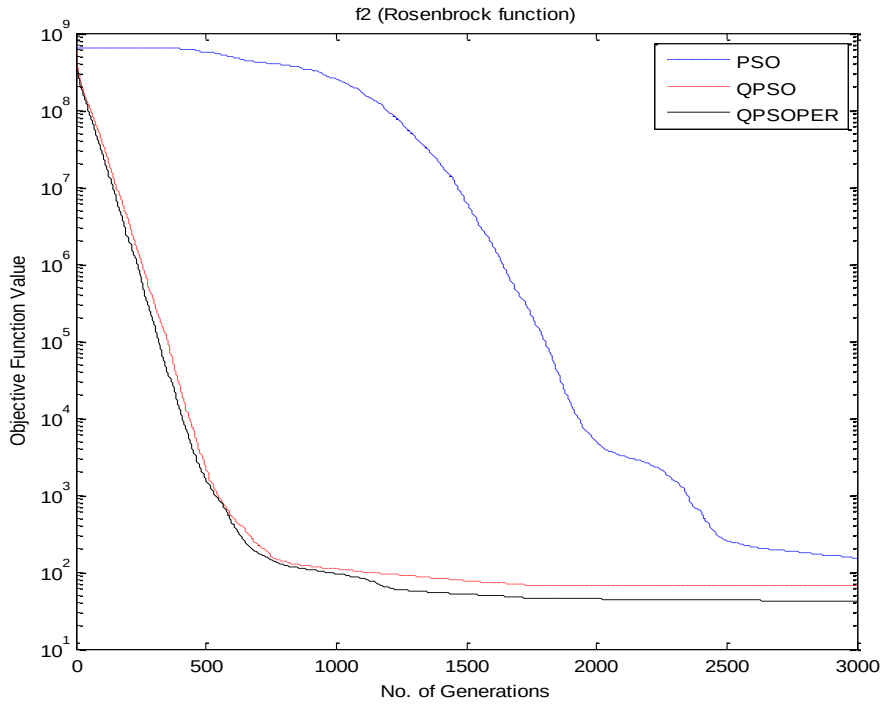
Function	Dimension	k_{max}	Mean Best Value (Standard Deviation)		
			PSO	QPSO	QPSO-PER
f_1	10	1000	5.42E-23 (1.99E-22)	**0.00** (0.00)	**0.00** (0.00)
	20	2000	3.90E-19 (1.30E-18)	2.49E-42 (0.00)	**1.40E-45** (0.00)
	30	3000	3.43E-15 (8.94E-15)	1.67E-34 (0.00)	**8.41E-45** (0.00)
f_2	10	1000	56.80 (85.75)	30.55 (60.17)	**21.86** (28.02)
	20	2000	113.03 (204.15)	55.41 (58.03)	**42.10** (34.63)
	30	3000	152.36 (228.74)	65.93 (64.45)	**41.75** (31.07)
f_3	10	1000	4.23 (2.11)	4.67 (2.59)	**3.78** (1.97)
	20	2000	17.73 (4.34)	14.35 (4.23)	**10.89** (3.43)
	30	3000	37.67 (9.73)	24.83 (6.74)	**19.99** (5.62)
f_4	10	1000	9.12E-02 (3.79E-02)	4.96E-02 (4.09E-02)	**3.36E-02** (3.28E-02)
	20	2000	2.40E-02 (1.62E-02)	1.76E-02 (1.61E-02)	**1.53E-02** (2.45E-02)
	30	3000	1.80E-02 (1.95E-02)	9.68E-03 (1.29E-02)	**4.93E-03** (8.53E-03)

The convergence history of the original PSO, QPSO and the perturbed QPSO are shown in Figure 4.9. All the benchmark functions are considered in 30 dimensions. In Figure 4.9a-Figure 4.9d, QPSO-PER performs almost the same as or even worse than QPSO in the

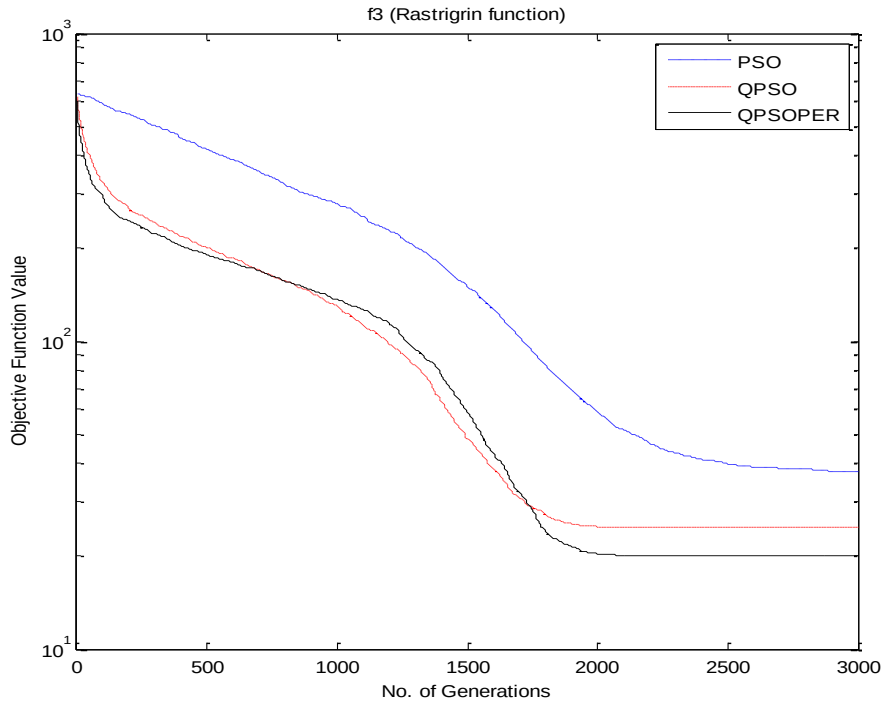
early stages of the computation. Due to the perturbation applied to the particles, the diversity of the population increases, prematurity of the particles can be avoided.



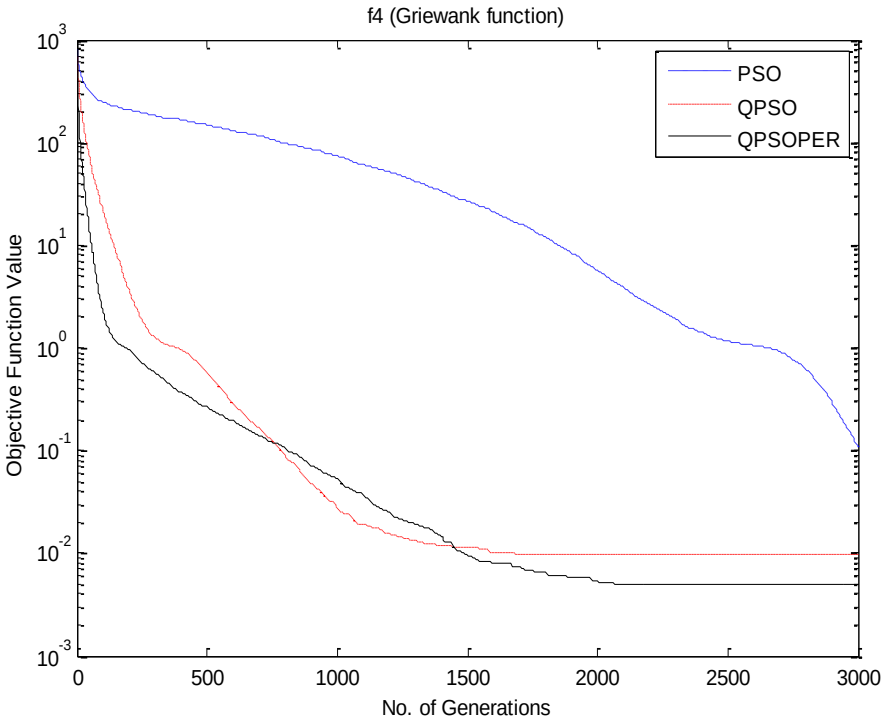
(a)



(b)



(c)



(d)

Figure 4.9: Convergence history of the original PSO, QPSO and QPSO-PER on benchmark functions.

4.4.2 Quantum-behaved particle swarm optimisation with ring topology model

In the original QPSO, the global topology (gbest) model (Figure 4.7a) was used, in which, the neighbour of each particle is the entire swarm. Each particle is connected to and able to obtain information from every other particle in the swarm. The advantage of the gbest model is its fast convergence and it will always result to better performance on simple unimodal problems. For more complex multimodal problems, the fast convergence of the gbest model leads to prematurity and prevents the particles escaping from local optima.

In contrast, the local topology model (lbest) (e.g. Figure 4.7b) connects each particle to a limited number of particles in its neighbourhood instead of all the particles. The advantage of the lbest model is its slower convergence rate relative to the gbest model so that the particle would have enough time to search a wide area instead of stopping with a premature convergence. A number of different limited communication topologies have been tested. The lbest model (ring topology) used here is perhaps the simplest form of the local topology, which was also adopted in the standard PSO (SPSO) [22]. In QPSO with lbest model (SQPSO for short), every particle i has its own lbest LBEST_i. Hence Equation (4.15) is modified as below

$$\mathrm{p}_i(\mathrm{k}) = \varphi \mathrm{P}_i(\mathrm{k}) + (1-\varphi)\mathrm{LBEST}_i(\mathrm{k}). \tag{4.33}$$

Simulations were carried out to observe the rate of convergence and the quality of the optimal solution of the SQPSO in comparison with the SPSO and the original QPSO. All benchmark functions in Table 4.1 were tested with dimension $\mathrm{D} = 30$. For each function, 50 trials were carried out and the average optimal value and the standard deviation are presented. In this paper, all empirical experiments were carried out with a population size of 50, which is the same value as that in SPSO.

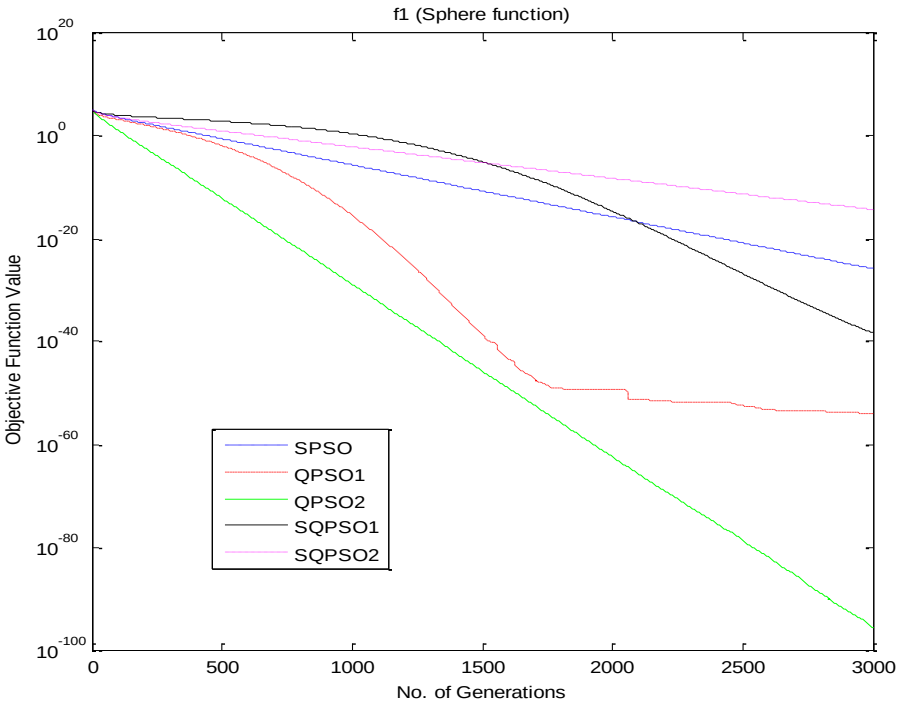
For QPSO and SQPSO, two kinds of contraction-expansion coefficient α are adopted. QPSO with linearly decreasing α is named as QPSO1 and QPSO with constant $\alpha = 0.75$ is named as QPSO2. SQPSO with linearly decreasing α is named as SQPSO1 and SQPSO with constant $\alpha = 0.75$ is named as SQPSO2.

The mean best values and standard deviation for 50 trials on the four benchmark functions are listed in Table 4.3. For the unimodal problem f_1, QPSO2 performed better than others. With finite number of generations, e.g. 3000, QPSO2 was able to converge quickly to

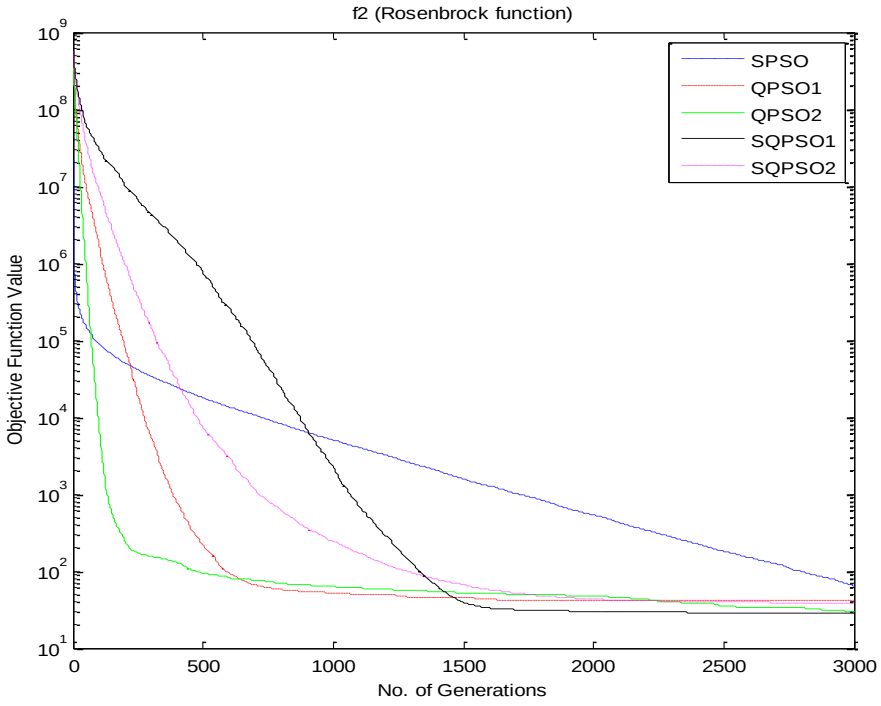
the unique minimum at origin. Optimisation of f_2 is a hard task although the function is unimodal. It is shown that SQPSO1 beat its competitors in performing this task. For the multimodal function f_3, QPSO1 yielded the best results. The difference between the performance of the algorithms on these two functions shows to be more significant than that on the other functions. f_4 is a multimodal function which is not so difficult to be optimised as f_3. The result obtained by SQPSO1 appears to be closer to the optimum, showing that SQPSO may be also an efficient tool for optimizing multimodal functions. It could be due to the slow convergence speed of SQPSO1 that the particles are able to search widely for the whole domain and can efficiently avoid trapping into the local optimum. Comparison of the convergences between all the algorithms over the eight benchmark functions is shown in Figure 4.10.

Table 4.3: Results of the benchmark functions using SPSO, QPSO and SQPSO.

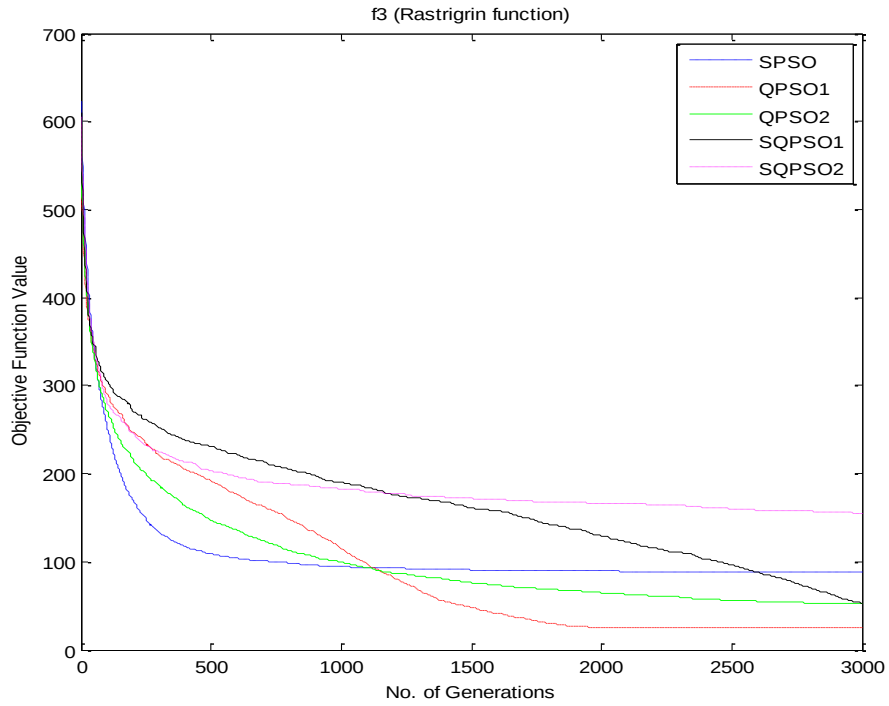
Function	Mean Best Value (Standard Deviation)				
	SPSO	QPSO		SQPSO	
		$\alpha \sim (1.0 \rightarrow 0.5)$	$\alpha = 0.75$	$\alpha \sim (1.0 \rightarrow 0.5)$	$\alpha = 0.75$
f_1	1.50E-26 (2.31E-26)	1.22E-54 (5.06E-54)	1.68E-96 (9.43E-96)	4.91E-39 (8.37E-39)	5.20E-15 (7.99E-15)
f_2	65.07 (42.02)	41.94 (28.56)	30.01 (33.63)	27.91 (13.38)	37.71 (24.24)
f_3	127.70 (31.54)	24.59 (11.63)	51.89 (24.60)	52.36 (23.78)	155.05 (13.45)
f_4	9.37E-04 (2.56E-03)	5.81E-03 (9.41E-03)	1.03E-02 (1.08E-02)	2.30E-04 (1.61E-03)	1.03E-02 (3.11E-02)


f1 (Sphere function)
Objective Function Value
No. of Generations
SPSO
QPSO1
QPSO2
SQPSO1
SQPSO2

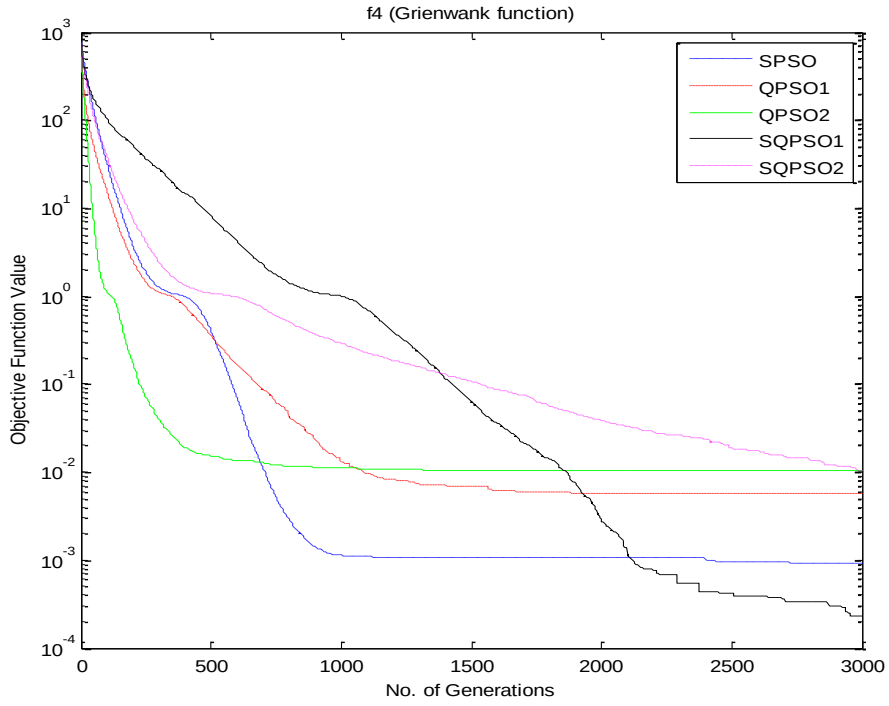
(a)


f2 (Rosenbrock function)
Objective Function Value
No. of Generations
SPSO
QPSO1
QPSO2
SQPSO1
SQPSO2

(b)



(c)



(d)

Figure 4.10: Convergence history of SPSO, QPSO and SQPSO on benchmark functions.

4.4.3 Quantum-behaved particle swarm optimisation with Gaussian mutation

PSO, QPSO and other population based evolutionary algorithms suffer from a loss of diversity in the population during the later stage of the search period. As a result, particles are clustered together gradually and the swarm is likely to be trapped into a local optimum. As in evolutionary programming, mutation operators may be used to prevent loss of diversity in the population and allow a greater region of the search space to be covered. Mutation operators introduce new individuals into a population by creating a variation of the current individual. The addition of variability into the population prevents stagnation of the search in local optima [47]. Works has been done in the investigation of PSO with mutation operators [48]-[50]. Andrews [51] gave an investigation and full comparison of the different mutation operators. Few literatures focused on QPSO with mutation operators.

Liu [52] used a Cauchy mutation operator to change the value of the mean best position or the global best position as

$$\text{mutate}(\text{mbest}_d) = \text{mbest}_d + \phi\delta, \tag{4.34}$$

$$\text{mutate}(P_{g,d}) = P_{g,d} + \phi\delta, \tag{4.35}$$

where ϕ is the mutation size, δ is a random variable of Cauchy distribution, the probability density function is

$$g(x) = \frac{a}{\pi(x^2 + a^2)}, \tag{4.36}$$

a is a scale parameter that determines the shape of the distribution, which is controlled by the annealing function

$$a = a_0(CR)^k, \tag{4.37}$$

where $a_0 = 2$, cooling rate $CR = 0.99$, and k is the current iteration. From numerical analysis and experience, ϕ and a_0 are problem dependent and require careful selection.

In this modification, the Gaussian distribution as depicted in Figure 4.11 instead of Cauchy distribution mutation operator is applied to QPSO. The Gaussian distribution has the advantage of having large probability to generate small perturbations to the position, the probability density function is

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right), \tag{4.38}$$

where the variance σ^2 controls the width of the distribution. Larger σ^2 has higher probability to make large perturbation, and smaller σ^2 gives higher probability to make smaller perturbation. Since the search domain is problem dependent, σ^2 can be set to relate to the dynamic range of the searching domain.

From Equation (4.31), the position of a particle for the next generation is decided by the current position $X_i(k)$, personal best position P_i, global best position P_g and the mean best position $\text{mbest}(k)$. If there is a premature convergence problem, the global best position P_g is the local optimum. If P_g was changed by mutation operation, the search direction of the particles would change. Hence new regions are likely to be searched and a new P_g may be found. Applying the mutation operation to the mean best position $\text{mbest}(k)$ will generate the same effect.

In the following benchmark function tests, each component of the mean best position or global best position is mutated according to the mutation probability $1/D$, and σ is set to be 0.1 times the length of the dynamic range of the particle dimension being mutated.

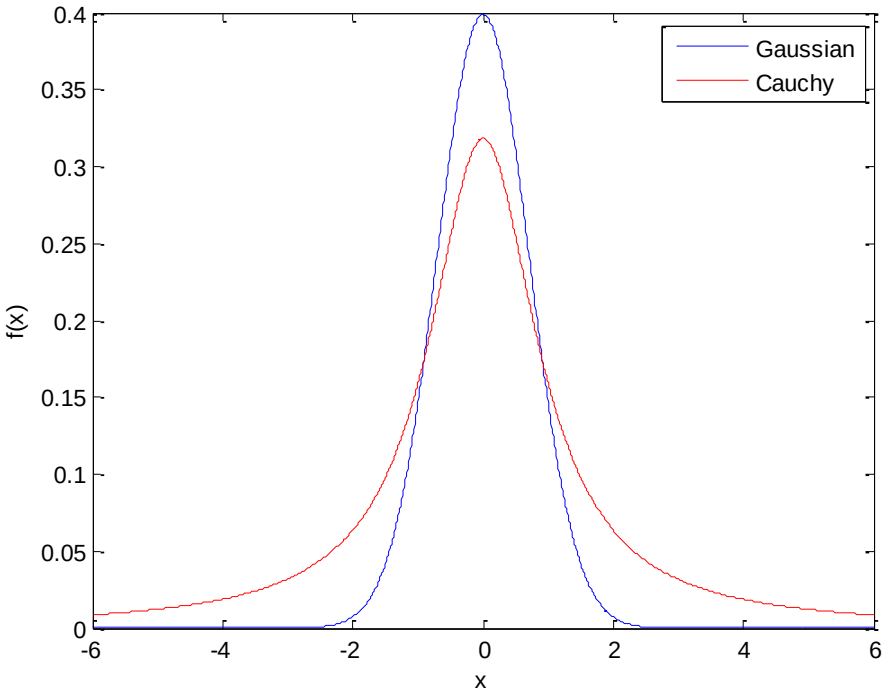


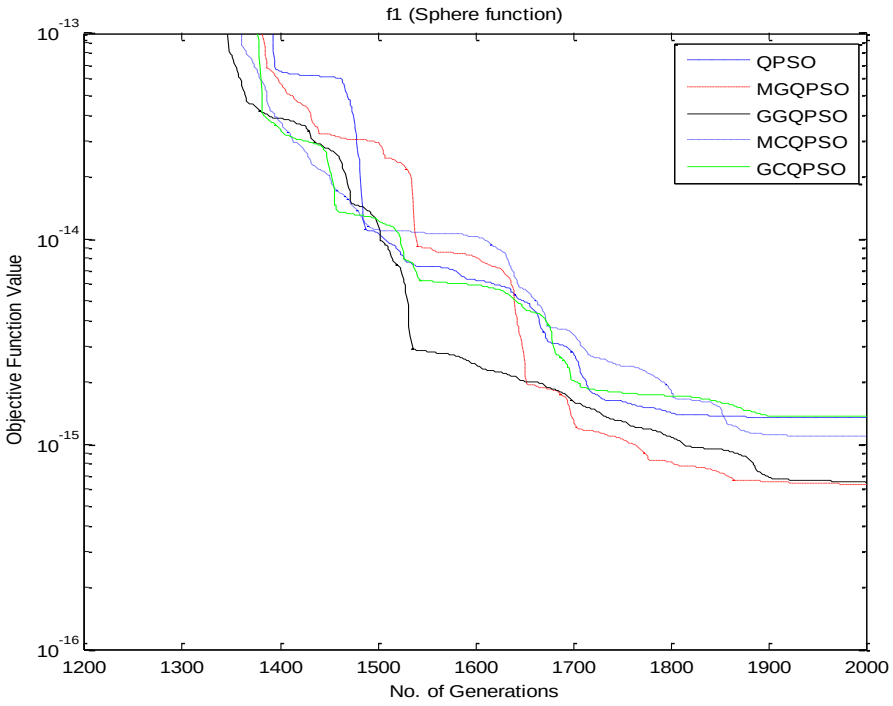
Figure 4.11: Standard distribution shape of Gaussian and Cauchy.

As in [51], for each function, the dimension size is set as 20 and the number of particles is 30. The maximum number of generations is set as 2000. A total of 50 runs were carried out and the mean best values of the optima are presented in Table 4.6, in which, the abbreviations GQPSO and CQPSO stand for QPSO with Gaussian Mutation and QPSO with Cauchy Mutation respectively. For the convenience of comparison, the scale parameter a is set to 0.1 times the range of the particle dimension.

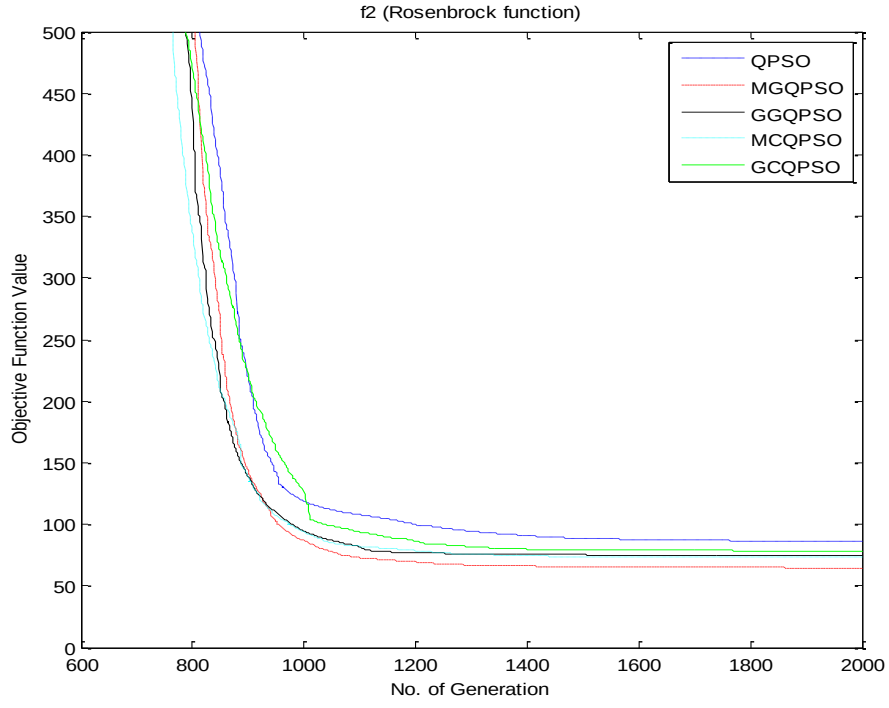
Table 4.4: Results of the benchmark functions test

functions	Mean best value				
	QPSO	GQPSO		CQPSO	
		mbest	gbest	mbest	gbest
f_1	1.34E-15	**6.33E-16**	6.49E-16	1.03E-15	1.36E-15
f_2	86.46	**64.74**	75.06	73.17	78.59
f_3	40.27	**34.52**	39.54	36.76	36.52
f_4	1.37E-02	**7.58E-03**	9.74E-03	9.95E-03	9.82E-03

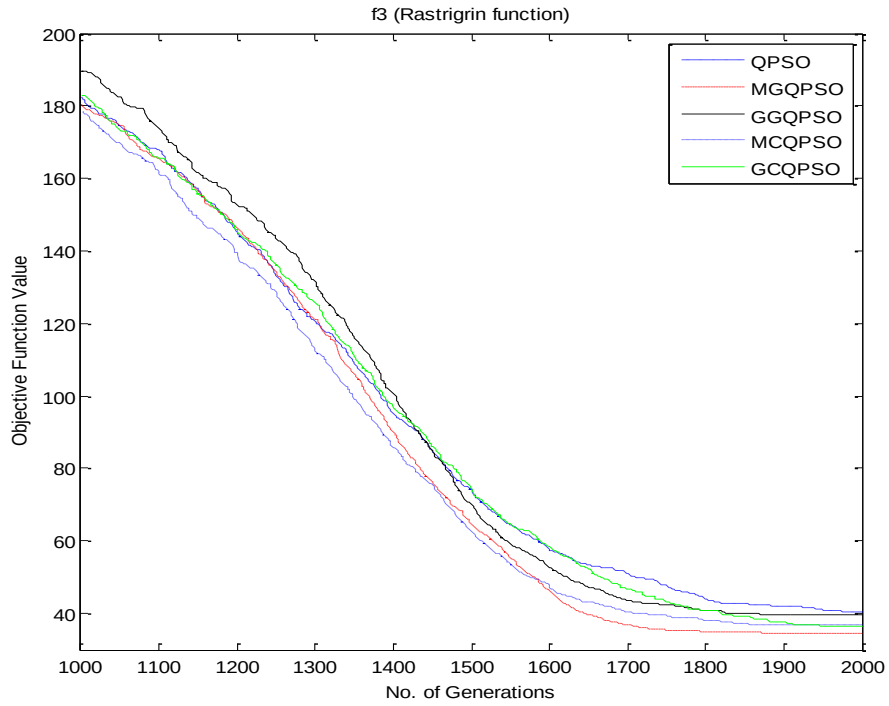
The convergence history of QPSO with different parameter selection methods is presented in Figure 4.12, in which, the abbreviations MGQPSO, GGQPSO, MCQPSO and GCQPSO stand for QPSO with Gaussian mutation in mbest, QPSO with Gaussian mutation in gbest, QPSO with Cauchy mutation in mbest and QPSO with Cauchy mutation in gbest respectively. Note that MGQPSO outperforms others in all the benchmark functions.


f1 (Sphere function)
QPSO
MGQPSO
GGQPSO
MCQPSO
GCQPSO
Objective Function Value
No. of Generations
10^-13
10^-14
10^-15
10^-16
1200
1300
1400
1500
1600
1700
1800
1900
2000

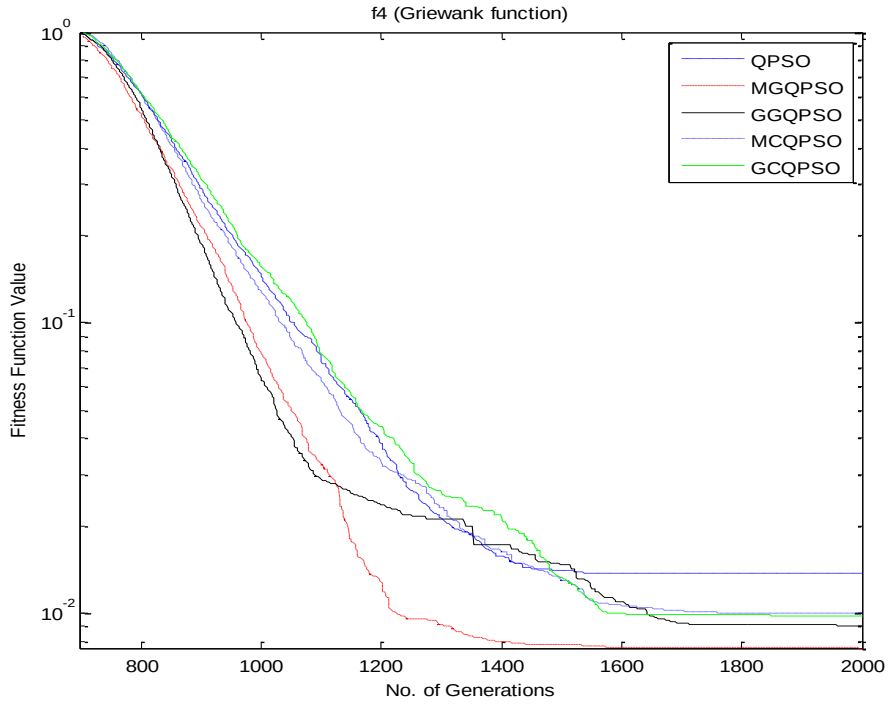
(a)


f2 (Rosenbrock function)
QPSO
MGQPSO
GGQPSO
MCQPSO
GCQPSO
Objective Function Value
No. of Generation
500
450
400
350
300
250
200
150
100
50
0
600
800
1000
1200
1400
1600
1800
2000

(b)



(c)



(d)

Figure 4.12: Convergence history of QPSO, MGQPSO, GGQPSO, MCQPSO and GCQPSO on benchmark functions.

4.4.4 Parameter control in quantum-behaved particle swarm optimisation

The contraction-expansion coefficient α, the only parameter in QPSO, influences the trade-off between global and local exploration ability of the particles. A larger α facilitates global exploration (searching new areas) while a smaller α tends to facilitate local exploitation to fine-tune the current search area. Suitable selection of α can provide a balance between global and local exploration ability and thus save searching time to find the optimum. $\alpha \leq 1.7$ must be satisfied to guarantee the convergence [25]. The linearly decreasing expansion-constriction from 1.0 to 0.5 is commonly used as Equation (4.28), this is because the larger α at the beginning helps to find good seeds and the later smaller α facilitates fine-tune search. A constant value $\alpha = 0.75$ is analyzed by Sun in [27] that it has better performance in simple unimodal problems. Many adaptive selection methods were proposed in order to enhance the performance of QPSO. In [25], an adaptive method on individual level is proposed as Equation (4.40), and proved to outperform the linearly decreasing method.

$$\alpha(z) = \begin{cases} 0.6, & z > 0 \\ 0.7, & -2 < z \leq 0 \\ 0.6 + 0.1 \times k, & -k-1 < z \leq -k \\ 1.0 + 0.2 \times (k-4), & -k-1 < z \leq k \\ 1.8, & z \geq -8 \end{cases} \tag{4.39}$$

where $z = \ln(\Delta F)$, ΔF is defined as below

$$\Delta F = \frac{(F_i - F_{gbest})}{MIN(ABS(F_i), ABS(F_{gbest}))} \tag{4.40}$$

where F_i is the fitness of the i th particle, F_{gbest} is the global best fitness of the swarm. However, z has to be computed according to Equation (4.41) before the value of α is obtained, which is time consuming and not easy to use.

In [28], Dong defined a self-adaptive Inertia weight function for a particle in terms of its fitness value, the swarm size and the dimension size as

$$\omega = \frac{1}{3 - \exp\left(-\frac{M}{200}\right) + \left(\frac{R}{8D}\right)^2} \tag{4.41}$$

where M is the swarm size, D is the dimension size of the solution space and R denotes the fitness rank of the given particle. But the value of ω is always smaller than 0.5, which is not suitable for QPSO.

Inspired by Simulated Annealing, the contraction-expansion coefficient α may be adjusted according to the annealing function

$$\alpha = \alpha_0 * (\mathrm{CR})^{\mathrm{k}}, \tag{4.42}$$

where CR is chosen as according to the maximum iteration.

For the multimodal problems, it's easy to get trapped into a local optimum because of the particles' premature convergence. So here, a new parameter control method is proposed in which α adjusts according to the cosine function as

$$\alpha = 0.5\cos\left(\frac{\pi \mathrm{k}}{2\mathrm{k}_{\max}}\right) + 0.5 \tag{4.43}$$

where k is the current generation, $\mathrm{k}_{\max}$ is the maximum generation. In Figure 4.13, one can note that the cosine function decreases slower than the linearly decreasing function and annealing function. For the complex multimodal, QPSO with cosine decreasing α may perform better in the global exploration.

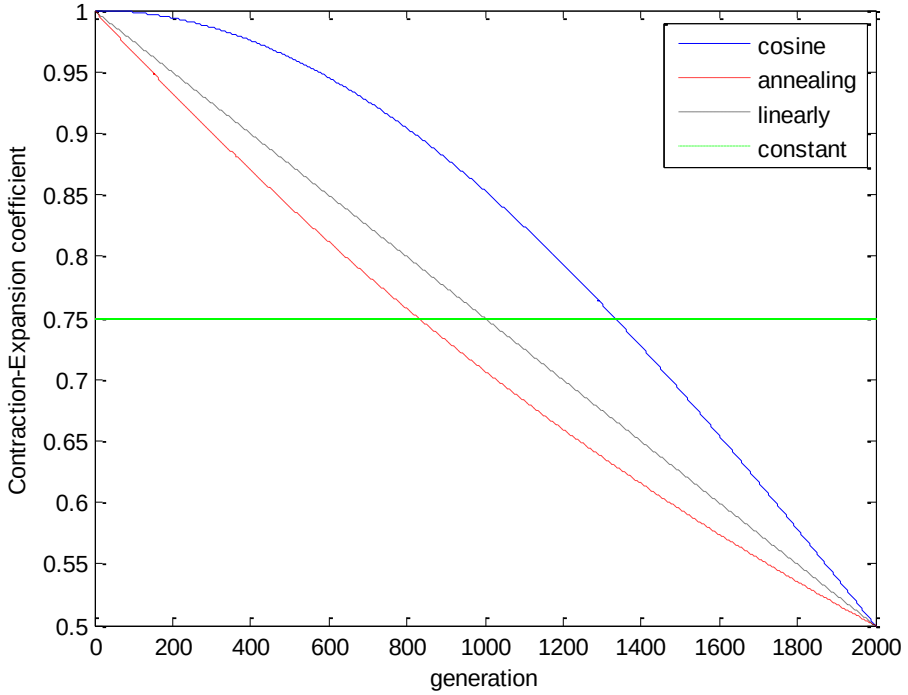


Figure 4.13: Contraction-Expansion coefficient decreases in different way.

As in [20], for each function, three different dimension sizes are tested, which are 10, 20 and 30. The maximum number of generations is set as 1000, 1500 and 2000 corresponding to the dimensions 10, 20 and 30, respectively. Population sizes of 20, 40 and 80 are used with different dimensions. A total of 50 runs were carried out and the average optimal value and the standard deviation are presented.

Four kinds of contraction-expansion coefficient α selection methods are tested. In the results tables, the abbreviations QPSO-CON, QPSO-LIN, QPSO-COS and QPSO-ANN stand for QPSO with constant α, QPSO with linearly decreasing α, QPSO with cosine decreasing α and QPSO with annealing decreasing α respectively.

The mean best values and standard deviation for 50 trials on the four benchmark functions are listed in Table 4.5-Table 4.8. For the unimodal function f_1, QPSO-CON outperforms others for all the dimension sizes and converges fast to the global optimum. For the simple multi-modal function f_2, QPSO-CON performs better in the dimension sizes 20 and 30 except $D=10$. Rastrigin function f_3 is a fairly difficult problem due to its large number of local minima, on which, QPSO-COS works better. Because the cosine function decreases more slowly than the linearly decreasing function and annealing function, the particles have more time to explore the big area before fine-tuning the small area. So the QPSO-COS can prevent the particles from being trapped into the local optimum and the global search ability is enhanced. For Griewank f_4, QPSO with annealing α QPSO-ANN outperforms others, either in terms of the optimum and convergence. Within finite number of iterations, e.g. 2000, QPSO-ANN converges faster than QPSO-COS and QPSO-LIN to a minimum.

Comparison of the convergence between all the algorithms over the five benchmark functions is shown in Figure 4.14. In Figure 4.14a-Figure 4.14b, QPSO-CON converges fastest to a minimum on the unimodal function and simple multimodal function. In Figure 4.14c, the convergence of QPSO-COS is the slowest, but the superiority appears at the later stage of the searching process. QPSO-ANN shows the best performance in Figure 4.14d. From Figure 4.14e, one can note that QPSO-COS has the best convergence, which may illustrate the ability of QPSO-COS in solving the complex problems.

Table 4.5: Sphere function.

M	D	k_{max}	**MEAN BEST VALUE (STANDARD DEVIATION)**			
			QPSO-CON	QPSO-LIN	QPSO-COS	QPSO-ANN
20	10	1000	**5.78E-76** (1.40E-75)	1.85-40 (1.29E-39)	8.44E-36 (5.90E-35)	2.24E-42 (1.30E-41)
	20	1500	**8.39E-57** (3.04E-56)	6.67E-21 (4.62E-20)	4.43E-19 (2.93E-18)	9.35E-22 (2.78E-21)
	30	2000	**1.82E-45** (6.27E-45)	2.04E-15 (4.28E-15)	8.19E-12 (3.15E-11)	6.65E-14 (1.92E-13)
40	10	1000	**2.66E-87** (1.15E-86)	5.30E-73 (3.64E-72)	1.94E-59 (1.36E-58)	2.13E-76 (1.47E-75)
	20	1500	**2.00E-71** (9.72E-71)	1.50E-43 (6.14E-43)	1.32E-34 (3.76E-34)	5.31E-42 (3.60E-41)
	30	2000	**1.24E-58** (3.22E-58)	1.50E-30 (3.95E-30)	3.02E-22 (1.48E-21)	2.92E-30 (8.39E-30)
80	10	1000	1.39E-97 (5.81E-97)	1.02E-100 (6.96E-100)	3.86E-79 (1.34E-78)	**2.28E-109** (1.59E-108)
	20	1500	**1.39E-85** (4.67E-85)	2.97E-70 (1.11E-69)	6.45E-53 (3.05E-52)	9.20E-72 (5.18E-71)
	30	2000	**3.48E-72** (9.35E-72)	1.57E-50 (5.28E-50)	5.05E-37 (2.06E-36)	2.22E-51 (1.31E-50)

Table 4.6: Rosenbrock function.

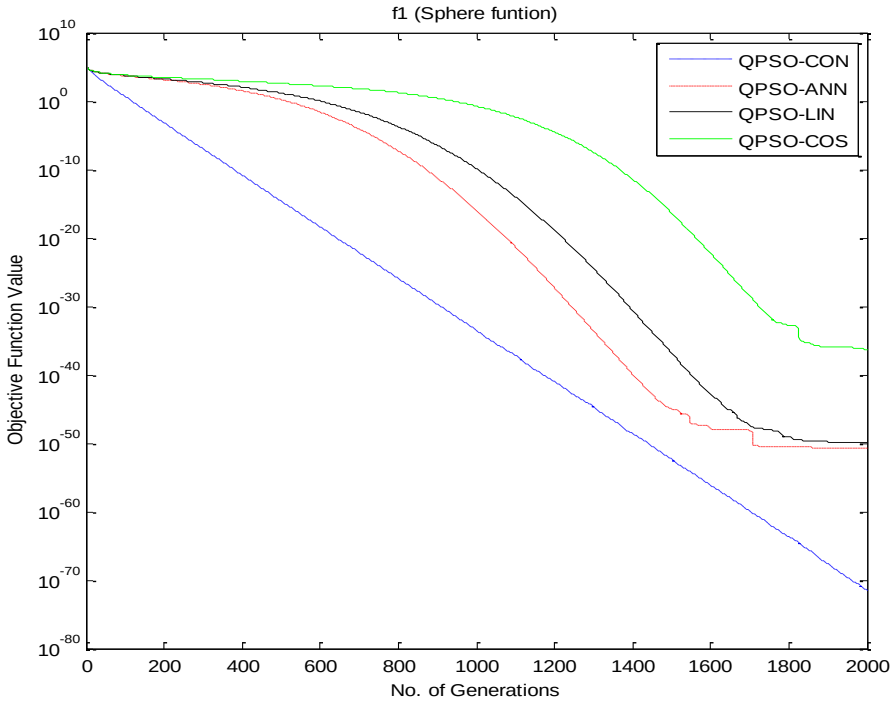
M	D	k_{max}	**MEAN BEST VALUE (STANDARD DEVIATION)**			
			QPSO-CON	QPSO-LIN	QPSO-COS	QPSO-ANN
20	10	1000	41.09 (67.08)	**11.75** (27.37)	14.79 (42.50)	28.60 (72.62)
	20	1500	**25.30** (31.36)	44.70 (44.34)	47.59 (68.83)	71.03 (87.45)
	30	2000	**64.56** (53.22)	72.15 (79.69)	85.34 (86.64)	69.90 (74.20)
40	10	1000	9.59 (16.06)	7.55 (11.07)	6.95 (5.21)	**5.10** (4.54)
	20	1500	**17.06** (26.66)	39.67 (31.26)	37.16 (31.75)	37.99 (31.50)
	30	2000	**28.35** (30.37)	47.56 (37.84)	48.46 (31.30)	53.72 (37.24)
80	10	1000	6.36 (3.91)	4.11 (3.37)	**3.82** (3.92)	4.17 (3.52)
	20	1500	**11.56** (17.16)	27.65 (26.76)	33.02 (30.05)	27.83 (27.54)
	30	2000	**31.02** (33.40)	41.28 (27.90)	34.18 (23.51)	41.36 (28.90)

Table 4.7: Rastrigrin function.

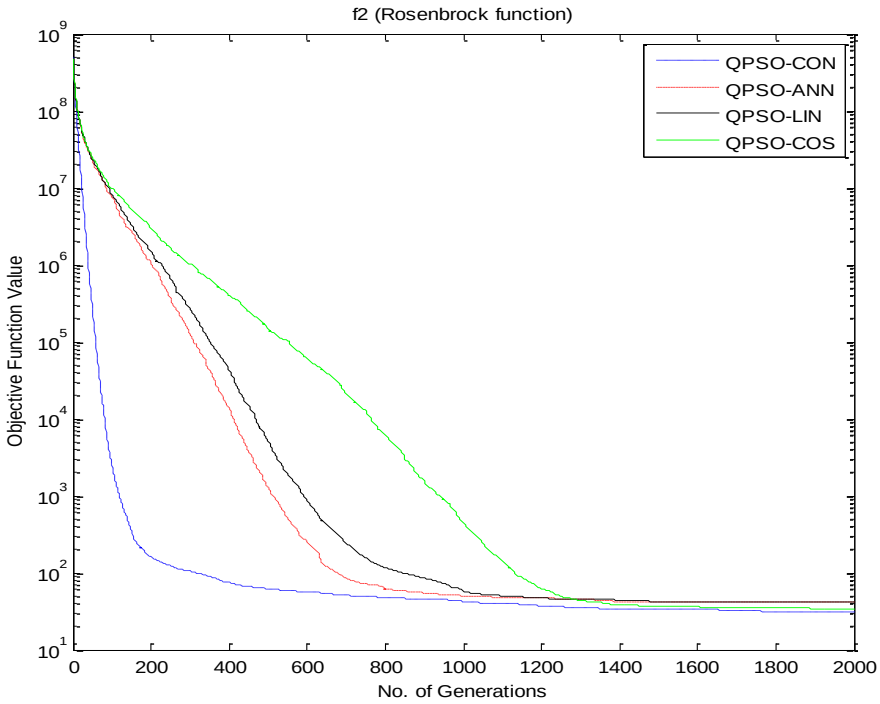
M	D	k_{max}	**MEAN BEST VALUE (STANDARD DEVIATION)**			
			QPSO-CON	QPSO-LIN	QPSO-COS	QPSO-ANN
20	10	1000	23.93 (13.12)	5.74 (3.23)	**4.64** (3.79)	5.20 (2.92)
	20	1500	67.69 (27.27)	19.06 (10.44)	**16.22** (8.10)	19.70 (11.17)
	30	2000	113.38 (34.69)	38.77 (14.36)	**38.52** (21.25)	46.10 (19.48)
40	10	1000	11.81 (9.16)	3.42 (2.00)	**3.29** (2.16)	3.52 (1.42)
	20	1500	33.27 (19.99)	13.03 (7.30)	**12.06** (5.13)	12.32 (4.28)
	30	2000	71.00 (30.14)	23.49 (8.81)	**23.07** (10.85)	23.31 (7.97)
80	10	1000	6.50 (4.90)	2.29 (1.40)	**2.14** (1.32)	2.26 (1.31)
	20	1500	24.35 (13.23)	9.48 (3.21)	**8.60** (6.42)	9.00 (2.98)
	30	2000	43.18 (19.42)	17.44 (5.86)	**16.83** (5.82)	17.85 (6.42)

Table 4.8: Griewank function.

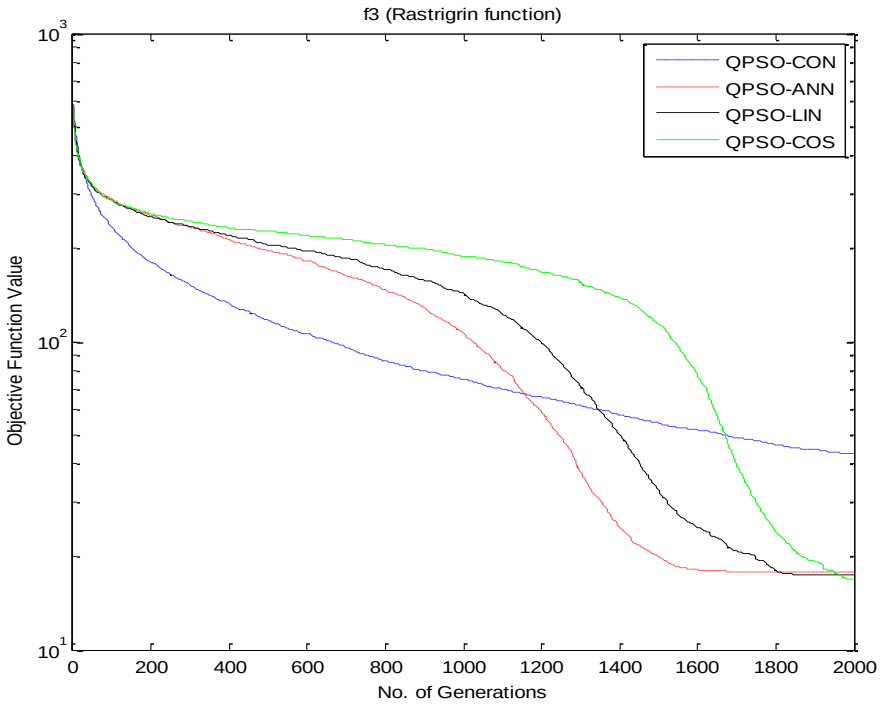
M	D	k_{max}	**MEAN BEST VALUE (STANDARD DEVIATION)**			
			QPSO-CON	QPSO-LIN	QPSO-COS	QPSO-ANN
20	10	1000	6.84E-02 (4.32E-02)	7.17E-02 5.22E-02	9.52E-02 (8.19E-02)	**5.70E-02** (4.25E-02)
	20	1500	2.27E-02 (1.96E-02)	2.09E-02 (2.03E-02)	2.45E-02 (3.10E-02)	**2.02E-02** (1.52E-02)
	30	2000	1.46E-02 (1.89E-02)	1.68E-02 (2.68E-02)	1.13E-02 (1.20E-02)	**9.37E-03** (1.31E-02)
40	10	1000	5.40E-02 (2.96E-02)	6.05E-02 (4.40E-02)	8.22E-02 (8.45E-02)	**4.47E-02** (2.74E-02)
	20	1500	1.78E-02 (1.74E-02)	2.08E-02 (1.91E-02)	1.85E-02 (2.02E-02)	**1.76E-02** (1.46E-02)
	30	2000	1.21E-02 (1.52E-02)	1.24E-02 (1.21E-02)	1.15E-02 (1.34E-02)	**1.06E-02** (1.40E-02)
80	10	1000	5.50E-02 (3.29E-02)	4.07E-02 (3.94E-02)	6.50E-02 (4.93E-02)	**3.78E-02** (2.69E-02)
	20	1500	1.59E-02 (1.59E-02)	1.29E-02 (1.40E-02)	1.61E-02 (1.55E-02)	**1.22E-02** (1.60E-02)
	30	2000	1.05E-02 (1.51E-02)	9.61E-03 (1.18E-02)	8.45E-03 (1.09E-02)	**7.88E-03** (1.07E-02)


f1 (Sphere funtion)
QPSO-CON
QPSO-ANN
QPSO-LIN
QPSO-COS
Objective Function Value
No. of Generations
10^10
10^0
10^-10
10^-20
10^-30
10^-40
10^-50
10^-60
10^-70
10^-80
0
200
400
600
800
1000
1200
1400
1600
1800
2000

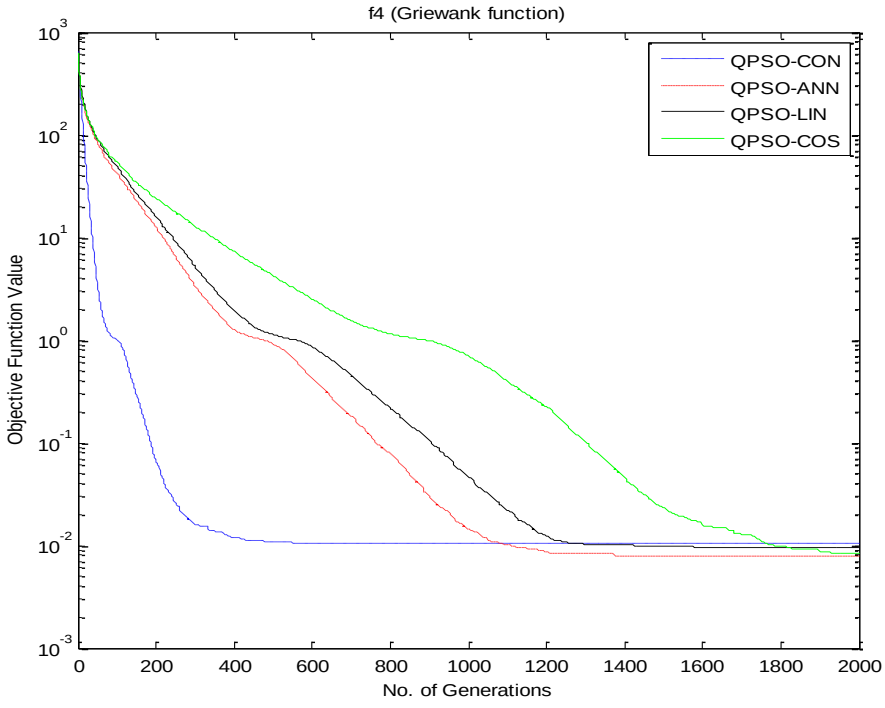
(a)


f2 (Rosenbrock function)
QPSO-CON
QPSO-ANN
QPSO-LIN
QPSO-COS
Objective Function Value
No. of Generations
10^9
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

(b)



(c)



(d)

Figure 4.14: Convergence history of different parameter selection methods on benchmark functions.

The methods proposed to control the parameter in QPSO, have been shown to be successful especially on the complex Rastrigin and Griewank functions. Beyond our expectations, QPSO-ANN performs better on the Griwank function. Since QPSO-COS converges slowly, it maybe need more iterations to achieve the global minimum. Here, we set the number of generation $k_{max} = 5000$ for $D = 30$ with 80 particles to compare the performance of different methods on the Griewank function. The results are shown in Table 4.9 and Figure 4.15, from which one can note that QPSO-COS performs better than other methods although it converges slowly.

Table 4.9: Griewank function with 5000 iterations.

M	D	k_{max}	**MEAN BEST VALUE (STANDARD DEVIATION)**			
			QPSO-CON	QPSO-LIN	QPSO-COS	QPSO-ANN
80	30	5000	6.75E-03 (7.37E-03)	7.28E-03 (1.15E-02)	6.03E-03 (9.37E-03)	8.46E-03 (1.25E-02)

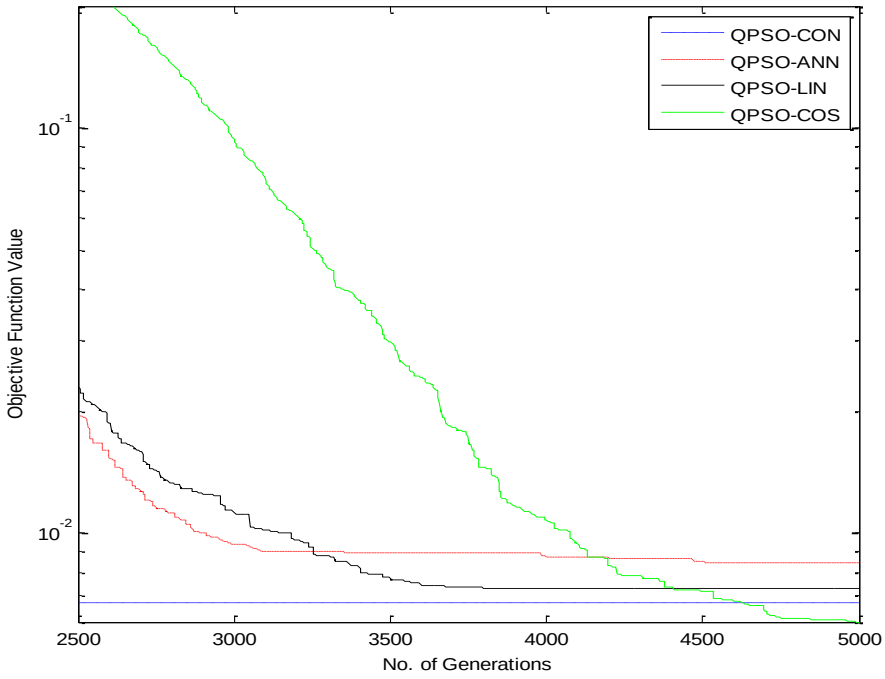


Figure 4.15: Griewank function with 5000 iterations.

4.4.5 Synchronous and asynchronous QPSO

Following the pseudo code for QPSO, it is possible to update the position $X(k+1)$ after all the particles are evaluated. This update is known as the synchronous QPSO. The algorithm itself involves computation of each particle that is independent from the others. The procedure is given in the following pseudo code.

Algorithm of Synchronous QPSO

Initialize: algorithmic parameters, positions of the population $X(0)$, personal best positions $P(0)$;

For each iteration $k = 1, 2, ..., k_{max}$

For each particle $i = 1, 2, ..., M$:

Evaluate the fitness $f(X_i(k))$;

If $f(X_i(k)) < f(P_i(k))$ then

$P_i(k) = X_i(k)$;

If $f(P_i(k)) < f(P_g(k))$ then

$P_g(k) = P_i(k)$;

End for

Compute the mean best position $C(k)$ according to Equation (4.30);

For each particle $i = 1, 2, ..., M$:

Compute $p_i(k)$ according to Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

End for

End for

On the contrary, the sequential asynchronous QPSO updates the positions based on the latest available information of each particle. The algorithm in pseudo code is given below.

Algorithm of Asynchronous QPSO

Initialize: algorithmic parameters, positions of the population $X(0)$, personal best positions $P(0)$;

For each iteration $k = 1, 2, ..., k_{max}$

For each particle $i = 1, 2, ..., M$:

Evaluate the fitness $f(X_i(k))$;

If $f(X_i(k)) < f(P_i(k))$ then

$P_i(k) = X_i(k)$;

If $f(P_i(k)) < f(P_g(k))$ then

$P_g(k) = P_i(k)$;

Compute the mean best position $C(k)$ according to Equation (4.30);

Compute $p_i(k)$ according to Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

End for

End for

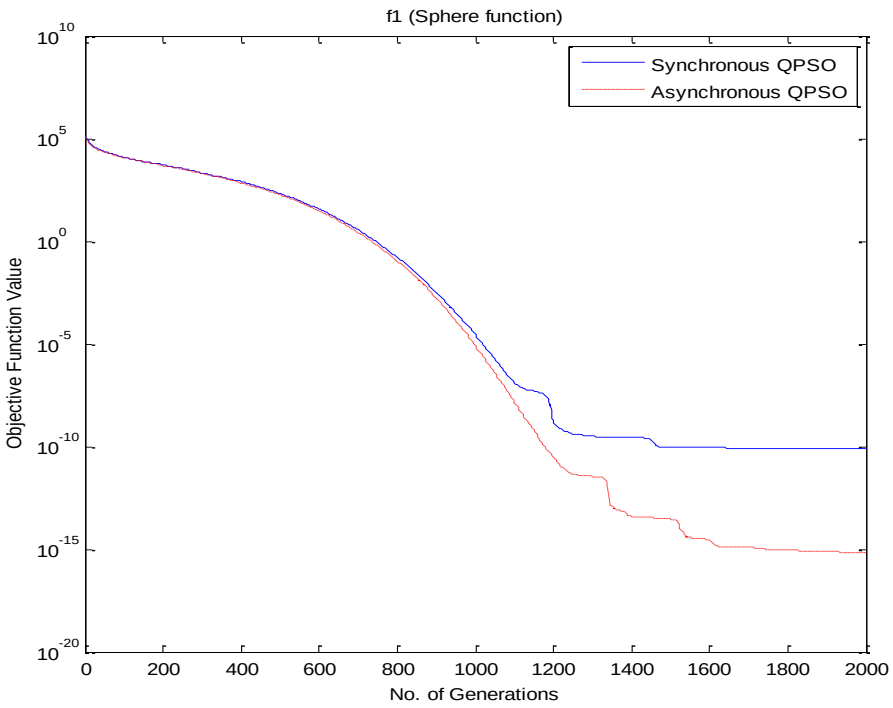
Note the difference between the two methods is similar to the difference in Jacobi (synchronous) and Gauss-Seidel (asynchronous) iterative methods for solving linear systems of equations.

First, in order to test the difference between synchronous and asynchronous methods in search ability of finding the global optimum, 20 particles with 2000 maximum iterations are used on the benchmark functions with dimensional size 30. The expansion-contraction coefficient is set to be decreasing from 1.0 to 0.5. A total of 50 runs were carried out and the mean best values and the deviation of the optima are presented in Table 4.10. The convergence of sequential synchronous and asynchronous QPSO is presented in Figure 4.16. One can note from Table 4.10 and Figure 4.16 that the synchronous QPSO outperforms asynchronous QPSO in complex multimodal functions e.g. Rastrigrin, Griewank. While for the simple unimodal functions (Sphere and Rosenbrock), asynchronous QPSO works better

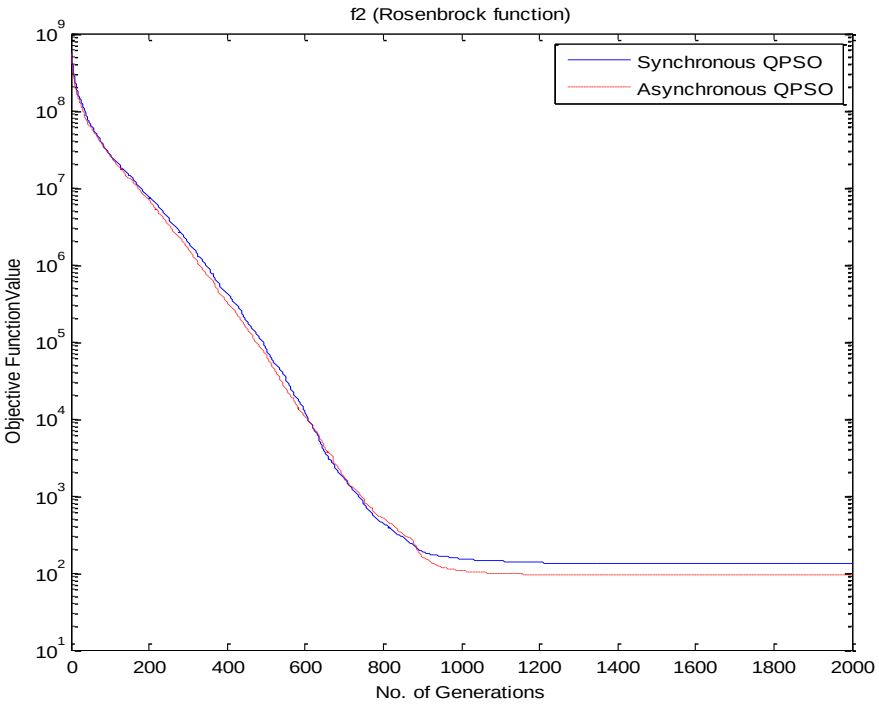
than synchronous QPSO. In asynchronous QPSO, the particles update the positions with the latest global best position P_g, which may lead to fast convergence and quick loss of diversity of the population. Therefore, the asynchronous QPSO easily locates the global minimum in the unimodal problems. However, several local minima exist in complex multimodal problems, lost of diversity will result to prematurity.

Table 4.10: Results of the benchmark functions tests

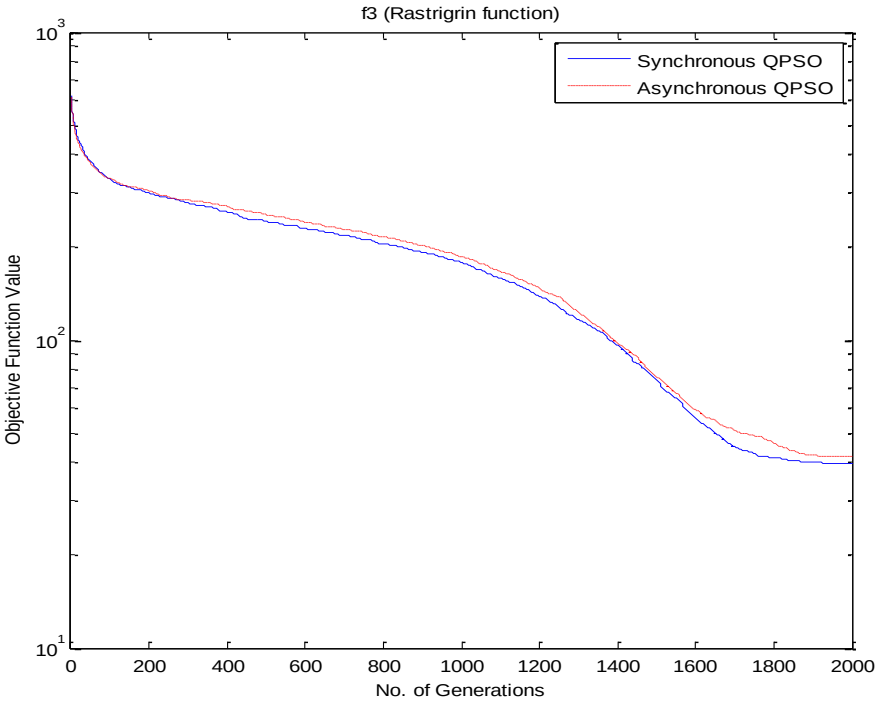
functions	Mean Best Value (Standard Deviation)	
	Synchronous	Asynchronous
Sphere	7.85E-11 (4.92E-10)	**6.84E-16** (2.53E-15)
Rosenbrock	130.23 (136.20)	**92.50** (117.21)
Rastrigrin	**39.68** (15.37)	41.90 (16.93)
Griewank	**6.85E-03** (8.28E-03)	1.17E-02 (1.56E-02)



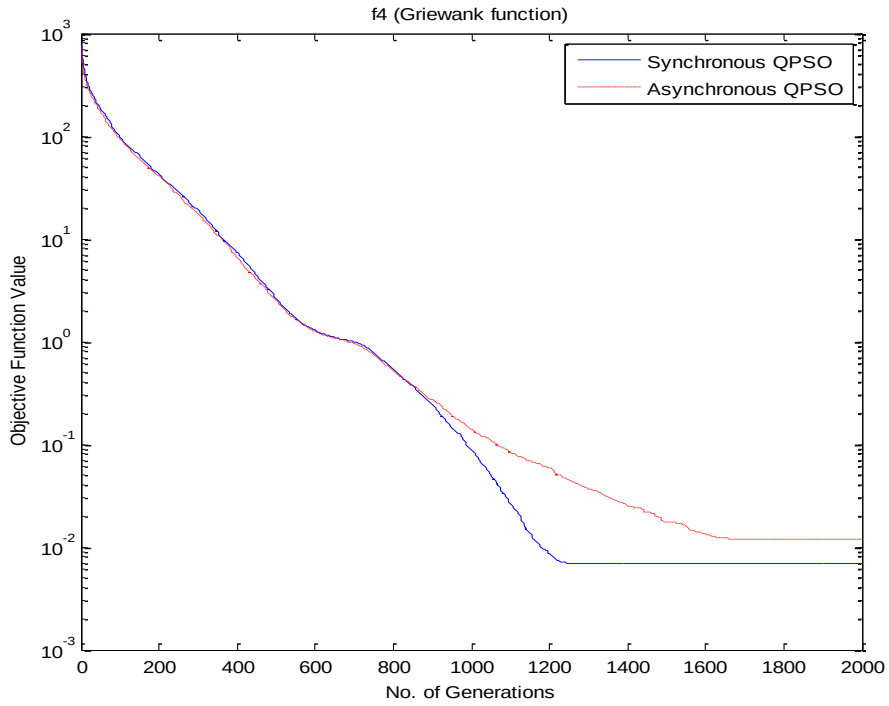
(a)


f2 (Rosenbrock function)
Synchronous QPSO
Asynchronous QPSO
Objective FunctionValue
No. of Generations
0
200
400
600
800
1000
1200
1400
1600
1800
2000

(b)


f3 (Rastrigrin function)
Synchronous QPSO
Asynchronous QPSO
Objective Function Value
No. of Generations
0
200
400
600
800
1000
1200
1400
1600
1800
2000

(c)



(d)

Figure 4.16: Convergence history of synchronous and asynchronous QPSO on benchmark functions.

Next, the efficieny of the two methods are tested, which is important for real-time problems. A new stopping criterion instead of maximum number of iterations is set as a predefined value that the search will stop when the global best fitness is less than this value. 20 particles are used on the benchmark functions with dimensionl size 30. The expansion-contraction coefficient is set as constant 0.75. The average number of iterations for 50 runs needed to achieve the predefined criterion is given in Table 4.11, from which, one can note that the asynchronous QPSO achieved the criterion faster than synchronous QPSO.

Table 4.11: Maximum iterations needed to achieve the predefined criterion.

Functions	Criterion	Iterations	
		Synchronous	Asynchronous
Sphere	1.0E-12	729.06	646
Rosenbrock	100.0	2227.64	1614
Rastrigrin	100.0	1368.14	1338
Griewank	0.1	292.9	264

4.5 Parallel Quantum-Behaved Particle Swarm Optimisation

The high computational cost of complex engineering optimisation problems prevents real-time computation. This motivated the development of parallel optimisation algorithms. QPSO usually involves a large amount of computation with thousands of generations for high dimensional or complicated problems. Therefore, investigation and exploitation of parallel properties of QPSO algorithm is important in order to produce fast solutions. Like other population-based evolutionary algorithms, QPSO algorithm is intrinsically parallel, as global best position gbest is the only shared information among the particles and the fitness evaluation requires only the particle's position. The particles have no communication during the fitness evaluation. Therefore, the parallel QPSO can be efficiently implemented on massively parallel processing architectures (MPP).

MPP is the coordinated processing of a program by multiple processors that work on different parts of the computation in which each processor using its own operating system and memory. Typically, MPP processor communicates using some message passing interface (MPI).

In this thesis, all parallel computational job was executed on the hardware Heracles of the University of Greenwich using FORTRAN with MPI. Heracles has one head node and six computing nodes. The head node is made up of two Dual Core of 2.4GHz AMD Opteron 2216, 8Gb Memory, with Intel compiler 10.1 and Sun Grid Engine 6.2. The computing node is made up of four Quad Cores of 2.2GHz AMD Opteron 8354, 32Gb Memory, which are connected with an Infinitband Memory channel.

The structure of QPSO is very close to being intrinsically parallel as other evolutionary algorithms, since each particle can be considered as an independent agent. A full review about the parallel genetic algorithm, which is classified into eight classes, is given in [139]. But the QPSO is different from the GA that the shared information named gbest exists among all the particles in QPSO. Hence in this work, two classes of parallel QPSO are given

(1) Master-Slave parallelization (distributed fitness evaluation)

   (a) Synchronous

   (b) Asynchronous

(2) Subpopulation parallelization

The details of each of the classes in terms of the parallel models are briefly discussed in the following.

4.5.1 Master slave parallelization

Compared to the computational time in fitness evaluations, the calculation of the particle's positions is just a small fraction of the entire course. Therefore, the most straightforward parallel model is to simply distribute fitness evaluations following a master-slave paradigm [53]. The evaluation of each particle's fitness is easily parallelized since it requires only the position of the particle being evaluated (not the whole population) without communication during the phase. The master processor holds the queue of particles ready to be sent to the slave processors and performs all decision making such as position updates and convergence checks. It does not perform any function evaluations. The slave processors repeatedly evaluate the fitness function using the particle's position received from the master. The MPI can be used to communicate between the master and slave nodes. Theoretically there should be as many slave nodes as the number of particles, however, due to the limitation in the system's availability, more than one particles may be distributed in one processor.

There are two approaches about the parallel quantum-behaved particle swarm optimisation: parallel synchronous QPSO and parallel asynchronous QPSO. In parallel synchronous QPSO, the master stops and waits to receive the fitness values of all the particles before updating the global best position and proceeding to the next iteration. A synchronous approach maintains consistency between sequential and parallel implementations, thereby avoiding alteration of the convergence characteristics of the algorithm. Thus, parallel synchronous QPSO should obtain exactly the same final solution as sequential synchronous QPSO.

The block diagram of the synchronous parallel QPSO is as follows:

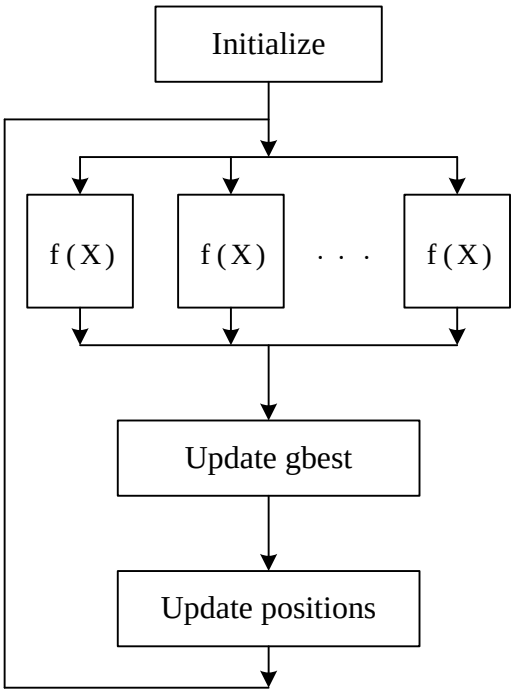


Figure 4.17: Block diagram for parallel synchronous QPSO.

The tasks performed by the master and slave nodes are as follows:

Master processor:

Initialize all parameters, positions of the population $X(0)$, personal best positions $P(0)$;

Do while ($k < k_{max}$)

Send the positions of every particle to the available slave processors;

Receive the evaluated fitness of every particle from slave processors;

For each particle $i = 1, 2, ..., M$

If $f(X_i(k)) < f(P_i(k))$ then

$P_i(k) = X_i(k)$;

If $f(P_i(k)) < f(P_g(k))$ then

$P_g(k) = P_i(k)$;

End for

Compute the mean best position $C(k)$ according to Equation (4.30);

For each particle $i = 1, 2, ..., M$

Compute $p_i(k)$ according to Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

End for

Update α according to Equation (4.28)

End do

Slave processor:

Receive data from master host;

Perform fitness evaluation;

Send the fitness result to master host;

The parallel synchronous master-slave QPSO is easy to implement and a significant speedup can be expected if the communication cost does not dominate the computational cost. It works better when there is no heterogeneity in either the computing environment or evaluation time for the fitness function. However, there is a bottle-neck effect such that the whole process has to wait for the slowest processor to finish the fitness evaluation. While most applications are run on distributed systems, asynchronous approach is therefore usually claimed to be better because it does not introduce idle waiting times, which are likely to occur in uneven environments [56]. In asynchronous QPSO, the global best position is allowed to be updated immediately after evaluation of each particle's fitness. Thus, the optimisation can proceed to the next generation without waiting for the completion of all function evaluations from the current generation. Similar with asynchronous PSO in [56], the block diagram of the parallel asynchronous master-slave QPSO is as follows.

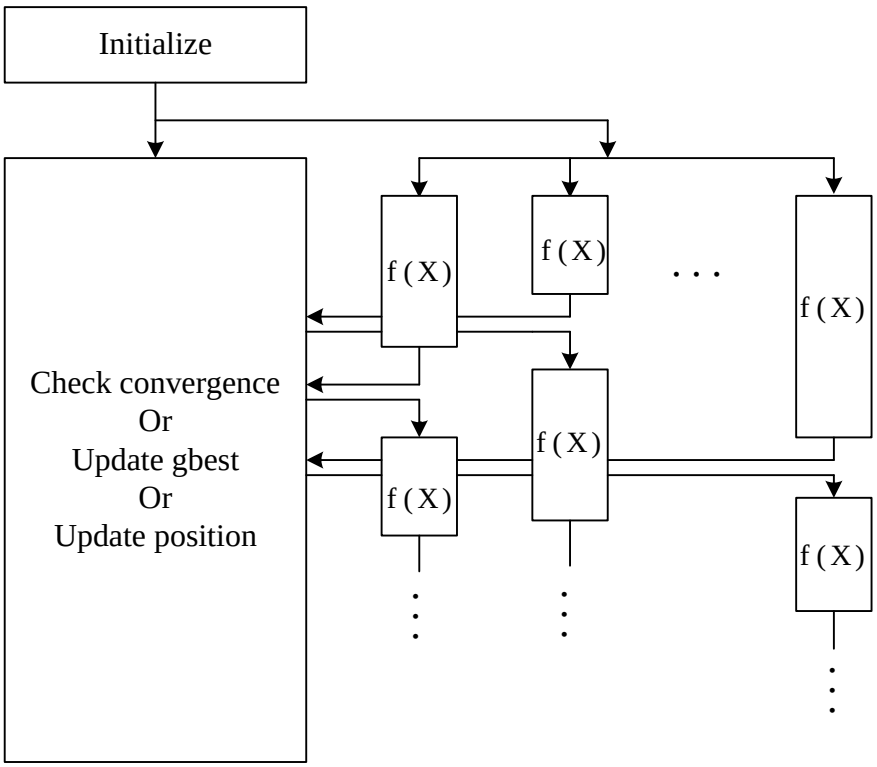


Figure 4.18: Block diagram for parallel asynchronous QPSO.

From Figure 4.18, the parallel asynchronous master-slave QPSO does not work exactly as a traditional QPSO because the predefined maximal number of iteration can't be used as the stopping criterion of the search.

In parallel asynchronous QPSO, a first-in-first-out task queue is used to determine the order in which particles are sent to the slave processors (Figure 4.19). Whenever a slave processor completes a function evaluation, it returns the fitness function value and the corresponding particle number r to the master processor, which places the particle number at the end of the task queue, and updates the global best position. Once a particle reaches the front of the task queue, the master processor updates its position and sends it to the next available slave processor.

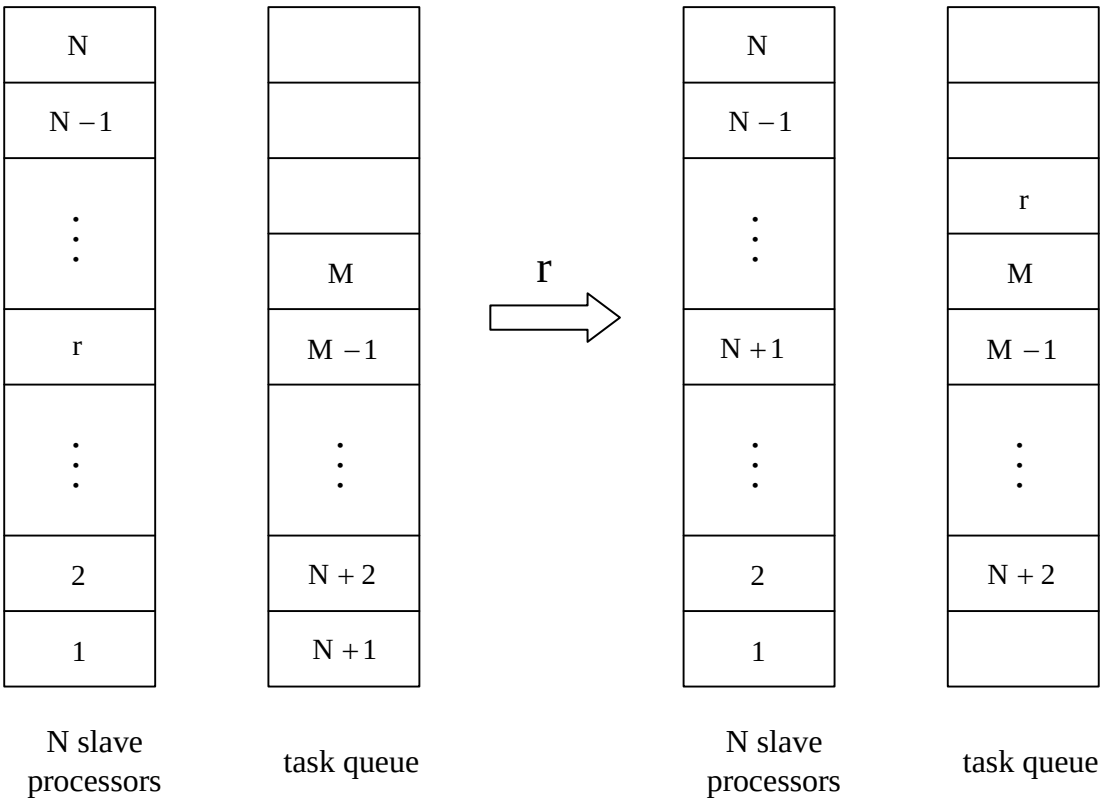


Figure 4.19: Block diagram for first-in-first-out task queue with M particles on a N-processor system.

4.5.2 Static subpopulation parallelization

The idea of static subpopulation has been used in parallel genetic algorithm [139]. This parallelization method requires the division of a population into some number of subpopulations, and every subpopulation executes as an independent QPSO. Subpopulations communicate to share their own global best particles after a predefined number of iterations. . There is a global loop to control the number of communications required for global convergence. Therefore, this algorithm is suitable for distributed memory computers and heterogeneous networks. This model is flexible in that you can decide the number of subpopulations according to your available computers or processors.

However, there are some problems with the scalability of this parallel algorithm. The number of particles in one subpopulation becomes very small when the the number of processors increases. But too small number of particles is not enough to get the global solution searching in high dimensional problem space.

The task performed by each processor node is as the following pseudo code.

Initialize all algorithmic parameters

If (myid==0)

Initialize positions of the population $X(0)$

Send M / N to each processor

End if

10 Do while ($k < k_{max}$)

For each particle $i = 1, 2, ..., M / N$

If $f(X_i(k)) < f(P_i(k))$ then

$P_i(k) = X_i(k)$;

If $f(P_i(k)) < f(P_g(k))$ then

$P_g(k) = P_i(k)$;

End for

Compute the mean best position $C(k)$ according to Equation (4.30);

Compute p according to Equation (4.15);

For each particle $i = 1, 2, ..., M / N$

Update the position $X_i(k+1)$ according to Equation (4.31);

End for

End do

Send P_g and $f(P_g)$ to 0^{th} processor node

If (myid==0)

Receive all the P_g and $f(P_g)$ from other nodes;

Send the best P_g and $f(P_g)$ to all the other nodes;

End if

Go to 10.

4.5.3 Performance metrics

Several metrics are used to quantify the performance, robustness and parallel efficiency of the parallel algorithms. Good parallelization strategies generate a good load balance amongst processors and reduce the time of inter-processors communications. In a perfect

world, a computational job that is split up among N processors would complete in $1/N$ time, leading to an N-fold increase in power. However, any given piece of parallelized work to be done will contain parts of the work that must be done sequentially. This part does not run any faster on a parallel collection of processors (and might even run more slowly). Only the part that can be parallelized runs as much as N-fold faster.

The speedup of a parallel program is defined to be the ratio of the time that it takes a task to run on one processor to the time that it takes on N processors.

Let $t(N)$ be the time required to complete the task on N processors. The speedup $S(N)$ is the ratio

$$S(N) = \frac{t(1)}{t(N)} \tag{4.44}$$

which depends on many things, but primarily depends on the ratio of communication to computation.

Efficiency is a measure of how much of your available processing power is being used. The simplest way to think of is as the speedup per processor. This is equivalent to defining efficiency as the time to run N tasks on N processors to the time to run one task on one processor

$$E = \frac{S(N)}{N} = \frac{t(1)}{Nt(N)}. \tag{4.45}$$

The performance evaluations of the parallel models of QPSO will be illustrated in section 5.7.

4.6 The Hybrid Method

As reported in **sections** 3.2 and 4.5, CGM is in general computationally fast, but it usually converges to a local optimum and depends strongly on the initial approximation used in the iterative process. QPSO requires large amount of computational time with hundreds of thousands generations, since one fitness function evaluation costs a long time for a complex problem. A hybrid method is proposed to integrate QPSO and CGM, which aims to combine the capacity of QPSO in avoiding local minima and the fast convergence of the CGM.

Since the CGM is able to converge to a better solution with properly selected smooth initial guess values, two methods are proposed to deal with the rough estimation obtained by QPSO, which are illustrated by the same example as in section 3.3.

The first method (HM1) is to use a polynomial equation to approximate the unknown heat flux function

$$q(t) = a_0 t^n + a_1 t^{n-1} + ... + a_i t^{n-i} + ... + a_{n-1} t + a_n . \quad (4.46)$$

The problem now becomes a parameter identification problem with unknown parameters $(a_0, a_1, ..., a_n)$, which is represented by a particle $X_i(k) = (X_{i1}(k), X_{i2}(k), ..., X_{iD}(k))$. A smooth function $\tilde{q}(t)$ can be achieved with the approximated parameters obtained by QPSO with a predefined number of generations. Then the approximation $\tilde{q}(t)$ is used as an initial guess q_0 in the CGM.

Procedure of HM1 for inverse problem of estimating heat flux $q(t)$:

A polynomial equation of n order is used to approximate heat flux;

QPSO is used to solve this parameter identification problem $(a_0, a_1, ..., a_n)$;

Result obtained from QPSO is used as initial guess q_0 of CGM;

CGM is used to solve the function estimation problem with initial guess to achieve the optimal estimation of $q(t)$.

In the second method (HM2), QPSO is used directly to estimate the unknown heat flux with no prior information on the functional form $q(t)$. Then an interpolation operation is done to the rough approximation $\tilde{q}(t)$, from which a smooth function q_0 is achieved and will be used as an initial guess for CGM.

Procedure of HM2 for inverse problem of estimating heat flux $q(t)$:

QPSO is used to solve the function estimation problem with large time step Δt to achieve a rough approximation $\tilde{q}(t)$;

An interpolation operation is applied to the rough approximation $\tilde{q}(t)$ to achieve a smooth function q_0

CGM is used to solve the function estimation problem with initial guess q_0 to achieve the optimal estimation of $q(t)$.

Both hybrid methods will be used to solve the inverse heat conduction problem of estimating the time-varying heat flux in section 5.2 and boundary shape in section 5.7.

4.7 Closure

This chapter provides an overview of three well-known stochastic heuristic algorithms, GA, PSO and QPSO. PSO and QPSO are much easier to use and execute than GA, because only simple mathematical operators are used instead of the complicated evolution operators in GA. As with most of the evolutionary algorithms, the loss of diversity in the population is also inevitable in QPSO. In the latter search period, the particles are investigated to cluster together gradually and the swarm is likely to get trapped into a local optimum. Therefore, several improvements have been proposed to apply to the QPSO, e.g. perturbation operator, Gaussian mutation. For the complex multi-modal problems, the QPSO with ring topology model with different parameter selection methods have been proposed. Considering the high computational cost of complex engineering optimisation problems, two parallel QPSO models (e.g. synchronous and asynchronous) have been proposed, and all the parallel programs are executed on a Heracles system. Finally, a hybrid method, which makes use of CGM and QPSO has been proposed, in which, there are two models of providing the initial guess to CGM with values from QPSO.

Chapter 5 NUMERICAL EXAMPLES

This chapter concerns several interesting applications related to heat engineering and ground water contaminant problems. In particular, external effects consist of applying heat fluxes and putting external sources of heat or contaminant to control the result of an engineering system. Material properties such as thermal conductivity and heat transfer coefficients were also included in the study. This chapter also examines the effect of boundary shape on a typical system. Finally, the numerical schemes developed are extended to handle more than one property.

5.1 Preliminary Setup

For the convenience of the numerical tests and illustrations in this chapter, some quantities are defined here. Δt is the temporal step size and Δx is the mesh size. N_t is the total number of time steps and N_x is the number of spatial grid points. t_f is the total computational time of the simulation process.

As described in section 2.2, the least squares method (Equation (2.42)) minimising the difference between measured and computed data is used to solve many inverse problems. These inverse problems are ill-posed because the unavoidable measurement noise and numerical computation errors often lead to an unstable solution. Therefore, a regularisation technique is adopted to stabilise the solution. Here, the Tikhonov regularisation method described in section 2.3 is used, and the objective function becomes

$$J[v] = \sum_{i=1}^{N} \int_{t=0}^{t_f} \left(u(v; x_i, t) - Y(x_i, t)\right)^2 dt + \lambda^2 \|Lv\|^2, \tag{5.1}$$

where N is the number of sensors, λ is the regularisation parameter and v is the unknown quantity involved in inverse problems. The first term on the right hand side of Equation (5.1) is the discrepancy term. The second term on the right hand side is the regularisation term. Lv is the general form of the regularisation operator which can be written as

$$\|Lv\|^2 = \int_{t=0}^{t_f} \left(\frac{d^n v(t)}{dt^n} \right)^2 dt \tag{5.2}$$

when v is time-varying, or

$$\|Lv\|^2 = \int_{x=0}^{L} \left(\frac{d^n v(x)}{dx^n} \right)^2 dx \tag{5.3}$$

when v is spatial varying.

The zeroth ($n = 0$) and first order ($n = 1$) regularisation term are commonly used.

The regularisation stabilises the solution. Minimisation of $J[v]$ in Equation (5.1) is a trade-off between the matching the data and stabilizing the solution. The values chosen for λ affect the stability of the solution. The L-curve method described in section 2.3.3 is used to find the best value of λ, in which, the regularisation term $\|Lv\|^2$ is plotted on a log-log plot against the residual term $\sum_{i=1}^{N} \int_{t=0}^{t_f} \left(u(x_i,t) - Y(x_i,t) \right)^2 dt$ for many values of the regularisation parameter λ. The value of the regularisation parameter at the corner of the L-curve is taken as the optimal parameter value.

To evaluate the accuracy of the algorithms used to solve inverse problems in this chapter, the average error of the estimated unknown quantitiy v_{error} is usually used. For example, if v is time-varying, we use

$$v_{error} = \frac{1}{N_t} \sqrt{\sum_{j=1}^{N_t} \left(v_j - v_j \right)^2}, \tag{5.4}$$

where v_j is the j^{th} component of the estimated unknown quantity, v_j is the j^{th} component of the exact unknown quantity. A similar expression can be used if v is space-varying, namely

$$v_{error} = \frac{1}{N_x}\sqrt{\sum_{i=1}^{N_x}\left(v_i - v_i\right)^2} \tag{5.5}$$

To examine cases involving random measurement errors, normal distribution uncorrelated errors with zero mean and constant standard deviation are assumed. The simulated inexact measurement data $Y(x_i,t)$ may be expressed as

$$Y(x_i,t) = u_{exact}(x_i,t) + \varepsilon\delta_i u_{exact}(x_i,t) \tag{5.6}$$

where $u_{exact}(x_i,t)$ is the solution of the direct problem with exact value $\tilde{v}$, $i = 1,2,...,N$, ε is the noise level, δ_i is a random number satisfying the standard normal distribution.

The numerical tests in this chapter were computed on a PC with Pentium (R) 4 CPU 3.6GHZ and 2.00GB of RAM, running Windows XP. Finally the QPSO algorithm with Tikhonov regularisation is given below as a reference.

Procedure of QPSO with Tikhonov regularisation for the estimation of v

Give an array of N_λ preselected regularisation parameters $\{\lambda_1, \lambda_2, ..., \lambda_{N_\lambda - 1}, \lambda_{N_\lambda}\}$ and two zero arrays with N_λ elements, $Residual$ and $Norm$;

For each regularisation parameter λ_j, ($j = 1,2,...,N_\lambda$)

Step 1. Initialization:

particle positions: $X(0) = \{X_1(0), X_2(0), ..., X_i(0), ..., X_M(0)\}$;

personal best positions: $P(0) = \{P_1(0), P_2(0), ..., P_i(0), ..., P_M(0)\}$;

global best position: P_g;

Contraction-Expansion coefficient $\alpha = 1.0$, $k = 0$, stopping criterion σ;

Step 2. While ($k < k_{max}$) or (σ is not reached)

Compute the mean best position $mbest$ by Equation (4.30);

For each particle $i \in \{1,2,...,M\}$

Compute the attractor $p_i(k)$ by Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

Evaluate the fitness $J[X_i(k+1)]$ according to Equation (5.1) with λ_j;

End for

Decrease α linearly;

$k = k + 1;$

Go to Step 2;

End while

$v = P_g$; $\text{Residual}(j) = \|u - Y\|^2$; $\text{Norm}(j) = \|Lv\|^2$;

End for

Step 3. Plot the corresponding elements of Residual and Norm to obtain the optimal regularisation parameter λ_{opt}. Go to Step 1.

Step 4. Output the optimal estimated v.

5.2 Estimation of Heat Fluxes in Heat Conduction Problems

During the past decades, many applications have been reported for the estimation of heat fluxes and their effects on heat conduction problems, including periodic heating in combustion chambers of internal combustion engines [117], solidification glass [118], indirect calorimetry for laboratory use [119], and transient boiling curve studies [120]. In [1] , Beck et al. used the sequential methods. Recently, the CGM became the most commonly used method in solving this problem [59]-[61]. In this section, a stochastic method known as QPSO, described in chapter 4 is used to solve the IHCP of estimating the time-varying heat flux. In order to remove the dependence on the initial guess of the CGM, the hybrid method proposed in section 4.6 is applied as well.

5.2.1 Mathematical description

One typical engineering problem is the estimation of the thermal history experienced by a shuttle or missile re-entering the Earth's atmosphere from space.The knowledge of the heat flux on the re-entering vehicle surface is vital in the design of such equipment. Figure 5.1 and Figure 5.2 depict a re-entering body and an enlarged section of its skin.

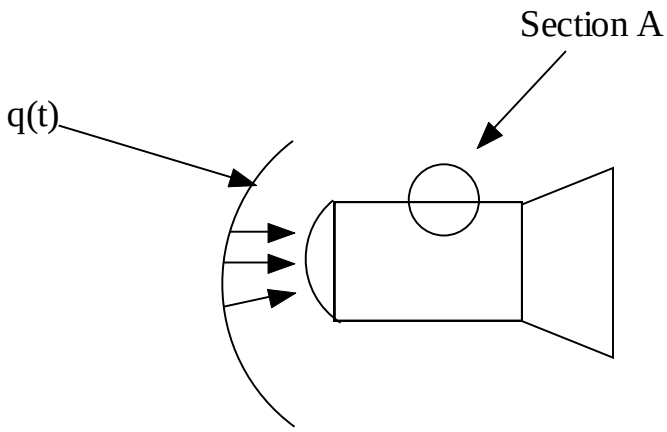


Figure 5.1: Re-entering vehicle schematic.

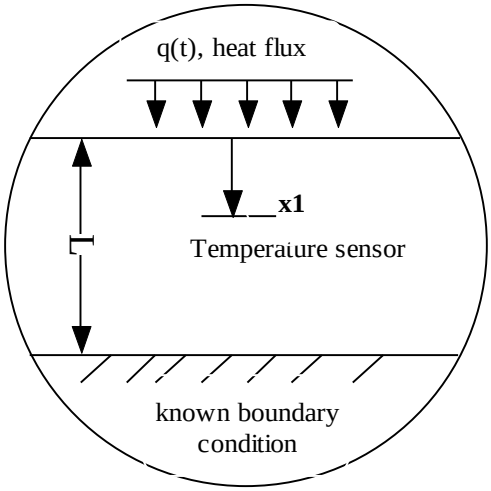


Figure 5.2: Transverse section of the re-entering vehicle.

At $x = 0$ the surface is heated by a time-varying heat flux $q(t)$ and at $x = L$ the surface is insulated. The heat flux may be estimated from measurements obtained from a surface or interior temperature sensor at $x = x_1$. The temperature measurements are usually taken at discrete times t_j, $j = 1, 2, \cdots, N_t$, and are denoted as Y_j. Assuming that the cross-section, as shown in Figure 5.2, is a homogeneous and isotropic slab of inifinite length, the mathematical model governing the heat conduction process may be reduced to a one-dimensional problem as given below

$$\begin{cases} \rho C \dfrac{\partial u(x,t)}{\partial t} = \dfrac{\partial}{\partial x}\left(K \dfrac{\partial u(x,t)}{\partial x} \right), \; 0 < x < L, \; 0 < t \le t_f & \text{(a)} \\ -K \left. \dfrac{\partial u}{\partial x} \right|_{x=0} = q(t), \quad 0 < t \le t_f & \text{(b)} \\ K \left. \dfrac{\partial u}{\partial x} \right|_{x=L} = 0, \quad 0 < t \le t_f & \text{(c)} \\ u(x,0) = u_0(x), \quad 0 \le x \le L & \text{(d)} \end{cases} \tag{5.7}$$

where $u(x,t)$ is is the temperature distribution at a spatial location x and time t, ρ, C, K are density, heat capacity and thermal conductivity, respectively. Equations (5.11b) and (5.11c) are two Neumann boundary conditions and Equation (5.11d) is the initial condition. For simplicity, the physical properties may be taken as $K = \rho C = L = 1$ which are the same as using non-dimensional data. Here, $q(t)$ is the unknown heat flux to be determined.

If the temperature at the heated surface is known, there exists an exact solution for the heat flux. Unfortunately, the physical situation at the surface may be unsuitable for attaching a sensor, or the accuracy of a surface measurement may be seriously impaired by the presence of the sensor. Although it is often difficult to measure the temperature history of the heated surface of such a re-entering body, the temperature history at an interior location or at the insulated surface of the body may be measured easily.

The solution process of the inverse problem involves solutions of the direct problem for every approximated heat flux. For simplicity and accuracy, the Crank-Nicholson implicit finite difference method is adopted and is given below

$$\rho C \frac{u_i^{j+1} - u_i^j}{\Delta t} = K \left(\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{2\Delta x^2} + \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{2\Delta x^2} \right), \tag{5.8}$$

where u_i^j is the temperature at the j^{th} time step, $j = 1, 2, ..., N_t$, along the i^{th} grid point, $i = 1, 2, ..., N_x$. A second order discretisation is used for the boundary condition Equation (5.11b) leading to

$$-K \frac{-3u_0^j + 4u_1^j - u_2^j}{2\Delta x} = q_j, \tag{5.9}$$

where $q_j = q(j\Delta t)$ is the discrete representation of the heat flux at time $t = j\Delta t$.

5.2.2 Numerical tests

In this section, the QPSO method is used to solve the IHCP of estimating the heat flux. The continuous function of the heat flux q(t) is discretised for numerical computation and simulated by a particle. The position of a particle represents a candidate solution of the heat flux. The dimension D of the position is equal to the number of time steps N_t. The fitness objective function $J[q]$ is defined as in Equation (5.1). At each generation k of finding the minimum of the objective functional $J[q]$, the particle i representing a feasible solution, is defined as

$$X_i(k) = (X_{i1}(k), X_{i2}(k), ..., X_{ij}(k), ..., X_{iD}(k)) = (q_1, q_2, ..., q_j, ..., q_{N_t}) \tag{5.10}$$

where $D = N_t$ represents the dimension of the particle's position and number of time steps required in the computation. Substituting $X_i(k)$ into Equation (5.11), the temperature u can be computed by solving the direct problem. Each feasible solution $X_i(k)$ is evaluated by computing the fitness function $J[q]$. At each generation, the positions of the particles are updated as according to Equation (4.31). This process is repeated until a pre-defined number of generations have reached or the solution converged. q_{error}, as defined in Equation (5.8), is used to evaluate the accuracy of the converged solution.

The numerical test in this section was to determine the heat flux defined by the triangle function $q(t) = \begin{cases} 0 & t \le 0 \\ t & 0 < t \le 0.6 \\ 1.2 - t & 0.6 < t \le 1.2 \\ 0, & 1.2 < t \le 1.56 \end{cases}$ as discussed in [1].

The parameters in the QPSO were set respectively as $M = 20$, $k_{max} = 2000$, $\alpha_0 = 1.0$, $\alpha_1 = 0.5$. Table 5.1 shows the results of average errors for the test cases with different mesh sizes and temporal step sizes. Note that unlike the effect on the solution of a direct problem, smaller mesh size does not necessarily produce better solution for an inverse problem. That is because of the ill-posedness of inverse problems, too small mesh size or temporal step size will lead to oscillations in the solution. The results produced with $\Delta x = 0.05$ and $\Delta x = 0.1$ are better than or have comparable accuracy with that obtained with $\Delta x = 0.01$ and $\Delta x = 0.02$. While for too large mesh sizes, i.e. $\Delta x = 0.2$ and $\Delta x = 0.5$, the accuracy of the results obviously deteriorates. Therefore, considering from both aspects of accuracy and

computational time, $\Delta x = 0.1$ is chosen. Furthermore, it can be seen from the data in the fifth row of Table 5.1, the smaller temporal step size, i.e. Δt =0.01, 0.02, 0.03, does not produce a better solution than that obtained with $\Delta t = 0.04$ or even $\Delta t = 0.06$ and $\Delta t = 0.12$. The reason may be due to the number of particles used in QPSO, i.e. $M = 20$, which is not sufficient to optimise a problem with dimensional size 53, 79, and 157 ($D = t_f / \Delta t$). Table 5.2 gives the results of average errors and computational time when the number of particles M increases to 30, 40 and 50 with different temporal step sizes except $\Delta t = 0.12$. For $\Delta t = 0.04$ and $\Delta t = 0.06$, increasing the number of particles does not improve the accuracy of the results because 20 particles are sufficient for dimensional size $D = 27$ and $D = 40$. While for $\Delta t = 0.01$, $\Delta t = 0.02$ and $\Delta t = 0.03$, the accuracy of the results improves when the number of particles increases. But the required computational time increases a lot as well. Considering both accuracy of the solution and the computational time, $\Delta t = 0.03$ is adopted in the subsequent tests and the number of particles M is set as 30 correspondingly.

Table 5.1: Effect of mesh size and temporal step size on the average error q_{error}.

Δx \ Δt	0.01	0.02	0.03	0.04	0.06	0.12
0.01	7.3E-03	5.3E-03	1.5E-03	9.37E-04	1.4E-03	3.5E-03
0.02	6.3E-03	3.3E-03	1.6E-03	8.20E-04	1.4E-03	3.5E-03
0.05	8.4E-03	3.1E-03	1.2E-03	7.78E-04	1.4E-03	3.5E-03
0.1	**8.3E-03**	**3.7E-03**	**1.1E-03**	**7.83E-04**	**1.3E-03**	**3.5E-03**
0.2	1.2E-02	3.0E-02	1.7E-03	1.1E-03	1.6E-03	4.0E-03
0.5	7.5E-03	7.3E-03	8.5E-03	9.6E-03	1.2E-02	1.5E-02

Table 5.2: Effect of number of particles on the average error and computational time.

M \ Δt	0.01	0.02	0.03	0.04	0.06
20	8.3E-03 (183.95)	3.7E-03 (102.14)	1.1E-03 (62.89)	**7.83E-04** **(47)**	1.3E-03 (34.44)
30	5.4E-03 (263.53)	1.8E-03 (132.55)	**6.49E-04** **(93.34)**	**8.08E-04** **(68.47)**	1.3E-03 (46.78)
40	4.6E-03 (528.84)	**8.06E-04** **(270.77)**	**5.47E-04** **(183)**	**7.84E-04** **(137.16)**	1.3E-03 (94.23)
50	3.8E-03 (865.69)	**6.31E-04** **(449.44)**	**5.42E-04** **(300.27)**	**7.84E-04** **(232.69)**	1.3E-03 (157.34)

Figure 5.3 shows the results obtained by QPSO with the sensor located at $x = 0.5$. The temperature measurements in this case contained no noise. Note that the agreement with the exact heat flux is very good. The only time that heat flux function has large discrepancy is that where abrupt changes appear. Figure 5.4 shows the results obtained for the same test case, but with the sensor located at $x = 1.0$. Note that the closer the sensor is to the heated boundary, the better the estimated result is. Figure 5.5 shows the results of the estimated heat flux by using QPSO with the sensor located at $x = 0.5$, and with noise level $\varepsilon = 0.01$ in the temperature measurement. The estimated heat flux converges to the exact heat flux with minor deviation. The average error of the estimated heat fluxes and the objective function values are listed in Table 5.3.

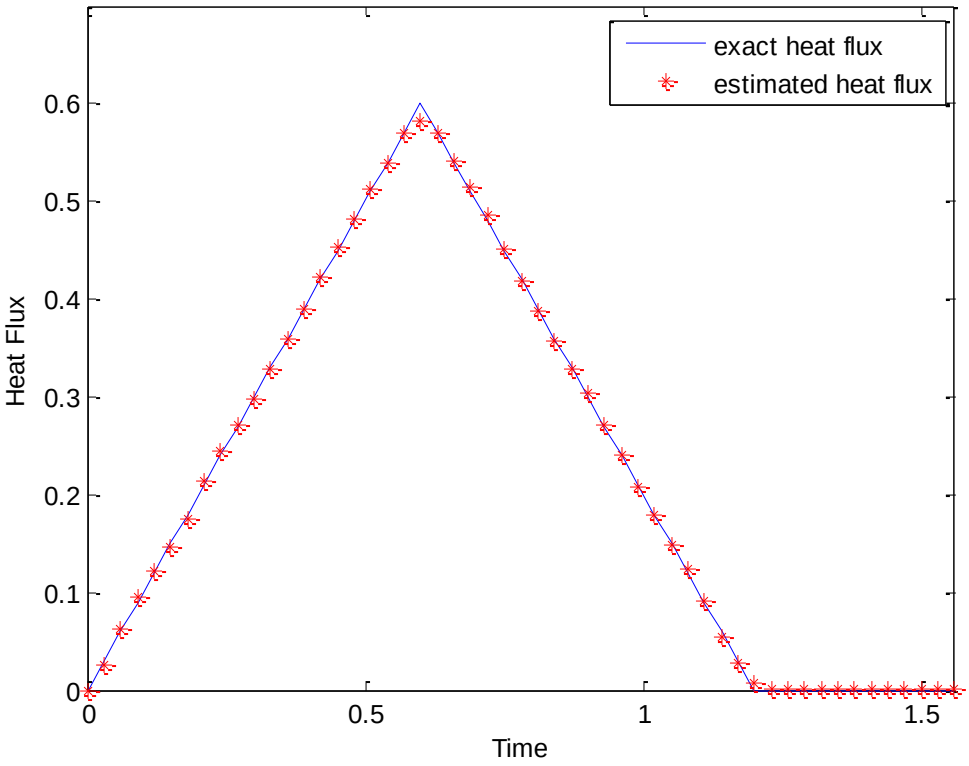


Figure 5.3: Estimated heat flux by using QPSO with sensor at $x = 0.5$ and exact measurements.

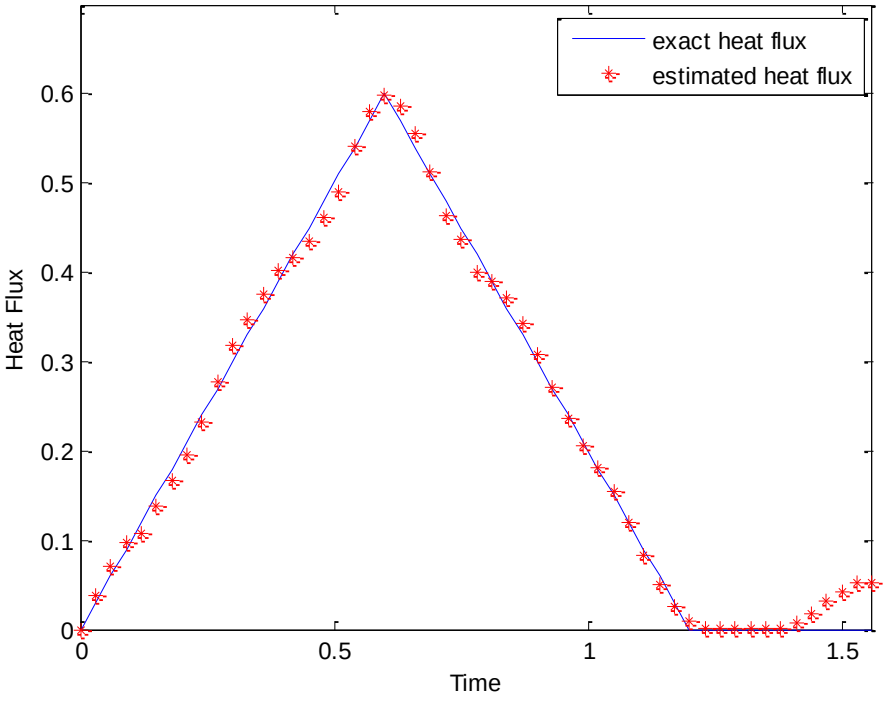


Figure 5.4: Estimated heat flux by using QPSO with sensor at $x = 1.0$ and exact measurements.

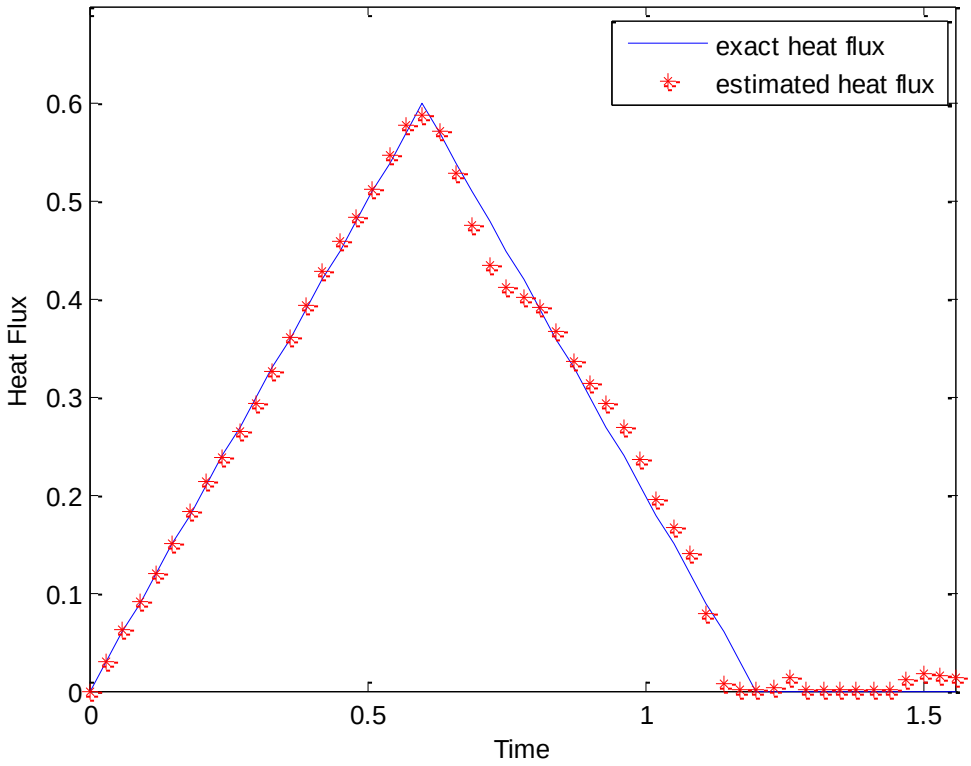


Figure 5.5: Estimated heat flux by using QPSO with sensor at $x = 0.5$ and noisy measurements with $\varepsilon = 0.01$.

Table 5.3: Tests of sensor locations and noise levels.

Sensor Location	Noise Level ε	q_{error}	$J[q]$
0.5	0.0	**6.49E-04**	3.53E-05
0.5	0.01	2.2E-03	2.99E-04
1.0	0.0	3.0E-03	4.21E-05
1.0	0.01	8.4E-03	2.54E-04

Figure 5.6 and Table 5.4 show the results obtained by using CGM with different initial guesses q_0, and with sensor located at $x = 0.5$. The temperature measurements in this case contained no noise. Note that different initial guesses may lead to different performance, in particular for the values near the final time t_f. Because the gradient $\nabla J[q]$ is always equal to zero at $t = t_f$, it is difficult to estimate the values near t_f unless an accurate initial guess is given. It can be also noted that the estimated heat flux with q_0 randomly chosen in [0, 0.6]

oscillates up and down from the exact value. However, the computational time required in CGM is only 0.8906 seconds. This suggests that CGM converges fast, and strongly depends on the choice of initial approximation. On the contrary, QPSO can obtain relatively good results with random initial populations while requiring much more computational time.

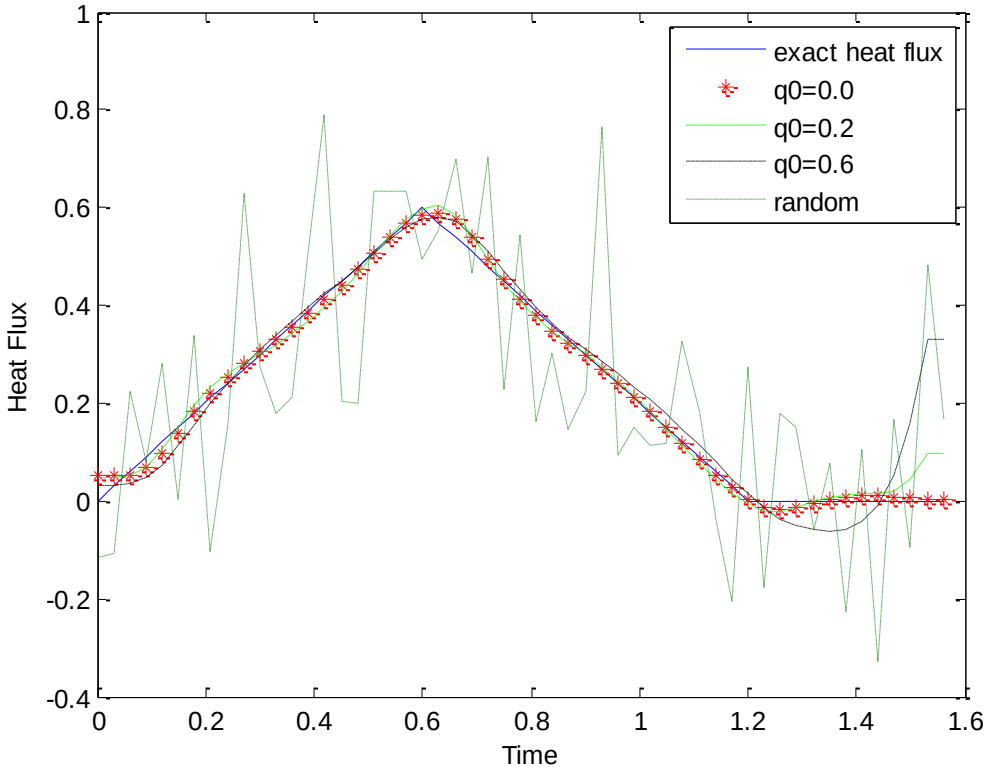


Figure 5.6: Estimated heat fluxes with different initial guesses by using CGM with sensor at $x = 0.5$ and exact measurements.

Table 5.4: Effects of initial guesses on CGM.

Initial guess q_0	q_{error}	$J[q]$
0.0	1.9E-03	3.0E-05
0.2	4.1E-03	1.0E-05
0.6	6.4E-03	5.0E-04
random	2.87E-02	8.00E-04

In order to overcome disadvantages of both QPSO and CGM, the hybrid method proposed in section 4.6 was used to estimate the unknown heat flux. In essence any function could be approximated by a polynomial function such as Equation (4.46). In such case the

inverse problem reduces from function estimation to parameter identification. The QPSO was used to solve the parameter identification problem with a small number of particles and generations, which can reduce the computational time. A rough estimation, but nevertheless a smooth curve of the heat flux is obtained with the estimated parameters $(a_0, a_1, ..., a_n)$. Then the value of the heat flux instead of the parameters may be used in CGM as its initial approximation and the solution iterate until the predefined tolerance is reached. In other words, the QPSO is first used to solve the parameter identification problem followed by CGM being used to obtain the results of the function estimation problem.

Figure 5.7 shows the results obtained by using the hybrid method (HM1). Note that good results are obtained by CGM with initial approximation with sufficient smoothness from QPSO. In this problem, $n = 3$ is used in the polynomial approximation.

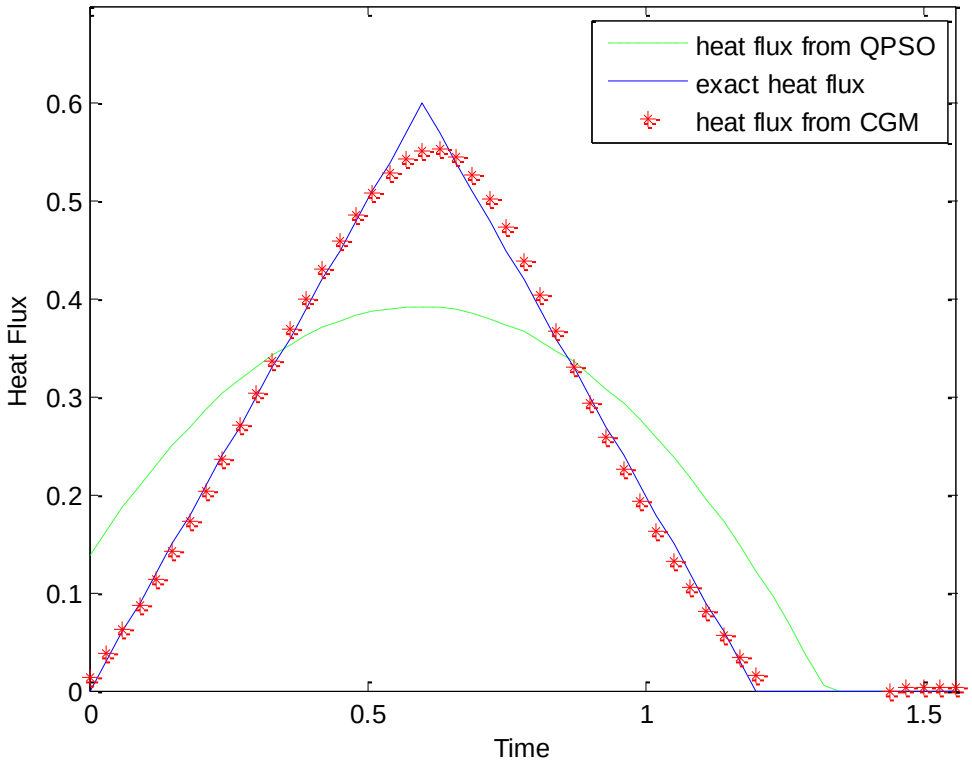


Figure 5.7: Estimated heat flux by using HM1 with exact temperature measurements.

The problem leading to the result in Figure 5.7 was tested again assuming standard normal distributed error with noise level $\varepsilon = 0.01$ (Figure 5.8) in the temperature measurements. Note that accurate estimation is obtained. The average error of the estimated heat fluxes is shown in Table 5.5. The computational time used in HM1 is 9.28 seconds. It is

suggested that the hybrid method is stable and efficient to solve the unknown heat flux function estimation problem. The estimated result by QPSO is only slightly better than that obtained by HM1. However, the saving in computational time achieved by using HM1 is tremendous due to the fact that smaller dimension is used in the positions of the particles, i.e. $D = 3$.

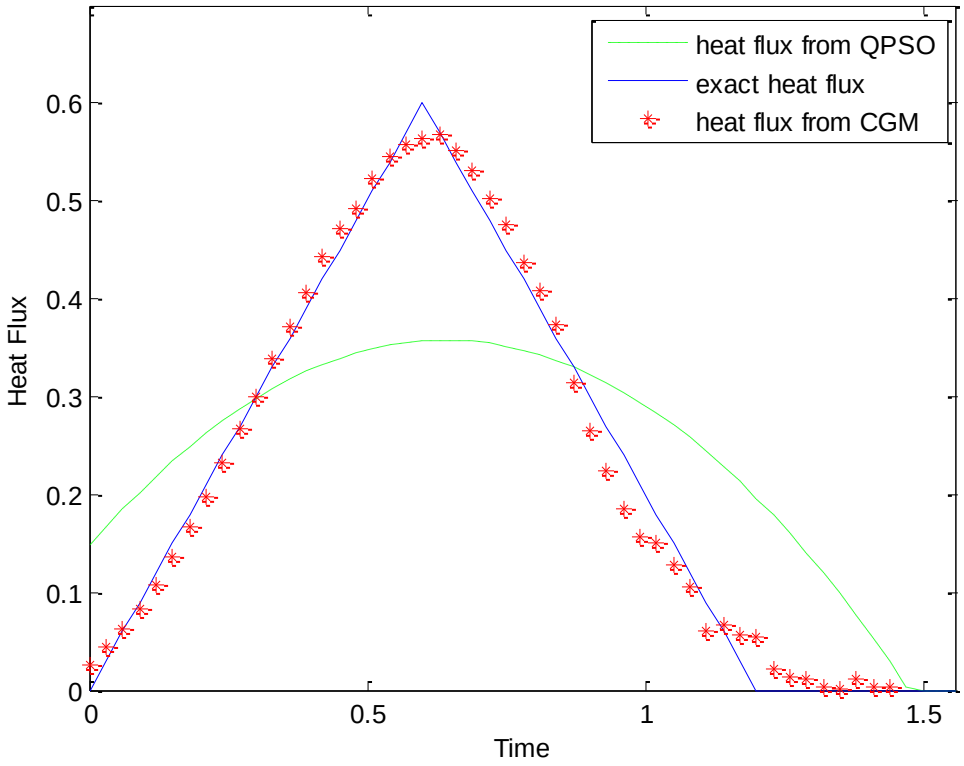


Figure 5.8: Estimated heat flux by using improved hybrid algorithm, measurements with noise level $\varepsilon = 0.01$.

Table 5.5: Numerical results of the hybrid method (HM1).

Noise Level ε	q_{error}	J[q]
0.0	1.4E-03	1.98E-05
0.01	2.8E-03	2.45E-04

5.2.3 Conclusion

In this section, the QPSO was used to estimate unknown heat flux functions in heat conduction problems. Numerical experiments demonstrated the viability, efficiency and stability of the QPSO for the solutions of IHCPs. The stochastic algorithm avoids complicated gradient computation in a gradient-based method but guarantees the global optimum. In order to overcome the high computational costs of the QPSO and the strong dependence on initial approximations of the CGM, the hybrid method proposed in section 4.6 combining advantages of the QPSO and the CGM is used. In order to avoid oscillation of the results, a polynomial function is used to approximate the unknown heat flux function, and the smoother curve obtained by the QPSO is used as an initial guess for use in the CGM. The results indicate the efficiency and stability of the hybrid algorithm in solving IHCPs.

5.3 Estimation of Heat Sources in Heat Conduction Problems

The inverse problem of determining an unknown heat source has received a lot of attention recently with several important applications in engineering, including the design of thermal equipments, systems and instruments, etc. [63], [77], [90]-[94], [121]-[127]. This problem also finds important applications in practice, e.g. in finding pollution source intensity and designing the final state in melting and freezing processes. The inverse source problem is concerned with the determination of the heat source term, from the knowledge of directly measureable temperature.

A variety of numerical and analytical methods for solving the inverse source problems have been proposed in the literature. The least-squares method with the addition of regularisation term was proposed by Beck et al. [1]. The CGMs have been widely used in inverse heat conduction and convection problems [63]. A stochastic search method known as genetic algorithm (GA) was used in [94] to estimate the plan heat source.

In this part of the thesis, another two stochastic methods PSO and QPSO, described in chapter 4, are used to solve this inverse problem of estimating the time-varying heat source. And the results are compared with those obtained by using GA and CGM.

5.3.1 Mathematical description

Consider a one-dimensional rod of length L with constant thermal properties. A plane surface heat source of strength G(t) located at a specified position $x = x_s$ generates heat. Assume boundaries at both ends of the rod are insulated. The longitudinal side of the rod is shown in Figure 5.9.

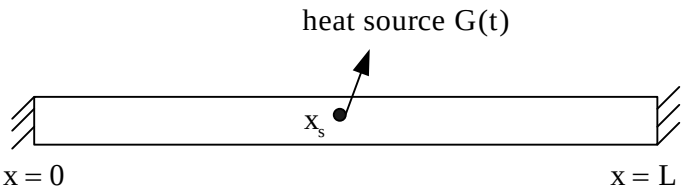


Figure 5.9: A one-dimensional rod with a heat source.

The mathematical formulation of this problem is defined as

$$\begin{cases} \rho C \dfrac{\partial u(x,t)}{\partial t} = \dfrac{\partial}{\partial x}\left(K \dfrac{\partial u(x,t)}{\partial x} \right) + G(t)\delta(x - x_s),\ 0 < x < L,\ 0 < t \le t_f & \text{(a)} \\ K \left.\dfrac{\partial u}{\partial x}\right|_{x=0} = 0,\ 0 < t \le t_f & \text{(b)} \\ K \left.\dfrac{\partial u}{\partial x}\right|_{x=L} = 0,\ 0 < t \le t_f & \text{(c)} \\ u\big|_{t=0} = u_0,\ 0 \le x \le L & \text{(d)} \end{cases} \tag{5.11}$$

where $\delta(\cdot)$ is the Dirac delta function, Equations (5.15b) and (5.15c) are two insulating Neumann boundary conditions, and Equation (5.11d) is the initial condition. For simplicity, the physical properties are taken as $K = \rho C = L = 1$ which are the same as using non-dimensional data. Here, G(t) is the unknown heat source to be determined. The determination of G(t) requires additional data such as temperature measurements $Y(x_i, t)$ $(i = 1, 2)$ obtained from the sensors located at both boundaries of the rod ($x_1 = 0.0$, $x_2 = 1.0$).

In order to solve the direct problem, the Crank-Nicolson implicit finite difference method is used to discretise Equation (5.15)

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{2(\Delta x)^2} + \frac{u_{i+1}^{j} - 2u_i^{j} + u_{i-1}^{j}}{2(\Delta x)^2} + G_{j+1}\delta(x_i - x_s), \tag{5.12}$$

where u_i^j is the temperature at the j^{th}, $j = 1, 2, ..., N_t$, time step and i^{th} mesh point, $i = 1, 2, ..., N_x$. G_j is the discretised approximation of $G(j\Delta t)$, $j = 1, 2, ..., N_t$.

5.3.2 Numerical tests

In this section, the QPSO is applied to solve inverse problem of estimating the time-varying heat source. The position of a particle represents a candidate solution of heat source. In a discretised computation, the dimension D of the position is equal to the number of time steps N_t involved in the simulation. The objective fitness function $J[G]$ is defined as in Equation (5.1). At each generation k of finding the minimum of $J[G]$, a particle is defined as

$$X_i(k) = (X_{i1}(k), X_{i2}(k), ..., X_{ij}(k), ..., X_{iD}(k)) = (G(t_1), G(t_2), ..., G(t_j), ..., G(t_{N_t})) \qquad (5.13)$$

where $D = N_t$ represents the dimension of the particle's position and number of time steps required in the computation. Substituting $X_i(k)$ into Equation (5.21), the temperature u can be computed by solving the direct problem. Each feasible solution $X_i(k)$ is evaluated by computing the objective fitness function $J[G]$. At each generation, the positions of the particles are updated as according to Equation (4.31). This process is repeated until a pre-defined number of generations have reached or the solution is converged. G_{error}, as defined in Equation (5.8) is used to evaluate the accuracy of the converged solution.

In order to verify the viability and efficiency of the QPSO and compare the performance of PSO, QPSO, GA and CGM for solving the inverse source problem of estimating the time-varying strength, the typical example

$$G(t) = \begin{cases} 1.8t, & 0 \le t < 0.5 \\ 1.9 - 2t, & 0.5 \le t < 0.8 \\ 0.3, & 0.8 \le t \le t_f \end{cases} \qquad (5.14)$$

described in [94] is examined here. The heat source is located at $x_s = 0.5$ and the total time of simulation is $t_f = 1.0$.

In the numerical tests, the temporal step size was chosen as $\Delta t = 0.02$ and the mesh size was set as $\Delta x = 0.05$. The parameters in PSO are set as $M = 20$, $D = N_t = t_f / \Delta t = 50$,

$V_{max} = 1.0$, $c_1 = c_2 = 2.0$, $\omega_0 = 0.9$, $\omega_1 = 0.4$, $k_{max} = 2000$. The crossover probability and mutation probability in GA are chosen as $P_c = 0.6$, $P_m = 0.05$. In order to compare the results under the same conditions, the same values were used in those common parameters in QPSO, PSO and GA. Due to the strong dependence of the CGM on the initial guess, the initial value of the unknown function needs to be chosen carefully. Here G(t) was initially set to zero ($0 < t \le t_f$). While in the stochastic methods, such as GA, PSO and QPSO, all the particles or chromosomes were initialized randomly instead of given a specific initial value.

All the test examples are listed in Table 5.6, each with different noise levels were used to examine the stability of the numerical methods. Three different noise levels were used for each method.

Table 5.6: Test scenarios for the comparison of the methods and noise levels.

Run Number	Method	Noise Level ε
Run 1	QPSO	0.0
Run 2	QPSO	0.03
Run 3	QPSO	0.05
Run 4	PSO	0.0
Run 5	PSO	0.03
Run 6	PSO	0.5
Run 7	GA	0.0
Run 8	GA	0.03
Run 9	GA	0.05
Run 10	CGM	0.0
Run 11	CGM	0.03
Run 12	CGM	0.05

Figure 5.10 shows the comparison of the estimated results obtained by different methods using measurements without noise. Note that the estimated heat source obtained by the QPSO is almost identical to the exact heat source, and is much better than those obtained by GA and PSO. Because the gradient of the objective function J[G(t)] at $t = t_f$ is always equal to zero in CGM [63], the estimated heat source near the final time is always difficult to

find, unless the initial guess at the final time is equal to the exact value. The values of average error G_{error} and the objective function $J[G]$ are listed in Table 5.7.

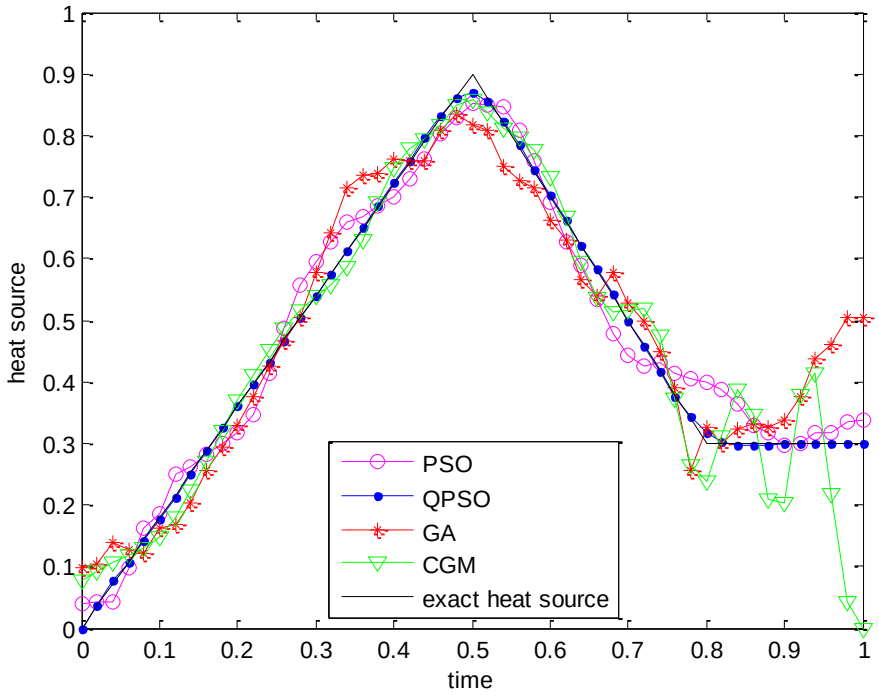


Figure 5.10: The estimated results by using different methods with exact measurements.

Table 5.7: Analysis of different methods with exact measurements.

Run Number	G_{error}	J[G]	Computational Time (s)
Run 1	**7.48E-04**	**1.12E-07**	5.13
Run 4	5.30E-03	9.64E-07	4.33
Run 7	1.67E-02	3.37E-06	4.40
Run 10	9.85E-03	1.14E-07	0.62

Figure 5.11 and Figure 5.12 illustrate the inverse solution of G(t) obtained by the same methods used in Figure 5.10, using the temperature measurements with noise level $\varepsilon = 0.03$ and $\varepsilon = 0.05$ respectively. Note that the estimated heat source by QPSO is in relatively good agreement with the exact heat source, and is also much better than those obtained by GA and PSO. Special attention should be paid to the result obtained by the CGM, large oscillations are exhibited in the estimated heat source with noisy measurements. It looks strange that the

estimated heat source with $\varepsilon = 0.03$ is even worse than that with $\varepsilon = 0.05$. The reason may be because the stopping criterion is set as the number of maximum iterations. The convergence history of CGM can be seen from Figure 5.13. The accuracy begins to deteriorate after several iterations. Correspondingly, the average error and the objective function value are listed in Table 5.8 and Table 5.9.

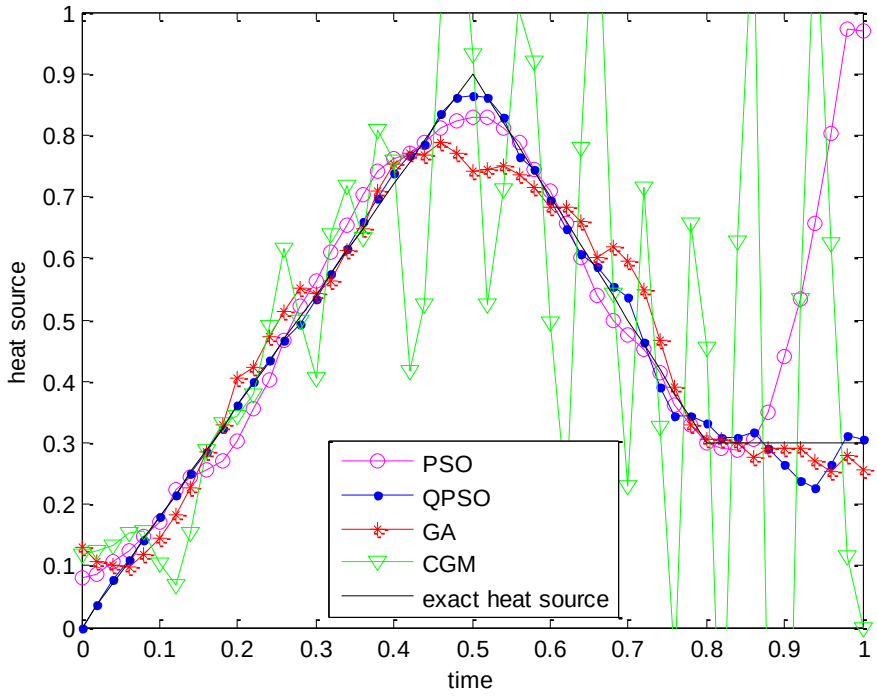


Figure 5.11: The estimated results using different methods with noise level $\varepsilon = 0.03$.

Table 5.8: Analysis of different methods with noise level $\varepsilon = 0.03$.

Run Number	G_{error}	J[G]	Computational Time (s)
Run 2	**2.33E-03**	1.91E-05	5.13
Run 5	5.69E-03	**1.50E-05**	4.27
Run 8	8.31E-03	1.72E-05	4.53
Run 11	5.68E-02	4.31E-06	0.70

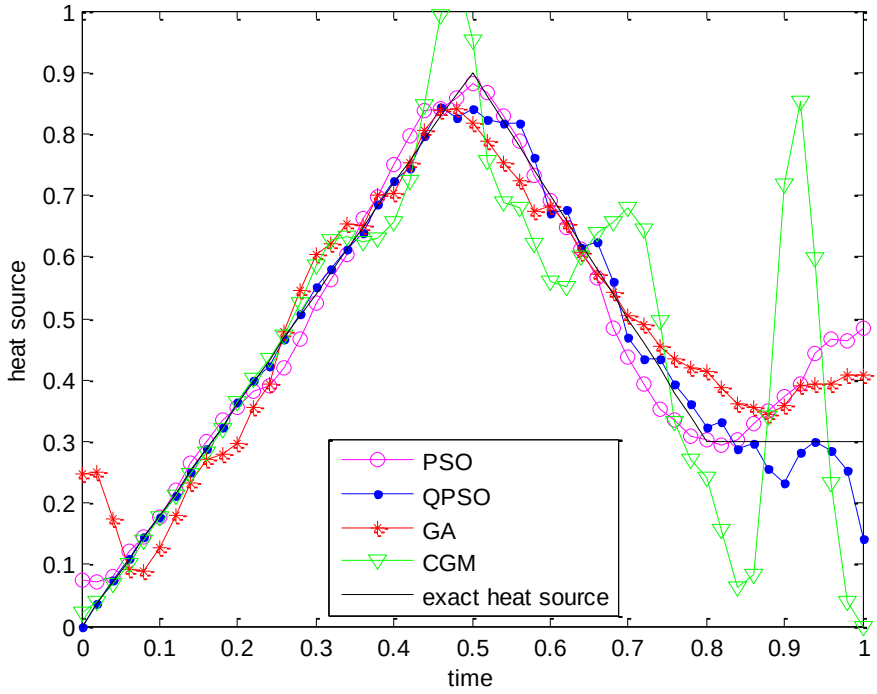


Figure 5.12: The estimated results using different methods with noise level $\varepsilon = 0.05$.

Table 5.9: Analysis of different methods with noisy level $\varepsilon = 0.05$.

Run Number	G_{error}	J[G]	Computational Time (s)
Run 3	2.85E-03	1.47E-05	5.11
Run 6	9.28E-03	3.98E-05	4.27
Run 9	6.76E-03	4.15E-05	4.50
Run 12	1.06E-02	3.18E-05	0.62

Figure 5.13 shows the convergence history of different methods used to solve the IHCP of estimating the heat source. It can be easily seen that CGM converges much faster than other methods, but it stagnates after only a few iterations. In addition, of the three stochastic methods, QPSO converges faster than GA and PSO, and the converged result obtained by using QPSO is much better than those obtained by using GA and PSO.

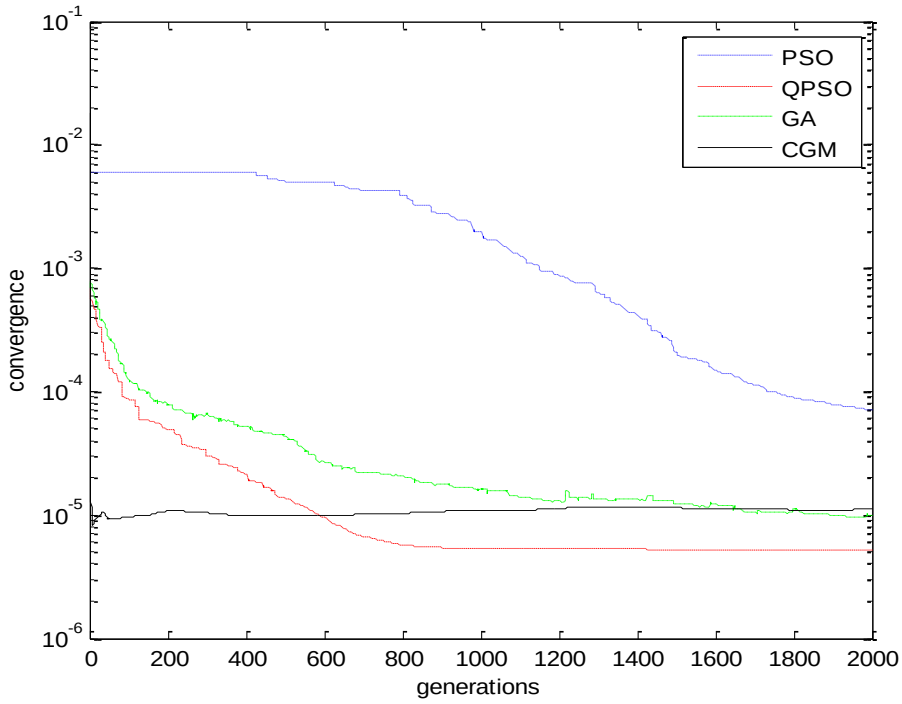


Figure 5.13: Convergence history of different methods after 2000 generations with exact measurements.

To check the robustness of the QPSO for estimating heat sources located at any position, two other cases with $x_s = 0.1$ and $x_s = 0.8$ were tested. The estimated sources are shown in Figure 5.14 and Figure 5.15, respectively. It can be seen that QPSO is robust enough to solve the inverse heat source problem with the heat source located at an arbitray position.

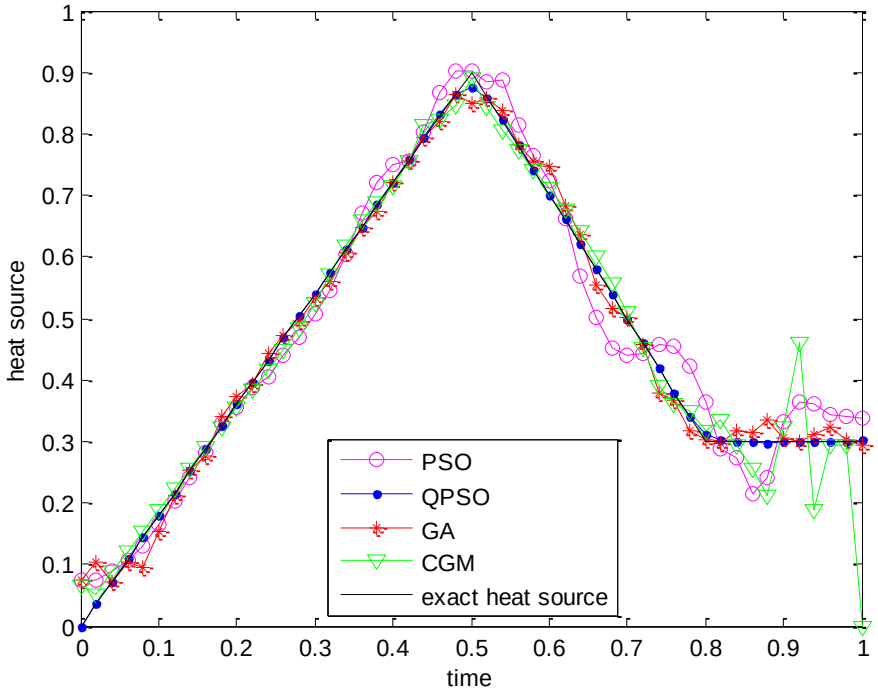


Figure 5.14: The estimated results using different methods with heat source located at $x_s = 0.1$, with exact measurements.

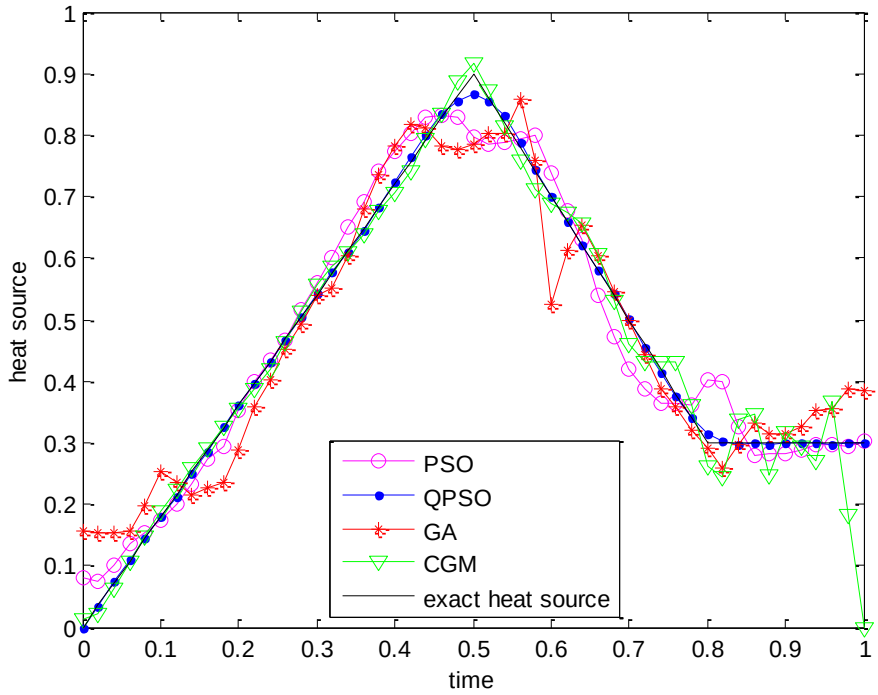


Figure 5.15: The estimated results using different methods with heat source located at $x_s = 0.8$, with exact measurements.

5.3.3 Conclusion

The QPSO method was used to solve the inverse heat source problem of estimating the time-varying heat source. No prior information about the functional form is needed. The numerical results validate the efficiency and stability of the QPSO method. Comparison with CGM, GA and PSO is also presented showing the robustness of QPSO. However, the algorithms encounter difficulties in estimating the value of heat source near the final time when the measurements contains noise.

5.4 Estimation of Groundwater Contaminant Sources in Advection-Dispersion Problems

Environmental contamination is a widespread problem that may affect the utility of resources such as a groundwater aquifer or a surface water body. Identifying contaminant sources in groundwater is important for developing effective remediation strategies and identifying responsible parties in a contamination incident. Groundwater contamination broadly defines any constituent that reduces the quality of groundwater. Contamination can be chemical, physical or biological. Chemical contamination can be broken down further into soluble components and non-aqueous phase liquid components. Soluble components are dissolved in the groundwater and are transported with the groundwater as it moves. Non-aqueous phase liquids are bodies of liquid that are separate from the water and are generally not transported with bulk groundwater movement. This work addresses transport of dissolved chemicals in water-saturated porous media. Transport of soluble chemicals is subjected to process of advection and dispersion. Advection describes the movement of a contaminant along with the bulk movement of groundwater. Dispersion describes the spreading of a contaminant as it moves through the porous media.

If the initial and boundary conditions, model parameters and contaminant release history are known, the advection-dispersion equation can be solved directly using analytical techniques or numerical simulations to obtain the distribution of contaminant concentration. This process is called forward advection-dispersion problem, which has a unique solution and is well-posed. In contrast, the inverse advection-dispersion problem for groundwater models considered here involves the determination of the unknown time-dependent contaminant release history from the knowledge of concentration measurements taken within the medium. The inverse problem of groundwater source identification is ill-posed. Since concentration data is sampled at finite discrete points, infinite number of source history functions can

produce the same set of measured data. As a result, the solution of this inverse problem is not unique. In addition, instability of the solution of the inverse problem caused by the unavoidable measurement noise and computation error makes the problem ill-posed.

In this part of the thesis, a source history reconstruction problem is studied using a point source of contamination at a known location in a one-dimensional flow field. The spatial distribution of the contaminant concentration is sampled at a specific time after the initial source release.

5.4.1 Mathematical description

The governing equation for contaminant transport in groundwater is described by the advection-dispersion equation. A one-dimensional contaminant solute transport through a saturated homogeneous porous medium can be written as

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = d\,\dfrac{\partial^2 u(x,t)}{\partial x^2} - V\,\dfrac{\partial u(x,t)}{\partial x}, \ 0 < x < L, 0 < t \le t_f & \text{(a)} \\ u(x,0) = u_0(x), \quad 0 \le x \le L & \text{(b)} \\ u(0,t) = C(t), \quad 0 < t \le t_f & \text{(c)} \\ u(L,t) = 0, \quad 0 < t \le t_f & \text{(d)} \end{cases} \tag{5.15}$$

where u is the contaminant concentration, d is a constant dispersion coefficient, V is a uniform steady pore velocity, $C(t)$ is the source located at $x = 0$, and $u_0(x)$ is the initial spatial distribution of the contaminant concentration. The first term on the right hand side of Equation (5.19a) is the dispersion term and the second term is advection term. Here $C(t)$ is the unknown contaminant source history to be determined.

Because a central difference approximation to the advection term of Equation (5.19) may lead to oscillation in the solution, an implicit upwind finite difference method is used to discretise the Equation (5.19), leading to

$$\frac{u_i^{j+1} - u_i^{j}}{\Delta t} = d\,\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{\Delta x^2} - V\,\frac{u_{i+1}^{j+1} - u_i^{j+1}}{\Delta x}. \tag{5.16}$$

5.4.2 Numerical tests

In this section, a modified QPSO with perturbation operator proposed in section 4.4.1 was applied to solve the inverse problem of estimating the time-varying contaminant source. The position of a particle represents a candidate solution of the contaminant source $C(t)$. In a discretised computation, the dimension of the position D is equal to the number of time steps N_t used in the temporal discretisation of $C(t)$. The objective fitness function $J[C]$ is defined as in Equation (5.1). At each generation k of finding the minimum of $J[C]$, a particle is defined as

$$X_i(k) = (X_{i1}(k), X_{i2}(k), ..., X_{ij}(k), ..., X_{iD}(k)) = (C(t_1), C(t_2), ..., C(t_j), ..., C(t_{N_t})) \qquad (5.17)$$

where $D = N_t$ represents the dimension of the particle's position and number of time steps required in the computation. Substituting $X_i(k)$ into Equation (5.19), the contaminant distribution $u(x,t)$ can be computed by solving the direct problem. Each feasible solution $X_i(k)$ is evaluated by computing the objective fitness function $J[C]$. At each generation, the positions of the particles are updated as according to Equation (4.31). This process is repeated until a pre-defined number of generations have reached or the solutions converged. C_{error}, as defined in Equation (5.8), is used to evaluate the accuracy of the converged solution.

In order to verify the viability and efficiency of the method and compare the performance of the original PSO, QPSO and modified QPSO with perturbation (QPSO-PER) to solve this inverse source problem, the typical source example

$$C(t) = \exp\left[-\frac{(t-130)^2}{2\cdot 5^2}\right] + 0.3\exp\left[-\frac{(t-150)^2}{2\cdot 10^2}\right] + 0.5\exp\left[-\frac{(t-190)^2}{2\cdot 7^2}\right] \qquad (5.18)$$

in the literature [104-109] was used in the numerical tests.

The contaminant source history of Equation (5.22) is plotted in Figure 5.16 from $t = 0$ up to $t = 300$ days showing three peaks within the period. This true release history is used to generate the contaminant concentration which may be used as sampling concentration in the subsequent computation. The sampling concentration distribution after 300 days at various locations of the measurements is shown in Figure 5.17.

The inverse problem is to reconstruct the source history using measured concentration sampling between $t \in [0,300]$ ($t_f = 300$), with $V = 1.0$ and $d = 1.0$. The temporal step size was chosen as $\Delta t = 3.0$ and the mesh size was set as $\Delta x = 1.0$.

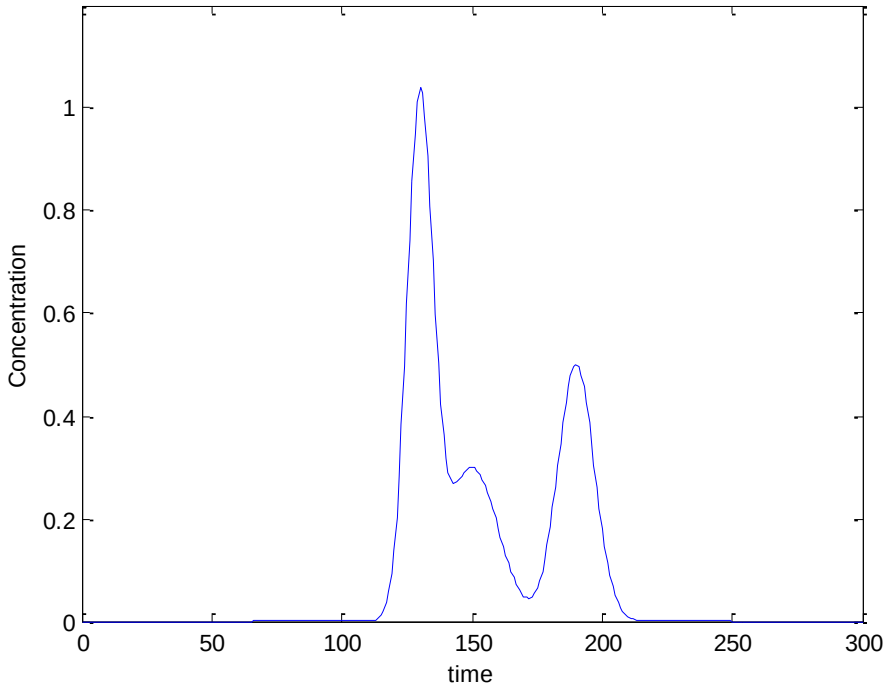


Figure 5.16: The time-varying contaminant release history

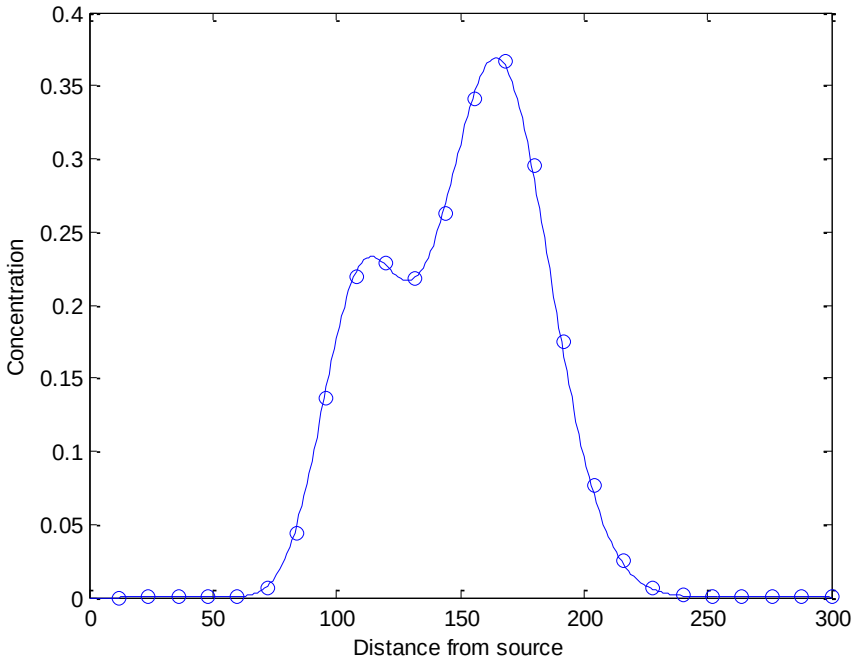


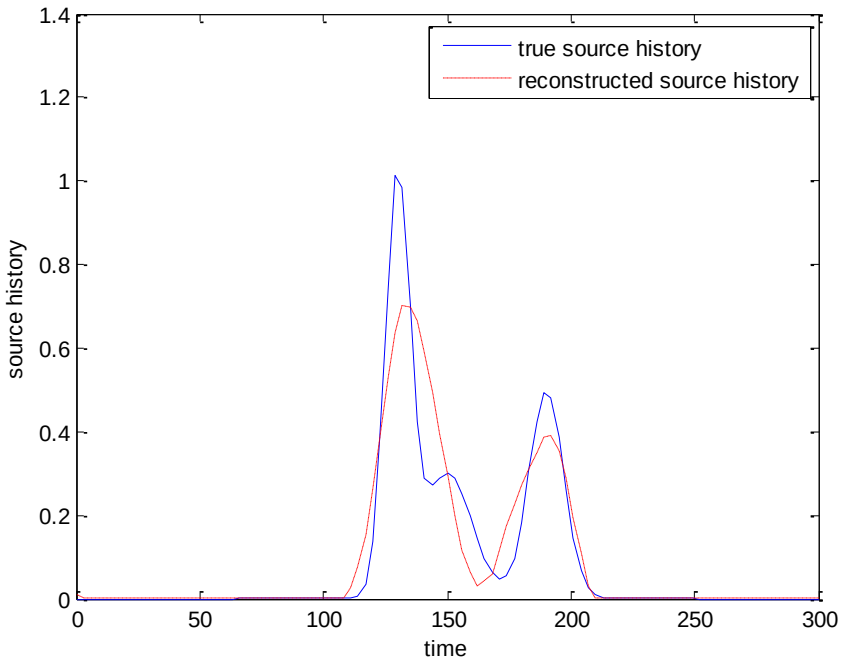
Figure 5.17: Contaminant plume after 300 days, with measurement locations denoted by circles.

The algorithmic parameters in the original PSO were set as, $M = 50$, $D = N_t = 300/\Delta t = 100$, $V_{max} = 1.2$, $c_1 = c_2 = 2.0$, $\omega_0 = 0.9$, $\omega_1 = 0.4$, $k_{max} = 2000$. In order to compare the results in the same condition, the same values were used in those common parameters in QPSO, QPSO-PER and PSO.

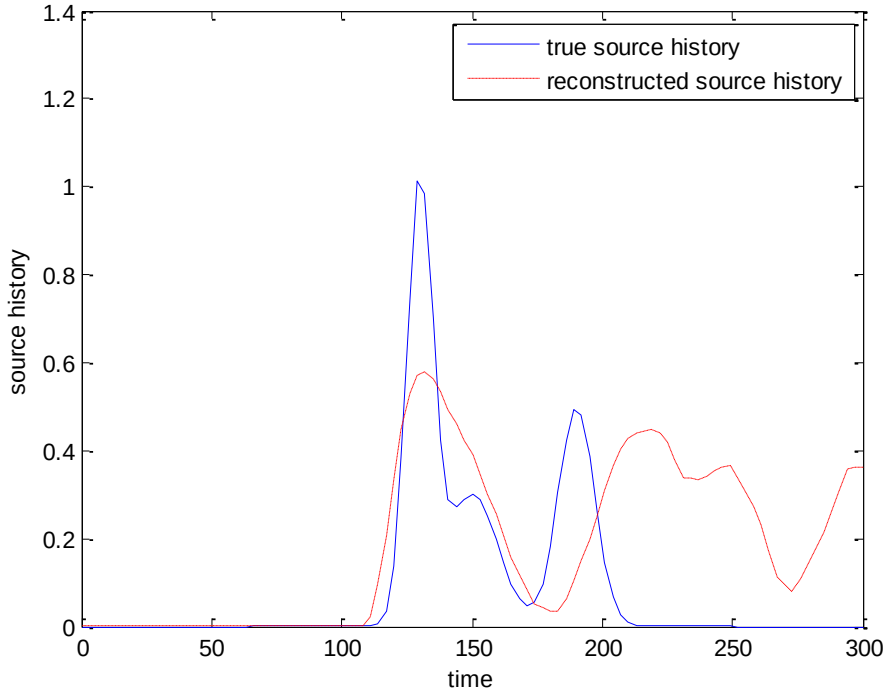
In the literature, three types of samples are included: samples taken at one location over a period of time, samples taken at many locations at one time, or a combination of the two. Numerical tests were performed to evaluate the effects of the sampling time (t_s) and location (x_s) on the solution of the inverse source problem. The sampling times and locations of these tests are listed in Table 5.10. Figure 5.18 shows the estimated contaminant source history for the three run cases. Their corresponding average errors C_{error} and objective function values $J[C]$ are listed in Table 5.11. Note that samples obtained at more locations over a period of time produce results with better accuracy as can be seen from Figure 5.19c.

Table 5.10: Test scenarios for the analysis of the sampling time and location.

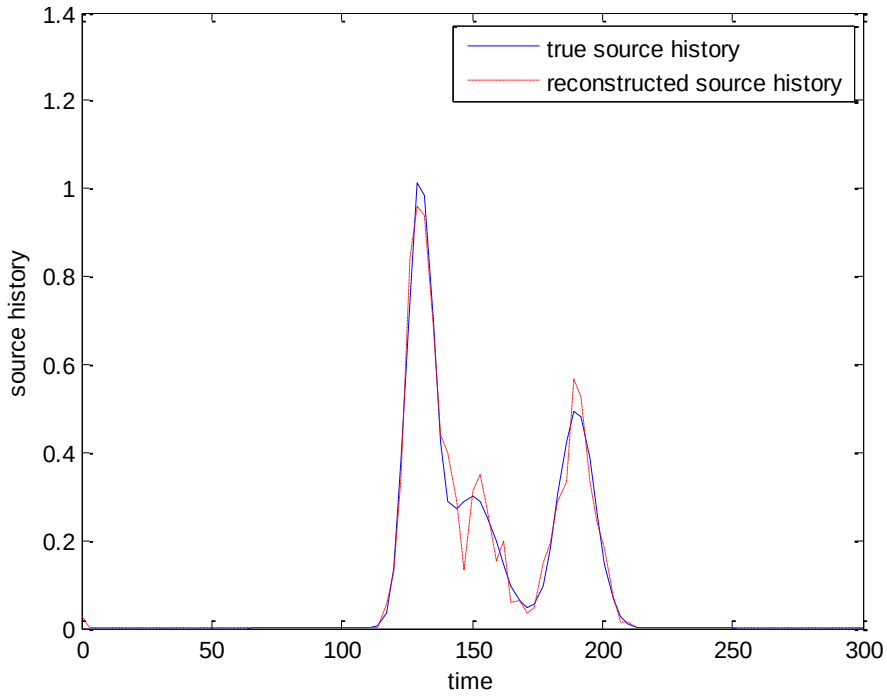
Run Number	Sampling Time	Sampling Location
Run1	$t_s = t_f$	25 locations as in Figure 5.17
Run2	$t_s = \Delta t, 2\Delta t, ..., N_t \Delta t$	$x_s = L/2$ (middle of the domain)
Run3	$t_s = \Delta t, 2\Delta t, ..., N_t \Delta t$	25 locations as in Figure 5.17



(a)



(b)



(c)

Figure 5.18: Reconstructed source history with data in (a) Run1, (b) Run2, (c) Run3.

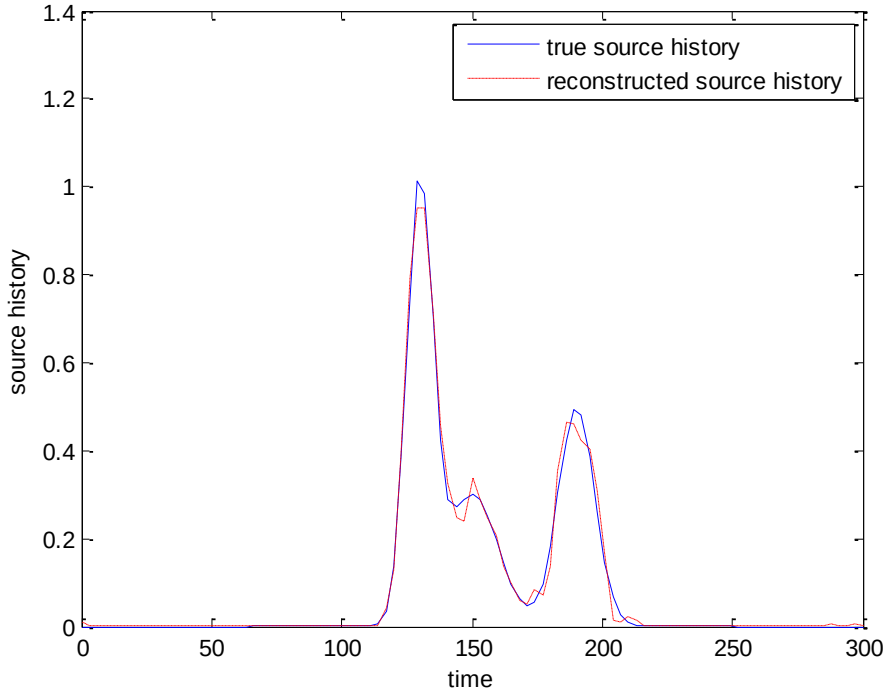
Table 5.11: Results of analysis of the sampling time and location.

Run Number	C_{error}	$J[C]$
Run1	7.96E-03	5.20E-06
Run2	2.12E-02	1.00E-04
Run3	**2.95E-03**	3.38E-04

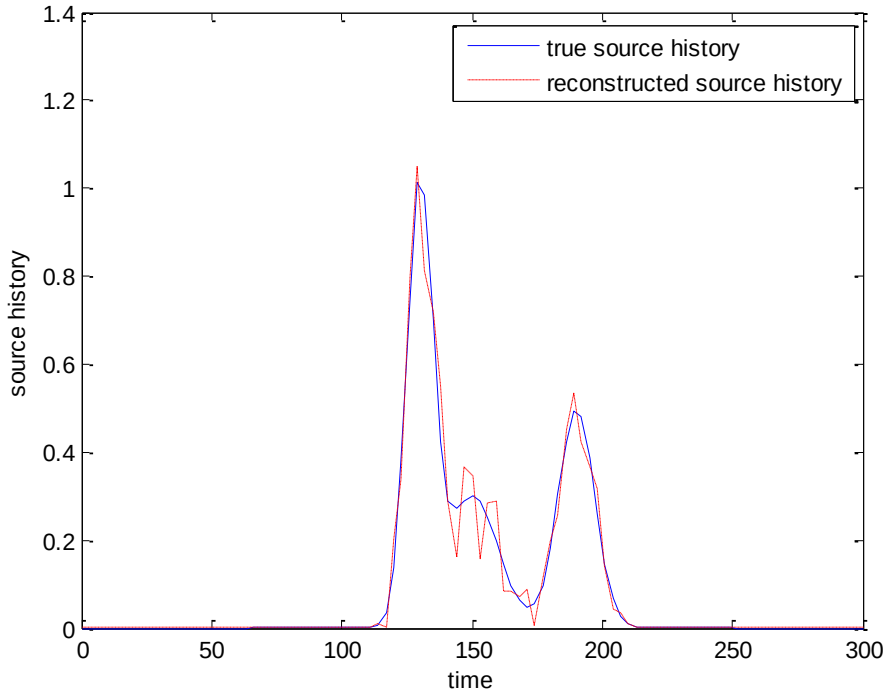
To analyse the effect of the regularisation term on the solution of the inverse source history problem, two different scenarios were tested. Various regularisation terms and noise level details, as outlined in Table 5.12 were used in the tests. Figure 5.19 shows the reconstructed source history by using QPSO, where exact measurements were used with first and zeroth order regularisation terms. The optimal regularisation parameter obtained by L-curve method is $\lambda = 6\times10^{-3}$. Table 5.13 shows the average error and the objective function value. It can be seen that the first-order regularisation can deal with the non-smooth function effectively, while the zeroth-order regularisation can not reduce the oscillatory feature of the input function.

Table 5.12: Test scenarios for the analysis of the regularisation term.

Run Number	Regularisation Order	Noise Level (ε)
Run4	First order	0
Run5	Zeroth order	0
Run6	First order	0.05
Run7	First order	0.2



(a)



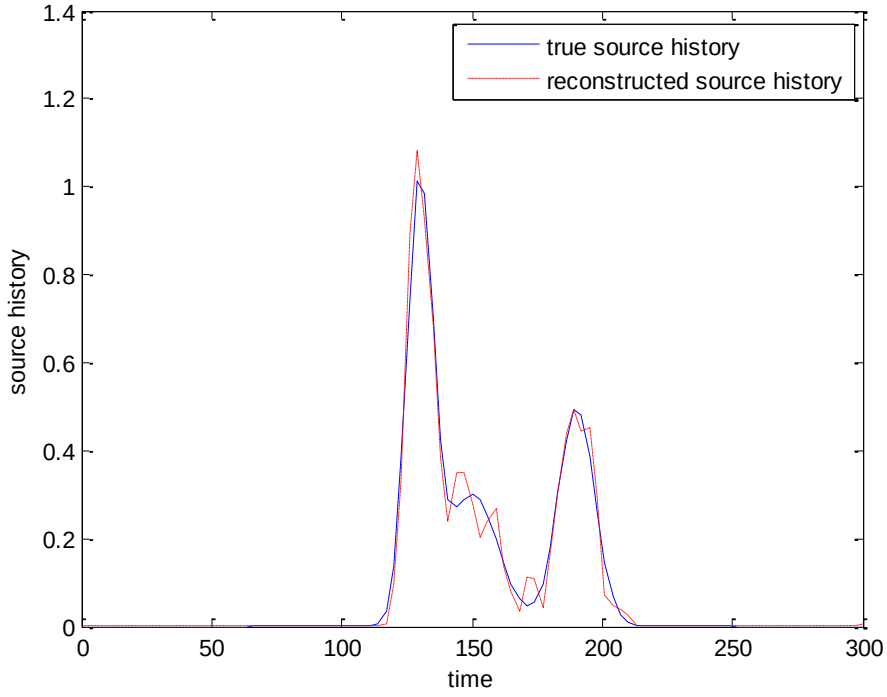
(b)

Figure 5.19: Reconstructed source history with (a) Run4 (b) Run5.

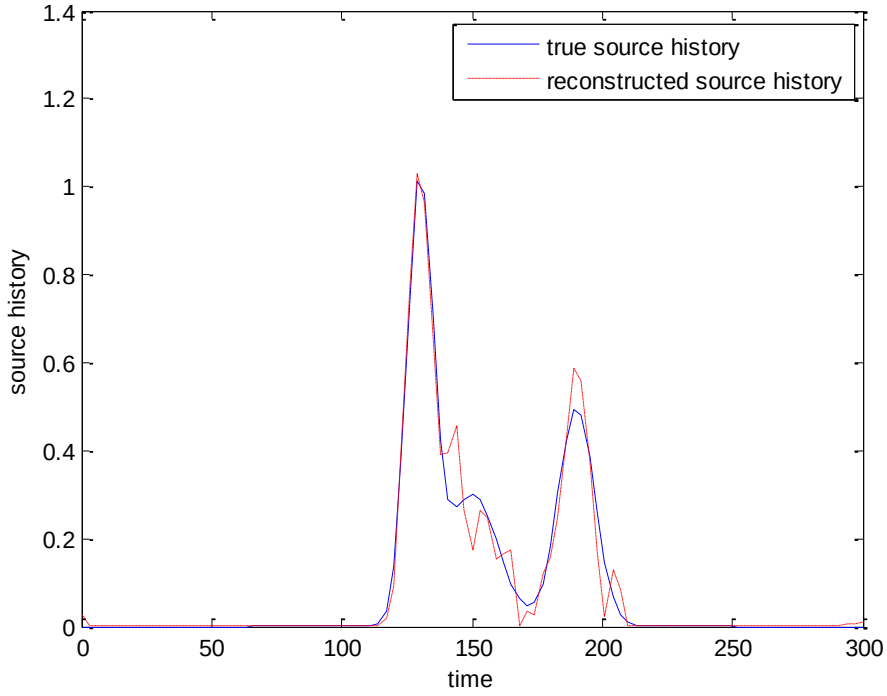
Table 5.13: Results of the analysis of regularisation.

Run Number	C_{error}	J[C]
Run4	1.81E-03	2.03E-04
Run5	3.49E-03	3.12E-04

In order to verify the stability of the QPSO for solving the inverse source history problem, measured concentrations with noise were used in scenarios of Run6 and Run7. The reconstructed contaminant source history is shown in Figure 5.20 and Table 5.14 listed the corresponding average error and objective function value. Note that the QPSO is able to reconstruct the source history with high accuracy even with noisy samples.



(a)



(b)

Figure 5.20: Reconstructed source history using data in (a) Run6 (b) Run7.

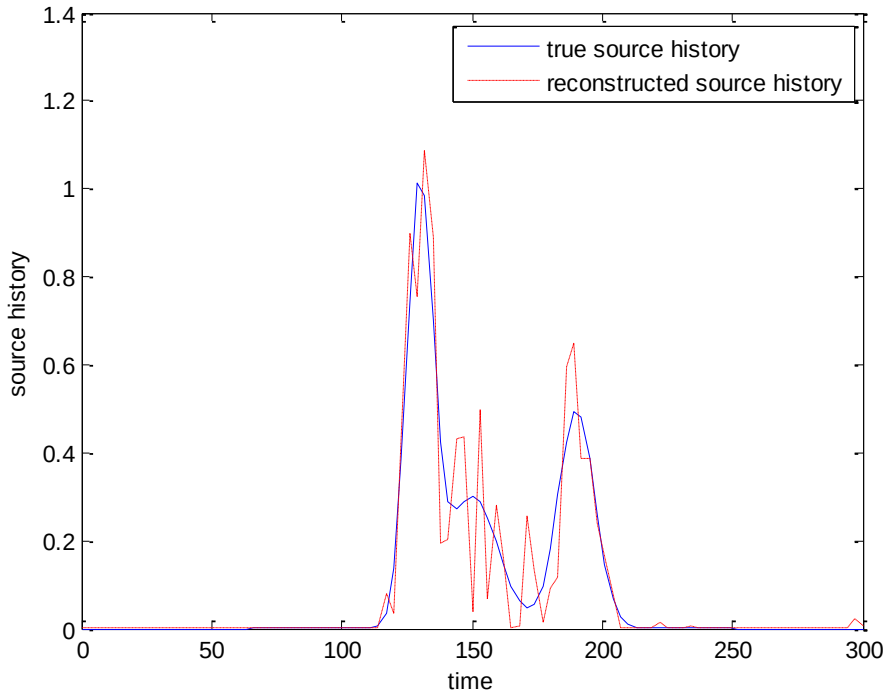
Table 5.14: Results of the analysis of noisy measurements.

Run Number	C_{error}	J[C]
Run6	2.94E-03	7.66E-02
Run7	2.07E-03	0.29

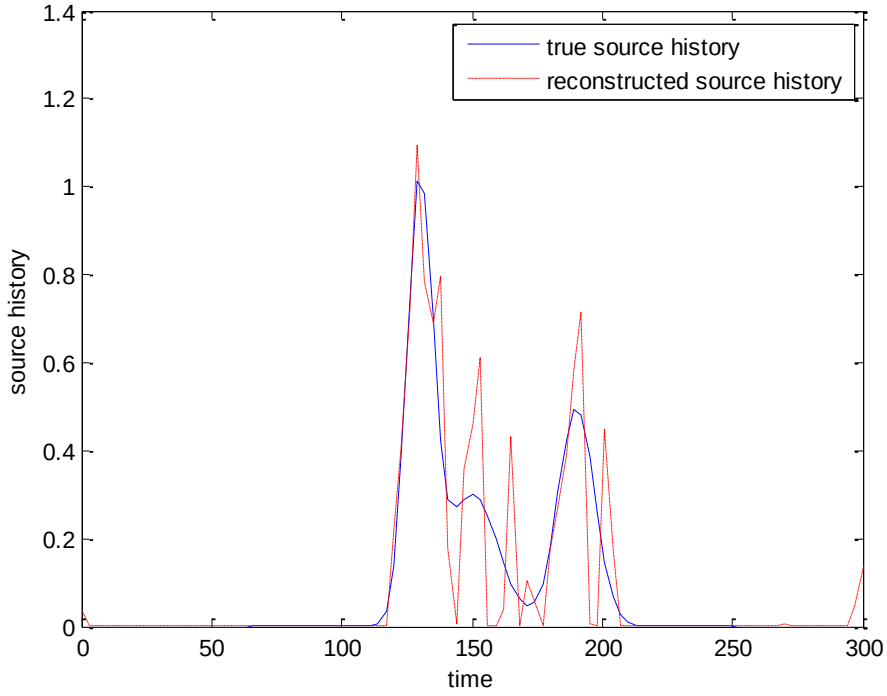
In order to examine the performance, PSO and GA were also used to solve the inverse source history problems for the test scenarios listed in Table 5.15. From Table 5.16 and Figure 5.21-Figure 5.22, it can be seen that the reconstructed source history and performance obtained by QPSO are much better than those from PSO and GA. In particular, QPSO is more robust in dealing with noisy samplings than its competitors.

Table 5.15: Test scenarios by PSO and GA.

Run Number	Methods	Noise Level (ε)
Run8	PSO	0.0
Run9	PSO	0.1
Run10	GA	0.0
Run11	GA	0.1

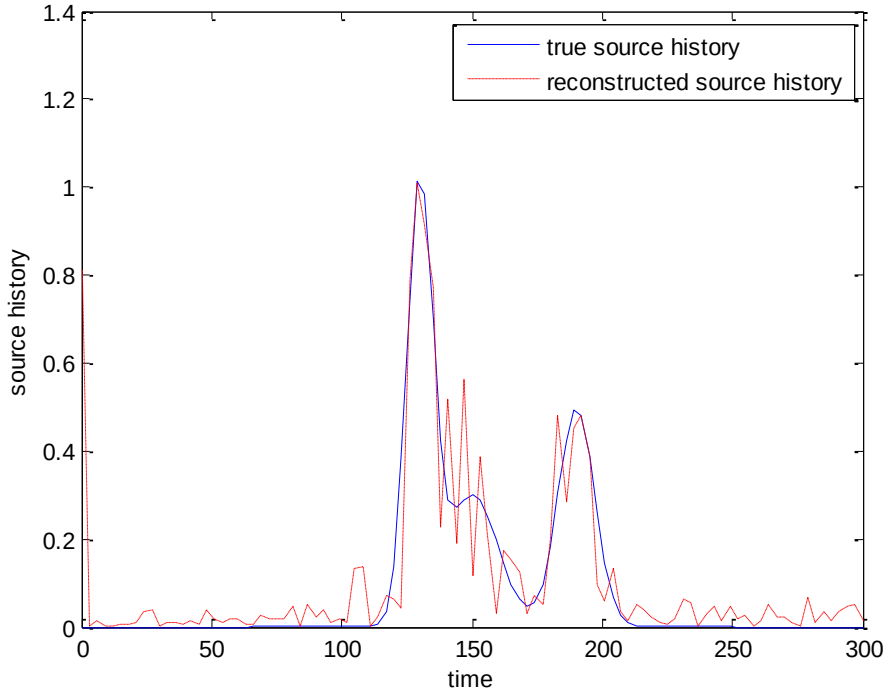


(a)

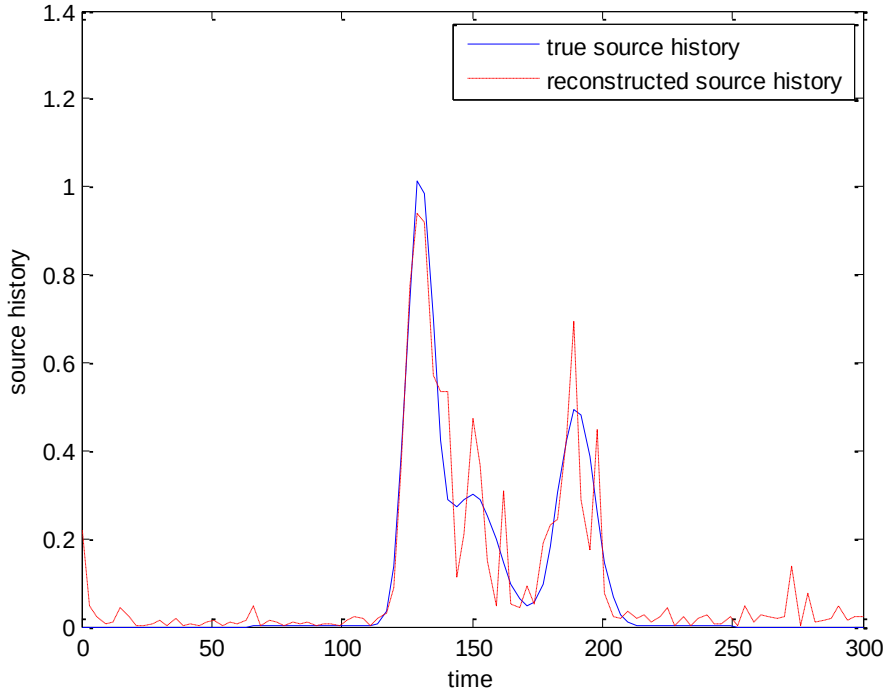


(b)

Figure 5.21: Reconstructed source history by using PSO in (a) Run8, (b) Run9.



(a)



(b)

Figure 5.22: Reconstructed source history by using GA in (a) Run10, (b) Run11.

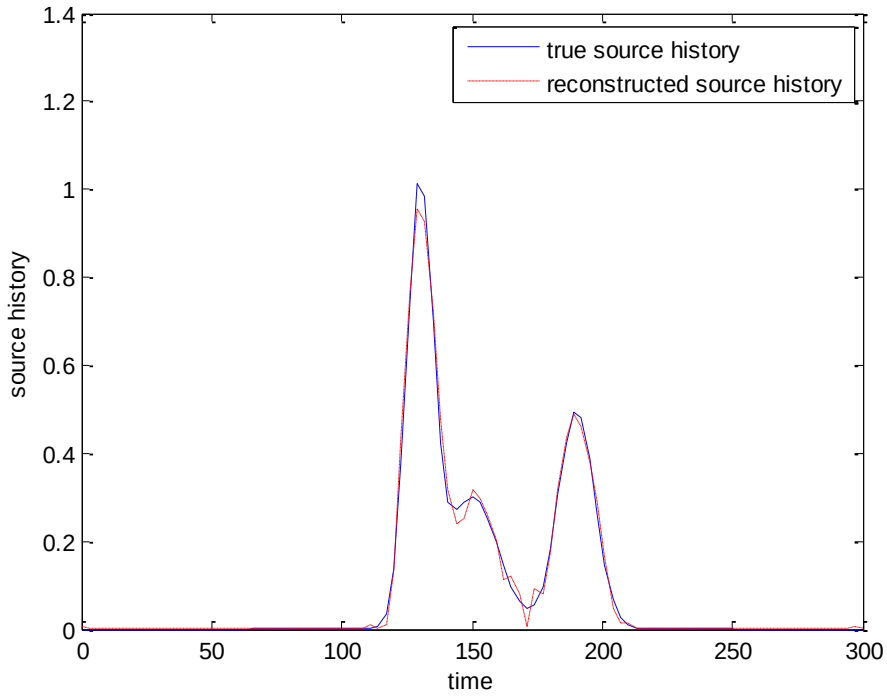
Table 5.16: Results of PSO and GA for solving the inverse source history problem.

Run Number	C_{error}	J[C)]
Run8	7.59E-03	2.58E-03
Run9	1.02E-02	0.31
Run10	1.10E-02	0.32
Run11	7.10E-03	0.46

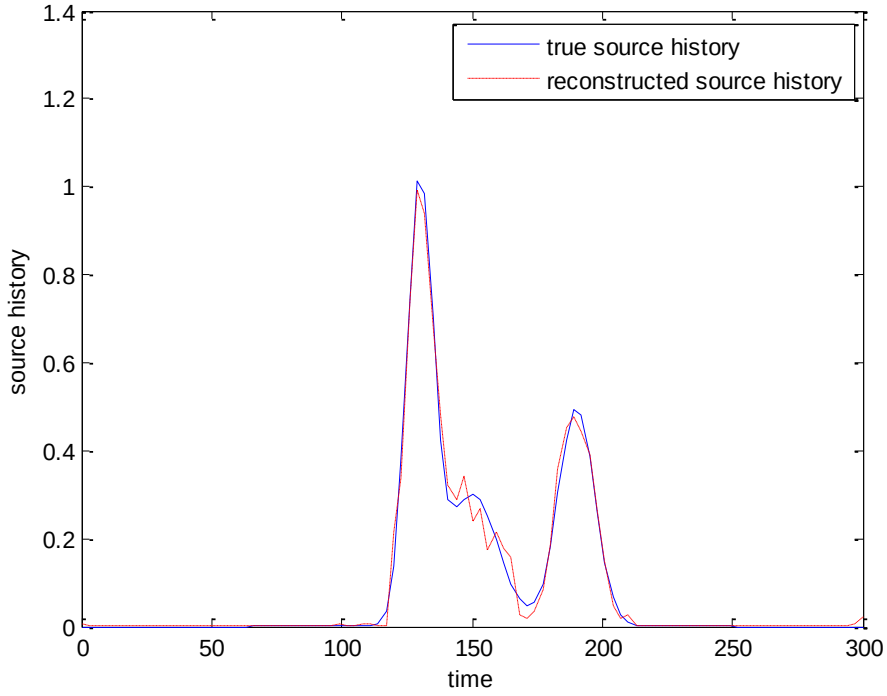
Finally, the modified QPSO with a perturbation operator (QPSO-PER) was applied to reconstruct the contaminant source history for the test scenarios listed in Table 5.17. The results are shown in Figure 5.23 and Table 5.18. It can be obviously observed that the QPSO-PER provides a better estimation of the unknown source history than other methods. The convergence history of the methods tested here used is shown in Figure 5.24. It can be seen that QPSO and QPSO-PER show better convergence rate.

Table 5.17: Test scenarios by using QPSO-PER.

Run Number	Methods	Noise Level (ε)
Run12	QPSO-PER	0.0
Run13	QPSO-PER	0.1



(a)



(b)

Figure 5.23: Reconstructed source history by using QPSO-PER in (a) Run12, (b) Run13.

Table 5.18: Results of QPSO-PER for solving the inverse source problems.

Run Number	C_{error}	J[C]
Run12	1.59E-03	1.89E-04
Run13	3.64E-03	0.28

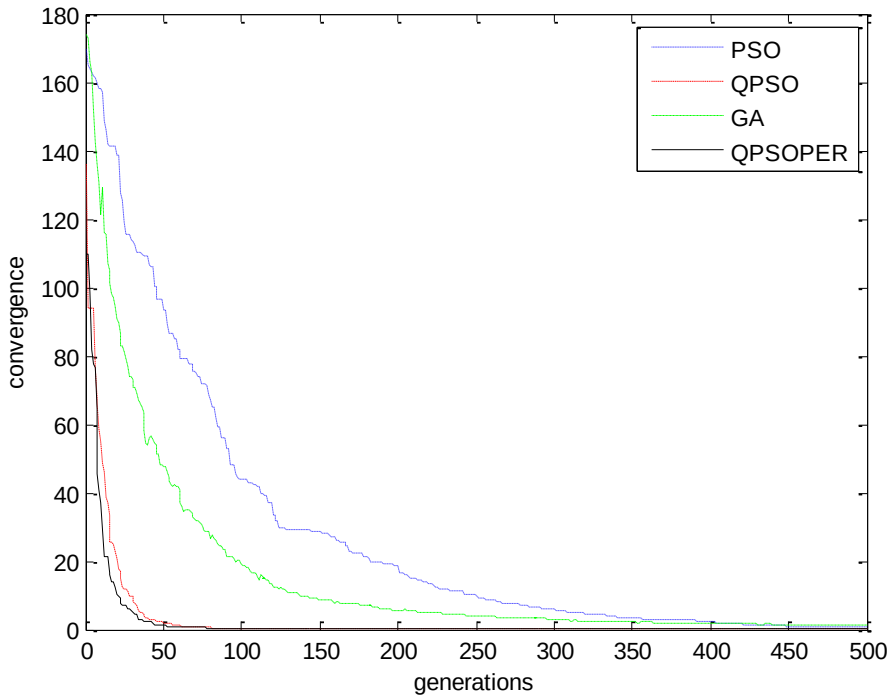


Figure 5.24: Convergence history of the methods used for solving the inverse source history problems.

5.4.3 Conclusion

A modified QPSO with perturbation operator (QPSO-PER) is applied to solve the inverse problem of reconstructing the contaminant source history. No prior information about the functional form is assumed and a number of peaks exist in the function. The numerical results suggest that these methods are all able to converge to the optimal or sub-optimal solutions. The QPSO-PER is more robust than the other methods in dealing with the noisy samplings.

It should be pointed out that deterministic methods such as MRE require a gradient calculation while stochastic methods such as QPSO developed in this thesis do not require this and are guaranteed to converge to the global optimum as seen in the tests above.

5.5 Estimation of Heat Transfer Coefficients in Heat Conduction Problems

The heat transfer coefficient, in thermodynamics, is used in calculating the heat transfer, typically by convection or phase change between a fluid and solid. Accurate knowledge of the heat transfer coefficient at the surface of the plate is important in many engineering applications, including the cooling of continuously cast slabs and electronic chips [1].

In the past decades, much research work has been contributed to the estimation of time-varying heat transfer coefficient. Su and Hewitt [69] used Alifanov's iterative regularisation method to estimate the time-varying heat transfer coefficient of forced-convective flow boiling over the outer surface of a heater tube. In this section, the QPSO described earlier is used to estimate the time-varying heat transfer coefficient. Other methods and studies are refered to in Chapter 1. The result is compared with that obtained by using CGM.

5.5.1 Mathematical description

A flat plate over which a fluid is flowing at a constant temperature u_∞ [1] as shown in Figure 5.25 is used as the test example here.

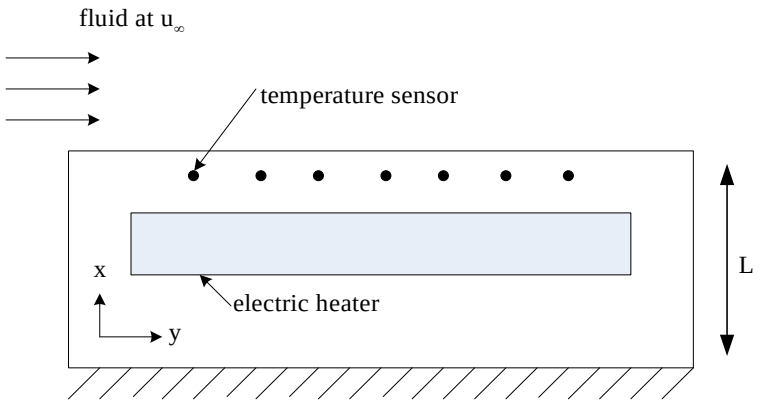


Figure 5.25: An electrically heated flat plate.

If the plate was suddenly heated by a certain electric heater inside it, the temperature of the plate would rise. Assume the material of the flat plate is homogeneous along the y-axis and that convective boundary conditions are specified at $x = 0$ and $x = L$. The mathematical model is given by

$$\begin{cases} \rho C \dfrac{\partial u}{\partial x} = \dfrac{\partial}{\partial x}\left(K \dfrac{\partial u}{\partial x} \right) + G(t)\delta(x - x_s), \quad 0 < x < L, 0 < t \le t_f & \text{(a)} \\ K \dfrac{\partial u}{\partial x}\Big|_{x=0} = 0, \quad 0 < t \le t_f & \text{(b)} \\ -K \dfrac{\partial u}{\partial x}\Big|_{x=L} = h(t)[u_\infty - u], \quad 0 < t \le t_f & \text{(c)} \\ u\big|_{t=0} = u_0, \quad 0 \le x \le L & \text{(d)} \end{cases} \tag{5.19}$$

where $u(x,t)$ is the temperature distribution at a spatial location x and time t. $G(t)$ is the strength of the heat source at $x = x_s$. u_∞ is the ambient temperature and u_0 is the initial temperature distribution. For simplicity, $\rho C = K = 1$ and $L = 1$, which is the same as using non-dimensional data. Here $h(t)$ is the unknown heat transfer coefficient to be determined.

Applying an implicit finite difference method to Equation (5.23) leads to

$$\rho C \frac{u_i^{j+1} - u_i^j}{\Delta t} = K \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{\Delta x^2} + G_{j+1}\delta(x_i - x_s). \tag{5.20}$$

At $x = L$, the second order discretisation of the convective boundary condition leads to

$$-K \frac{3u_{N_x}^{j+1} - 4u_{N_x-1}^{j+1} + u_{N_x-2}^{j+1}}{2\Delta x} = h_{j+1}(u_\infty - u_{N_x}^{j+1}) \tag{5.21}$$

5.5.2 Numerical tests

QPSO is used to estimate the time-varying heat transfer coefficient $h(t)$. Every particle $X_i(k)$ is treated as a candidate solution of $h(t)$, namely

$$X_i(k) = \{X_{i1}(k), X_{i2}(k), ..., X_{iD}(k)\} = \{h(t_1), h(t_2), ..., h(t_{N_t-1}), h(t_{N_t})\}, \tag{5.22}$$

where $D = N_t$. Substituting $X_i(k)$ into Equation (5.23), the temperature u can be computed by solving the direct problem. Each feasible solution $X_i(k)$ is evaluated by computing the objective fitness function $J[h]$ as defined in Equation (5.1). At each generation, the positions of the particles are updated according to Equation (4.31). The process is repeated until a pre-defined number of generations have reached or the solution converged. h_{error}, as defined in Equation (5.8), is used to evaluate the accuracy of the converged solution.

In the subsequent numerical tests, the square-wave function [69]

$$h(t)=\begin{cases}1.0, & t\le 1\\ 2.0, & 1<t\le 2\\ 1.0, & t>2\end{cases} \tag{5.23}$$

is used. It is the most difficult case to inversely estimate. The problem settings are $x_s=0.5$, $t_f=3.0$, $u_0=0.0$, $G(t)=10.0$, $u_\infty=100.0$, $\Delta x=0.05$ and $\Delta t=0.05$.

In order to examine the effect of the number of sensors and their location on the estimation, a set of different cases as listed in Table 5.19 were used. Figure 5.26 shows the results of the inversely determined heat transfer coefficient using one, three and five sensors. Note that the estimated results obtained with 3 and 5 sensors are not better than that obtained with only one sensor. Therefore, it is suggested that one sensor is sufficient to obtain the satisfactory estimation for this problem.

Table 5.19: Different number of sensors and their locations.

No. of sensors	Locations	J[h]	h_{error}
1	0.9	1.37E-05	2.52E-04
3	0.3, 0.6, 0.9	3.15E-04	1.60E-03
5	0.1, 0.3, 0.5, 0.7, 0.9	7.37E-03	2.87E-03

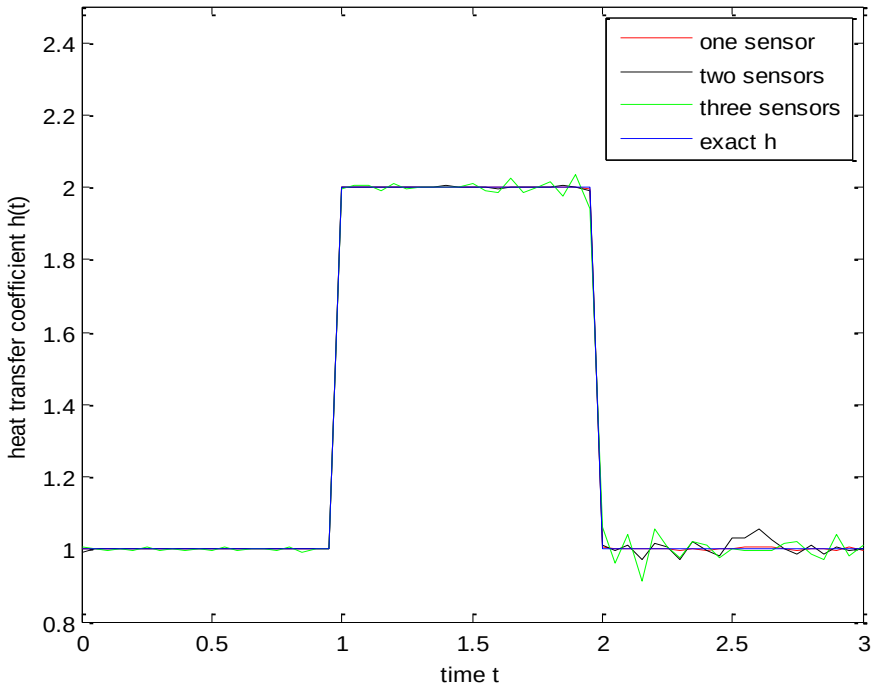


Figure 5.26: Estimated $h(t)$ by using QPSO with different number of sensors.

Therefore, only one sensor is used in the subsequent tests. The effect of sensor locations on the heat transfer coefficient estimation was examined as shown in Figure 5.27. Table 5.20 lists four cases of sensor locations and their corresponding average error h_{error} and objective function value $J[h]$. Note that the closer the sensor is to the boundary with convection, the better the result is, see also Figure 5.27.

Table 5.20: Effects of sensor locations.

Sensor location	$J[h]$	h_{error}
0.0	0.11	3.52E-02
0.3	8.32E-02	3.20E-02
0.6	3.79E-02	1.84E-02
0.9	1.37E-05	2.52E-04

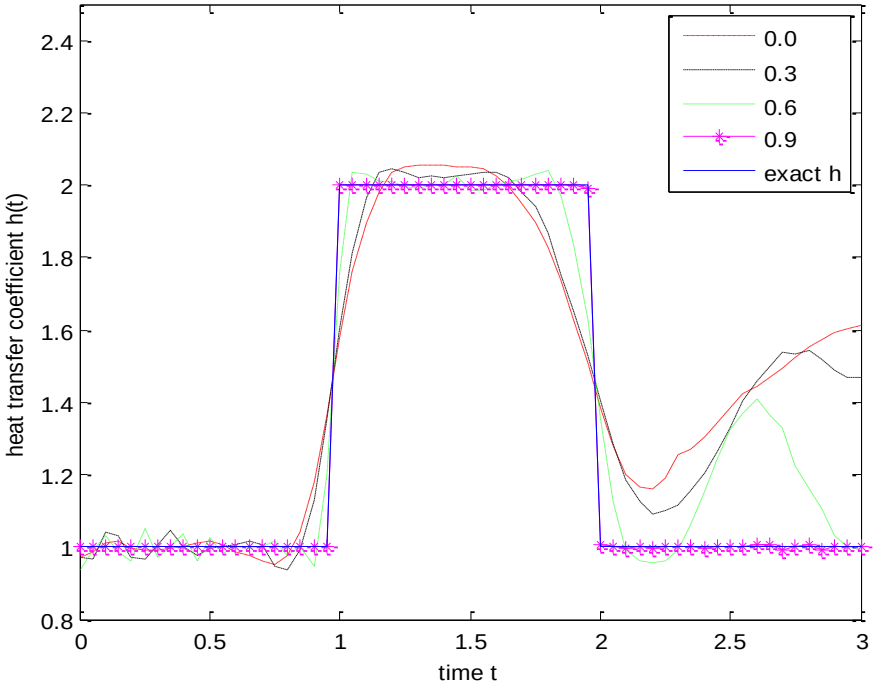


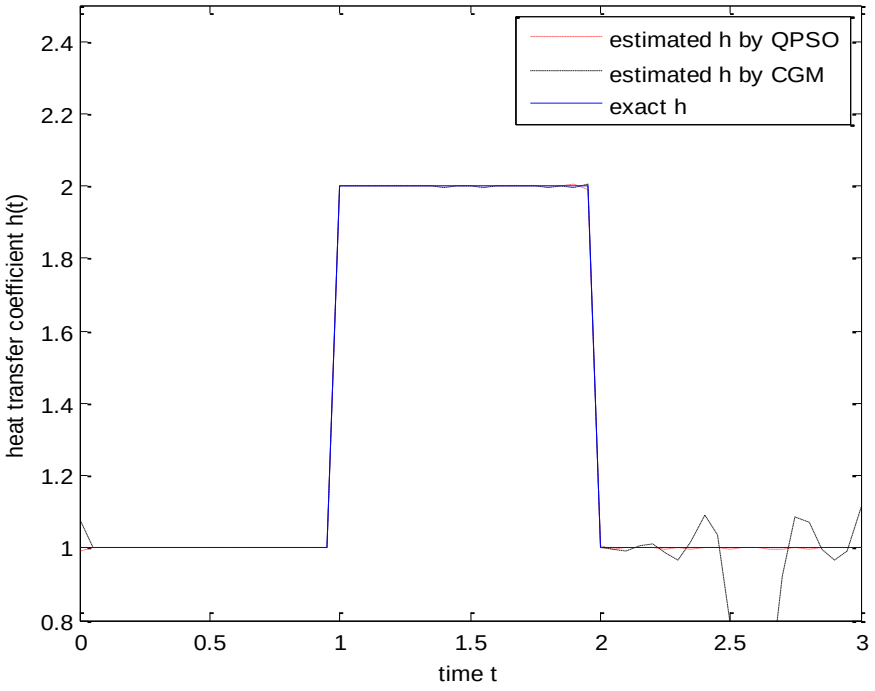
Figure 5.27: Estimated $h(t)$ by QPSO with different sensor locations.

The simulated experimental temperatures with different noise levels were used to examine the effect of measurement errors. Three different noise levels were used in the test. Table 5.21 shows the average error and objective function value obtained by QPSO and CGM. Figure 5.28 shows the comparison of results obtained by QPSO and CGM. Note the excellent result by QPSO, especially for the values near the final time. The comparison of

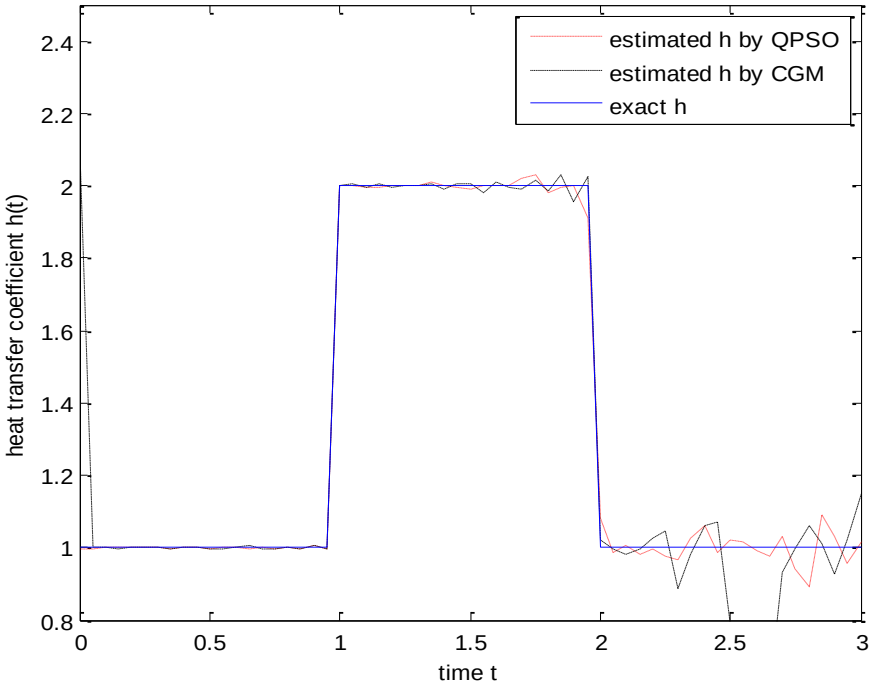
convergence history of QPSO and CGM in estimating the heat transfer coefficient is shown in Figure 5.29. Note that CGM converges faster in the early stage of the iteration process but stagnates after several iterations. Although the QPSO with randomly initialised particles converges slowly in the early stage, the global optimum is ensured.

Table 5.21: Comparison QPSO and CGM with different noise levels.

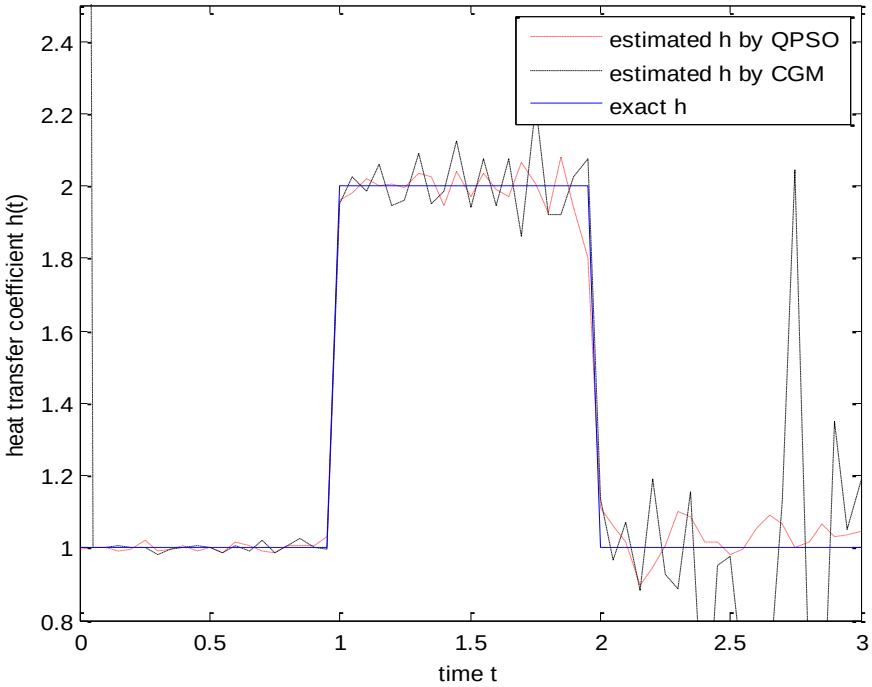
Noise level	QPSO		CGM	
	J[h]	h_{error}	J[h]	h_{error}
0.0	1.37E-05	2.52E-04	3.23E-03	1.61E-02
0.01	6.46E-04	3.74E-03	5.58E-03	1.65E-01
0.05	4.17E-03	6.26E-03	2.94E-02	6.69E-01



(a)



(b)



(c)

Figure 5.28: Estimated results by using QPSO and CGM with different noise levels. (a) $\varepsilon = 0.0$ (b) $\varepsilon = 0.01$ (c) $\varepsilon = 0.05$.

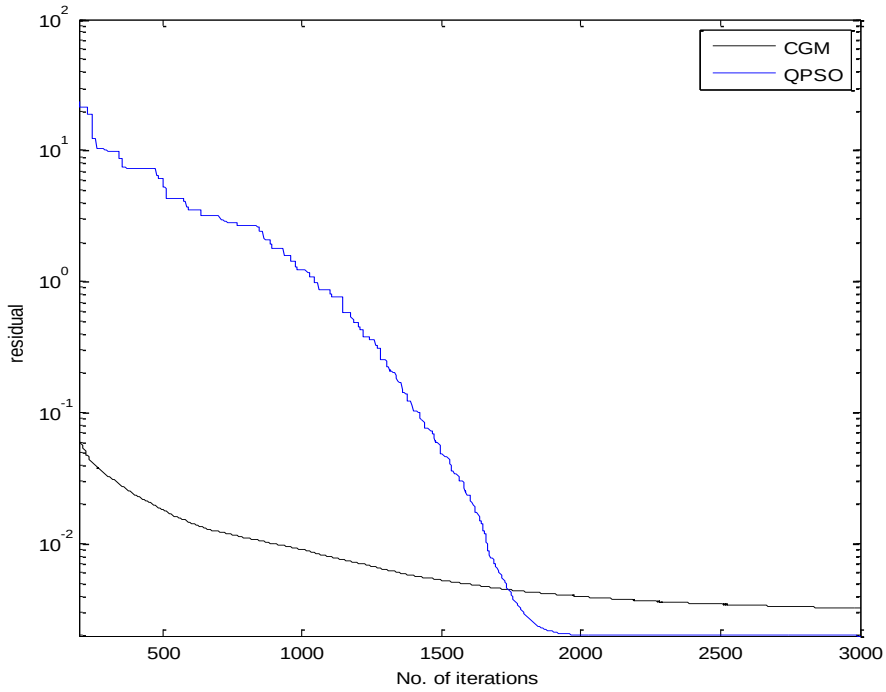


Figure 5.29: Convergence history of CGM and QPSO in estimating the heat transfer coefficient, $\varepsilon = 0$.

In order to test the robustness of the QPSO, a square wave with smaller duration

$$h(t) = \begin{cases} 1.0, & t \le 1 \\ 2.0, & 1 < t \le 1.1 \\ 1.0, & t > 1.1 \end{cases} \tag{5.24}$$

and a piecewise linear waveform

$$h(t) = \begin{cases} t + 1, & t \le 1 \\ t, & 1 < t \le 2 \\ t - 1, & t > 2 \end{cases} \tag{5.25}$$

are used to represent the heat transfer coefficient variation.

The estimated results by QPSO and CGM are shown in Figure 5.30 and Figure 5.31. It can be seen that the time-varying heat transfer coefficients estimated by QPSO agree very well with the exact heat transfer coefficient.

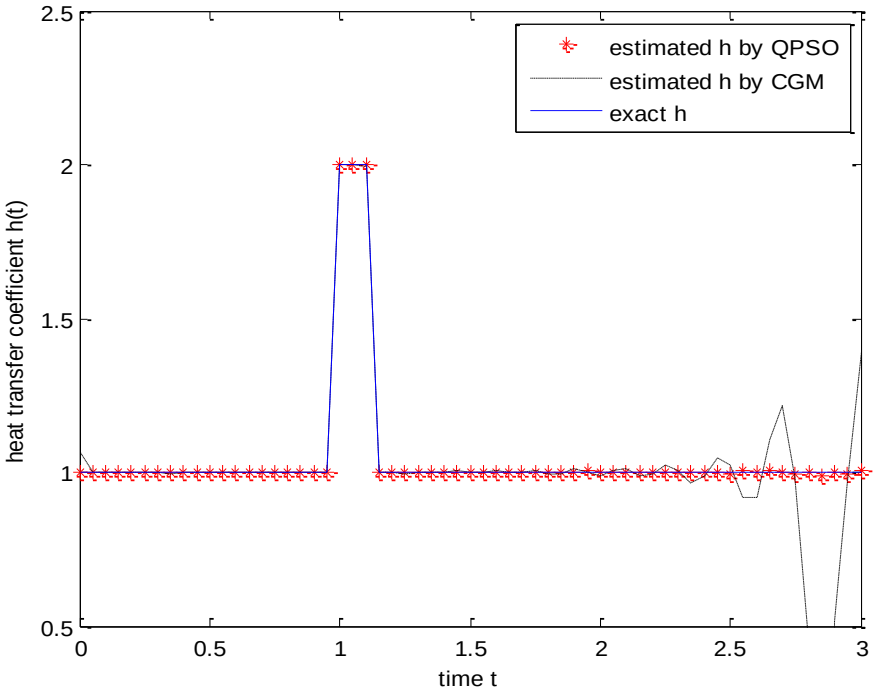


Figure 5.30: Estimated h(t) with smaller wave duration by using QPSO and CGM.

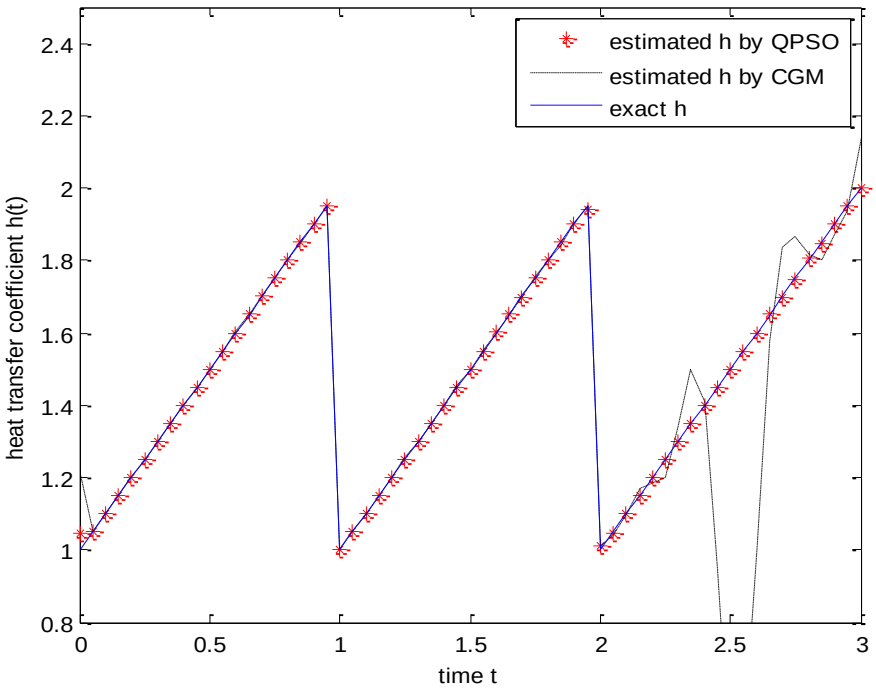


Figure 5.31: Estimated waveform h(t) by using QPSO and CGM with exact measurements.

5.5.3 Conclusion

In this section, the inverse problem of estimating the time-varying heat transfer coefficient at the surface of a plate was solved by using QPSO. Results obtained with simulated measurements demonstrated the viability and stability of the method with respect to measurement error. The effect of number of sensors and sensor locations on the accuracy of the estimation was also investigated. The comparisons of QPSO and CGM illustrate the superiority of the QPSO in estimating the heat transfer coefficient. CGM has difficulty to estimate the values near the final time because of the gradient computation.

5.6 Estimation of Temperature-Dependent Thermal Conductivities in Heat Conduction Problems

The determination of the thermal conductivity of certain material from a measured temperature profile is also an important IHCP [1], [3]. In most practical engineering problems the thermal conductivity is temperature dependent. Thus, the governing heat conduction equation is a nonlinear equation which is harder to solve. A brief overview is given of methods developed to determine the temperature dependent thermal conductivity of materials. Huang et al. [64]-[68] used the CGM with an adjoint equation to search for the thermal properties in one or two dimensional inverse problems. The main problem they saw was the initial guess of the unknown quantities that must be chosen carefully to guarantee convergence of this method. In addition convergence rate could stagnate near the final computing time. Terrola [71] applied the Davidon-Fletcher-Powell method to determine the temperature dependent thermal conductivity. Kim et al. [72] formulated the problem to find the solution through the direct integral method. One limitation of this method is that it requires the material to be homogeneous. Other studies are mentioned in chapter 1.

5.6.1 Mathematical description

Consider a typical one-dimensional homogeneous heat conduction medium with length L as shown in Figure 5.32. It is initially at a constant temperature u_0. When $t > 0$, the left end of the medium is heated by a constant heat flux q_1, while another heat flux q_2 is applied to the right end.

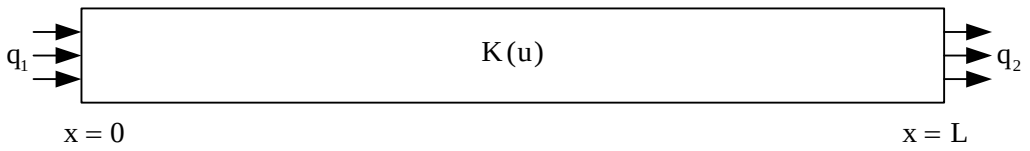


Figure 5.32: Model of a heated slab.

For the case of temperature dependent thermal conductivity $K(u)$, constant heat capacity per unit volume C and density ρ, the heat conduction process is governed by

$$\begin{cases} \rho C \dfrac{\partial u(x,t)}{\partial t} = \dfrac{\partial}{\partial x}[K(u)\dfrac{\partial u(x,t)}{\partial x}], \ 0 < x < L, 0 < t \le t_f & \text{(a)} \\ -K(u)\dfrac{\partial u}{\partial x}\Big|_{x=0} = q_1, \quad 0 < t \le t_f & \text{(b)} \\ K(u)\dfrac{\partial u}{\partial x}\Big|_{x=L} = q_2, \quad 0 < t \le t_f & \text{(c)} \\ u(x,0) = u_0, \quad 0 \le x \le L & \text{(d)} \end{cases} \tag{5.26}$$

For simplicity, $\rho C = 1$ and $L = 1.0$, which is the same as using the non-dimensional data. Here the thermal conductivity $K(u)$ is the unknown to be determined from temperature measurements $Y(x_i, t)$ ($i = 1, 2$) obtained from the sensors located at the both ends, i.e. $x_1 = 0.0$, $x_2 = L$, of the heated medium .

When generating simulated temperature measurements $Y(x,t)$ with a predefined $K(u)$, the nonlinear direct problem defined by Equation (5.30) is required to be solved. An iterative technique is needed in solving the problem in conjunction with an implicit finite difference method.

As the temperature $u(x,t)$ is approaching its converged result by using an iterative technique under some specified initial and boundary conditions, the values of K at any time and position (x,t) are fixed, because temperature $u(x,t)$ is known and fixed at any (x,t). Under this situation, Equation (5.30a) is discretised as

$$\rho C \frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{K_{i+1}^{j+1} - K_{i-1}^{j+1}}{2\Delta x} \frac{u_{i+1}^{j+1} - u_{i-1}^{j+1}}{2\Delta x} + K_i^{j+1} \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2}, \tag{5.27}$$

where u_i^j and K_i^j are the temperature and thermal conductivity at the j^{th} time step $j = 1, 2, ..., N_t$, along the i^{th} grid point $i = 1, 2, ..., N_x$. For the boundaries, the second order discretisation is used

$$-K_0^j \frac{-3u_0^j + 4u_1^j - u_2^j}{2\Delta x} = q_1, \tag{5.28}$$

$$K_{N_x}^j \frac{3u_{N_x}^j - 4u_{N_x-1}^j + u_{N_x-2}^j}{2\Delta x} = q_2. \tag{5.29}$$

The average error value, which intends to evaluate the accuracy of the estimated thermal conductivity, is defined as

$$K_{error} = \frac{1}{N_x N_t} \sqrt{\sum_{i=1}^{N_t} \sum_{j=1}^{N_x} \left(K_i^j - \tilde{K}_i^j \right)^2}, \tag{5.30}$$

where K_i^j is the estimated thermal conductivity and $\tilde{K}_i^j$ is the exact thermal conductivity.

The Tikhonov regularisation method, as described in section 2.3, is used to address the ill-posedness of the inverse problem and stabilise the solution. The objective function given by Equation (5.1) needs to be modified as

$$J[K(x,t)] = \sum_{i=1}^{2} \int_{t=0}^{t_f} \left(u(x_i,t) - Y(x_i,t) \right)^2 dt + \lambda^2 \left\| LK(x,t) \right\|^2. \tag{5.31}$$

In the numerical solution of the direct problem, $u(x,t_{j+1})$ is computed from $u(x,t_j)$. Therefore in the inverse problem, the thermal conductivity may be computed step by step from t_0 to t_{N_t}. Hence, to determine $K(x,t_j)$, the objective function is given by

$$J[K(x,t_j)] = \sum_{i=1}^{2} \int_{t=t_{j-1}}^{t_j} \left(u(x_i,t_j) - Y(x_i,t_j) \right)^2 dt + \lambda^2 \left\| LK(x,t_j) \right\|^2, \tag{5.32}$$

which can be solved in the same way as $v(x)$ defined in Equation (5.1).

5.6.2 Numerical tests

Because the inverse problem of determining the temperature dependent thermal conductivity is highly nonlinear and ill-posed, the QPSO with global topology converges too fast and the diversity of the population decreases quickly. In order to avoid being trapped into a local

optimum, the QPSO with ring topology (SQPSO), proposed in section 4.4.2, was applied to solve this inverse problem. Every particle $X_i(k)$ of the QPSO is treated as a candidate solution of the thermal conductivity $K(x,t_j)$ given by

$$X_i(k) = \{X_{i1}(k), X_{i2}(k), ..., X_{iD}(k)\} = \{K(x_1,t_j), K(x_2,t_j), ..., K(x_{N_x},t_j)\}. \quad (5.33)$$

The dimension size D of particle's position equals to the number of space nodes N_x. Equation (5.36) is used as the objective function to evaluate the particles at time t_j. Two different methods may be used to select the contraction-expansion coefficient α as described in section 4.4.2. Substituting $X_i(k)$ into Equation (5.30), the temperature u can be computed by solving the direct problem. At each generation, the positions of the particles are updated according to Equation (4.31). The process is repeated until a pre-defined number of generations have reached or the solution converged. K_{error}, as defined in Equation (5.34), is used to evaluate the accuracy of the converged solution.

The solution procedure of SQPSO for estimating thermal conductivity is shown as below.

For $j = 1, 2, \ldots N_t - 1$

Initialize particles with random positions $X(0)$;

Initialize pbest $P(0)$, lbest $LBEST(0)$, α, $k = 0$, stopping criteria σ;

While ($k < k_{max}$) or (σ is not reached)

Compute the Mean Best Position $C(k)$ by Equation (4.30);

For each particle $i = 1, 2, ..., M$:

Compute $p_i(k)$ by Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

Evaluate all the particles $J[X_i(k)]$, according to Equation (5.36);

Update pbest $P_i(k)$ and lbest $LBEST_i(k)$;

End for

Update α according to Equation (4.28) or keep constant;

$k = k + 1;$

End while

$K(:, t_{j+1}) = P_g(1: N_x)$

End for

The polynomial variation thermal conductivity

$$K(u) = a_0 + a_1\left(\frac{u}{a_7}\right) + a_2\left(\frac{u}{a_7}\right)^2 + a_3\left(\frac{u}{a_7}\right)^3 + a_4\left(\frac{u}{a_7}\right)^4 + a_5\left(\frac{u}{a_7}\right)^5 + a_6\left(\frac{u}{a_7}\right)^6 \qquad (5.34)$$

defined in [76] is used as the reference solution in the numerical tests. Here $a_0 = a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = 1.0$ and $a_7 = 50.0$. This slab material has an initial temperature $u_0 = 1.0$. When $t > 0$, the left and right ends of the slab are subjected to constant heat fluxes $q_1 = 20$ and $q_2 = -14$, respectively. The total simulation time is $t_f = 0.8$. The spatial mesh size and temporal step size are $\Delta x = 0.05$ and $\Delta t = 0.05$. To illustrate the viability and stability of the QPSO and SQPSO in predicting temperature dependent thermal conductivity, no prior information of the functional form is assumed.

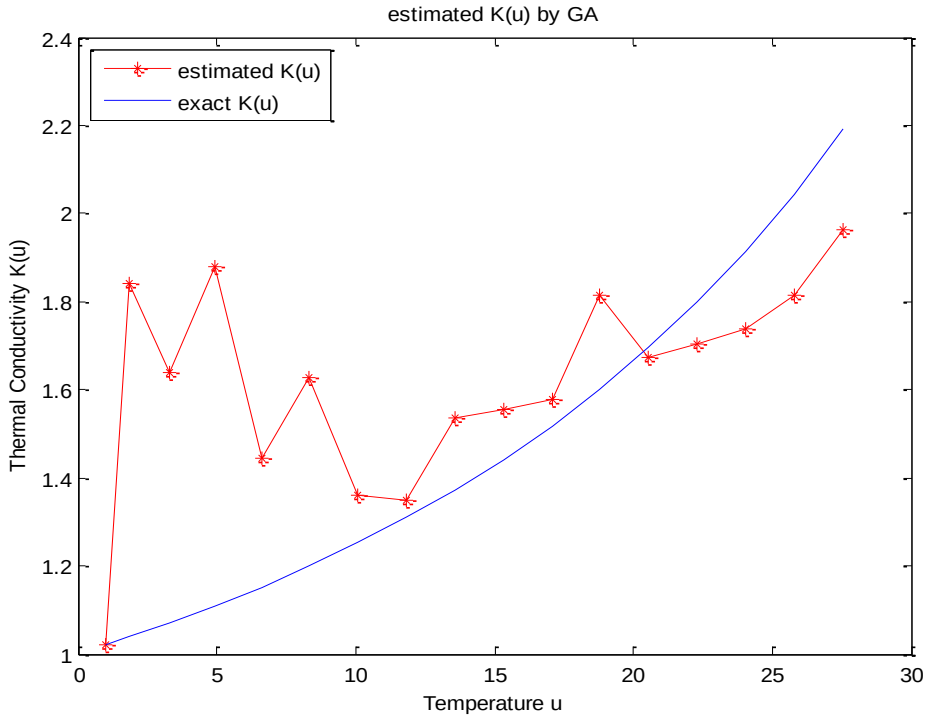
The parameters in SPSO are set as $M = 20$, $D = 21$, $c_1 = c_2 = 2.05$, $\chi = 0.72984$, $k_{max} = 3000$. In GA, crossover probability and mutation probability are $P_c = 0.6$, $P_m = 0.05$. In order to compare the results under the same conditions, the same values are used for those common parameters in QPSO, SQPSO, SPSO and GA. The value of α was set as that in section 4.4.2. Different from gradient-based methods, in the stochastic methods used here, GA, SPSO, QPSO and SQPSO all the particles or chromosomes are initialized randomly instead of being given a specific initial value.

The estimated results by GA, SPSO, QPSO and SQPSO are shown in Table 5.22, and Figure 5.33-Figure 5.35. The meaning of QPSO1, QPSO2, SQPSO1 and SQPSO2 are QPSO with linearly decreasing α, QPSO with constant α, SQPSO with linearly decreasing α and SQPSO with constant α respectively. Figure 5.33 shows the comparison of the estimated results using exact measurements. Note that the estimated thermal conductivity obtained by QPSO1 and QPSO2 are almost identical to the exact value, a little better than SQPSO1 and SQPSO2. With exact measurements, QPSO with global topology model converges fast and the converged results are good. The comparison of convergence history with exact

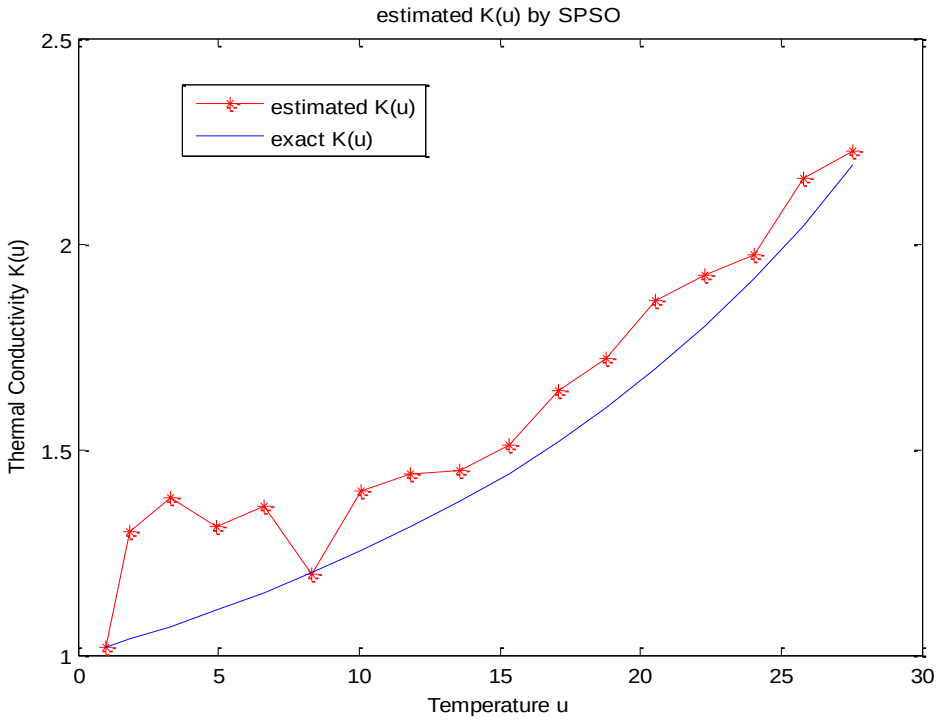
measurements is illustrated in Figure 5.36, in which, QPSO2 obviously outperforms others in both convergence rate and quality of converged solution. The estimated results with noisy measurements are shown in Figure 5.34-Figure 5.35, from which, it can be noted that the result obtained by SQPSO1 is better than others with noise level $\varepsilon = 0.01$, while for noise level $\varepsilon = 0.02$, QPSO2 outperforms others. Actually, as in [22], despite the advantages of a local topology, it is important to note that it should not always be considered to be the optimal choice in all situations. Even on some very complex multimodal problems a gbest model swarm can deliver performance competitive with the lbest model, given proper circumstances.

Table 5.22: Estimated results of the test Example

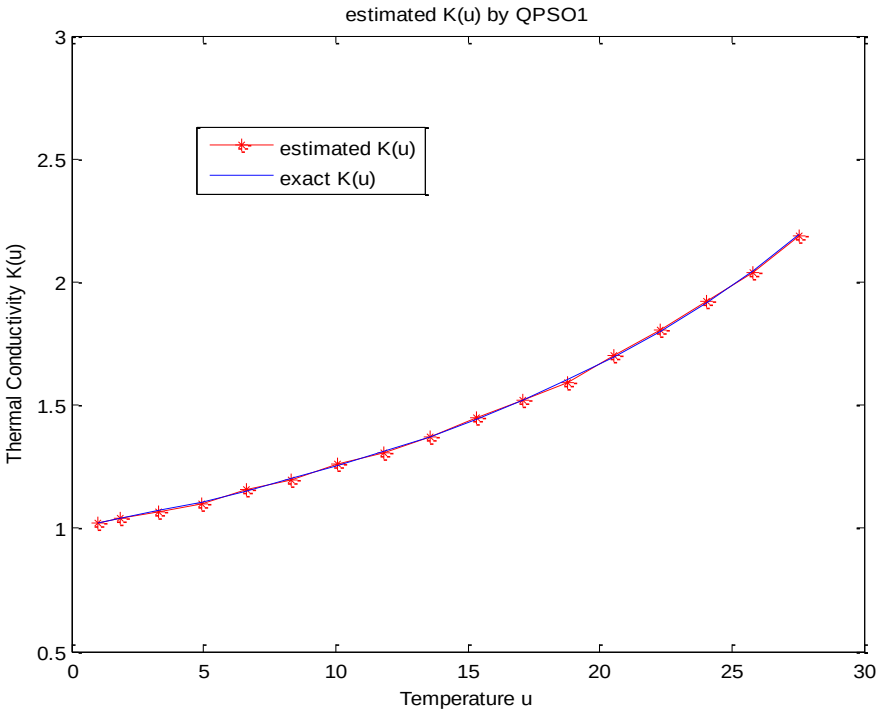
Methods		Average Error K_{error} (Objective Function Value $J[K]$)		
		$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$
GA		1.18E-02 (0.18)	1.23E-02 (0.31)	1.36E-02 (0.35)
SPSO		5.73E-03 (5.09E-02)	6.66E-03 (0.11)	9.18E-03 (0.11)
QPSO	$\alpha \sim (1.0 \to 0.5)$	2.41E-04 (3.24E-05)	2.59E-03 (1.39E-02)	4.22E-03 (5.88E-02)
	$\alpha = 0.75$	1.73E-04 (1.69E-05)	2.66E-03 (1.35E-02)	4.64E-03 (7.64E-02)
SQPSO	$\alpha \sim (1.0 \to 0.5)$	3.89E-04 (7.05E-05)	2.43E-03 (1.49E-02)	4.31E-03 (7.17E-02)
	$\alpha = 0.75$	5.29E-04 (1.60E-04)	2.94E-03 (1.73E-02)	5.77E-03 (6.40E-02)


estimated K(u) by GA
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
5
10
15
20
25
30

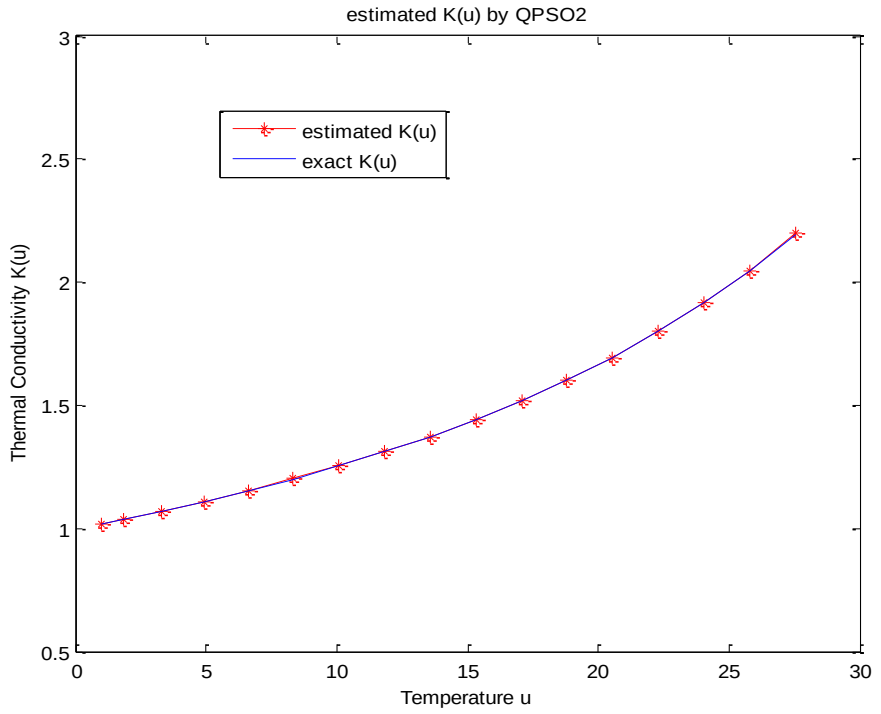
(a)


estimated K(u) by SPSO
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
2.5
2
1.5
1
0
5
10
15
20
25
30

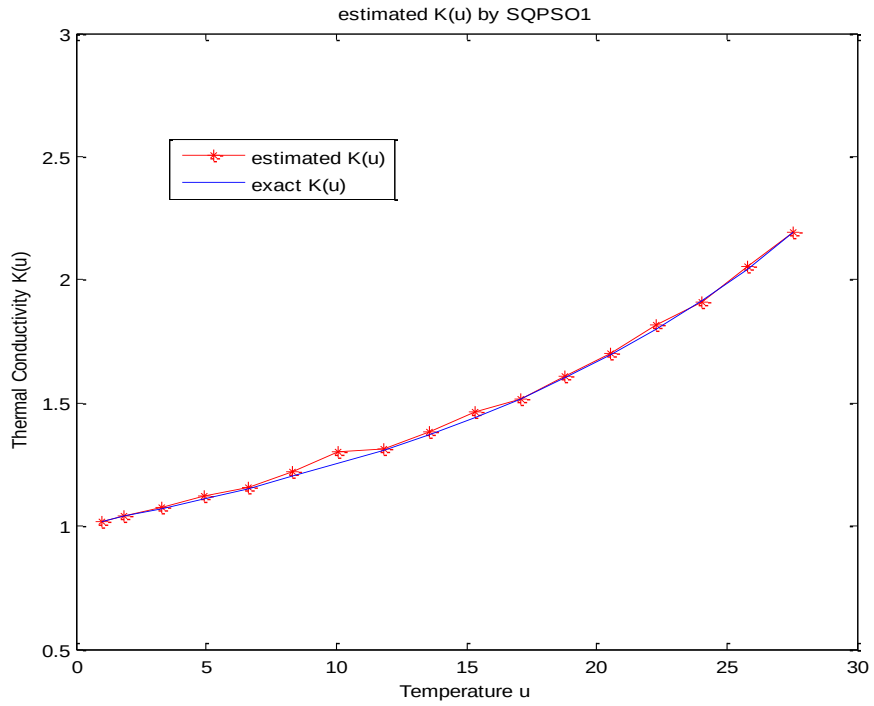
(b)


estimated K(u) by QPSO1
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30

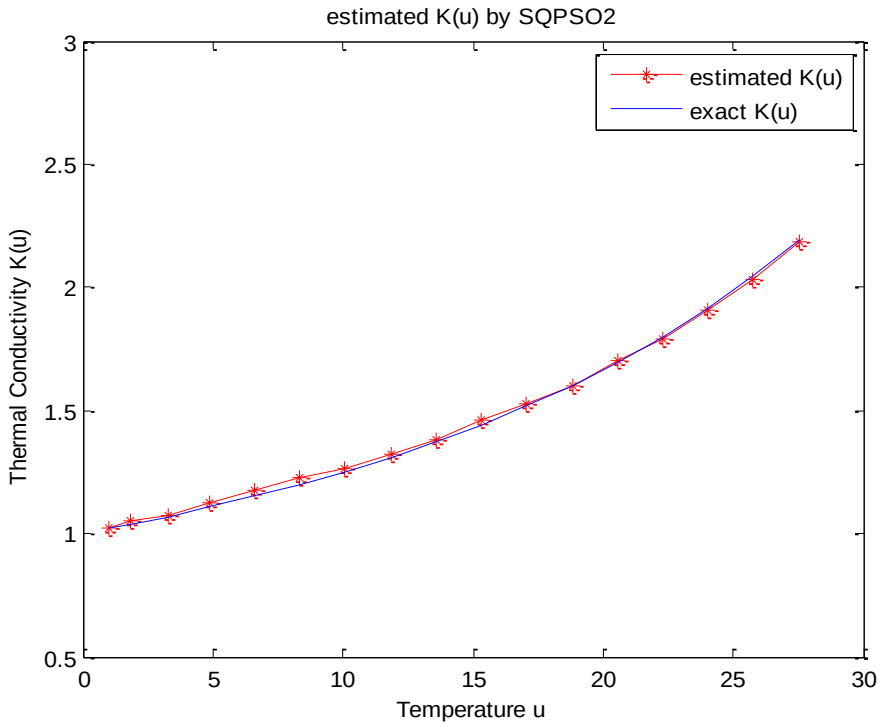
(c)


estimated K(u) by QPSO2
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30

(d)

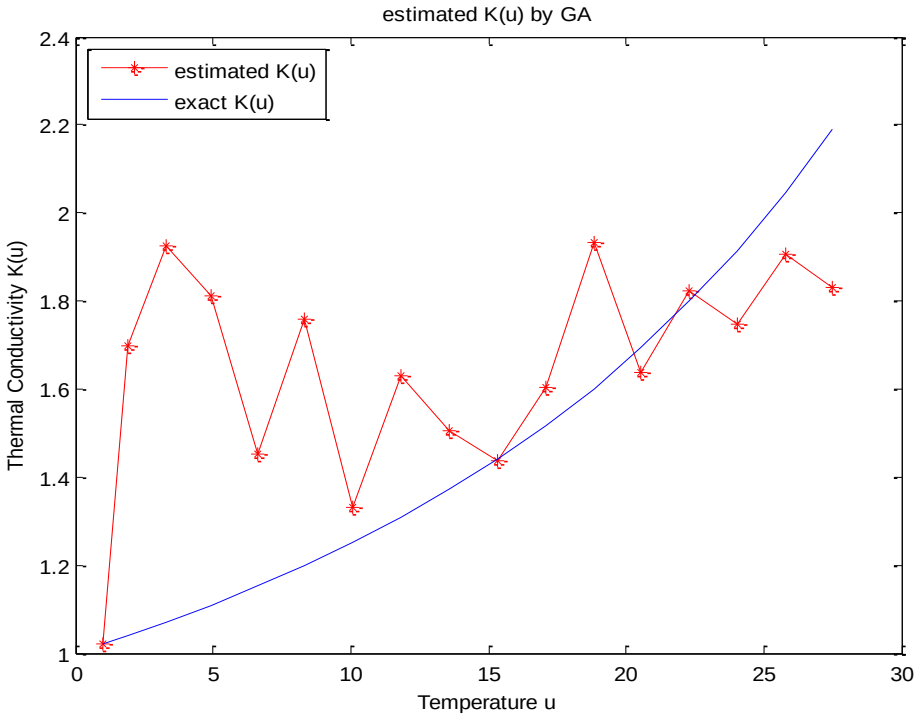


(e)

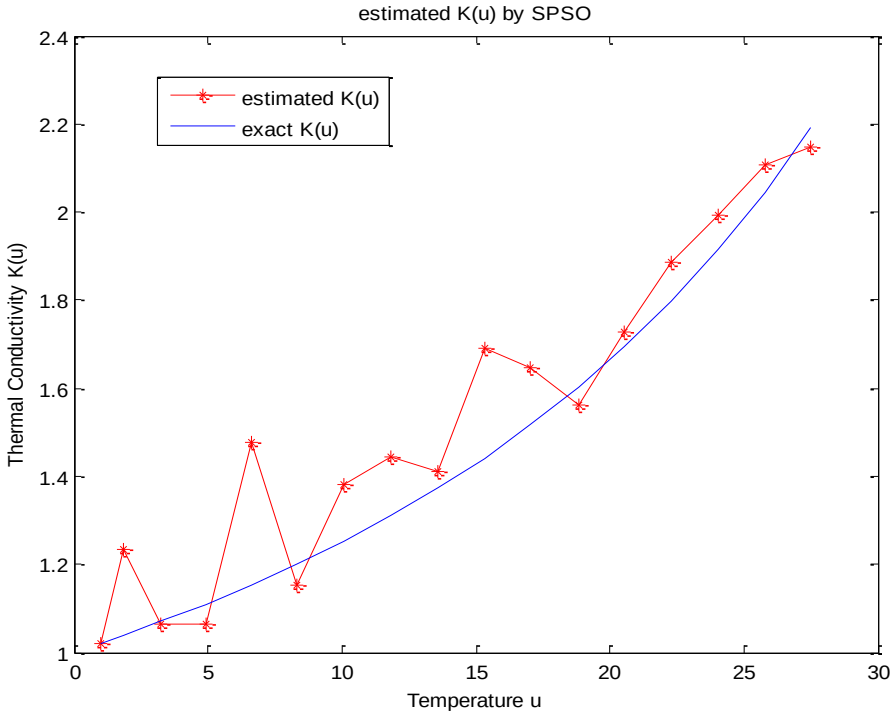


(f)

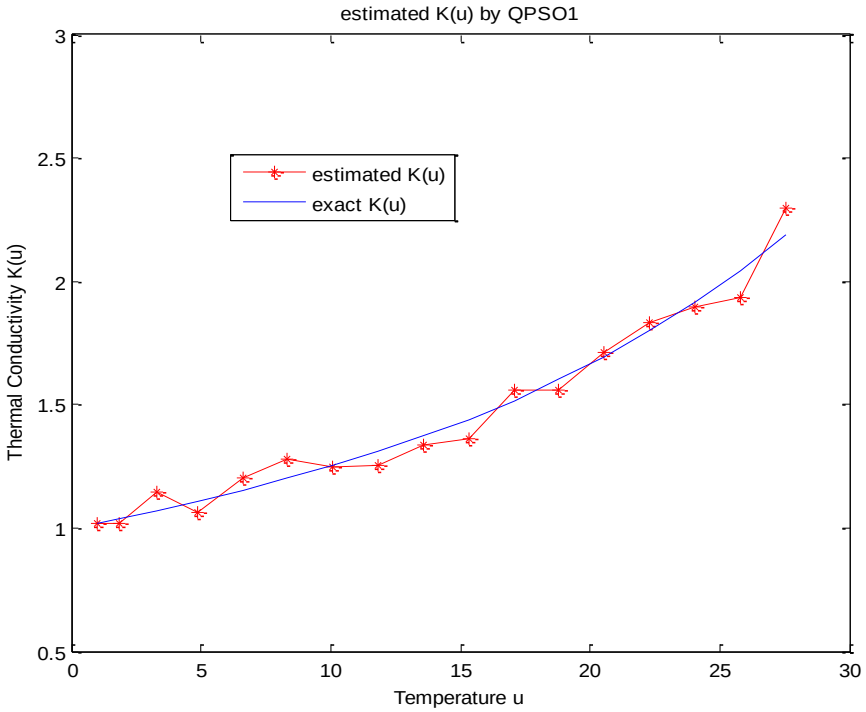
Figure 5.33: Estimated thermal conductivity using exact measurements.


estimated K(u) by GA
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
5
10
15
20
25
30

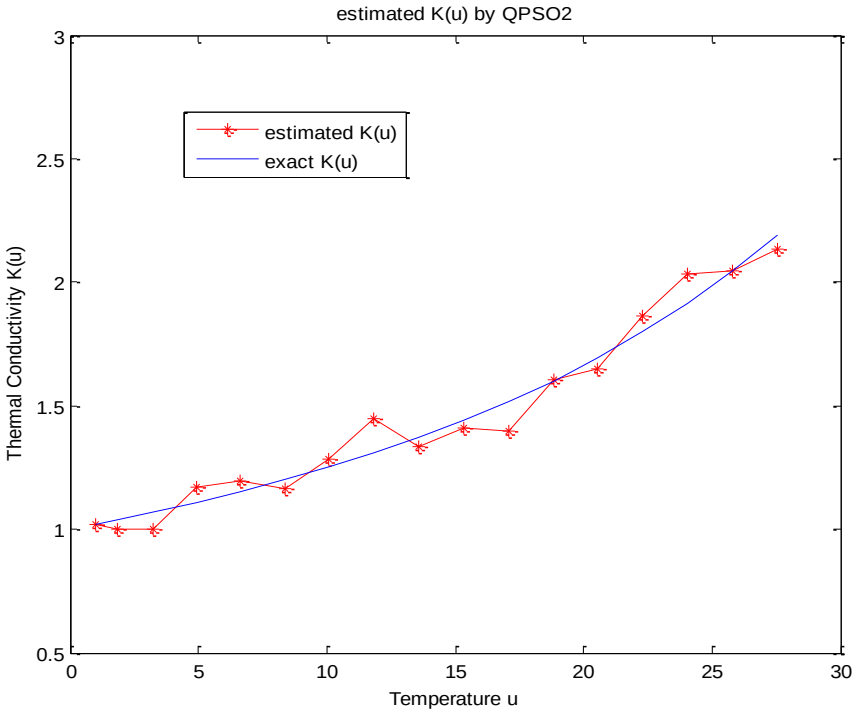
(a)


estimated K(u) by SPSO
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
5
10
15
20
25
30

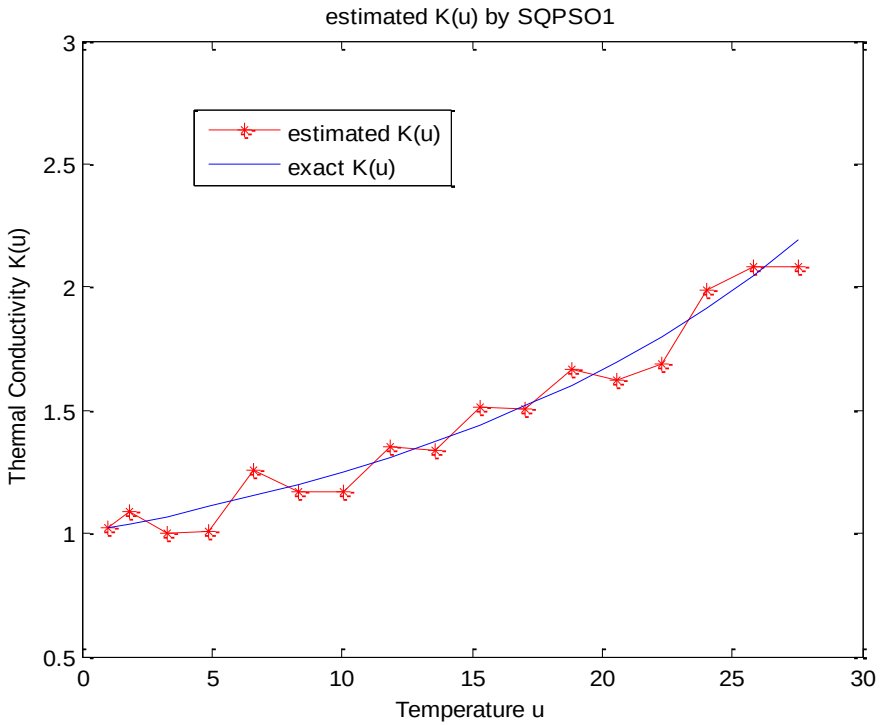
(b)


estimated K(u) by QPSO1
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30

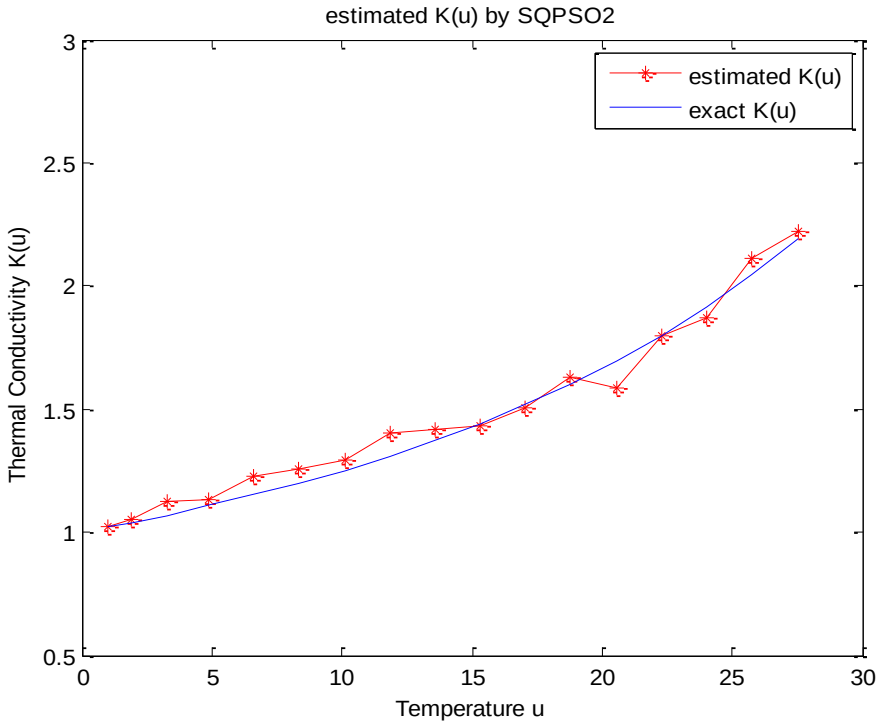
(c)


estimated K(u) by QPSO2
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30

(d)

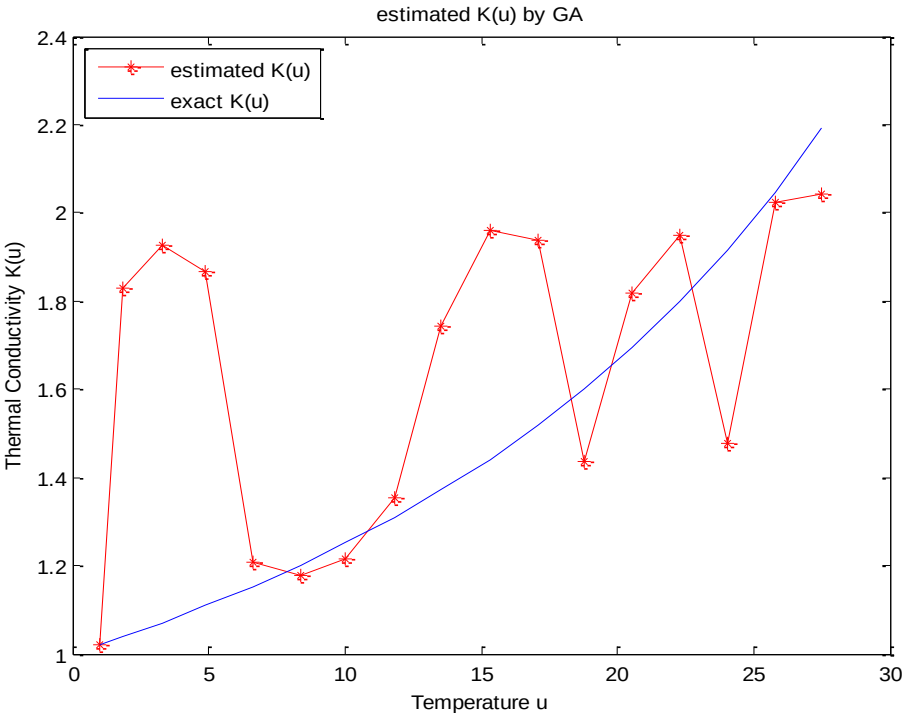


(e)

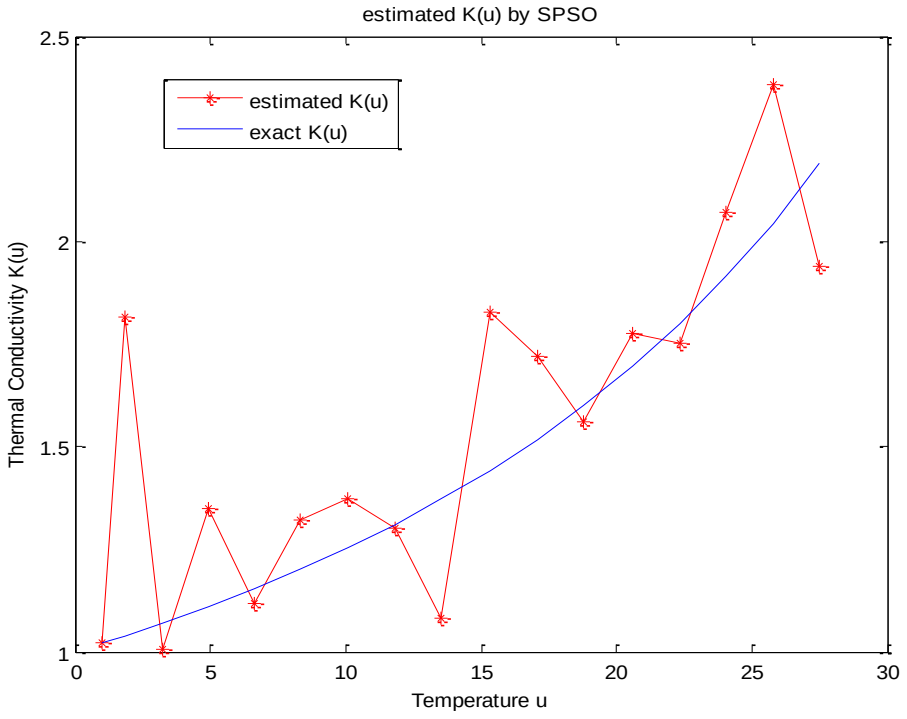


(f)

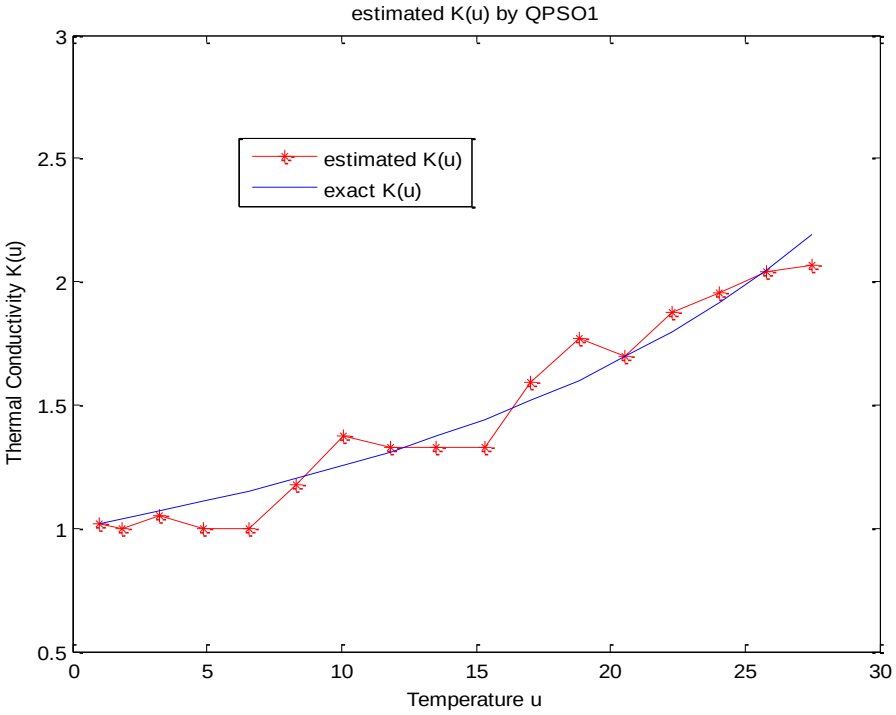
Figure 5.34: Estimated thermal conductivity using measurements with noise level $\varepsilon = 0.01$.


estimated K(u) by GA
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
5
10
15
20
25
30

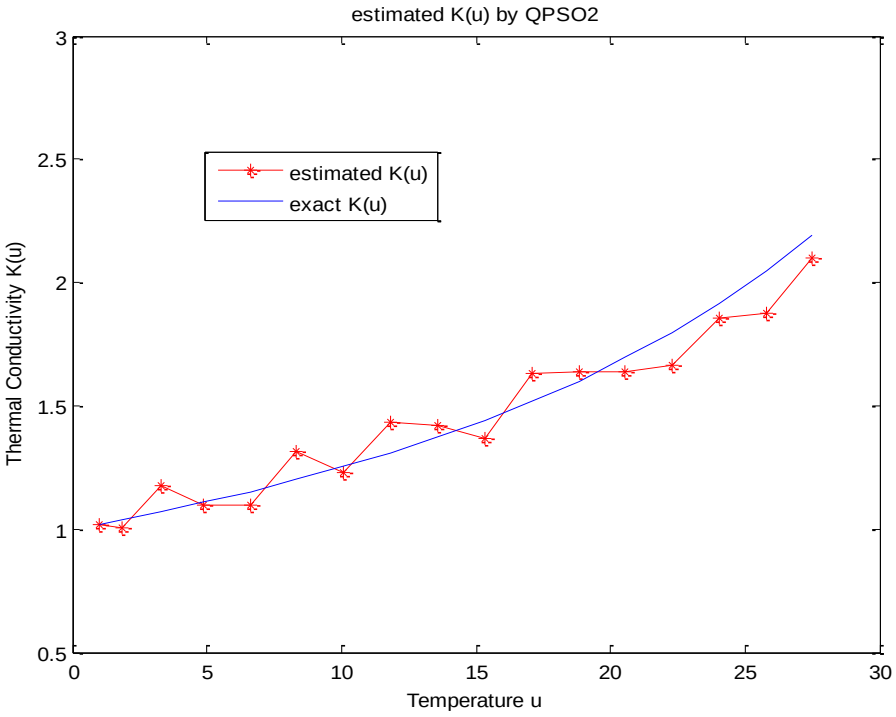
(a)


estimated K(u) by SPSO
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
2.5
2
1.5
1
0
5
10
15
20
25
30

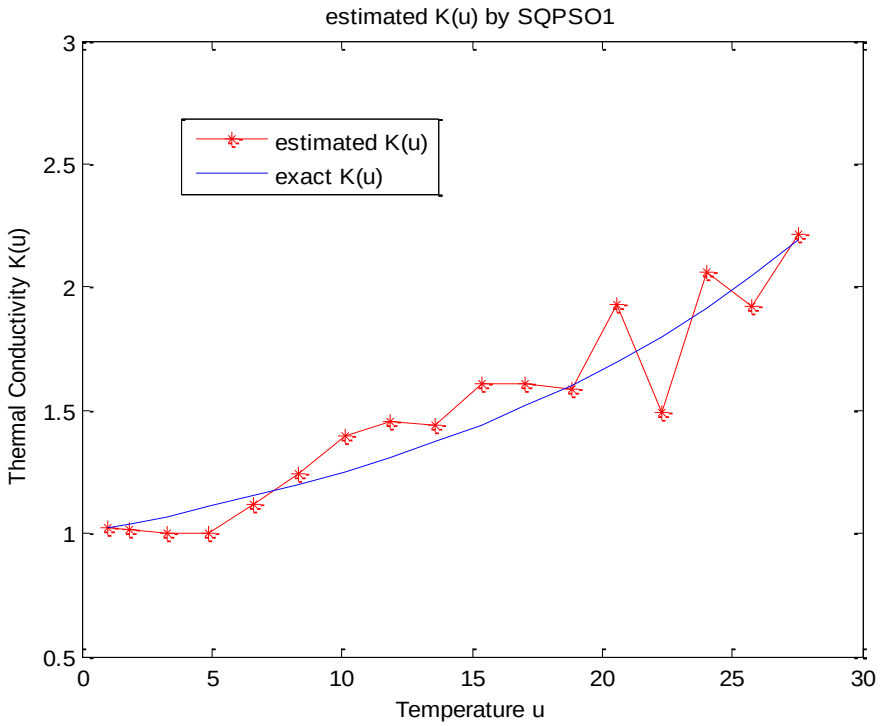
(b)


estimated K(u) by QPSO1
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30

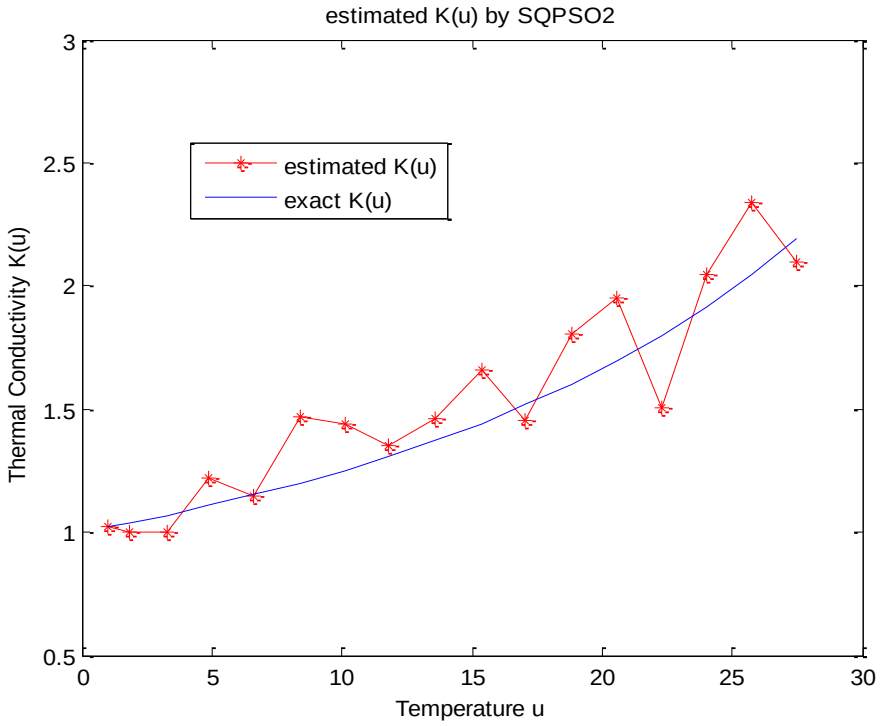
(c)


estimated K(u) by QPSO2
estimated K(u)
exact K(u)
Thermal Conductivity K(u)
Temperature u
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30

(d)



(e)



(f)

Figure 5.35: Estimated thermal conductivity using measurements with noise level $\varepsilon = 0.02$.

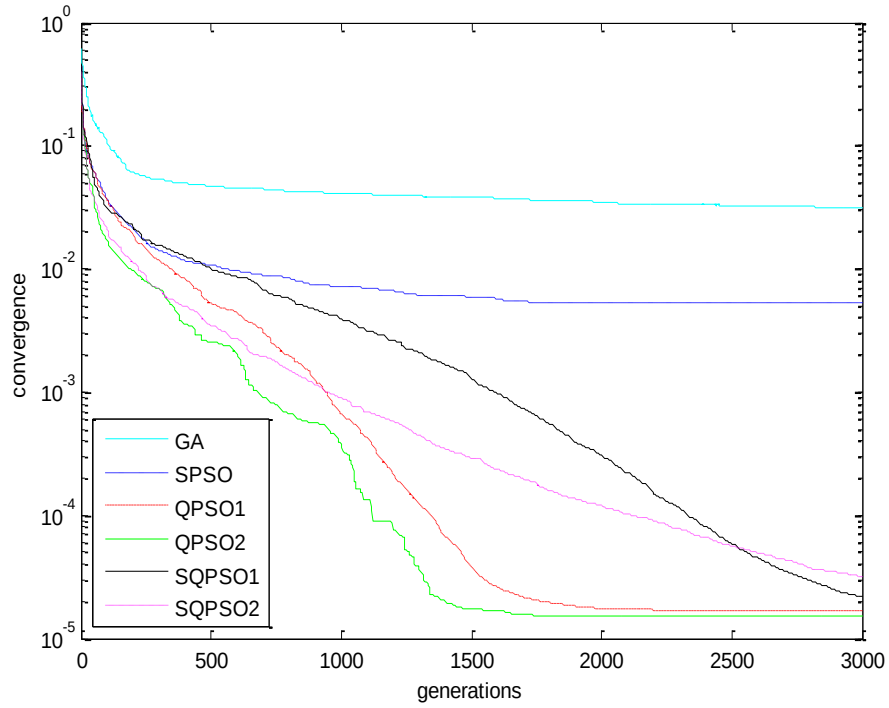


Figure 5.36: Convergence history of the algorithms for estimating thermal conductivity with exact measurements.

5.6.3 Conclusion

In this section, QPSO with ring topology (SQPSO) with both linearly decreasing and constant contraction-expansion coefficient was used estimate the temperature-dependent thermal conductivity with no priori information of the functional form. The numerical results demonstrate that SQPSO shows competitive performance compared to QPSO when measurements are without noise. However, SQPSO shows its superiority in handling noisy measurements.

5.7 Identification of Boundary Shapes in Steady Heat Conduction Problems

Shape identification problems involve estimating the shape of a part of the boundary of the domain, which arise in many branches of science and engineering [138]. They also have important applications in industrial design and maintainance such as optimal design of aircraft, ships and engines, and non-destructive evaluation of structured behaviour [128]. Such identification problems find applications in material loss defect determination [129], electromagnetic crack detection [131] and corrosion detection [132], [135]. These problems

are nonlinear and ill-posed, which require special techniques in order to accurately and stably solve them numerically. In [95], Nachaoui estimated the boundary shape using the CGM. Mera et al. used GA to solve the boundary detection problem in [96], which requires the information of the functional form of the boundary shape. In [97] and [98], the inclusion detection problems are investigated, in which the singularity of the BEM and the expensive computational cost are the most difficult points.

In this section, an inverse steady heat conduction problem of estimating the shape of a part of the boundary from the measured boundary temperature on the remaining known part is investigated. The hybrid method (HM2) proposed in section 4.6 is used in order to overcome the disadvantages of both CGM and QPSO.

5.7.1 Mathematical description

Consider a steady state heat conduction problem in a bounded domain Ω, depicted in Figure 5.37, where $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ is the boundary of the domain. A heat flux q(x) is imposed on the boundary Γ_1, while boundaries Γ_2 and Γ_4 are insulated. The boundary Γ_3 has the Dirichlet condition $u = u_0$ and the shape of this boundary $y = f(x)$ is unknown.

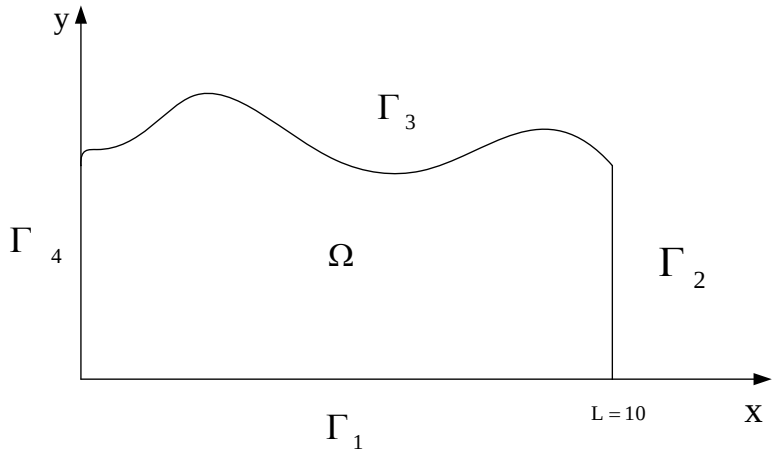


Figure 5.37: A two-dimensional steady state heat conduction problem.

For simplicity, the Laplace equation is treated in this section, i.e.

$$\begin{cases} \dfrac{\partial^2 u(x,y)}{\partial x^2} + \dfrac{\partial^2 u(x,y)}{\partial y^2} = 0, & \text{in } \Omega \quad \text{(a)} \\ \dfrac{\partial u(x,y)}{\partial y} = q(x), & \text{on } \Gamma_1 \quad \text{(b)} \\ \dfrac{\partial u(x,y)}{\partial x} = 0, & \text{on } \Gamma_2 \cup \Gamma_4 \quad \text{(c)} \\ u(x,y) = u_0, & \text{on } \Gamma_3 \quad \text{(d)} \end{cases} \tag{5.35}$$

where $u(x, y)$ is the temperature distributed over the domain. In the formulation of direct problems, the geometry boundary shape is assumed to be known. Then the temperature distribution over the entire domain can be determined using BEM. In the inverse-geometry problems, the boundary shape Γ_3 is unknown and needs to be determined from the extra temperature measurements obtained on Γ_1.

Assume experimental measurements of temperature are available at a set of points on Γ_1, denoted as Y_i ($i = 1, 2, ..., m$, m is the number of survey points). In the numerical tests, simulated measurements which are computed by solving the direct problem Equations (5.39) with a predefined exact boundary shape, were used.

5.7.2 Numerical tests

The BEM [41], [134] is adopted to solve the direct problem, since the geometry of the system changes for every possible solution during the optimisation process. Further, BEM does not require any meshing of the domain but only needs the discretisation of the boundary, which reduces the modeling effort to a minimum. However, one disadvantage of BEM is that the resulting matrix is dense and non-symmetric. Numerical solution of a large dense system is expensive. Therefore, the hybrid method (HM2) proposed in section 4.6 was used to solve this steady IHCP of identifying the boundary shape. Initially, a smaller number of boundary elements were used in QPSO to ensure small computational time. The continuous function $f(x)$ was discretised for numerical computation and simulated by a particle. The position of a particle represents a candidate solution of the unknown boundary shape. The dimension D of the position is equal to the number of space steps N_x of $f(x)$. The objective fitness function $J[f]$ is defined as in Equation (5.1). At each generation k of finding the minimum of $J[f]$, a particle is defined as

$$X_i(k) = (X_{i1}(k), X_{i2}(k), ..., X_{ij}(k), ..., X_{iD}(k)) = (f(x_1), f(x_2), ..., f(x_j), ..., f(x_{N_x})), \quad (5.36)$$

where $D = N_x$ represents the dimension of the particle's position and number of space nodes required in the computation. Substituting $X_i(k)$ into Equation (5.39), the temperature $u(x, y)$ can be computed by solving the direct problem. Each feasible solution $X_i(k)$ is evaluated by computing the fitness function $J[f]$. At each generation, the positions of the particles are updated as according to Equation (4.31). A roughly estimated boundary shape is achieved after a pre-defined small number of generations.

The smooth estimated shape obtained by a spline interpolation on the rough estimation is used as the initial value for CGM, in which a large number of boundary elements are used to ensure the accuracy of the estimated solution. f_{error}, as defined in Equation (5.9), is used to evaluate the accuracy of the converged solution.

To illustrate the hybrid method in estimating the unknown boundary shape $f(x)$ from the knowledge of measurements obtained on the bottom boundary Γ_1, two test examples were considered. In the first one, the exact boundary and the exact solution $u(x, y)$ of the governing Equation (5.39) are known and the exact temperature on boundary Γ_1 is used as the desired measurement. In the second one, the exact boundary is given, the desired measurements on Γ_1 are obtained from solving Equation (5.39) using BEM.

For all numerical experiments described below, $L = 10$. In the first stage of QPSO estimating the boundary shape, the boundary is coarsely discretised into $N = 18$ boundary elements (Figure 5.38), which can largely reduce the computational time. While in the process of CGM, the number of boundary elements is $N = 220$ (Figure 5.39) to guarantee accuracy. The first order regularisation term is used in the following examples to ensure the smoothness of the estimated shape. And the L-curve method is applied to choose the regularisation parameter.

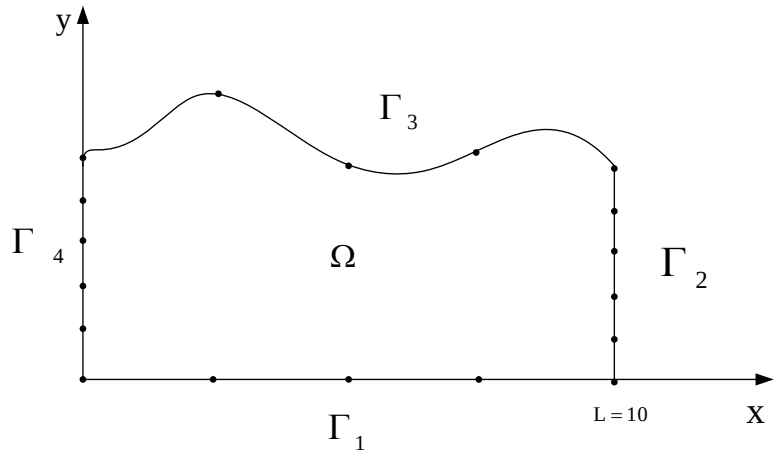


Figure 5.38: Coarse element discretisation with $N = 28$.

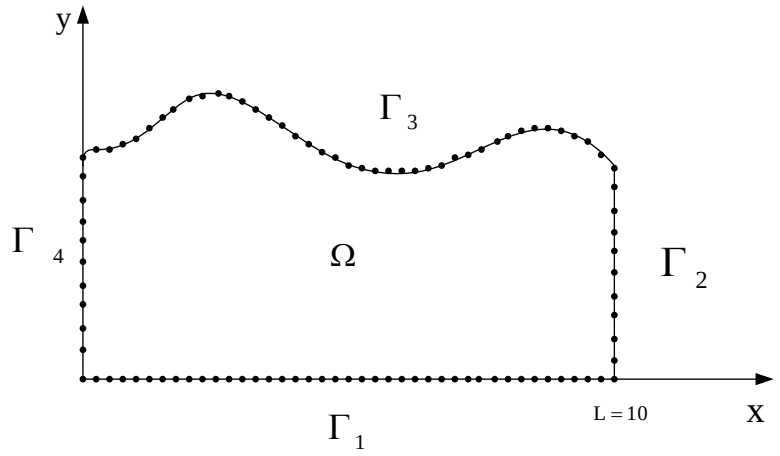


Figure 5.39: Dense element discretisation with $N = 220$.

Example 1. The temperature over the domain is given by $u(x, y) = y$, which implies that $q(x) = -1$, $u_0(x) = f(x)$ and $Y(x) = 0$ for all $x \in [0, L]$. The exact boundary shape Γ_3 is given by $f(x) = 1 + 0.4\exp\left(-(x - L/2)^2/3\right)$, $x \in [0, L]$.

From Figure 5.40, it can be observed that the final estimated shape is close enough to the exact boundary shape, which shows the viability and accuracy of the hybrid method in the boundary shape identification. Figure 5.41a) shows the convergence history of the QPSO, in which the objective function value $J[f]$ decrease quickly even within 6 iterations. Then the CGM produces a convergent optimal solution after 20 iterations with the initial value from QPSO (Figure 5.41b)).

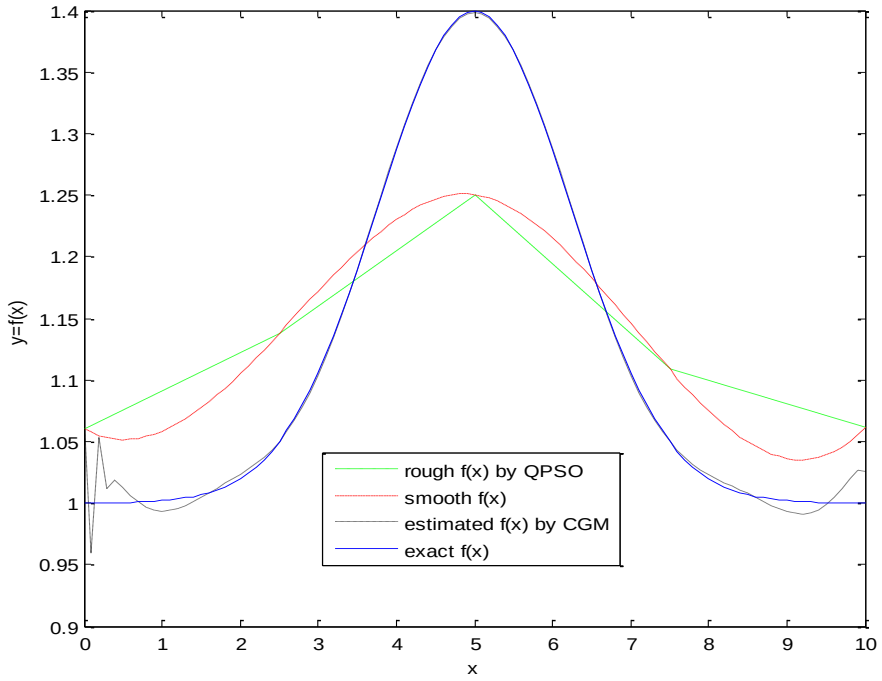
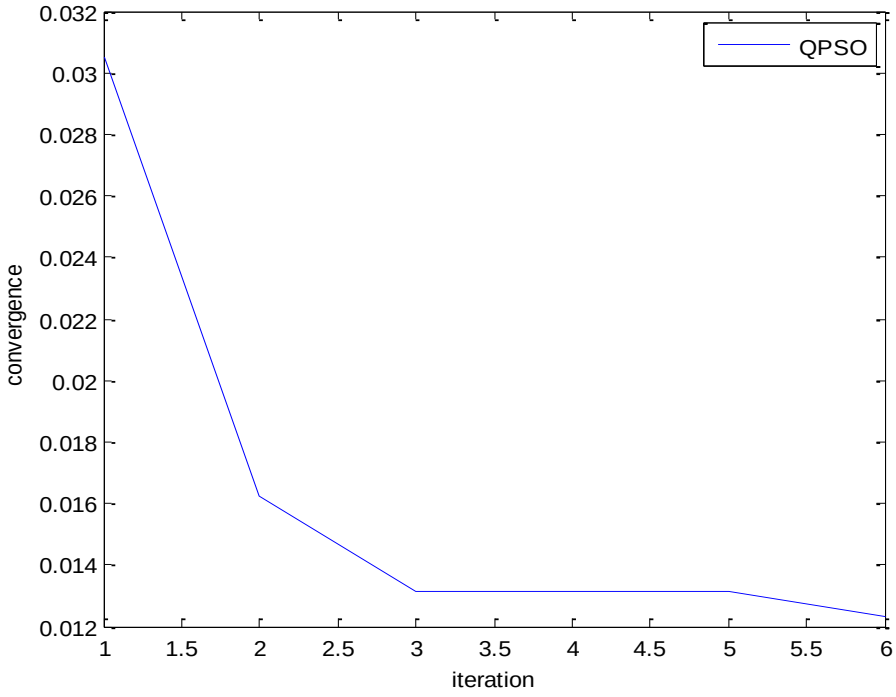
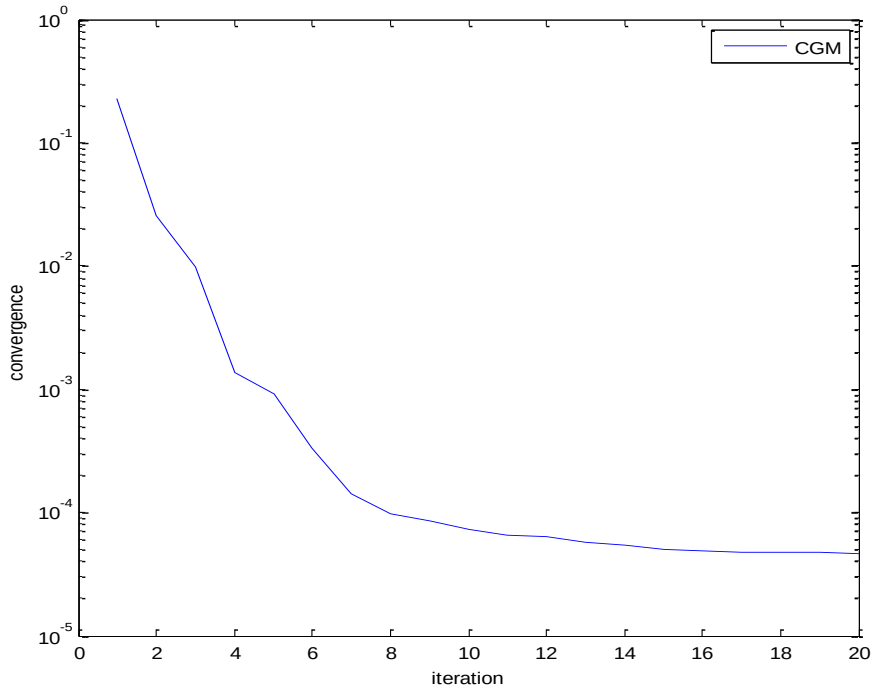


Figure 5.40: Estimated boundary shape in Example 1 by using HM2 with exact measurements.



(a)



(b)

Figure 5.41: Convergence history of the hybrid method (HM2).

Example 2. The temperature on boundary Γ_3 is $u_0(x)=100$, heat flux on boundary Γ_1 is $q(x)=20$. The exact boundary shape function Γ_3 is $f(x)=1.5+0.8\sin(0.2\pi x)$, $x\in[0,L]$.

The effect of the number of boundary elements on the accuracy of the estimated results is examined. Figure 5.42 and Figure 5.43 show the estimated boundary shape by using HM2 with the number of boundary elements $N=110$ and $N=220$. The computational CPU time, average error of the estimated $f(x)$ and the objective function value are listed in Table 5.23. Note that, more boundary elements produce better estimated results, but require large computational time.

.

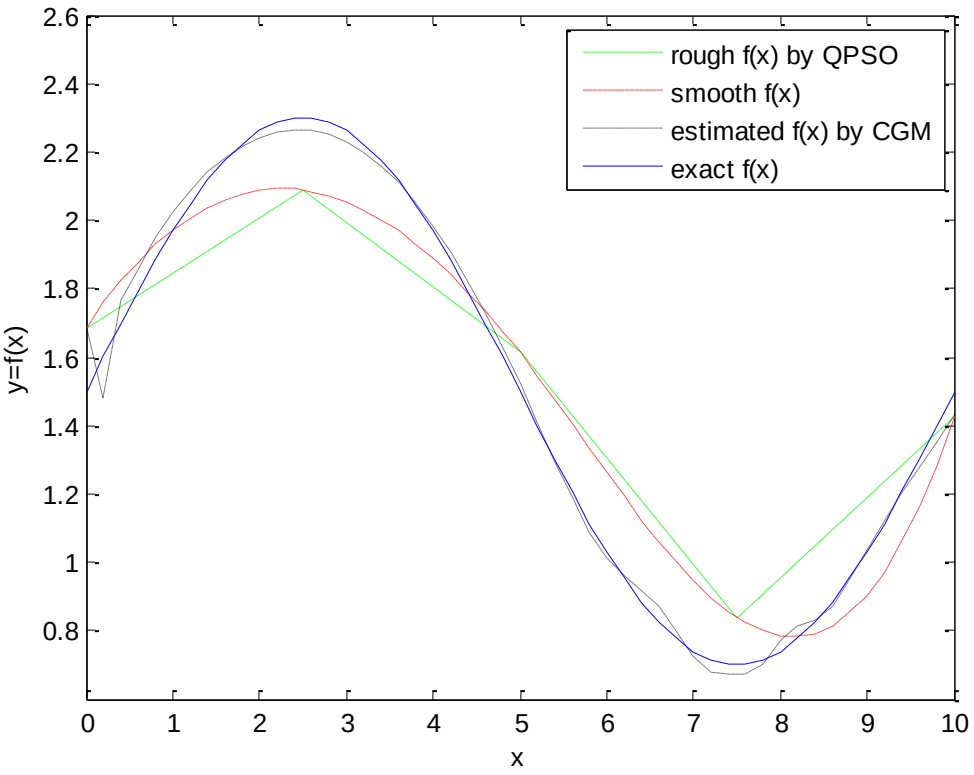


Figure 5.42: Estimated boundary shape in example 2 by using HM2 with $N = 110$.

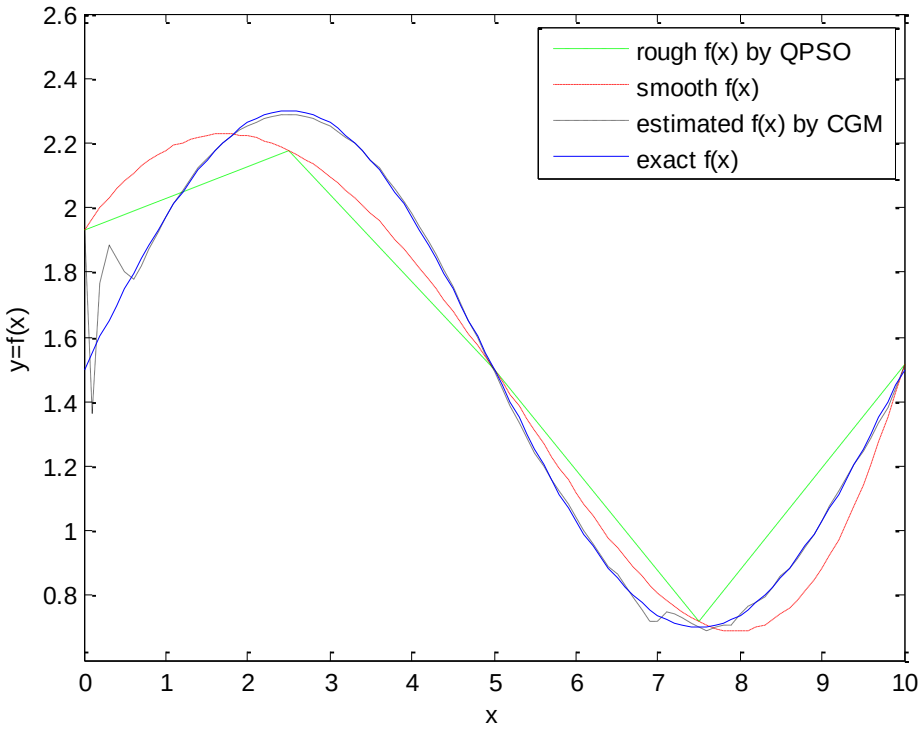
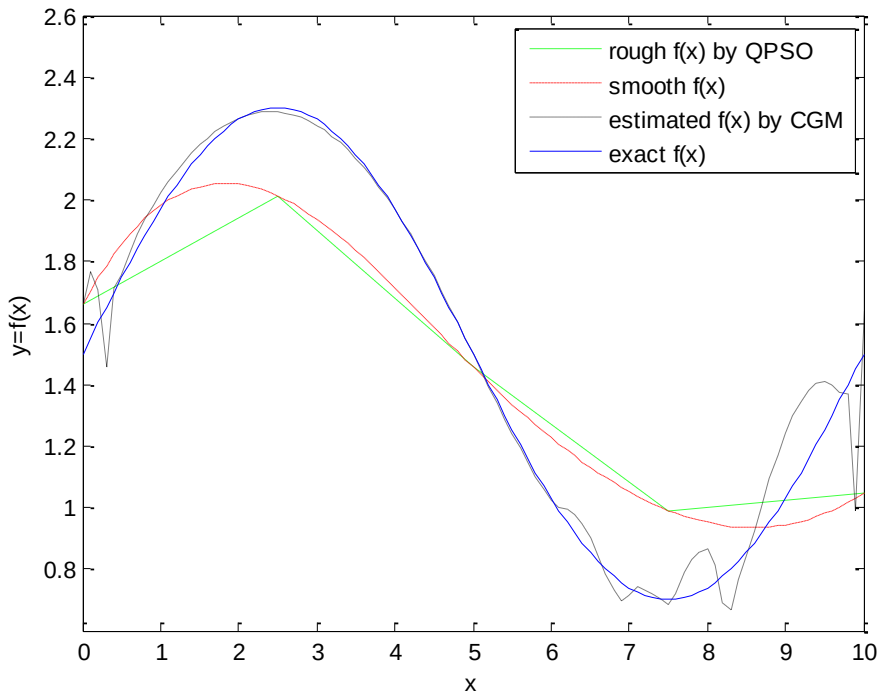


Figure 5.43: Estimated boundary shape in example 2 by using HM2 with $N = 220$.

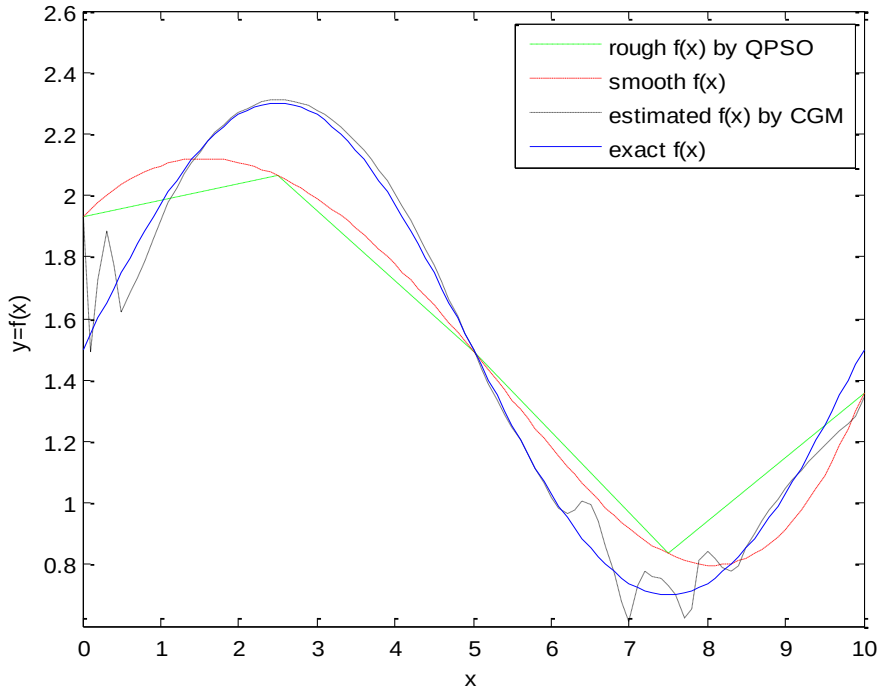
Table 5.23: Effect of number of elements on the results.

No. of Elements N	CPU time (s)	f_{error}	J[f]
110	612.38	9.90E-03	0.31
220	4520	3.10E-03	1.02E-02

Then the effect of the measurement errors on the inverse solutions was also discussed. Two different noise levels of the temperature measurement error $\varepsilon = 0.01$ and $\varepsilon = 0.05$ were studied. The estimated boundary shapes of the two test cases are presented in Figure 5.44. Note that the estimated results are seriously affected by the error from the temperature measurements. The most important thing from the figures is the oscillations in the estimated results near $x = 0$ and $x = L$, the cause of which lies in the CGM. The gradient of the objective function $\nabla J = \frac{\partial u}{\partial y}\frac{\partial \lambda}{\partial y}$, as defined in section 3.3, is always zero at $x = 0$ and $x = L$, because of $\frac{\partial u}{\partial y} = 0$ and $\frac{\partial \lambda}{\partial y} = 0$ at both Γ_2 and Γ_4. This is the inherent problem of CGM, unless the exact initial value of $f(x)$ is given.



(a)



(b)

Figure 5.44: Estimated boundary shape in Example 2 by usingthe hybrid method with noise measurement. (a) $\varepsilon = 0.01$ (b) $\varepsilon = 0.05$.

To overcome this stagnated problem of CGM and reduce the computational CPU time of QPSO in estimating the boundary shape, the parallel QPSO proposed in section 4.5 running on Heracles system was used. Table 5.24 shows the comparison of the performance between synchronous and asynchronous parallel QPSO for example 1. Note that the average error of the estimated shape obtained by synchronous QPSO (both sequential and parallel) is better than that obtained by asynchronous QPSO. However, the computational CPU time required in synchronous QPSO is much longer than that in asynchronous QPSO. It seems strange that the efficiency of the asynchronous parallel QPSO exceeds 100%. That is because the process of parallel asynchronous QPSO is not exactly the same as the sequential asynchronous QPSO (Figure 4.18 and Figure 4.19). The parallel asynchronous QPSO makes full use of the computational resources without idle waiting, which saves much computational CPU time.

Table 5.25 shows the average error of the estimated boundary shape obtained by the parallel static subpopulation QPSO, which is not as good as that obtained by parallel master-slave QPSO. The result becomes worse when the number of processors increases. The reason

is because the small number of particles in the subpopulation is not enough to produce good result in a large problem space, e.g. 4 particles in search space with dimensional size 50. Figure 5.46 shows that the estimated boundary shape is very close to the exact shape and much better than that estimated by CGM, especially at the both ends of the curve.

Table 5.24: Performance of parallel synchronous and asynchronous QPSO.

Number of Processors	f_{error}		Computational Time (s)		Speedup		Efficiency	
	Syn	Asyn	Syn	Asyn	Syn	Asyn	Syn	Asyn
1	4.47E-04	5.53E-04	690.98	650.18	1	1	100%	100%
4	4.56E-04	4.78E-04	222.75	80.63	3.10	8.06	77.5%	201.59%
8	4.22E-04	4.37E-04	105.76	53.44	6.53	12.17	81.6%	152.08%
16	3.73E-04	4.32E-04	53.97	18.60	12.80	34.96	80.0%	218.47%

Table 5.25: Performance of static subpopulation QPSO.

No. of Processors	f_{error}	Computational Time (s)	Speedup	Efficiency
1	6.75E-04	653.99	1	100%
4	6.76E-04	164.62	3.97	99.3%
8	8.75E-04	82.82	7.89	98.7%
16	1.79E-03	41.88	15.62	97.6%

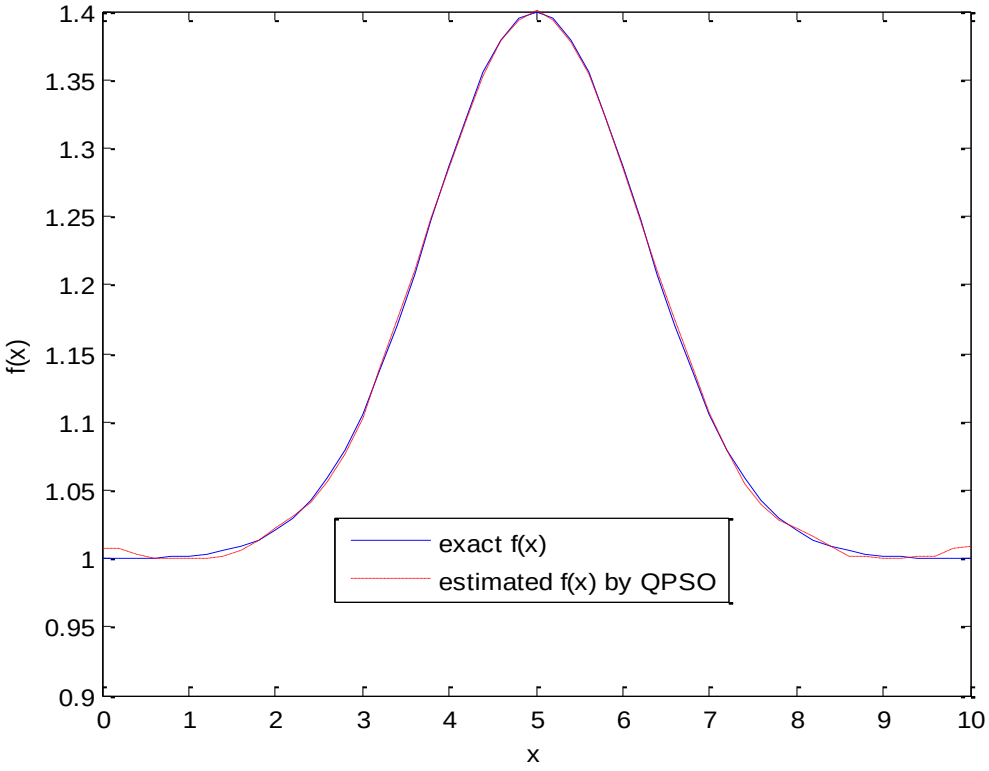


Figure 5.45: Estimated boundary shape in Example 1 by using parallel QPSO.

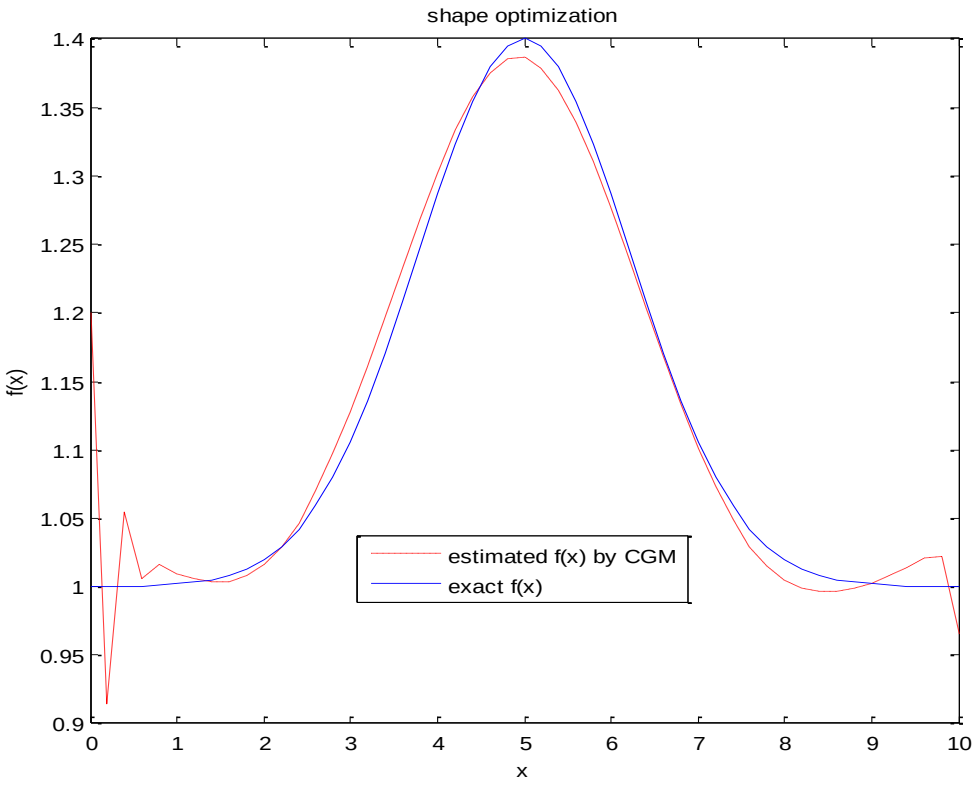


Figure 5.46: Estimated boundary shape in Example 1 by using CGM.

5.7.3 Conclusion

In this section, the geometric shape of the boundary in heat conduction problems is identified using the hybrid method (QPSO + CGM) together with BEM, which is based on the minimisation of the squared errors between the measured and calculated temperatures at some observation points. Tikhonov regularisation method with first-order regularisation term is used to stabilise the solution. Two numerical examples are tested, which demonstrate the viability and effectiveness of the hybrid method in solving the problem of identifying the geometric boundary shape. Besides, the hybrid method avoids the careful choice of initial value as in gradient-based method and reduces the computational time as in the QPSO, which is more adaptive and much easier to use.

To overcome the stagnated problem of CGM to identify the values near both ends of the boundary and avoid expensive computational CPU time in QPSO, the parallel QPSO is used.

5.8 Simultaneous Estimation of Two unknown Quantities in Heat Conduction Problems

The determination of the thermal properties from a measured temperature profile is a coefficient inverse problem of heat conduction [1], [3]. In the literature, several methods have been developed to address this problem, see the discussion in chapter 1.

In this section, the QPSO and the modified QPSO with Gaussian mutation are applied to simultaneously estimate the temperature dependent thermal conductivity and heat capacity with no prior information about the functional form.

5.8.1 Mathematical description

Consider a one-dimensional homogeneous slab as shown in Figure 5.47, with thickness L and initial temperature u_0. Two constant heat fluxes q_1 and q_2 are imposed on the both boundaries.

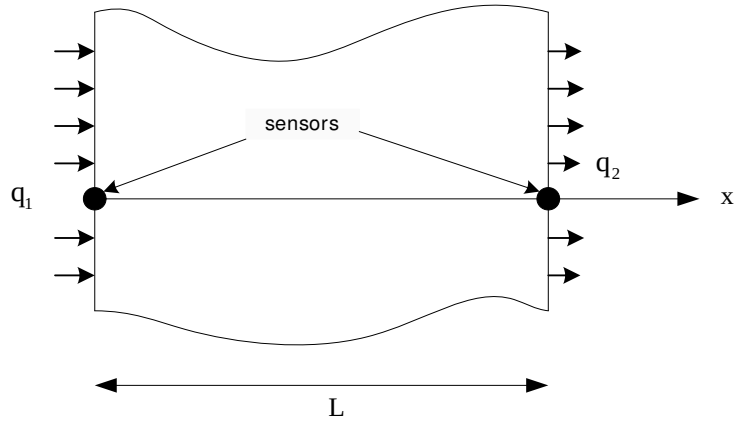


Figure 5.47: One-dimensional heated slab.

For the case of temperature dependent thermal conductivity $K(u)$ and heat capacity per unit volume $C(u)$ and density ρ, the heat conduction process is governed by

$$\begin{cases} \rho C(u)\dfrac{\partial u(x,t)}{\partial t} = \dfrac{\partial}{\partial x}[K(u)\dfrac{\partial u(x,t)}{\partial x}],\ 0 < x < L,\ 0<t \le t_f & \text{(a)} \\ -K(u)\left.\dfrac{\partial u}{\partial x}\right|_{x=0} = q_1, \quad 0<t \le t_f & \text{(b)} \\ K(u)\left.\dfrac{\partial u}{\partial x}\right|_{x=L} = q_2, \quad 0<t \le t_f & \text{(c)} \\ u(x,0) = u_0, \quad 0 \le x \le L & \text{(d)} \end{cases} \tag{5.37}$$

For simplicity, $\rho = 1$ and $L = 1$, which is the same as using non-dimensional data.

The task is to find $C(u)$ and $K(u)$, with which the temperature $u(x_i,t)$ ($i = 1,2$) computed from Equation (5.41) is as close to measured temperature $Y(x_i,t)$ as possible. The nonlinear least squares method is used as

$$J[C(u),K(u)] = \sum_{i=1}^{2}\int_{t=0}^{t_f}\left(u(x_i,t) - Y(x_i,t)\right)^2 dt. \tag{5.38}$$

When generating simulated temperature measurements $Y(x,t)$ with a predefined $C(u)$ and $K(u)$, the nonlinear direct problem defined by Equation (5.41) is required to be solved. An iterative technique is needed in solving the problem in conjunction with an implicit finite difference method.

As the temperature $u(x,t)$ is approaching its converged result by using an iterative technique under some specified initial and boundary conditions, the values of K and C at

any time and position (x,t) are fixed, because temperature $u(x,t)$ is known and fixed at any (x,t). Under this situation, Equation (5.41a) is discretised as

$$\rho C_i^{j+1}\frac{u_i^{j+1}-u_i^{j}}{\Delta t}=\frac{K_{i+1}^{j+1}-K_{i-1}^{j+1}}{2\Delta x}\frac{u_{i+1}^{j+1}-u_{i-1}^{j+1}}{2\Delta x}+K_i^{j+1}\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{(\Delta x)^2} \tag{5.39}$$

where u_i^j, C_i^j and K_i^j are temperature, heat capacity and thermal conductivity at the j^{th} time step $j=1,2,...,N_t$, along the i^{th} grid point $i=1,2,...,N_x$. For the boundaries, the second order discretisation Equations (5.28) and (5.29) are used.

The average error values, which intend to evaluate the both of the estimated heat capacity and thermal conductivity, are defined as Equation (5.34) and

$$C_{error}=\frac{1}{N_x N_t}\sqrt{\sum_{i=1}^{N_t}\sum_{j=1}^{N_x}(C_i^j-C_i^j)^2}\ , \tag{5.40}$$

where K_i^j and C_i^j are the estimated thermal conductivity and heat capacity, $\tilde{K}_i^j$ and C_i^j are the exact thermal conductivity and heat capacity.

The Tikhonov regularisation method as described in section 2.3 is used to address the ill-posedness of the inverse problem and stabilise the solution. The objective function becomes

$$\begin{aligned}J[C(x,t),K(x,t)]=&\sum_{i=1}^{2}\int_{t=0}^{t_f}\left(u(x_i,t)-Y(x_i,t)\right)^2 dt\\&+\lambda_1^2\left\|LK(x,t)\right\|^2+\lambda_2^2\left\|LC(x,t)\right\|^2\end{aligned}\ , \tag{5.41}$$

where λ_1 and λ_2 are the regularisation parameters.

In the numerical solution of direct problem, $u(x,t_{j+1})$ is computed from $u(x,t_j)$, therefore in the inverse problem, $C(u)$ and $K(u)$ may be computed step by step from t_0 to t_{N_t}. Then for $C(x,t_j)$ and $K(x,t_j)$, the objective function becomes

$$\begin{aligned}J[C(x,t_j),K(x,t_j)]=&\sum_{i=1}^{2}\int_{t=t_{j-1}}^{t_j}\left(u(x_i,t_j)-Y(x_i,t_j)\right)^2 dt\\&+\lambda_1^2\left\|LC(x,t_j)\right\|^2+\lambda_2^2\left\|LK(x,t_j)\right\|^2\end{aligned}\ . \tag{5.42}$$

The regularisation operator is the same as $\|Lv(x)\|$, which becomes

$$\min J[C(x,t_j),K(x,t_j)] = \sum_{i=1}^{2}\int_{t=t_{j-1}}^{t_j}\left(u(x_i,t_j)-Y(x_i,t_j)\right)^2 dt + \lambda_1^2\sum_{i=1}^{N_x}\left(C(x_i,t_j)\right)^2 + \lambda_2^2\sum_{i=1}^{N_x}\left(K(x_i,t_j)\right)^2 \quad , \tag{5.43}$$

for the zeroth-order regularisation and

$$\min J[C(x,t_j),K(x,t_j)] = \sum_{i=1}^{2}\int_{t=t_{j-1}}^{t_j}\left(u(x_i,t_j)-Y(x_i,t_j)\right)^2 dt + \lambda_1\sum_{i=1}^{N_x}\left(C'(x_i,t_j)\right)^2 + \lambda_2\sum_{i=1}^{N_x}\left(K'(x_i,t_j)\right)^2 \quad , \tag{5.44}$$

for the first-order regularisation.

5.8.2 Numerical tests

In this section, the modified QPSO with Gaussian mutation proposed in section 4.4.3 is used to solve the inverse problem of simultaneously estimating temperature dependent thermal conductivity and heat capacity, in which, every particle $X_i(k)$ is treated as a candidate solution of $C(x,t_j)$ and $K(x,t_j)$ ($j=1,2,...,N_t$),

$$\begin{aligned} X_i(k) &= \{X_{i1}(k), X_{i2}(k),..., X_{iD}(k)\} = \\ &\{C(x_1,t_j),C(x_2,t_j),...,C(x_{N_x},t_j),K(x_1,t_j),K(x_2,t_j),...,K(x_{N_x},t_j)\} \end{aligned} \quad , \tag{5.45}$$

where $D=2N_x$. Equation (5.49) is used as the objective function to evaluate the particles at time t_j. Two different methods were used to select the contraction-expansion coefficient α as described in section 4.4.2. Substituting $X_i(k)$ into Equation (5.41), the temperature u can be computed by solving the direct problem. At each generation, the positions of the particles are updated according to Equation (4.31). The process is repeated until a pre-defined number of generations have reached or the solution converged. K_{error} and C_{error} as defined in Equations (5.46) and (5.47), are used to evaluate the accuracy of the converged solutions.

Procedure of QPSO with Gaussian Mutation for the simultaneous estimation of $C(u)$ **and** $K(u)$

For $j = 1: N_t$

Initialization:

particle positions: $X(0) = \{X_1(0), X_2(0), ..., X_i(0), ..., X_M(0)\}$;

personal best positions: $P(0) = X(0)$;

global best position: P_g;

contraction-expansion coefficient α, $k = 0$, stopping criteria σ;

while ($k < k_{max}$) or (σ is not reached)

Compute the mean best position $C(k)$ by Equation (4.30);

For each particle $i = 1, 2, ..., M$

Compute the attractor $p_i(k)$ by Equation (4.15);

Update the position $X_i(k+1)$ according to Equation (4.31);

Evaluate the fitness $J[X_i(k+1)]$ according to Equation (5.42);

End for

Update $P(k)$ and $P_g(k)$;

Decrease contraction-expansion coefficient α linearly or keep constant;

$k = k + 1$;

End while

$C(:, t_{j+1}) = P_g(1: N_x)$;

$K(:, t_{j+1}) = P_g(N_x + 1: D)$.

End for

To demonstrate the viability, accuracy and stability of the modified QPSO with Gaussian mutation, a typical example used in [67] is considered. The predefined functional forms of $K(u)$ and $C(u)$ are defined as

$$K(u) = 1.0 + 4.5\exp\left(\frac{u}{80}\right) + 2.5\sin\left(\frac{u}{3}\right) \tag{5.46}$$

$$C(u) = 1.2 + 0.02u + 0.00001u^2 \tag{5.47}$$

The material has initial temperature $u_0 = 1$. When $t > 0$, the two boundaries are subjected to a constant heat flux, $q_1 = 17$ and $q_2 = 6$, respectively. The total simulated time is assumed to be $t_f = 1.2$, which is also dimensionless. The temporal step size and mesh size are set as $\Delta t = 0.02$ and $\Delta x = 0.05$ for the numerical tests.

Before addressing the simultaneous estimation of $C(u)$ and $K(u)$, special cases involving the estimation of either $C(u)$ or $K(u)$ are considered, assuming the other function as exactly known. Figure 5.48 shows the estimated $K(u)$ obtained by QPSO with exact measurements, by assuming $C(u)$ known as in Equation (5.47). The estimated $C(u)$ with known $K(u)$ as in Equation (5.46) is shown in Figure 5.49. The average errors of the estimated $K(u)$ and $C(u)$ are 8.84E-04 and 4.05E-05, respectively. Note that accurate results were obtained.

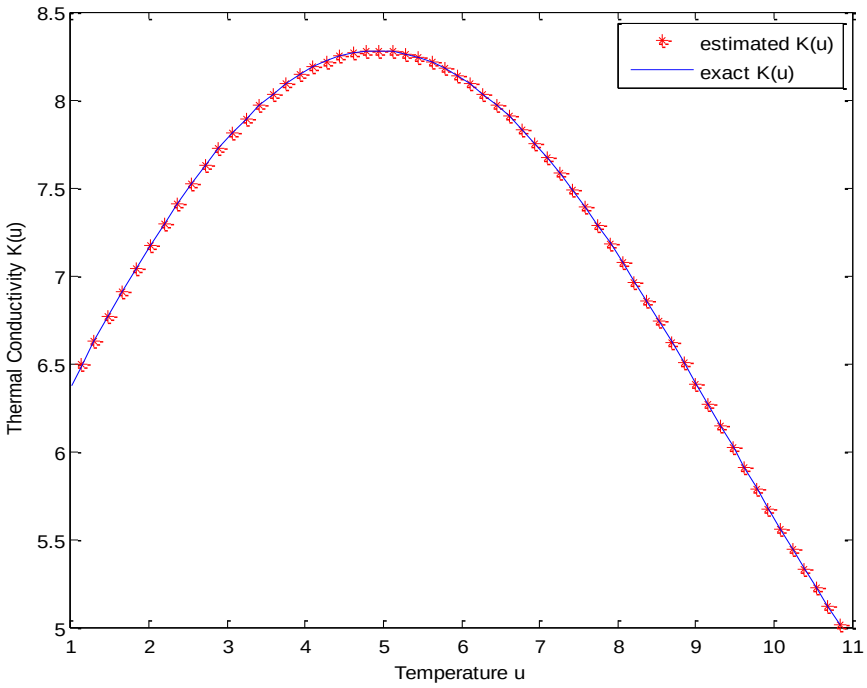


Figure 5.48: Estimated $K(u)$ at $x = 0.5$ with known $C(u)$ by using QPSO with exact measurements.

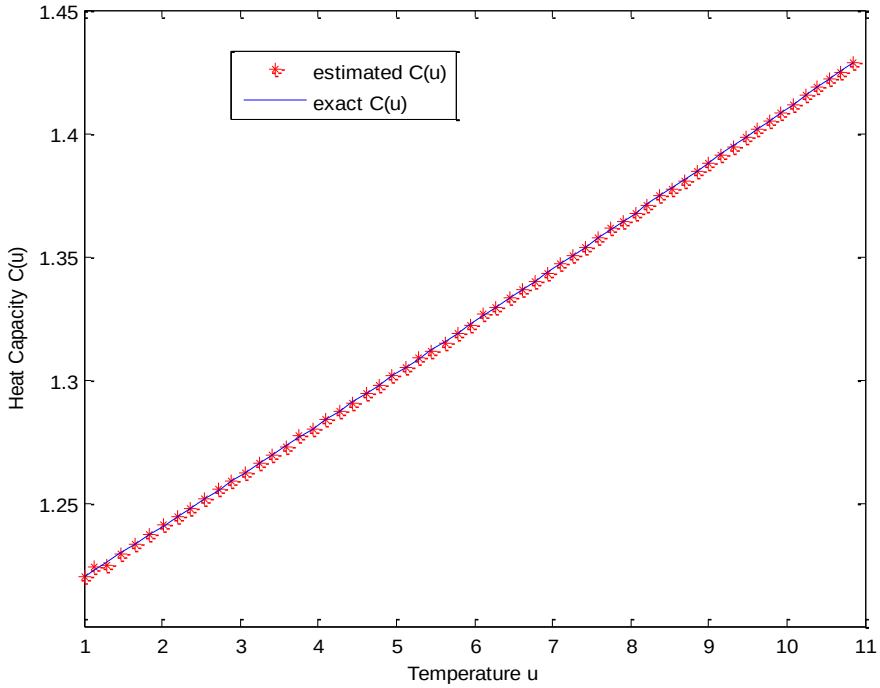
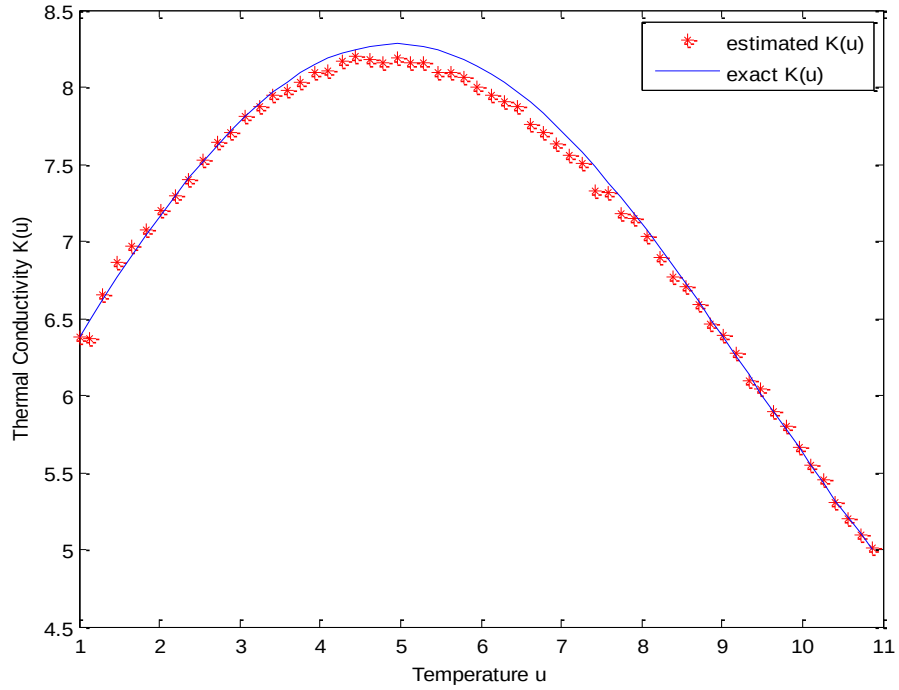
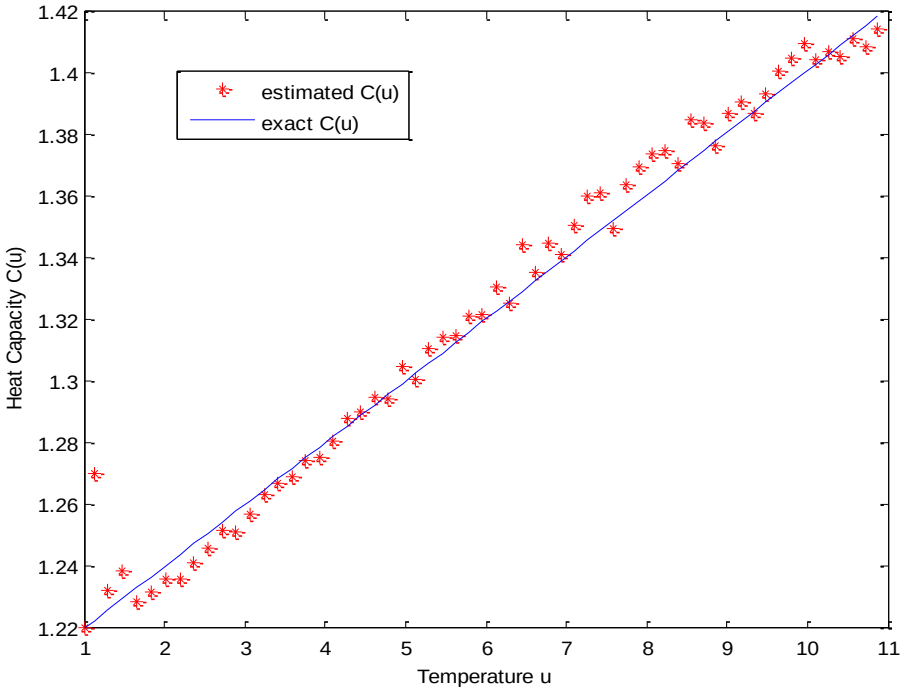


Figure 5.49: Estimated $C(u)$ at $x = 0.5$ with known $K(u)$ by using QPSO with exact measurements.

Then, the case of simultaneous estimation of $K(u)$ and $C(u)$ is examined by using QPSO, in which, the significant aspect is how the solutions are represented. The first N_x components of each particle's position represent the heat capacity $C(u)$ which are initialized in the range [4.0, 9.0], and the next N_x components represent the thermal conductivity $K(u)$ initialized in the range [1.0, 1.5]. Figure 5.50 shows the estimated results obtained by using QPSO with exact measurements. It can be seen that the estimated results are reasonably accurate.



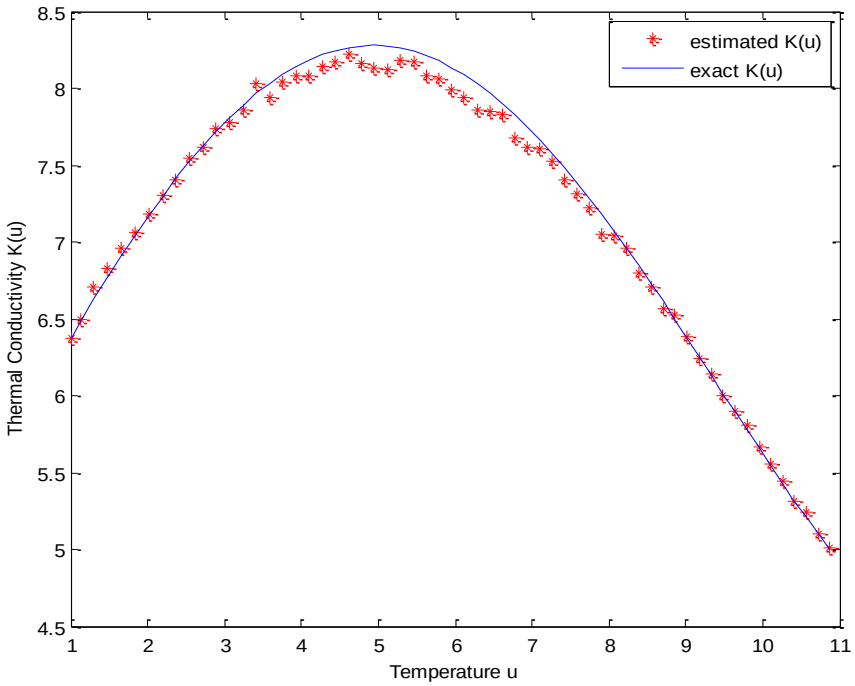
(a)



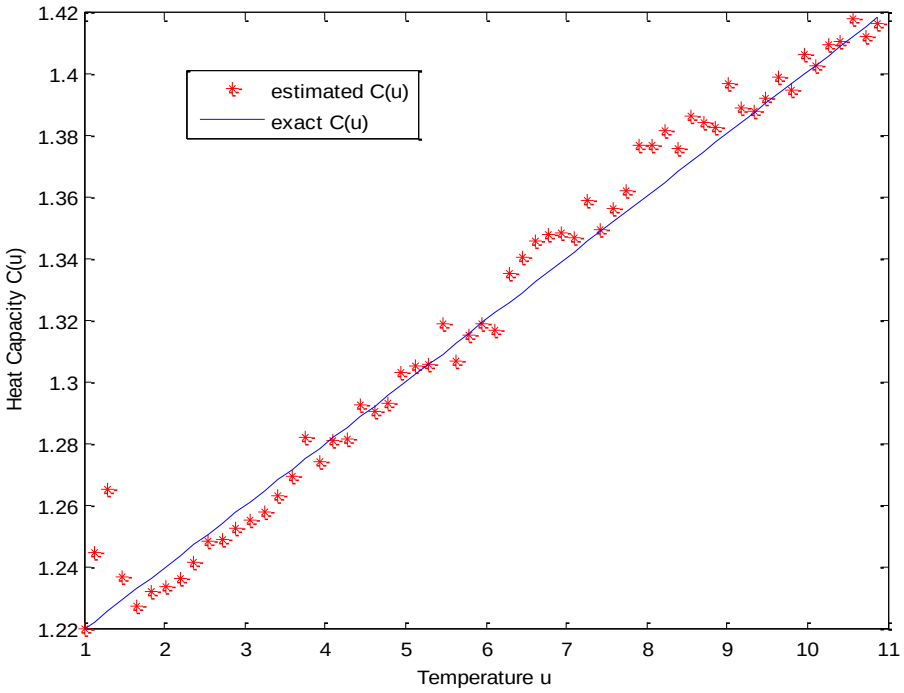
(b)

Figure 5.50: Simultaneous estimated results at $x = 0.5$ by using QPSO with exact measurements.

The estimated results using measurements with noise level $\varepsilon = 0.001$ and $\varepsilon = 0.005$ are shown in Figure 5.51 and Figure 5.52, respectively. As expected, increases in the measurement errors cause decreases in the accuracy of the inverse solution. Especially for the estimated $C(u)$ with noise level $\varepsilon = 0.005$ large oscillations exhibit around the exact value.

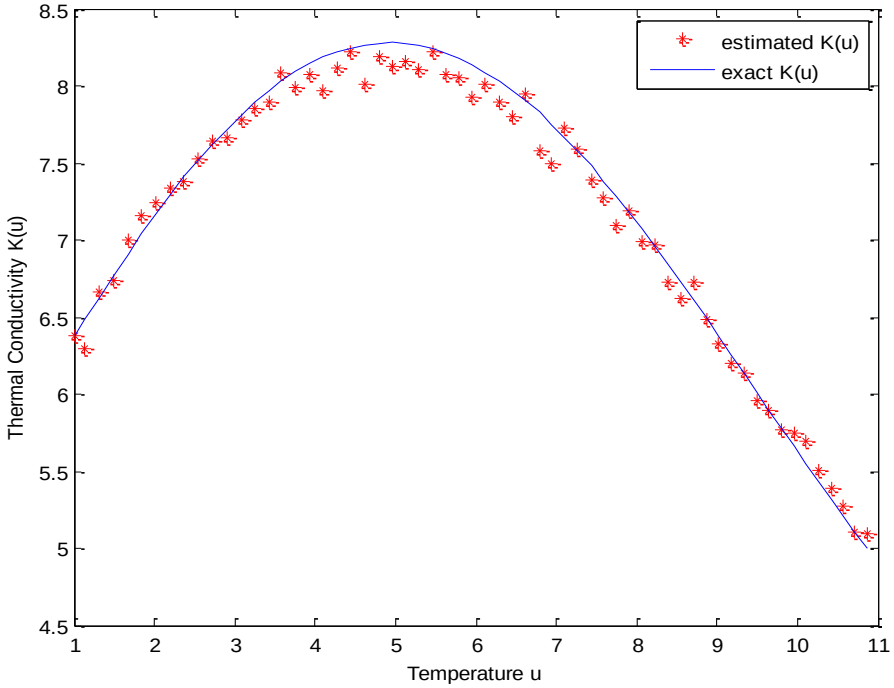


(a)

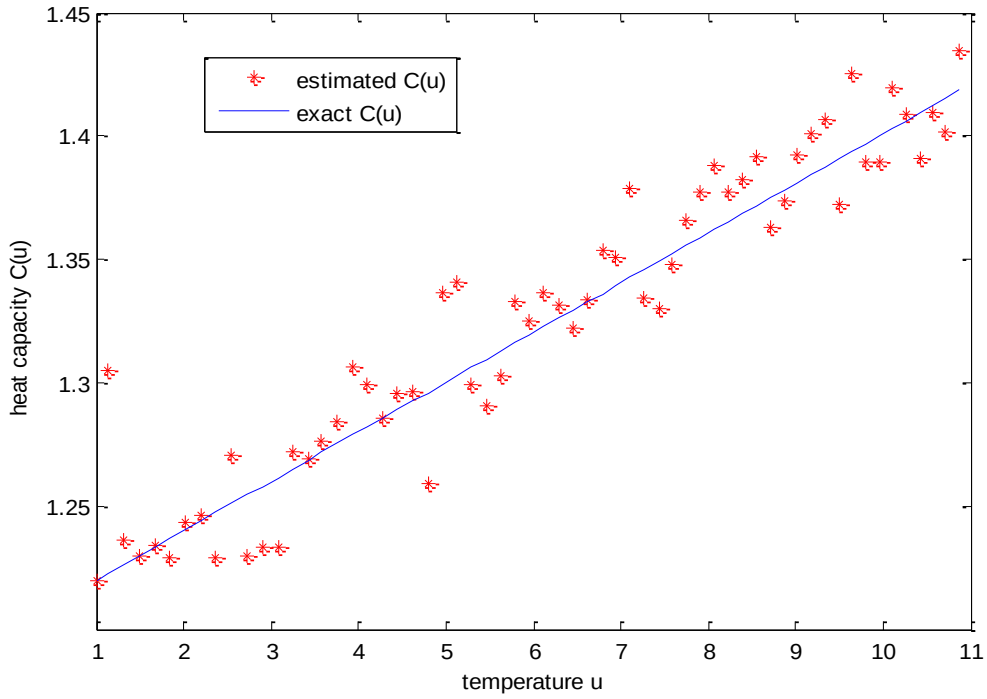


(b)

Figure 5.51: Simultaneous estimated results at $x = 0.5$ by using QPSO with noisy measurements $\varepsilon = 0.001$



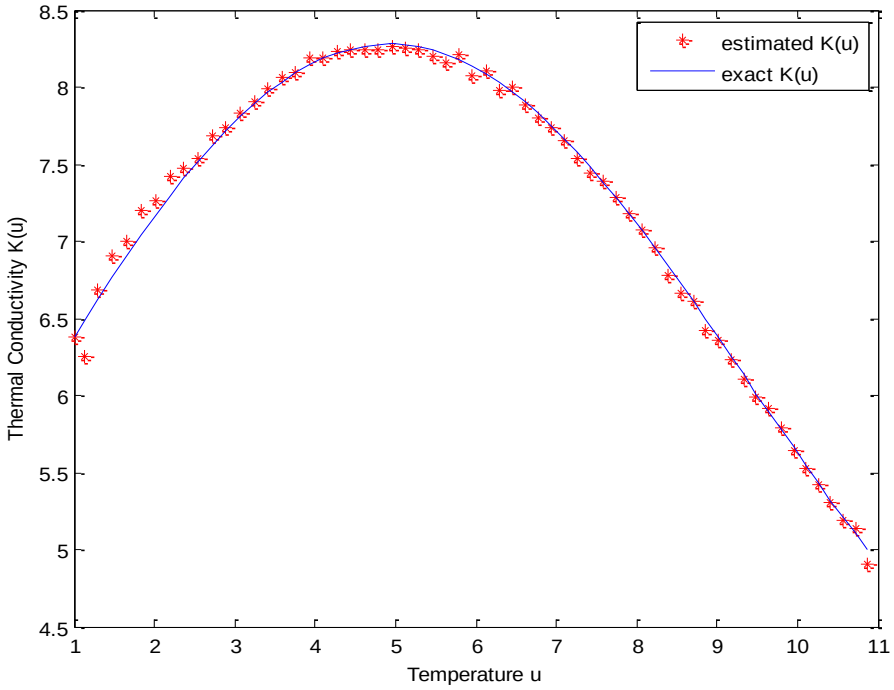
(a)



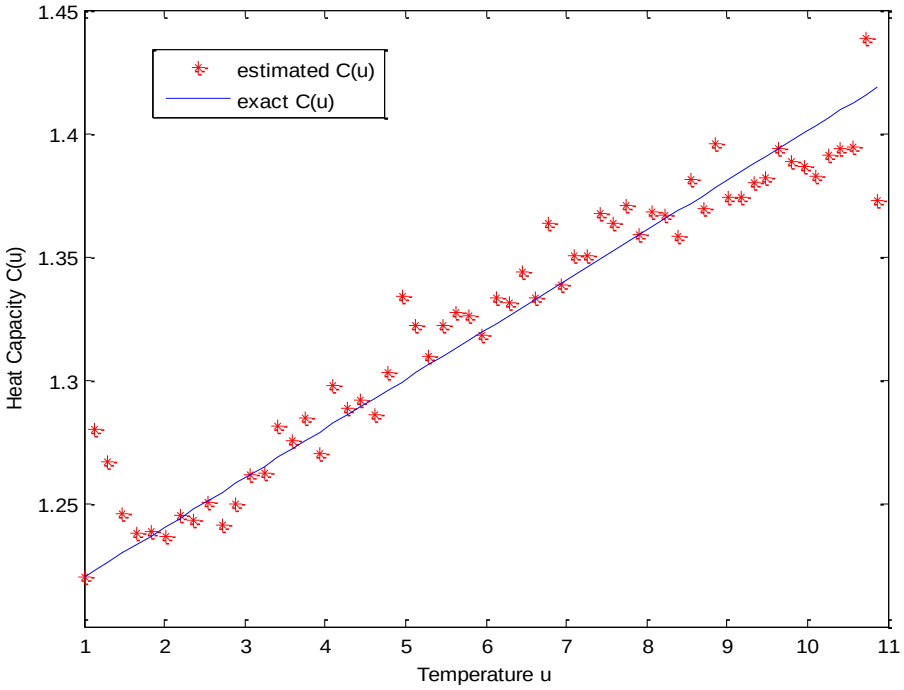
(b)

Figure 5.52: Simultaneous estimated results at $x = 0.5$ by using QPSO with noisy measurements $\varepsilon = 0.005$.

To improve the quality of the estimation results, QPSO with Gaussian mutation to mbest (MGQPSO), as described in section 4.3, is used to solve the inverse problem. The estimated results are shown in Figure 5.53-Figure 5.55. Note that the quality of the estimated $K(u)$ was improved to get closer to the exact value. On the other hand, from the view of figures, the quality of the estimated $C(u)$ is not improved at all. But the average error of $C(u)$ has improved significantly, as shown in Table 5.26. The reason may be because the range of $C(u)$ is in $[1.0,\ 1.45]$, a small perturbation looks like a large oscillation from the exact $C(u)$.

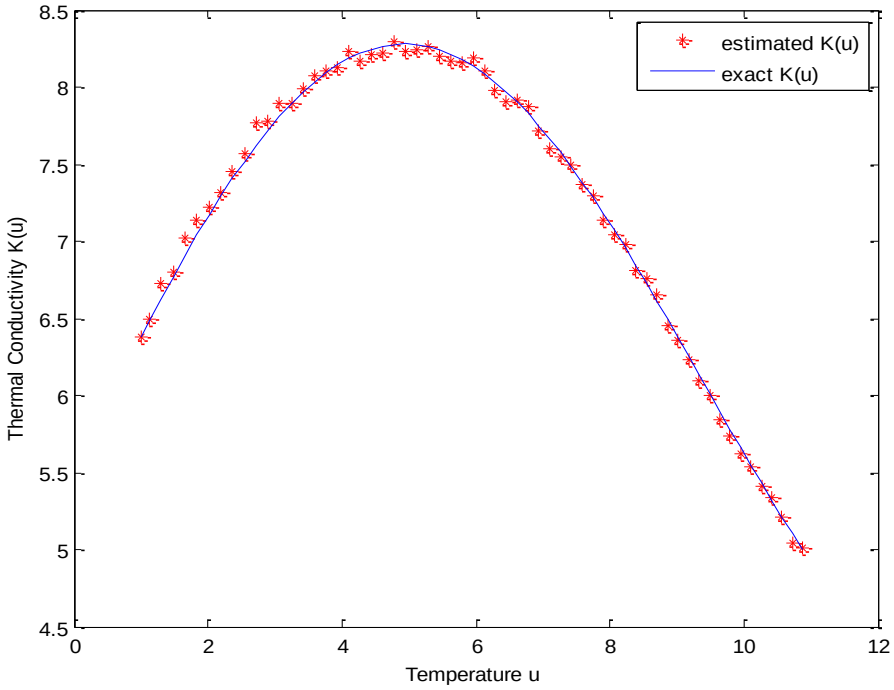


(a)

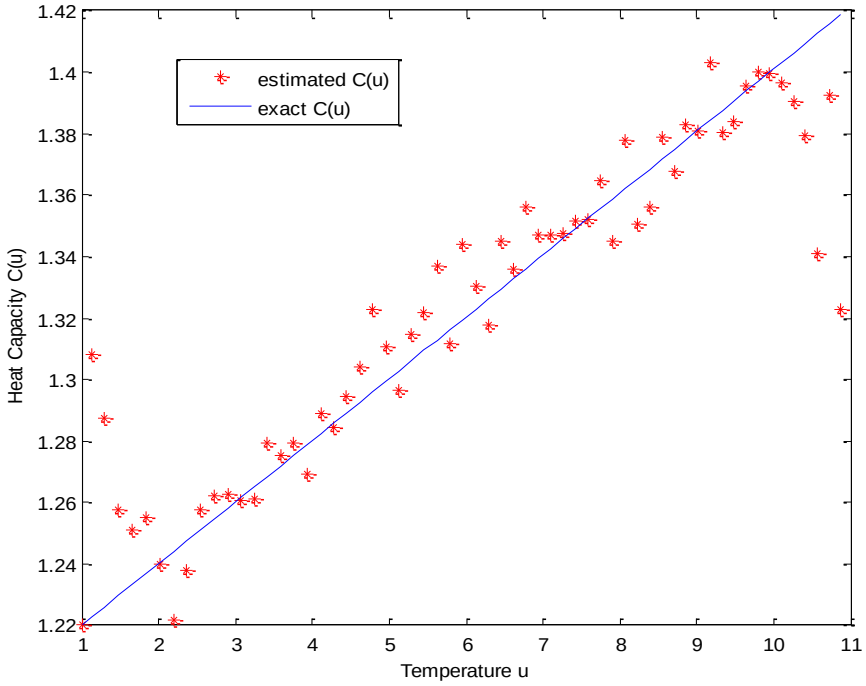


(b)

Figure 5.53: Simultaneous estimated results at $x = 0.5$ by using MGQPSO with exact measurements.

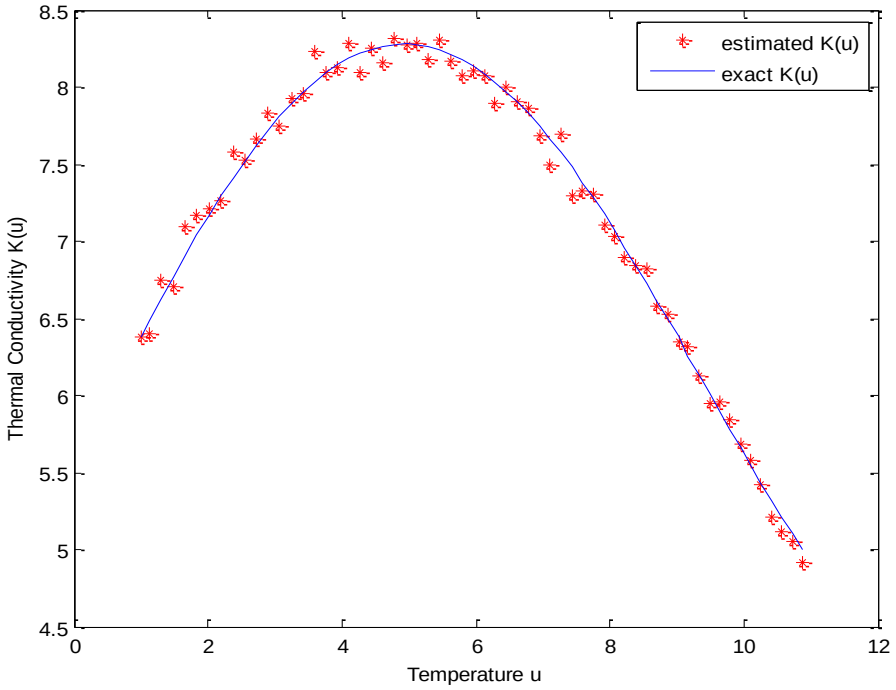


(a)

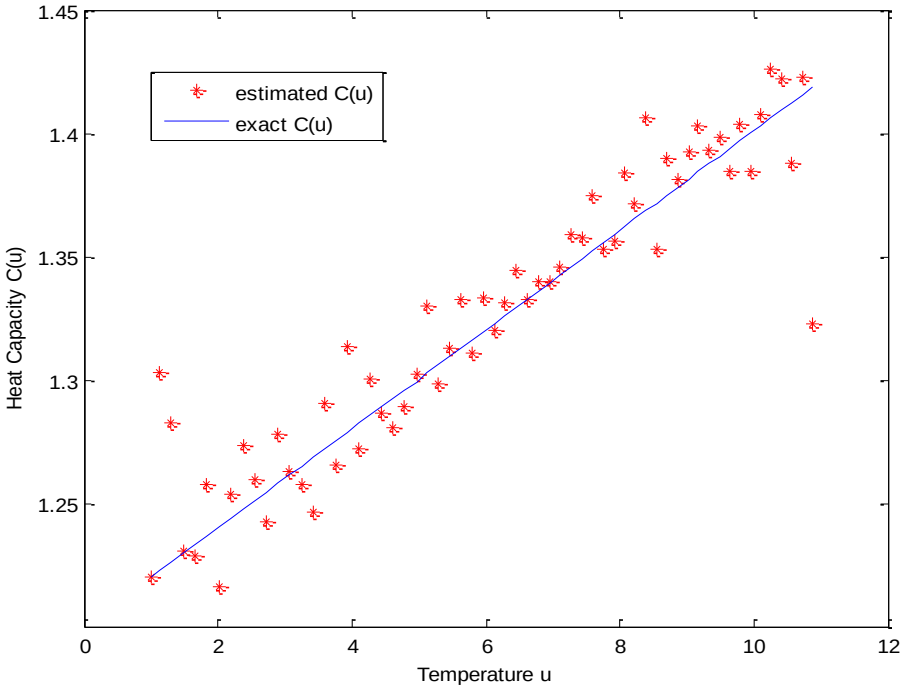


(b)

Figure 5.54: Simultaneous estimated results at $x = 0.5$ by using MGQPSO with noisy measurements $\varepsilon = 0.001$.



(a)



(b)

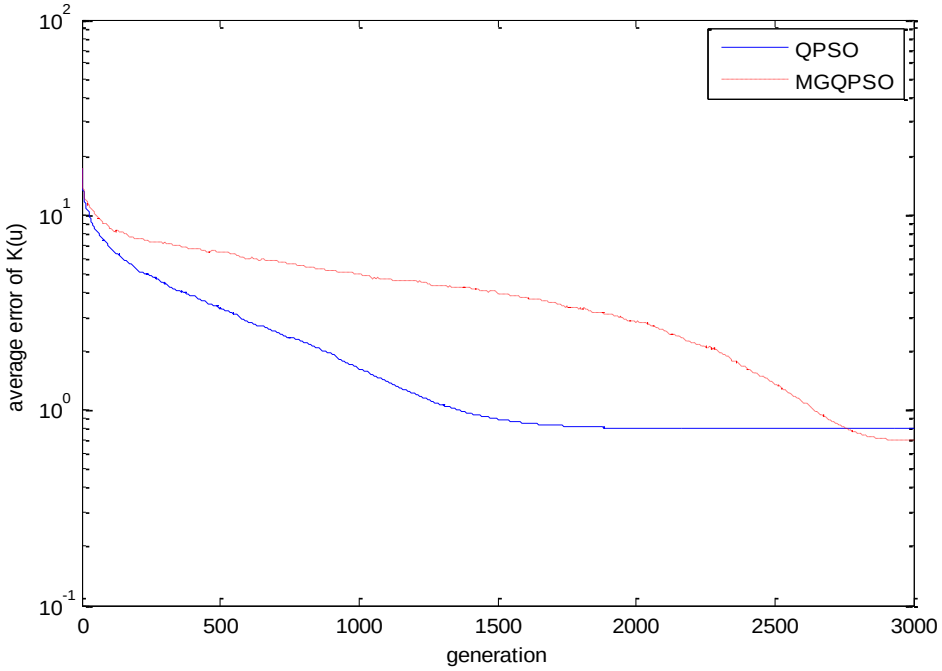
Figure 5.55: Simultaneous estimated results at $x = 0.5$ by using MGQPSO with noisy measurements $\varepsilon = 0.005$.

Table 5.26 gives the average error of the estimated C(u) and K(u) with different noise levels by using QPSO and MGQPSO, from which, one can see that the results obtained by using MGQPSO are better than that obtained by using QPSO.

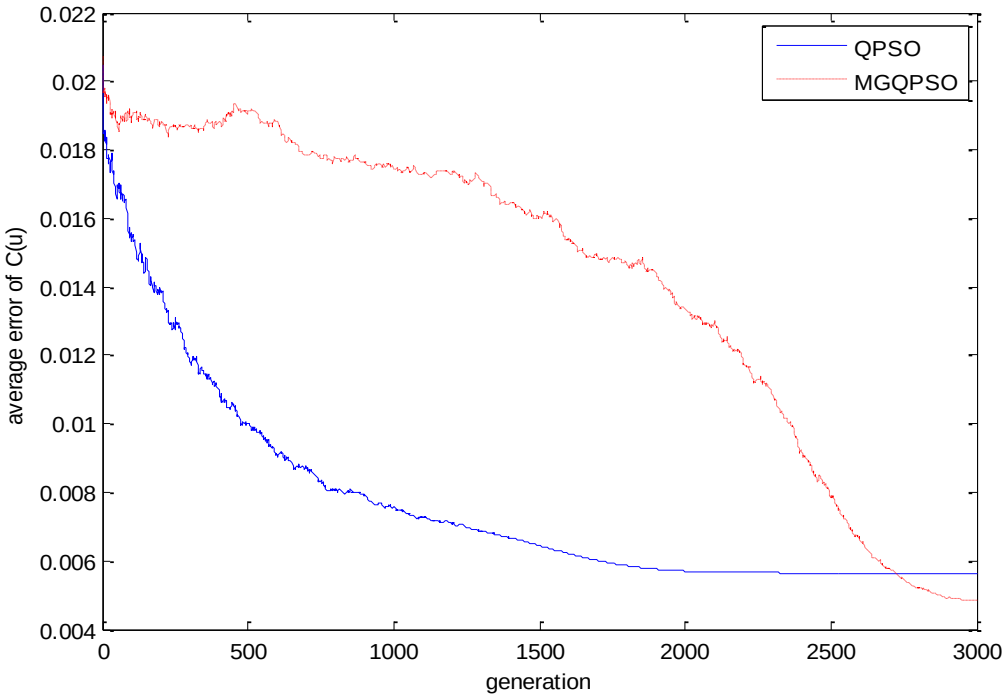
Table 5.26: Average error of different tests.

Method	Noise level ε	Average error	
		C_{error}	K_{error}
QPSO	0.000	8.53E-04	1.93E-03
	0.001	8.43E-04	1.96E-03
	0.005	9.74E-04	3.06E-03
MGQPSO	0.000	7.05E-04	1.53E-03
	0.001	7.09E-04	1.60E-03
	0.005	8.13E-04	2.90E-03

Figure 5.56 presents the average error of K(u) and C(u) with respect to the number of generations obtained by QPSO and MGQPSO. Note that QPSO converges much faster than MGQPSO at the beginning of the search process. But QPSO stagnates and gets trapped into a local optimum in the latter stage. On the other hand, the Gaussian mutation helps MGQPSO escape from the local optimum and have chance to search for the global optimum.



(a)



(b)

Figure 5.56: Convergence history of QPSO and MGQPSO for the simultaneous estimation of (a) $K(u)$, (b) $C(u)$.

5.8.3 Conclusion

The QPSO method with Tikhonov regularisation was applied to simultaneously estimate the temperature-dependent thermal conductivity and heat capacity with no prior information about the functional forms. The numerical experimental results demonstrate the viability of QPSO even with sensors only at the boundaries. In order to enhance the global search ability, a modified QPSO with Gaussian mutation was applied to the inverse estimation problem. The results illustrate the efficiency and stability of the proposed method, especially for the ill-posed problem with noisy measurements.

5.9 Closure

This chapter gave six applications of IHCPs in estimating time-varying heat flux, heat source, heat transfer coefficient, temperature-dependent thermal conductivity, boundary shape and simultaneously thermal conductivity and heat capacity. In addition, one application of inverse advection-dispersion problem of estimating the contaminant source was also examined. Both stochastic and deterministic methods were tested. To improve the estimated solution and the performance of the algorithms, the proposed hybrid method was applied to solve the inverse problems. In each application, the comparison of performance between different algorithms was presented and discussed.

Chapter 6 CONCLUSIONS AND FUTURE WORK

This thesis provided an in depth study of solving inverse problems numerically using deterministic and stochastic algorithms. Two types of engineering applications, including heat conduction problems and groundwater contamination problems, were examined in detail with unknown quantities such as boundary heat flux, heat or pollutant source, thermal properties, boundary shape. Measurement data such as temperature or concentration was used to provide the additional information in the process of determinating the unknown quantities.

Finite difference method was used to solve the direct problems for regular computational domains, while the boundary element method was used for irregular domains. The computing time for large numbers of boundary elements becomes very high.

Inverse problems are ill-posed because the solution is sensitive to random errors in the measured input data. Regularisation techniques are used to improve the stability. Nonlinear Tikhonov regularisation method was discussed in Chapter 2. The L-curve method for the regularisation parameter selection was used in the numerical tests presented in this thesis.

The conjugate gradient method (CGM) is a useful technique for inverse function estimation problems because regularisation is implicitly built into the algorithm and the number of iterations can be used as the regularisation parameter. The combination of a minimisation algorithm with an adjoint equation that provides the gradient to be used in the minimisation iterative procedure is the basis of the CGM. However, one main disadvantage of this method is that the initial guess for the iterative procedure must be chosen carefully in order to ensure convergence to the global minimum. In essence, the initial guess must fall into a very restricted region containing the exact solution. Another disadvantage of CGM is

that stagnation of the convergence often appears with the value close to the final time in estimating time-varying unknown quantities (e.g. heat flux) or with the value near the two boundaries in estimating spatial-dependent quantities (e.g. boundary shape).

A population-based heuristic algorithm known as quantum-behaved particle swarm optimisation (QPSO) was re-visited and examined. This thesis provided a new insight into the quantum behaviour of the particles, in particular the reasons behind using such behaviour and the application of the delta-well. The QPSO method is robust and easier to use than CGM, because the particles which represent the candidate solutions may be initialized randomly and do not require computing the complicated gradient of the objective function. In other words, the QPSO method is a derivative free method.

One disadvantages of a QPSO system is the loss of diversity in the population as in other population-based evolutionary algorithms. In the latter part of the search period, the particles are clustered together gradually and the swarm is likely to be trapped into local optima. To enhance the global search ability of QPSO for complex multi-modal problems, several modifications to QPSO are proposed. These are perturbation operation, Gaussian mutation, ring topology model and several methods for algorithmic parameter selection. Benchmark functions were used to test the performance of the proposed modified algorithms.

In complex engineering optimisation problems, every fitness evaluation costs long computational time, e.g. BEM for solving a two-dimensional steady heat conduction problem. It requires a large amount of CPU time in QPSO with thousands of generations. QPSO, as other population-based methods, is intrinsically parallel and can be effectively implemented on massively parallel processing architectures. Two parallel models of QPSO, master-slave parallelizaiton (synchronous and asynchronous) and static subpopulation parallelization are developed and applied to estimate the boundary shape. From the estimated results and performance of the parallel QPSOs, it can be concluded that the scalability of the static subpopulation parallel QPSO is limited and not suitable for systems with large number of processors, the asynchronous parallel QPSO makes full use of the computing resources without idle waiting.

A hybrid method, which makes use of deterministic (CGM) and stochastic (QPSO) methods, was proposed in this thesis to enhance the quanlity of the estimated solutions and the performance of the algorithms for solving the inverse problems numerically. Two novel

data transfer methods were proposed to pass a rough estimated solution from QPSO as an initial approximation to CGM.

Finally, the proposed methods were used to solve typical examples in the research literature of estimating time-varying heat fluxes, heat sources, heat transfer coefficient and contaminant sources, temperature-dependent thermal conductivities and two simultaneous unknown quantities. The numerical results demonstrate the feasibility and efficiency of QPSO and the global search ability and stability of the improved versions of QPSO. The two hybrid methods were applied to the estimation of heat flux and boundary shape, respectively. The results obtained are accurate, especially for the boundary shape determination problem. The simultaneous estimation of temperature dependent thermal conductivity and heat capacity is addressed by using QPSO with Gaussian mutation, the estimated results of which are much better than CGM and more stable with noisy measurements.

All of the objectives defined in Chapter 1 of the thesis have been achieved.

Possible future work may begin by applying the proposed algorithms to higher dimensional problems. This may not lead to significant complexities but simply increase the computational time and require more measurements to obtain accurate solution. Therefore, migrating the parallel QPSO algorithms onto other massively parallel processing architectures, such as GPU, or other computing resources, such as Cloud Computing on the internet to save computational time is one way out.

The random error existing in data measurements inevitably affects the accuracy of the numerical inverse determination of the unknown quantities. Methods of quantifying such inaccuracy would be useful in comparing different methods. In addition, such knowledge would be able to feed into a heuristic optimisation technique as a stopping criterion of the iterative algorithm. One way to understand such error manifestation is to examine the stochastic response due to a controlled random disturbance to a chosen property, such as thermal conductivity, on the direct solution of the field variable.

The methods developed in this thesis can be easily generalised to other science and engineering applications. In particular the wave equation falls into similar category of problems as investigated in this thesis. On the other hand tomographic reconstruction of hidden surface seems to be a challenging area coupling image processing and inverse problems. For example, in an inverse scattering problem for a time-dependent elastic wave equation, one seeks to determine unknown material parameters with variation in space from

measured wave reflection data. Typical applications involve many types of inverse problems occurring in seismic wave propagation, non-destructive testing and medical imaging.

Heuristic algorithms without evaluation of derivatives would be very useful in inverse problems. In particular methods such as Plant Propagation Algorithm, Bacteria Growth Algorithm, Predator model, etc.

References

[1] J.V. Beck, B. Blackwell and C.R. St-Clair Jr., Inverse Heat Conduction: Ill-Posed Problems, Wiley-Interscience, New York, 1985.

[2] J. Taler and P. Duda, Solving Direct and Inverse Heat Conduction Problems, Springer, Berlin, 2006.

[3] O.M. Alifanov, Inverse Heat Transfer Problems, Springer-Verlag, Berlin, 1994.

[4] A.N. Tikhonov and V.Y. Arsenin, Solution of Ill-posed Problems, V.H. Winston, Washington, D.C., 1977.

[5] J.V. Beck and K.J. Arnold, Parameter Estimation in Engineering and Science, Wiley, New York, 1977.

[6] J.V. Beck, Criteria for comparison of methods of solution of the inverse heat conduction problem, Nuclear Engineering and Design, vol. 53, pp. 11-22, 1979.

[7] H.W. Engl, K. Kunisch and A. Neubauer, Convergence rates for Tikhonov regularisation of nonlinear ill-posed problems, Inverse Problems, vol. 5, pp. 523-540, 1989.

[8] A. Neubauer, Tikhonov regularisation for non-linear ill-posed problems: optimal convergence rates and finite-dimensional approximation, Inverse Problems, vol. 5, pp. 541-557, 1989.

[9] T.I. Seidman and C.R. Vogel, Well-posed and convergence of some regularisation methods for nonlinear ill-posed problems, Inverse Problems, vol. 5, pp. 227-238, 1989.

[10] A. Binder, H.W. Engl, A. Neubauer, O. Scherzer and W.G. Charles, Weakly closed nonlinear operators and parameter identification in parabolic equations by Tikhonov regularisation, Applicable Analysis, vol. 55, pp. 215-234, 1994.

[11] P.C. Hansen, Analysis of discrete ill-posed problems by means of the L-curve, SIAM Review, vol. 34, pp. 561-580, 1992.

[12] G.H. Golub, M. Heath and G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics, vol. 21, pp. 215-223, 1979.

[13] J. Kennedy and R.C. Eberhart, Particle swarm optimisation, IEEE International Conference on Neural Networks, Perth, Australia, 1995.

[14] F. Van den Bergh, An analysis of particle swarm optimizers, Ph.D. diss., University of Pretoria, South Africa, 2001.

[15] R.C. Eberhart, Y. Shi and J. Kennedy, Swarm Intelligence. Morgan Kaufmann Series in Evolutionary Computation, 1st ed., Morgan Kaufmann, 2001.

[16] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimisation, Proceeding of Congress on Evolutionary Computation, IEEE Press, Washington D.C., pp. 1951-1957, 1999.

[17] R.C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimisation, in: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE Press, San Diego, USA, pp. 84–88, 2000.

[18] Y. Shi and R.C. Eberhart, Parameter selection in particle swarm optimisation, in: Proceedings of the Seventh Annual Conference on Evolutionary Programming, New York, USA, pp.591–600, 1998.

[19] Y. Shi and R.C. Eberhart, A modified particle swarm optimizer, Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ, pp. 69-73, 1998.

[20] Y. Shi and R.C. Eberthart, Empirical study of particle swarm optimisation, Proceeding of IEEE Congress on Evolutionary Computation, IEEE press, Piscataway, NJ, pp. 1945-1950, 2000.

[21] V.A. Morozov, On the solution of functional equations by the method of regularisation, Soviet Math. Dokl, vol. 7, pp. 414-417, 1996.

[22] D. Bratton and J. Kennedy, Defining a standard for particle swarm optimisation, Proceeding of IEEE Swarm Intelligence Symposium 2007, IEEE Press, Honolulu, Hawaii, pp. 120-127, 2007.

[23] J. Sun, B. Feng and W.B. Xu, Particle swarm optimisation with particles having quantum behaviour, Proceedings of Congress Evolutionary Computation, IEEE Press, Portland, USA, vol. 1, pp. 325-331, 2004.

[24] J. Sun, W.B. Xu and B. Feng, A global search strategy of quantum-behaved particle swarm optimisation, Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, IEEE Press, Singapore, vol. 1, pp. 111-116, 2004.

[25] J. Sun, W.B. Xu and B. Feng, Adaptive parameter control for quantum-behaved particle swarm optimisation on individual level, Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Hawaii, vol. 4, pp. 3049-3054, 2005.

[26] M. Clerc and J. Kennedy, The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space, IEEE Transaction on Evolutionary Computation, vol. 6, pp. 58-73, 2002.

[27] J. Sun, W. Fang, X.J. Wu, Z.P. Xie and W.B. Xu, Quantum-behaved particle swarm optimisation: analysis of the individual particle's behaviour and parameter selection, Evolutionary Computation (MIT Press), in press.

[28] C. Dong, G.F. Wang, Z.Y. Chen and Z.Q. Yu, A method of self-adaptive inertia weight for PSO, Proceeding of International Conference Computer Science and Software Engineering, vol. 1, pp. 1195-1198, 2008.

[29] J. Holland, Outline for a logical theory of adaptive systems, Journal of the ACM, vol. 3, pp. 297-314, 1962.

[30] J. Holland, Adaption in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.

[31] D.E. Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, Reading, MA, USA, 1989.

[32] L.J. Eshelman and J.D. Schaffer, Real-coded genetic algorithms and interval schemata, In D. Whitley, editor, Foundations of Genetic algorithm II, pp. 187-202, Morgan Kaufmann, San Mateo, CA, USA, 1993.

[33] M.J. Colaco, H.R.B. Orlande and G.S. Dulikravich, Inverse and optimisation problems in heat transfer, Invited Lecture, 10th Brazilian Congress of Thermal Science and Engineering – ENCIT, Rio de Janeiro, RJ, 2004.

[34] A. Thom and C.J. Apelt, Field Computation in Engineering and Physics, D. Van Nostrand, London, 1961.

[35] A. R. Mitchell, The Finite Difference Method in Partial Differential Equations, John Wiley & Sons Inc, New York, 1980.

[36] O. Scherzer, The use of Morozov's discrepancy principle for Tikhonov regularisation of nonlinear ill-posed problems, Computing, vol. 51, pp. 45-60, 1993.

[37] P.C. Hansen and D.P. O'Leary, The use of the L-curve in the regularisation of discrete ill-posed problem, SIAM Journal of Scientific Computing, vol. 14, pp. 1487-1503, 1993.

[38] P.C. Hansen, The L-curve and its use in the numerical treatment of inverse problem, Computational Inverse Problems in Electrocardiology, WIT Press, Southampton, pp. 119-142, 2000.

[39] C.R. Vogel, Non-convergence of the L-curve regularisation parameter selection method, Inverse Problems, vol. 12, pp. 535-547, 1996.

[40] G. Wahba, Practical approximation solutions to linear operator equations when the data are noisy, SIAM Journal of Numerical Analysis, vol. 14, pp. 651-667, 1977.

[41] K.C. Ang, Introducing the boundary element method with MATALB, International Journal of Mathematical Education in Science and Technology, vol. 39, pp. 505-519, 2008.

[42] O.M. Alifanov, Solution of an inverse problem of heat conduction by iteration methods. Journal of Engineering Physics, vol. 26, pp. 471-476, 1974.

[43] H.X. Long, J. Sun, X.G. Wang, C-H. Lai and W.B. Xu, Using selection to improve quantum-behaved particle swarm optimisation，International Journal of Innovative Computing and Applications, vol. 2, pp. 100-114, 2009.

[44] J. Sun, W.B. Xu and W. Fang, A diversity-guided quantum-behaved particle swarm optimisation algorithm, Simulated Evolution and Learning, Hefei, China, 2006.

[45] J. Sun, W.B. Xu and W. Fang, Quantum-behaved particle swarm optimisation algorithm with controlled diversity, ICCS, Reading, UK, 2006.

[46] T.V. Bharat, V.S. Puvvadi and M.A Mehter, Swarm intelligence based inverse model for characterization of groundwater contaminant source, Electronic Journal of Geotechnical Engineering, 14, 2006.

[47] D. Dumitrescu, B. Lazzerini, L.C. Jain and A. Dumitrescu, Evolutionary Computation, CRC Press, Florida, 2000.

[48] Z.S. Lu, Z.R. Hou and J. Du, Particle swarm optimisation with adaptive mutation, Frontiers of Electrical and Electronic Engineering in China, vol. 1, pp. 99-104, 2006.

[49] A. Stacey, M. Jancic and I. Grundy, Particle swarm optimisation with mutation, IEEE Congress on Evolutionary Computation, 2003.

[50] Y.L. Gao and Z.H. Ren, Adaptive particle swarm optimisation algorithm with genetic mutation operation, Third International Conference Natural Computation, vol.2, pp. 211-215, 2007.

[51] P.S. Andrews, An investigation into mutation operators for particle swarm optimisation, IEEE Congress of Evolutionary Computation, pp. 1044-105, Vancouver, BC, 2006.

[52] J. Liu, J. Sun and W.B. Xu, Quantum-behaved particle swarm optimisation with adaptive mutation operator, Advances in Natural Computation, Lecture Notes in Computer Science, vol. 4221, pp. 959-967, 2006.

[53] D. Velduizen, J. Zydallis and G. Lamount, Considerations in engineering parallel multiobjective evolutionary optimisations, IEEE Transaction Evolutionary Computing, vol. 7, pp. 144-173, 2002.

[54] A.Eiben and J. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, 2003.

[55] L. Mussi, F. Daolio and S. Cagnoni, Evaluation of parallel particle swarm optimisation algorithms within the CUDA^{TM} architecture, Information Science, 2010.

[56] B. Koh, A.D. George, R.T. Haftka and B.J. Fregly, Parallel asynchronous particle swarm optimisation, International Journal for Numerical Methods in Engineering, vol. 67, pp. 578-595, 2006.

[57] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Quarterly Journal of Applied Mathematics, vol. 2, pp. 164-168, 1944.

[58] J.F. Schutte, J.A. Reinbolt, B.J. Fregly, R.T. Haftka and A.D. George, Parallel global optimisation with particle swarm algorithm, International Journal of Numerical Methods in Engineering, vol. 61, pp. 2296-2315, 2004.

[59] C.H. Huang and C.C. Tsai, An inverse heat conduction problem of estimating boundary fluxes in an irregular domain with conjugate gradient method, International Journal of Heat and Mass Transfer, vol. 34, pp. 47-54, 1998.

[60] C.H. Huang and C.W. Chen, A boundary element-based inverse problem in estimating transient boundary conditions with conjugate gradient method, International Journal for Numerical Methods in Engineering, vol. 42, pp. 943-965, 1998.

[61] C.H. Huang and C.W. Chen, A three-dimensional inverse forced convection problem in estimating surface heat flux by conjugate gradient method, International Journal of Heat and Mass Transfer, vol. 43, pp. 3171-3181, 2000.

[62] C. H. Huang and J.H. Hsiao, A non-linear fin design problem in determining the optimum shape of spine and longitudinal fins, Communications in Numerical Methods in Engineering, vol. 19, pp. 111–124, 2003.

[63] J. Su and A.J.S. Neto, Heat source estimation with the conjugate gradient method in inverse linear diffusive problems, Journal of Brazilian Society of Mechanical Sciences, vol. 23, pp. 321-334, 2001.

[64] C.H. Huang, J.Y. Yan and H.T. Chen, Function estimation in predicting temperature-dependent thermal conductivity without internal Measurements, AIAA Journal of Thermophysics and Heat Transfer, vol. 9, pp. 667-673, 1995.

[65] C.H. Huang and M.N. Ozisik, Conjugate gradient method for determining unknown contact conductance during metal casting, International Journal of Heat and Mass Transfer, vol. 35, pp. 1779-1786, 1992.

[66] C.H. Huang and J.Y. Yan, An inverse problem in predicting temperature dependent heat capacity per unit volume without internal measurements, International Journal for Numerical Methods in Engineering, vol. 39, pp. 606–618, 1996.

[67] C.H. Huang and J.Y. Yan, An inverse problem in simultaneously measuring temperature dependent thermal conductivity and heat capacity, International Journal of Heat and Mass Transfer, vol. 18, pp. 3433-3441, 1995.

[68] C.H. Huang and S.C. Chin, A two-dimensional inverse problem in imaging the thermal conductivity of a non-homogeneous medium, International Journal of Heat Mass Transfer, vol. 43, pp. 4061-4071, 2000.

[69] J. Su and G.F. Hewitt, Inverse heat conduction problem of estimating time-varying heat transfer coefficient, Numerical Heat Transfer, Part A, vol. 45, pp. 777-789, 2004.

[70] R. Fletcher, Practical Methods of Optimisation, Wiley, New York, 1987.

[71] P. Terrola, A method to determine the thermal conductivity from measured temperature profiles, International Journal of Heat and Mass Transfer, vol. 32, pp. 1425-1430, 1989.

[72] S. Kim, M.C. Kim and K.Y. Kim, Non-iterative estimation of temperature-dependent thermal conductivity without internal measurements, International Journal of Heat and Mass Transfer, vol. 46, pp. 1801-1810, 2003.

[73] W.K. Yeung and T.T. Lam, Second-order finite difference approximation for inverse determination of thermal conductivity, International Journal of Heat and Mass Transfer, vol. 39, pp. 3685-3693, 1996.

[74] C.L. Chang and M. Chang, Non-iterative estimation of thermal conductivity using finite volume method, International Communications Heat and Mass Transfer, vol. 33, pp. 1013-1020, 2006.

[75] C.L. Chang and M. Chang, Inverse estimation of the thermal conductivity in a one-dimensional domain by Taylor series approach, Heat Transfer Engineering, vol. 29, pp. 830-838, 2008.

[76] C.Y. Yang, Estimation of the temperature-dependent thermal conductivity in inverse heat conduction problems, Applied Mathematical Modelling, vol.23, pp. 469-478, 1999.

[77] A. Shidfar, A. Zakeri and A. Neisi, A two-dimensional inverse heat conduction problem for estimation heat source, International Journal of Mathematics and Science, vol. 10, pp. 1633-1641, 2005

[78] B. Czel and G. Grof, Genetic algorithm-based method for determination of temperature-dependent thermo-physical properties, International Journal of Thermophysics, vol. 30, pp. 1975-1991, 2009.

[79] A. Imani, A.A. Ranjbar and M. Esmkhani, Simultaneous estimation of temperature-dependent thermal conductivity and heat capacity based on modified genetic algorithm, Inverse Problems in Science and Engineering, vol. 14, pp. 767-783, 2006.

[80] S. Garcia and E. P. Scott, Use of genetic algorithms in thermal property estimation: Part I – experimental design optimisation, Numerical Heat Transfer, Part A: Applications, vol. 33, pp. 135-147, 1998.

[81] S. Garcia, J. Guynn and E.P. Scott, Use of genetic algorithms in thermal property estimation: Part II – simultaneous estimation of thermal properties, Numerical Heat Transfer, Part A: Applications, vol. 33, pp. 149-168, 1998.

[82] M. Raudensky, J. Horsky, J. Krejsa and L. Slama, Usage of artificial intelligence methods in inverse problems for estimation of material parameters, International Journal of Numerical Methods for Heat & Fluid Flow, vol. 6, pp. 19-29, 1996.

[83] O. Cortes, G. Urquiza and J.A. Hernandez, Inverse heat transfer using Levenberg-Marquardt and particle swarm optimisation methods for heat source estimation, Applied Mechanics and Materials, vol. 15, pp. 35-40, 2009.

[84] V.C. Mariani, V.J. Neckel and R.B. Grebogi, Cauchy particle swarm optimisation with dynamic adaption applied to inverse heat transfer problem, IEEE International Conference Systems Man and Cybernetics, Istanbul, pp. 3730-3734, 2010.

[85] S. Kim, B.J. Chuang, M.C. Kim and K.Y. Kim, Inverse estimation of temperature-dependent thermal conductivity and heat capacity per unit volume with the direct integration approach, Numerical Heat Transfer, Part A: Applications, vol. 44, pp. 521-535, 2003.

[86] C.H. Huang and M.N. Ozisik, Direct integration approach for simultaneously estimating temperature dependent thermal conductivity and heat capacity, Numerical heat transfer, Part A: Applications, vol. 20, pp. 95-110, 1991.

[87] G.P. Flach and M.N. Ozisik, Inverse heat conduction problem of simultaneously estimating spatially varying thermal conductivity and heat capacity per unit volume, Numerical Heat Transfer, Part A: Application, vol. 16, pp. 249-266, 1989.

[88] H.T. Chen and J.Y. Lin, Simultaneous estimation of temperature-dependent thermal conductivity and heat capacity, International Journal of Heat and Mass Transfer, vol. 41, pp. 2237-2244, 1998.

[89] A.A Ranjbar, M. Famouri and A. Imani, A transient inverse problem in simultaneous estimation of TDTP based on MEGA, International Journal of Numerical Methods for Heat & Fluid Flow, vol. 20, 201-217, 2010.

[90] C.Y. Yang, Non-iterative solution of inverse heat conduction problems in one dimension, Communications in Numerical Methods in Engineering, vol. 13, pp. 419-427, 1997.

[91] M. Raudensky, K.A. Woodbury, J. Kral and T. Brezina, Genetic algorithm in solution of inverse heat conduction problems, Numerical Heat Transfer, Part B, vol. 28, pp. 293-306, 1995.

[92] G.R. Liu, J.-J. Zhou and J.G. Wang, Coefficient identification in electronic system cooling simulation through genetic algorithm, Computers and Structures, vol. 80, pp. 23-30, 2002.

[93] Q.P. Guo, D. Shen, Y. Guo and C.-H. Lai, Parallel genetic algorithms for the solution of inverse heat conduction problems, International Journal of Computer Mathematics, vol. 84, pp. 241-249, 2007.

[94] F.B. Liu, A modified genetic algorithm for solving the inverse heat transfer problem of estimating plane heat source, International Journal of Heat and Mass Transfer, vol. 51, pp. 3745-3752, 2008.

[95] A. Nachaoui, An improved implementation of an iterative method in boundary identification problems, Numer. Algorithms, vol. 33, pp. 381-398, 2003.

[96] N.S. Mera, L. Elliott and D.B. Ingham, Numerical solution of a boundary detection problem using genetic algorithm, Engineering Analysis with Boundary Elements, vol. 28, pp. 405-411, 2004.

[97] D.A. Mohsen and K. Mahmud, Identification of thermal conductivity and the shape of an inclusion using the boundary element method and the particle swarm optimisation algorithm, Inverse Problems in Science and Engineering, vol. 17, pp. 855-870, 2009.

[98] K. Mahmud and D.A. Mohsen, Inclusion identification by inverse application of boundary element method, genetic algorithm and conjugate gradient method, American Journal of Applied Science, vol. 5, pp.1158-1166, 2008.

[99] M.J. Colaco and H.R.B. Orlande, Comparison of different versions of the conjugate gradient method of function estimation, Numerical Heat Transfer, Part A, vol. 36, pp. 229-249, 1999.

[100] H.T. Chen and X.Y. Wu, Investigation of heat transfer coefficient in two-dimensional transient inverse heat conduction problems using the hybrid inverse scheme, International Journal for Numerical Methods in Engineering, vol. 73, pp. 107-122, 2008.

[101] M. Slodicka, D. Lesnic and T.T.M. Onyango, Determination of a time-dependent heat transfer coefficient in a nonlinear inverse heat conduction problem, Inverse Problems in Science and Engineering, vol. 18, pp. 65-81, 2010.

[102] S. Chantasiriwan, Inverse heat conduction problem of determining time-dependent heat transfer coefficient, International Journal of Heat and Mass Transfer, vol. 42, pp. 4275-4285, 2000.

[103] P. Sidauruk, Parameter determination for multi-layered aquifer and groundwater contaminant transport, Ph.D. diss., University of Delaware, 1996.

[104] S.M. Gorelick, B. Evans and I. Remson, Identifying sources of groundwater pollution: An optimisation approach, Water Resources Research, vol. 19, pp. 779-790, 1983.

[105] B.J. Wagner, Simultaneous parameter estimation and contaminant source characterization for coupled groundwater flow and contaminant transport modelling, Journal of Hydrology, vol. 135, pp. 275-300, 1992.

[106] S. Zou and A. Parr, Estimation of dispersion parameters for two dimensional plumes, Ground Water, vol. 31, pp. 389-392, 1993.

[107] H.T. Skaggs and Z.J. Kabala, Recovering the release history of a groundwater contaminant, Water Resources Research, vol. 30, pp. 71-79, 1994.

[108] H.T. Skaggs and Z.J. Kabala, Limitations in recovering the history of a groundwater contaminant plume, Journal of Contaminant Hydrology, vol. 33, pp. 347-359, 1998.

[109] H.T. Skaggs and Z.J. Kabala, Recovering the history of a groundwater contaminant plume: method of quasi-reversibility, Water Resources Research, vol. 31, pp. 2669-2673, 1995.

[110] A.D. Woodbury and T.J. Ulrych, Minimum relative entropy inversion: Theory and application to recovering the release history of a groundwater contaminant, Water Resources Research, vol. 32, pp. 2671-2681, 1996.

[111] A.D. Woodbury, E. Sudicky, T.J. Ulrych and R. Ludwig, Three-dimensional plume source reconstruction using minimum relative entropy inversion, Journal of Contaminant Hydrology, vol. 32, pp. 131-158, 1998.

[112] M.F. Snodgrass and P.K. Kitanidis, A geostatistical approach to contaminant source identification, Water Resources Research, vol. 33, pp. 537-546, 1997.

[113] N.Z. Sun, Inverse Problems in Groundwater Modeling, Dordrecht, The Netherlands: Kluwer, 1994.

[114] T.H. Nguyen, Optimisation approach to the inverse convection problem, Proc. Int. Workshop Inverse Problems, HoChiMinh City, Vietnam, 1995.

[115] C. Liu and W.P. Ball, Application of inverse methods to contaminant source identification from aquitard diffusion profiles at Dover AFB, Delaware, Water Resources Research, vol. 35, pp. 1975-1985, 1999.

[116] T.V. Bharat, P.V. Sivapullaiah and M.M. Allam, Accurate parameter estimation of contaminant transport inverse problem using particle swarm optimisation, IEEE Swarm Intelligence Symposium, St Louis, 2008.

[117] A.L. Alkidas, Heat transfer characteristics of a spark-ignition engine, Journal of Heat Transfer, vol. 102, pp. 189-193, 1980.

[118] T.K.J. Howse, R. Kent and H. Rawson, The determination of glass-mould heat fluxes from mould temperature measurements, Glass Technology, vol. 12, pp. 91-93, 1971.

[119] E.M. Sparrow, A. Haji-Sheikh and T.S. Lundgren, The inverse problem in transient heat conduction, Journal of Applied Mechanics, vol. 86, pp. 369-375, 1964.

[120] D.Y. T. Lin and J.W. Westwater, Effect of metal thermal properties on boiling curves obtained by the quenching method, Heat Transfer 1982- Munchen Conference Proceedins, Hemisphere Publ. Corp., vol. 4, pp. 155-160, 1982.

[121] V.V. Solov'ev, Solvability of the inverse problem of finding a source, using overdetermination on the upper base for a parabolic equation, Journal of Differential Equations vol. 25, pp. 1114-1119, 1990.

[122] E.G. Savateev, On problems of determining the source function in a parabolic equation, Journal of Inverse and Ill-posed Problems, vol. 3, pp. 83-102, 1995.

[123] J.R. Cannon and P. DuChateau, An inverse problem for determination an unknown source term in a heat conduction, Journal of Mathematical Analysis and Applications, vol. 75, pp. 465-485, 1980.

[124] J.R. Cannon, Determination of an unknown heat source from overspecified boundary data, SIAM Journal on Numerical Analysis, vol. 5, pp. 275-286, 1968.

[125] T. Johansson and D. Lesnic, Determination of a spacewise dependent heat source, Journal of Computational and Applied Mathematics, vol. 209, pp. 66-80, 2007.

[126] A. Shidfar, G.R. Karamali and J. Damirchi, An inverse heat conduction problem with a nonlinear source term, Nonlinear Analysis, vol. 65, pp. 615-621, 2006.

[127] A. Farcas and D. Lesnic, The boundary-element method for the determination of a heat source dependent on one variable, Journal of Engineering Mathematics, vol. 54, pp. 375-338, 2006.

[128] O. Pironneau, Optimal Shape Design for Elliptic Systems, Springer-Verlag, New York, 1984

[129] L. Rondi, Optimal stability estimates for determination of defects by electrostatic measurements, Inverse Problems, vol. 15, pp. 1193-1212, 1999.

[130] N.D. Aparicio and M.K. Pidcock, The boundary inverse problem for the Laplace equation in two dimensions, Inverse Problems, vol. 12, pp. 565-577, 1996.

[131] S. Peneau, Y. Jarny and A. Sardra, Isotherm shape identification for a two dimensional heat conduction problem, Inverse Problems in Engineering Mechanics, pp. 47-53, 1994.

[132] E. Beretta and S. Vessella, Stable determination of boundaries from Cauchy data, SIAM Journal on Mathematical Analysis, vol. 30, pp. 220-232, 1998.

[133] Y.C. Hon and Z. Wu, A numerical computation for inverse boundary determination problem, Engineering Analysis with Boundary Elements, vol. 24, pp. 599-606, 2000.

[134] P. Hunter and A. Pullan, FEM/BEM Notes, The University of Auckland, New Zealand, 2001

[135] P.K. Kythe, An Introduction to Boundary Element Methods, CRC Press, Boca Raton, 1995.

[136] F.S. Levin, An Introduction to Quantum Theory, Cambridge University Press, Cambridge, 2002.

[137] H.T. Banks and F. Kojima, Boundary shape identification problems in two-dimensional domains related to thermal testing of materials, Hampton, Columbus, 1988.

[138] Y. Sunahara and F. Kojima, Boundary identification for a two dimensional diffusion system under noisy observations, Proc. 4th IFAC Symp. Control of Distributed Parameter Systems, UCLA, California, 1986, Pergamon Press, New York.

[139] M. Nowostawski and R. Poli, Parallel genetic algorithm taxonomy, Third International Conference Knowledge-Based Intelligent Information Engineering Systems, Adelaide, SA, Australia, pp. 88-92, 1999.

[140] V. Isakov, Inverse Problems for Partial Differential Equations, Springer, Berlin, 1998.

[141] J. Fliege, Steepest descent methods for multicriteria optimisation, Mathematical Methods of Operations Research, vol. 51, pp. 479-494, 2000.

[142] M. Renardy and R.C. Rogers, An Introduction to Partial Differential Equations, Springer-Verlag, Berlin, 1993.

[143] V. Agarwal, Total variation regularization and L-curve method for the selection of regularization parameter, ECE599, 2003.

[144] A.S. Householder, Principles of Numerical Analysis, McGraw-Hill, New York, 1953.

[145] N. Tian, J. Sun, W. B. Xu, C-H. Lai, An improved quantum-behaved particle swarm optimization with perturbation operator and its application in estimating groundwater contaminant source. Inverse Problems in Science and Engineering, vol. 19 (2), pp. 181-202, 2011.