國立臺灣師範大學

An ordinary differential equation approach for nonlinear programming and nonlinear complementarity problem

by

Irfan Nurhidayat

A thesis submitted in partial satisfaction of the
requirements for the degree of
Master of Science in Mathematics

An optimization research group

in the

Department of Mathematics

of

National Taiwan Normal University, Taipei City

Supervisor:
Professor Jein-Shan Chen

January 7, 2019

Contents

List of Tables

List of Figures

Acknowledgement

I would first like to thank my thesis advisor Professor Jein-Shan Chen of the Department of Mathematics, National Taiwan Normal University. The door to Professor Chen office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work but steered me in the right the direction whenever he thought I needed it.

I would also like to thank the experts who were involved in the validation survey for this research thesis to examiners team, especially Professor Chun-Hsu Ko from the Department of Electrical Engineering, I-Shou University, and Professor Yu-Lin Chang from the Department of Mathematics, National Taiwan Normal University. Without their passionate participation and input, the validation survey could not have been successfully conducted.

I would also like to acknowledge Professor Chu-Chin Hu of the Department of Mathematics, National Taiwan Normal University, as the second reader of this thesis, and I am gratefully indebted to her for her very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my parents and to my research partners over providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Dedication

I dedicate this thesis to my parents,
Moh. Hidayat and Dedeh Nuridah,
over their affections, loves, and prayers for my success in my life.

Saya mempersembahkan tesis ini untuk orang tua saya,
Moh. Hidayat dan Dedeh Nuridah,
atas kasih sayang, cinta, dan do'a mereka untuk kesuksesan saya dalam hidup saya.

Taipei City, January 7, 2019
Irfan Nurhidayat

Abstract

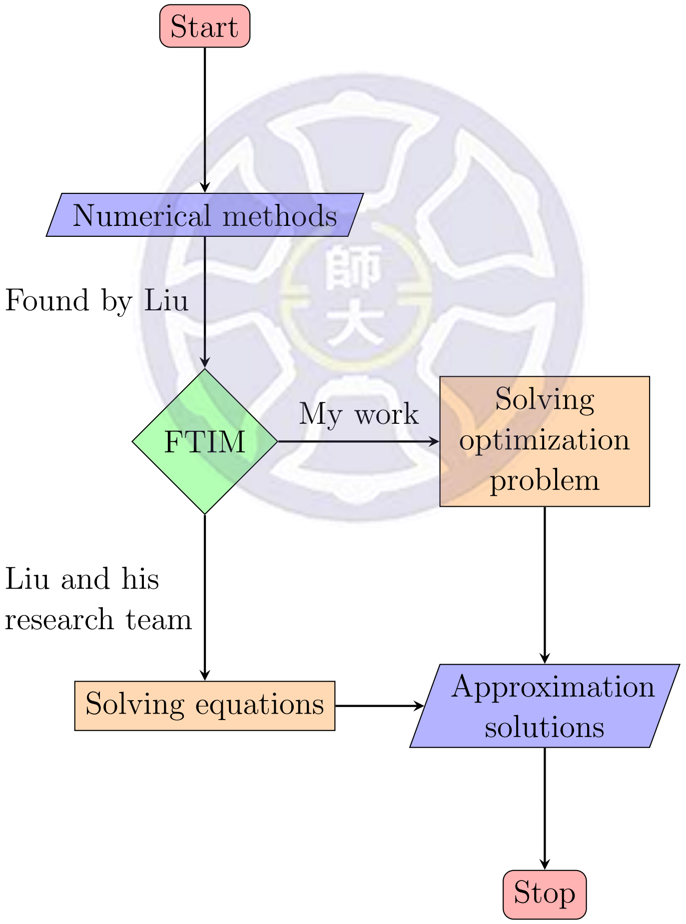
In this thesis, we consider an ordinary differential equation (ODE) approach for solving nonlinear programming (NLP) and nonlinear complementarity problem (NCP). The Karush-Kuhn Tucker (KKT) optimality conditions of NLP and NCP are used to get the new NCP-functions. A special technique is employed to reformulate of the NCP as the system of nonlinear algebraic equations (NAEs) later on reformulated once more by force of an original time-like function into an ordinary differential equation (ODE). Afterwards, a group preserving scheme (GPS) is a package to reformulate an ODE into the new numerical equation in a way the ODEs system is designed into a nonlinear dynamical system (NDS) and is continued to a discovery the new numerical equation through activating the Lorentz group $\mathbf{SO}_0(n,1)$ and its Lie algebra $\mathfrak{so}(n,1)$. Lastly, the fictitious time integration method (FTIM) is utilized into this new numerical equation to determine an approximation solution in numerical experiments area.

Keywords. NLP, NCP, ODE, FTIM.

AMS classifications. 49K15, 65K05, 65B99.

Research history

The flowchart in this page guarantees this research is newest and being initial research to apply the FTIM inside optimization areas deeply. Even if Liu and Atluri (2008a) had proposed the idea concerning the FTIM for solving a nonlinear optimization problem formerly, and they awfully succeeded, yet afterwards, they did not exactly develop their FTIM at optimization areas deeply.



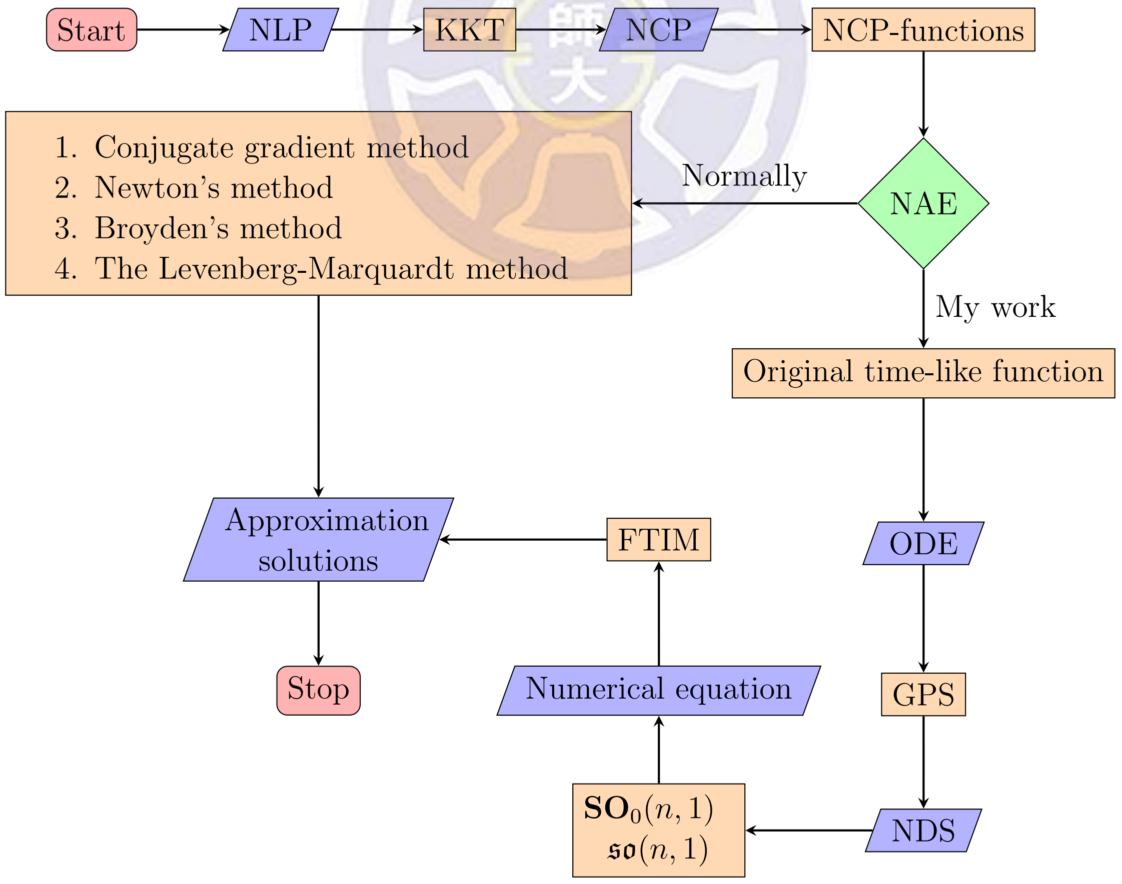
Chapter 1

Motivation and Introduction

This chapter consists of two sections, that is motivation which provides us with information by way of presented a flowchart to open up the main idea in this research, what the goal will be achieved in this research, and why this research must be carried out. Temporarily, the introduction section is a short description from the motivation section encompassing into the relevant background, a past related work, and the proposed solution where the reader is pushed up to fascinate concerning this research.

1.1 Motivation

The researchers have been deeply discussing concerning solving the nonlinear algebraic equations (NAEs) directly by using the optimization approaches. For instance, in this research, we employed the conjugate gradient method, Newton's method, Broyden's method, and the Levenberg-Marquardt method to find out the solutions of NAEs. Nowadays, lots of researchers have been solving NAE problems directly by the olden way without reformulated into ODE as other ways to find approximation solutions of numerical methods. Actually, Liu and his colleague have been using this FTIM in various sciences such as engineering, even now continuously developed in plentiful areas. They found the fictitious time integration method (FTIM) and became their popular methods to solve a variety of the numerical problems currently, but in several ways, his research was not exactly focused into the optimization areas. We purpose to adopt and introduce their ideas of the fictitious time integration method (FTIM) in the optimization areas as the newest alternative to solve numerical problems in optimization problems. The discovery of this research is to offer the fictitious time integration method (FTIM) as a better method than the methods were normally used by researchers in undertaking numerical approximations to solve the optimization problems. A reformulating from numerical algebraic equations (NAEs) into an ordinary differential equation (ODE) is a way to get a new numerical equation before applying FTIM as a numerical method for solving optimization problems by means of the approximation solutions. The flowchart below will expose the main idea of the research and also summarizes the important information through keywords away from every single chapter which will be discussed later on.



1.2 Introduction

We consider an ordinary differential equation approach for solving nonlinear programming and nonlinear complementarity problem. The nonlinear programming (NLP) is

$$\begin{aligned} &\text{Minimize} && f(\mathbf{x}) \\ &\text{subject to} && -g_i(\mathbf{x}) \le 0 \ , \ [i = 1, 2, ..., l], \\ &&& h_i(\mathbf{x}) = 0 \ , \ [i = 1, 2, ..., k], \\ &&& \mathbf{x} \in X, \end{aligned}$$

where X be a nonempty open set in $\mathbb{R}^n$, and let $f : \mathbb{R}^n \to \mathbb{R}$, $g_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, 2, ..., l$, and $h_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, 2, ..., k$, are differentiable functions. The Lagrange function is given by

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) - \langle \boldsymbol{\lambda}, \mathbf{g}(\mathbf{x}) \rangle + \langle \boldsymbol{\mu}, \mathbf{h}(\mathbf{x}) \rangle \tag{1.1}$$

under some constraint qualifications (e.g. linear independence constraint qualification (LICQ) and Mangasarian-Fromovitz constraint qualification (MFCQ)), there exist $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ such that the Karush-Kuhn-Tucker (KKT) conditions of nonlinear programming (NLP) and nonlinear complementarity problem (NCP) are described as follows

$$\begin{array}{lr} \text{optimality condition:} & \nabla_x L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0}, \\ \text{feasible condition:} & -g_i(\mathbf{x}) \le 0, \ h_i(\mathbf{x}) = 0, \\ \text{multiplier condition:} & \lambda_i \ge 0, \\ \text{complementarity condition:} & \langle \lambda_i, g_i(\mathbf{x}) \rangle = 0. \end{array} \tag{1.2}$$

In the above system, these constraints

$$\lambda_i \ge 0, \ g_i(\mathbf{x}) \ge 0, \ \langle \lambda_i, g_i(\mathbf{x}) \rangle = 0 \Longleftrightarrow \phi_k(\lambda_i, g_i(\mathbf{x})) = 0 \tag{1.3}$$

consist of a nonlinear complementarity problem.

In the past decades, nonlinear complementarity problem (NCP) has attracted much attention due to its various applications in operations research, economics, engineering, game theory (Ferris and Pang, 1997; Harker and Pang, 1990; Pang, 1994), which aims to find a point $x \in \mathbb{R}^n$ such that

$$x \geq 0,\ F(x) \geq 0,\ \langle x, F(x)\rangle = 0, \tag{1.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and $F := (F_1, ..., F_n)^T$ is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$. Until now, there exist four primary means for solving these complementarity problems, which are the merit function approach, nonsmooth approach, smoothing function approach, and regularization approach. A moment ago, Chen and his research team are extending a new method that is a neural networks approach as another alternative way for solving nonlinear complementarity problem. Nevertheless, nonlinear complementarity function has an extraordinary impact on these methods to deal with practical problems modelling by NCPs, which satisfies the key condition

$$\phi(a,b) = 0 \Longleftrightarrow a \geq 0,\ b \geq 0,\ ab = 0, \tag{1.5}$$

where $\phi : \mathbb{R}^2 \to \mathbb{R}$ is the NCP function. Focusing on this topic, plenty of NCPs have been investigated (Sun and Qi, 1999; Chen, 2007; Kanzow *et. al.*, 1997; Tseng, 1996; De Luca *et. al.*, 1996; Huang *et. al.*, 2017). Recalling to Chen (2007) and Huang *et. al.* (2017), we are going to construct a variety of definitions from the new NCP-functions that is $\phi_k(a,b),\ [k = 1,2,3,4,5]$, such as

$$\phi_1(a,b) := \sqrt[p]{|a|^p + |b|^p} - (a+b), \tag{1.6}$$

$$\phi_2(a,b) := \left(\sqrt{a^2+b^2}\right)^p - (a+b)^p, \tag{1.7}$$

$$\phi_3(a,b) := a^p - (a-b)_+^p, \tag{1.8}$$

$$\phi_4(a,b) := \begin{cases} \phi_3(a,b), & \text{if } \ a > b, \\ a^p = b^p, & \text{if } \ a = b, \\ \phi_3(b,a), & \text{if } \ a < b, \end{cases} \tag{1.9}$$

$$\phi_5(a,b) := \begin{cases} \phi_3(a,b)b^p, & \text{if } \ a > b, \\ a^p b^p = a^{2p}, & \text{if } \ a = b, \\ \phi_3(b,a)a^p, & \text{if } \ a < b. \end{cases} \tag{1.10}$$

Many approaches that have been shown in (Burden and Faires, 2011; Mangasarian, 1976; Yamashita and Fukushima, 1997), are to reformulate the NCP as a system of NAE. Pourrajabian *et. al.* (2013), solving of nonlinear algebraic equations is a prominent problem in science and engineering. The definition of the system of nonlinear equations is given below

$$F_i(x_j) = 0, \ [i, j = 1, 2, ..., n], \tag{1.11}$$

where $x_j \in \mathbb{R}$ and each $F_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a nonlinear real function. According to Liu and Atluri (2008a), we have a new method to reformulate of the complementarity problem (1.4) as the system of nonlinear algebraic equations (NAEs) like as the equation below

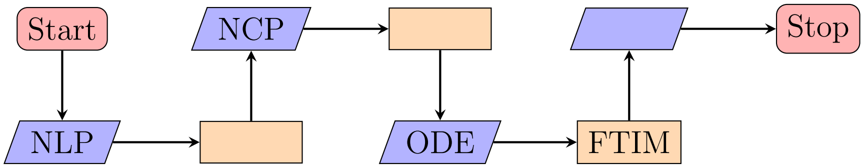
$$F_i(x_1, ..., x_n) := \phi_k(P_i(x_1, ..., x_n), Q_i(x_1, ..., x_n)), \tag{1.12}$$

for some mapping $\phi_k : \mathbb{R}^2 \rightarrow \mathbb{R}$. In other words, this property guarantees that a vector $(x_1, ..., x_n) \in \mathbb{R}^n$ is a solution of the complementarity problem (1.4) if and only if $(x_1, ..., x_n)$ solves the equations system (1.11) (Liu and Atluri, 2008a). The system of nonlinear algebraic equations (NAEs) is able to be transformed into an ordinary differential equation (ODE) by force of a time-like function. As a fabulous discovery by Liu and Atluri in (2008), they have been using the time-like function for the first time within their papers that is in (2008b) and one of the popular paper is in (2009). Their researches team, Ku and friends by way of writing a paper entitled applications of the FTIM using a new time-like function are as well making clear of time-like function on its paper. By the ideas of Liu and their colleagues, in this thesis too, we will bring off a transformation away from nonlinear algebraic equations (NAEs) into ordinary differential equations (ODEs) by way of using a time-like function. Since did a study by ways of seeing, reading, and other literature reviews in Liu's paper is in most cases used a group preserving scheme inside committing a grouping of the ordinary differential equations. A group preserving scheme (GPS) is used to preserve ordinary differential equations (ODEs) such that the transformation of ODEs into numerical equation happens. In this thesis, it recently will be alluded to step by step how to derive the system of ordinary differential equations into numerical equations by utilizing a group preserving scheme.

Numerical methods for ordinary differential equations are methods used to find numerical approximations to the solution of ordinary differential equations (ODEs). Ordinary differential equations occur in many scientific disciplines, for instance in physics, chemistry, biology, engineering, and economics. As well as previously we have been knowing that is the numerical methods of ordinary differential equations consist of Picard's theorem, one-step methods, linear multi-step methods, secant method, bisection method, Newton's method, Broyden's method which is sometimes called a Quasi-Newton method where it is derived from Newton's method, finite difference method that is used to solve boundary value problems of nonlinear ordinary differential equations, and probably there are still many methods will be found in the next future. One decade ago, Liu and Atluri (2008a) proposed the idea of using the FTIM for solving a nonlinear optimization problem (NOP) under multiple equality and inequality constraints. A novel time integration method is named the fictitious time integration method (FTIM) has been proposed by Liu and Atluri (2008b). The fictitious time integration method (FTIM) was the first used to solve an nonlinear equation by introducing fictitious time (Liu, [2008a, 2008b]). The most leading paper of FTIM that is a solution two-dimensional quasilinear elliptic boundary value problems (BVPs) has been worked by Liu (2008c). Liu (2008d), also has introduced how to use FTIM to solve the nonlinear obstacle problems based on a time-marching algorithm. The using of FTIM for solving the discretized inverse Sturm-Liouville problems and m-point boundary value problems (BVPs) also has been proposed by Liu (2008b, 2009).

The goal of this thesis is to offer the fictitious time integration method (FTIM) as an alternative simulation better in nonlinear optimization problems (NOPs). The contribution of this thesis is a great advantage to easily extend to higher dimensional nonlinear optimization problems with nonlinear equality and inequality constraints. The organizations of this thesis follow, chapter 1, we focus on introducing some references related to NLP, NCP, ODE, FTIM, and several stories from previous researchers, chapter 2, is the basic theories like an introduction of the original time-like or fictitious time function, the Lorentz group $\mathbf{SO}_0(n,1)$ and the Lie algebra $\mathfrak{so}(n,1)$, and lastly pertaining to the performance profile concepts. Chapter 3, is the ODE reformulation to give us specific knowledge out of how to derive NAEs to ODEs and also a GPS to preserve ODEs such that obtained the numerical equation. Chapter 4, focuses on the numerical experiments by using MATLAB R2015a to compare ϕ_k^p to each other together as well with performed the performance profiles analysis, all of the results of the functions are gotten in the form of the graphics and tables. Finally, chapter 5, is the conclusions of this research covering a tidy package of main contents in this thesis.

The research methods of this thesis is a literature study, namely by studying the material from various references related to study such as journals, books, thesis even dissertation, and others. The research methodology of this thesis like a systematic way to solve the problem will be emphasized inside under these topics i.e.



Chapter 2

Preliminaries

We recall some basic concepts regarding the original time-like function summarized from (Liu *et. al.*, 2008). The basic theory of the Lorentz group $\mathbf{SO}_0(n,1)$ and the Lie algebra $\mathfrak{so}(n,1)$ summarized and is adapted from (Gallier and Quaintance, 2017) while the performance profile theory in the numerical test adapted from (Dolan and More, 2002). These theories will become the fundamental concepts to construct the main topics of this thesis in the next chapter.

2.1 Original time-like function

Based on the idea of introducing a fictitious time, the FTIM can transform a system of linear or nonlinear algebraic equation into an ODEs system. The fictitious time function that used in the transformation was named time-like function (Ku *et. al.*, 2009). The numerical examples are given to confirm that this fictitious time integration method (FTIM) is highly efficient to find the true solutions with residual errors being much smaller. The FTIM is used as well to study the attracting sets of fixed points, when multiple roots exist (Liu and Atluri, 2008b). In their studies, a more general form of the time-like function i.e.

$$q(\tau) =(1+\tau)^{\gamma},\ 0 \leq \gamma \leq 1, \tag{2.1}$$

with τ is a fictitious time variable. Liu and Atluri (2008b), have the first shown that the time-like function (2.1), has to be differentiable, that is $q(0) = 1$ and $q(\infty) = \infty$. In this thesis, we used the original time-like function by $\gamma = 1$ to transform an algebraic equation into the first order ordinary differential equation (FOODE).

2.2 The Lorentz group $\mathbf{SO}_0(n, 1)$ and the Lie algebra $\mathfrak{so}(n, 1)$

We presently would forward to see a little bit relating to the Lie algebra of the Lorentz group. Started by the Lorentz group $\mathbf{SO}_0(n, 1)$ is the proper orthochronous Lorentz group and its Lie algebra is $\mathfrak{so}(n, 1)$. Both of them are able to be defined as below

$$\begin{aligned}\mathbf{SO}_0(n, 1) &= \{\mathbf{Y} = (y_{ij}) \in \mathbf{SO}(n, 1) | y_{n+1\ n+1} > 0 \text{ for } y_{i=1,\dots,n+1\ j=1,\dots,n+1} \in \mathbb{R}\},\\ \mathfrak{so}(n, 1) &= \{\mathbf{Y} \in \mathrm{M}_{(n+1)\times(n+1)}(\mathbb{R}) | \mathbf{Y}^T\mathbf{J} + \mathbf{J}\mathbf{Y} = \mathbf{0}\},\end{aligned}$$

for $\mathrm{M}_{(n+1)\times(n+1)}(\mathbb{R})$ is a matrix with its sizes $(n + 1) \times (n + 1)$ in $\mathbb{R}$. Based on those definitions are gained a couple of sets as $\mathbf{SO}(n, 1)$ is a special orthogonal group and $\mathbf{GL}(n + 1, \mathbb{R})$ is a general linear group, namely

$$\begin{aligned}\mathbf{SO}(n, 1) &= \{\mathbf{G} \in \mathbf{GL}(n + 1, \mathbb{R}) | \mathbf{G}^T\mathbf{J}\mathbf{G} = \mathbf{J} \text{ and } \det \mathbf{G} = 1\},\\ \mathbf{GL}(n + 1, \mathbb{R}) &= \{\mathbf{G} \in \mathrm{M}_{(n+1)\times(n+1)}(\mathbb{R}) | \mathbf{G} \text{ invertible}\}.\end{aligned}$$

Here is alluded slightly pertaining to skew-symmetric $\mathbf{JY}$ in Lie algebra $\mathfrak{so}(n, 1)$ by reason of $\mathbf{J} = \mathbf{J}^T$ as below

$$\mathfrak{so}(n, 1) = \left\{ \begin{bmatrix} B & u \\ u^T & 0 \end{bmatrix} \in \mathrm{M}_{(n+1)\times(n+1)}(\mathbb{R}) \middle| u \in \mathbb{R}^n,\ B^T = -B \right\}.$$

There are many materials to be spoken about Lorentz group and its Lie algebra from (Gallier and Quaintance, 2017), yet at this thesis will not be discussed deeply concerning those. All of the materials which were written here just related to constructing a group preserving scheme (GPS).

2.3 Performance profile

The performance profile is the popular theory in the numerical test from Dolan and More (2002) where has been utilized by researchers to compare the performance of the functions ϕ_k. From the inside of its paper, we aim to summarize a little bit pertaining to the performance profile theory and use it in these numerical experiments. In consequence, we assume the functions of ϕ_k as a solver, there is n_s solver, and also n_p is test problem from the test set $\mathcal{P}$ which is generated randomly. For each problem, p and solver s, are given by

$$f_{p,s} = \text{iteration number is required to solve problem } p \text{ by solver } s.$$

We are using the performance ratio

$$r_{p,s} := \frac{f_{p,s}}{\min\{f_{p,s} : s \in \mathcal{S}\}},$$

where $\mathcal{S}$ is the five solvers set. Assuming of a parameter $r_{p,s} \leq r_M, \forall p, s$ which is chosen, and $r_{p,s} = r_M$ as well if and only if solver s does not solve the problem p. In order to obtain an overall approximation for each solver, we now define

$$\rho_s(\tau) := \frac{1}{n_p}\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\}$$

which is called the performance profile of the number of iteration for solver s. Then, $\rho_s(\tau)$ is the probability for solver $s \in \mathcal{S}$, that is a performance ratio $f_{p,s}$ inside a factor $\tau \in \mathbb{R}$ of the best possible ratio.

Chapter 3

ODE reformulation

This chapter is devoted to the main topic in this thesis which covers a transformation into ordinary differential equations (ODEs) system and a group preserving scheme (GPS) for differential equations system is the important requirements to derive a numerical formula, then lastly, the numerical equation which will be employed in the next chapter is given in this chapter.

3.1 Transformation into an ODEs system

Let us consider the following nonlinear algebraic equations (NAEs) system

$$F_i(x_j) = 0,\ [i, j = 1, 2, ..., n]. \tag{3.1}$$

Recalling (2.1), we redefine an original time-like function below

$$\begin{aligned} y_i(\tau) &= q(\tau)x_j,\ [i, j = 1, 2, ..., n], \\ q(\tau) &= 1 + \tau, \end{aligned} \tag{3.2}$$

where τ is a variable which is independent of x_j. Taking the derivative of (3.2) with respect to τ, we have

$$\dot{y}_i = \frac{dy_i}{d\tau} = x_j. \tag{3.3}$$

If $z \neq 0$, then (3.1) is equivalent to

$$0 = -zF_i(x_j). \tag{3.4}$$

By using (3.2), we obtain

$$0 = -zF_i\left(\frac{y_i}{1+\tau}\right). \tag{3.5}$$

Adding (3.3), we get

$$\dot{y_i} = x_j - zF_i\left(\frac{y_i}{1+\tau}\right). \tag{3.6}$$

By (3.2), we have $x_j = y_i/(1+\tau)$ that means we can write (3.6) as an ODEs system for y_i like

$$\dot{y_i} = \frac{y_i}{1+\tau} - zF_i\left(\frac{y_i}{1+\tau}\right). \tag{3.7}$$

Multiplying of each equation by the integrating factor $1/(1+\tau)$ and used (3.7) once more so that

$$\frac{\dot{y_i}}{1+\tau} = \frac{y_i}{(1+\tau)^2} - \frac{z}{1+\tau}F_i\left(\frac{y_i}{1+\tau}\right). \tag{3.8}$$

Others were known that

$$\frac{d}{d\tau}\left(\frac{y_i}{1+\tau}\right) = \frac{\dot{y_i}}{1+\tau} - \frac{1}{1+\tau}\left(\frac{y_i}{1+\tau}\right). \tag{3.9}$$

By utilizing (3.2) and (3.9), then (3.8) can be rewritten becoming

$$\frac{d}{d\tau}x_j = -\frac{z}{1+\tau}F_i(x_j),\ [i, j = 1, 2, ..., n]. \tag{3.10}$$

Further (3.10) becomes

$$\dot{x_i} = -\frac{z}{1+\tau}F_i(x_1, ..., x_n). \tag{3.11}$$

The above idea is the first proposed by Liu (2008e) to treat an inverse Sturm-Liouville problem by transforming an ODE into a PDE. Then, (Liu, [2008a, 2008f]) and (Liu *et. al.*, 2008), extended this idea to develop new methods for estimating parameters in the inverse vibration problems. We now may employ (3.11) to develop a more stable group preserving scheme (GPS) where will be discussed in the next section, formerly we are going to rewrite (3.11) as a vector form that is

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \tau) = -\frac{z}{1+\tau}\mathbf{F}(\mathbf{x}),\ [\mathbf{x} = (x_1, ..., x_n, \mu_1, ..., \mu_k, \lambda_1, ..., \lambda_l) \in \mathbb{R}^N,\ \tau > 0], \tag{3.12}$$

where $N = n + k + l$ is the dimensional state vector of algebraic equations. We currently define the system of nonlinear algebraic equations (NAEs) of (3.1) as below

$$F_i(x_j) := \phi_k(P_i(x_j), Q_i(x_j)),\ [i, j = 1, 2, ..., n \text{ and } k = 1, 2, 3, 4, 5]. \tag{3.13}$$

In other words, we get

$$F_i = \phi_k(P_i, Q_i) = 0.$$

We rewrite (3.11) as well as systems of ODE i.e.

$$\begin{aligned} Q_1 : \dot{x}_i &= -\frac{z_1}{1+\tau}F_1(x_i),\ [x_i \in \mathbb{R}^n], \\ Q_2 : \dot{\mu}_i &= -\frac{z_2}{1+\tau}F_2(\mu_i),\ [\mu_i \in \mathbb{R}^k], \\ H_k : \dot{\lambda}_i &= -\frac{z_3}{1+\tau}F_3(\lambda_i),\ [\lambda_i \in \mathbb{R}^l]. \end{aligned}$$

The different coefficients of $z_1, z_2,$ and z_3 can be used to enhance the stability of numerical integrations of their equations. More far away,

$$Q_1 = -\frac{z_1}{1+\tau}\left[\frac{\partial f}{\partial x_i} + \sum_{j=1}^{k}\mu_j\frac{\partial h_j}{\partial x_i} - \sum_{j=1}^{l}\lambda_j\frac{\partial g_j}{\partial x_i}\right], \tag{3.14}$$

$$Q_2 = -\frac{z_2}{1+\tau}h_i, \tag{3.15}$$

$$H_k = -\frac{z_3}{1+\tau}\phi_k(\lambda_i, g_i). \tag{3.16}$$

The equations (3.14)-(3.16) are henceforward called as systems of ordinary differential equations (ODEs) which one will be used to construct the next chapters of this thesis.

3.2 GPS for differential equations system

We separate into two topics that are group preserving scheme (GPS) and one-step GPS which both of them respectively would be discussed here.

3.2.1 Group preserving scheme

The above equations (3.14)-(3.16) can be combined together into the nonlinear dynamical system (NDS) as follows

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \tau) \Longleftrightarrow \dot{\mathbf{x}}^T = [\mathbf{f}(\mathbf{x}, \tau)]^T, \ \mathbf{x} \in \mathbb{R}^N, \tag{3.17}$$

with $\mathbf{x} = (x_1, ..., x_n, \mu_1, ..., \mu_k, \lambda_1, ..., \lambda_l)$, $\mathbf{x}$ is a $(N = n + k + l)$-dimensional state vector, τ is a time variable, and $\mathbf{f} \in \mathbb{R}^N$ is a vector valued function of $\mathbf{x}$ and τ. The augmented vectors are defined by

$$\mathbf{X} := (\mathbf{x}^T, \|\mathbf{x}\|)^T = (\mathbf{x}, \|\mathbf{x}\|) \in \mathbb{R}^{N+1}. \tag{3.18}$$

A group preserving scheme (GPS) can preserve the internal symmetry group of the considered ODEs system (Liu and Atluri, 2008a). Although we do not know previously the symmetry group of differential equations system. In fact, Liu (2001), has embedded it into an augmented differential equations system, which concerns with not only the evolution of state variables themselves but also the evolution of the magnitude of the state variables vector. By referring to the Lorentz group $\mathbf{SO}_0(n, 1)$ and its Lie algebra $\mathfrak{so}(n, 1)$ in the previous chapter, let us define again more detail that $\mathbf{G}(r) \in \mathbf{SO}_0(n, 1)$ is the group value of $\mathbf{G}$ at a time t_r. $\mathbf{X}_r$ denotes the numerical value of $\mathbf{X}$ at discrete time t_r. Liu (2001) has been writing on his papers concerning to the connecting of $\mathbf{G}(r)$ and $\mathbf{X}_r$ i.e.

$$\mathbf{X}_{r+1} = \mathbf{G}(r)\mathbf{X}_r. \tag{3.19}$$

Referring to Gallier and Quaintance (2017), specifically, we are able to redefine (3.17) as follows

$$\omega := \|\dot{\mathbf{x}}\| = \|\mathbf{f}_r(\mathbf{x}, \tau)\|, \ r \in \mathbb{Z}^+ \cup \{0\},$$

so that

$$\mathbf{Y}(r) = \begin{bmatrix} 0 & \mathbf{f}_r(\mathbf{x}, \tau) \\ [\mathbf{f}_r(\mathbf{x}, \tau)]^T & 0 \end{bmatrix} \in \mathfrak{so}(n, 1).$$

The Lie group $\mathbf{G}(r)$ can be generated from $\mathbf{Y}(r) \in \mathfrak{so}(n,1)$ by an exponential mapping

$$\exp : \mathfrak{so}(n,1) \to \mathbf{SO}_0(n,1),$$

wherever

$$\begin{aligned} e^{\mathbf{Y}(r)} &= \begin{bmatrix} I_n + \frac{(\cosh\omega - 1)}{\omega^2}\mathbf{f}_r\mathbf{f}_r^T & \frac{\sinh\omega}{\omega}\mathbf{f}_r \\ \frac{\sinh\omega}{\omega}\mathbf{f}_r^T & \cosh\omega \end{bmatrix} \\ &= \begin{bmatrix} I_n + \frac{(\cosh\|\mathbf{f}_r\| - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r\mathbf{f}_r^T & \frac{\sinh\|\mathbf{f}_r\|}{\|\mathbf{f}_r\|}\mathbf{f}_r \\ \frac{\sinh\|\mathbf{f}_r\|}{\|\mathbf{f}_r\|}\mathbf{f}_r^T & \cosh\|\mathbf{f}_r\| \end{bmatrix}. \end{aligned} \tag{3.20}$$

We are allowed to modify (3.20) into

$$e^{\frac{\Delta t}{\|\mathbf{x}_r\|}\mathbf{Y}(r)} = \begin{bmatrix} I_n + \frac{\left(\cosh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) - 1\right)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r\mathbf{f}_r^T & \frac{\sinh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)}{\|\mathbf{f}_r\|}\mathbf{f}_r \\ \frac{\sinh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)}{\|\mathbf{f}_r\|}\mathbf{f}_r^T & \cosh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) \end{bmatrix}.$$

For some $\Omega := \frac{\Delta t}{\|\mathbf{x}_r\|}$, we have

$$e^{\Omega\mathbf{Y}(r)} = \begin{bmatrix} I_n + \frac{(\cosh(\Omega\|\mathbf{f}_r\|) - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r\mathbf{f}_r^T & \frac{\sinh(\Omega\|\mathbf{f}_r\|)}{\|\mathbf{f}_r\|}\mathbf{f}_r \\ \frac{\sinh(\Omega\|\mathbf{f}_r\|)}{\|\mathbf{f}_r\|}\mathbf{f}_r^T & \cosh(\Omega\|\mathbf{f}_r\|) \end{bmatrix}.$$

More far away,

$$\mathbf{G}(r) := e^{\Omega\mathbf{Y}(r)} = \begin{bmatrix} I_n + \frac{(a_r - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r\mathbf{f}_r^T & \frac{b_r\mathbf{f}_r}{\|\mathbf{f}_r\|} \\ \frac{b_r\mathbf{f}_r^T}{\|\mathbf{f}_r\|} & a_r \end{bmatrix} \in \mathbf{SO}_0(n,1), \tag{3.21}$$

where

$$a_r := \cosh(\Omega\|\mathbf{f}_r\|),$$

$$b_r := \sinh(\Omega\|\mathbf{f}_r\|).$$

Substituting (3.21) into (3.19) and then exploiting (3.18) as well, are obtained

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{r+1} \\ \|\mathbf{x}_{r+1}\| \end{bmatrix} &= \begin{bmatrix} I_n + \frac{(a_r - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r\mathbf{f}_r^T & \frac{b_r\mathbf{f}_r}{\|\mathbf{f}_r\|} \\ \frac{b_r\mathbf{f}_r^T}{\|\mathbf{f}_r\|} & a_r \end{bmatrix} \begin{bmatrix} \mathbf{x}_r \\ \|\mathbf{x}_r\| \end{bmatrix} \\ \Longleftrightarrow \begin{bmatrix} \mathbf{x}_{r+1} \\ \|\mathbf{x}_{r+1}\| \end{bmatrix} &= \begin{bmatrix} \mathbf{x}_r + \frac{\mathbf{x}_r(a_r - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r\mathbf{f}_r^T + \frac{b_r\mathbf{f}_r}{\|\mathbf{f}_r\|}\|\mathbf{x}_r\| \\ \frac{b_r\mathbf{f}_r^T}{\|\mathbf{f}_r\|}\mathbf{x}_r + a_r\|\mathbf{x}_r\| \end{bmatrix}. \end{aligned}$$

Since we know that $\mathbf{f}_r^T = \mathbf{f}_r$, accordingly

$$\begin{aligned}
\mathbf{x}_{r+1} &= \mathbf{x}_r + \frac{\mathbf{x}_r(a_r - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r\mathbf{f}_r^T + \frac{b_r\mathbf{f}_r}{\|\mathbf{f}_r\|}\|\mathbf{x}_r\| \\
&= \mathbf{x}_r + \frac{\mathbf{x}_r(a_r - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r \cdot \mathbf{f}_r + \frac{b_r\mathbf{f}_r}{\|\mathbf{f}_r\|}\|\mathbf{x}_r\| \\
&= \mathbf{x}_r + \left[\frac{\mathbf{x}_r(a_r - 1)}{\|\mathbf{f}_r\|^2}\mathbf{f}_r + \frac{b_r\|\mathbf{x}_r\|}{\|\mathbf{f}_r\|}\right]\mathbf{f}_r \\
&= \mathbf{x}_r + \left[\frac{(a_r - 1)\mathbf{f}_r \cdot \mathbf{x}_r + b_r\|\mathbf{x}_r\|\|\mathbf{f}_r\|}{\|\mathbf{f}_r\|^2}\right]\mathbf{f}_r \\
&= \mathbf{x}_r + \eta_r\mathbf{f}_r,
\end{aligned} \tag{3.22}$$

whereupon

$$\eta_r := \frac{(a_r - 1)\mathbf{f}_r \cdot \mathbf{x}_r + b_r\|\mathbf{x}_r\|\|\mathbf{f}_r\|}{\|\mathbf{f}_r\|^2}.$$

Besides that, we have possession of

$$\begin{aligned}
\|\mathbf{x}_{r+1}\| &= \frac{b_r\mathbf{f}_r^T}{\|\mathbf{f}_r\|}\mathbf{x}_r + a_r\|\mathbf{x}_r\| \\
&= a_r\|\mathbf{x}_r\| + \frac{b_r}{\|\mathbf{f}_r\|}\mathbf{f}_r \cdot \mathbf{x}_r.
\end{aligned} \tag{3.23}$$

As well as we have been knowing previously that η_r is an adaptive factor. Before we get in to prove $\eta_r > 0$, let us take the time to retrieve one simple hyperbolic property below which will be used in the proof

$$\begin{aligned}
&\cosh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) - \sinh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) \\
&= \frac{\exp\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) + \exp\left(-\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)}{2} - \frac{\exp\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) - \exp\left(-\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)}{2} \\
&= \frac{\exp\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) + \exp\left(-\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) - \exp\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) + \exp\left(-\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)}{2} \\
&= \frac{2\exp\left(-\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)}{2} \\
&= \exp\left(-\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right).
\end{aligned}$$

Next, we are going to give a proof for $\eta_r > 0$ started by the definition of Schwartz inequality that is $\mathbf{f}_r \cdot \mathbf{x}_r \geq -\|\mathbf{f}_r\|\|\mathbf{x}_r\|$ so as

$$
\begin{aligned}
&(a_r - 1)\mathbf{f}_r \cdot \mathbf{x}_r \geq -(a_r - 1)\|\mathbf{f}_r\|\|\mathbf{x}_r\| \\
\Longrightarrow\ & b_r\|\mathbf{x}_r\|\|\mathbf{f}_r\| + (a_r - 1)\mathbf{f}_r \cdot \mathbf{x}_r \geq b_r\|\mathbf{x}_r\|\|\mathbf{f}_r\| - (a_r - 1)\|\mathbf{f}_r\|\|\mathbf{x}_r\| \\
\Longrightarrow\ & b_r\|\mathbf{x}_r\|\|\mathbf{f}_r\| + (a_r - 1)\mathbf{f}_r \cdot \mathbf{x}_r \geq [1 - a_r + b_r]\|\mathbf{f}_r\|\|\mathbf{x}_r\| \\
\Longrightarrow\ & \frac{b_r\|\mathbf{x}_r\|\|\mathbf{f}_r\| + (a_r - 1)\mathbf{f}_r \cdot \mathbf{x}_r}{\|\mathbf{f}_r\|^2} \geq \frac{[1 - a_r + b_r]\|\mathbf{f}_r\|\|\mathbf{x}_r\|}{\|\mathbf{f}_r\|^2} \\
\Longrightarrow\ & \frac{b_r\|\mathbf{x}_r\|\|\mathbf{f}_r\| + (a_r - 1)\mathbf{f}_r \cdot \mathbf{x}_r}{\|\mathbf{f}_r\|^2} \geq [1 - a_r + b_r]\frac{\|\mathbf{x}_r\|}{\|\mathbf{f}_r\|} \\
&\Longrightarrow \eta_r \geq \left[1 - \cosh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) + \sinh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)\right]\frac{\|\mathbf{x}_r\|}{\|\mathbf{f}_r\|} \\
&\Longrightarrow \eta_r \geq \left[1 - \left\{\cosh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right) - \sinh\left(\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)\right\}\right]\frac{\|\mathbf{x}_r\|}{\|\mathbf{f}_r\|} \\
&\Longrightarrow \eta_r \geq \left[1 - \exp\left(-\frac{\Delta t\|\mathbf{f}_r\|}{\|\mathbf{x}_r\|}\right)\right]\frac{\|\mathbf{x}_r\|}{\|\mathbf{f}_r\|} > 0,\ [\forall \Delta t > 0].
\end{aligned}
$$

The ending of the proof, and this scheme preserves the group properties for all $\Delta t > 0$, further is called the group preserving scheme.

3.2.2 One-step GPS

The review related to one-step GPS virtually had been peeled in Liu (2008e) or another his paper, and in this occasion, we probably would like to change some notations and make clear again by wielding another sentence and version. We presently assume that the total time T is divided by K steps, that is the time step size $\Delta t = T/K$. $\mathbf{X}_0$ is an initial condition, $\mathbf{X}_T$ is our calculating value at the desired time T. By harnessing (3.19), we secure

$$\mathbf{X}_T = \mathbf{G}_K(\Delta t)...\mathbf{G}_1(\Delta t)\mathbf{X}_0,$$

where $\mathbf{X}_T$ approximates the exact value within a certain accuracy depending on Δt and every single element $\mathbf{G}_i \in \mathbf{SO}_0(n, 1),\ i = 1, ..., K$. Next, by means of the closure property of Lie group, $\mathbf{G}_K(\Delta t)...\mathbf{G}_1(\Delta t)$ are a Lie group as well denoted by $\mathbf{G}(T)$. Therefore, we have

$$\mathbf{X}_T = \mathbf{G}(T)\mathbf{X}_0,$$

and outright this is a transformation one-step from $\mathbf{X}_0$ to $\mathbf{X}_T$.

In accordance of survey, the researchers who have been studied it, they shared their observations to us it is awfully hard to find out an exact solution of $\mathbf{G}(T)$. However, the researchers found out an alternative way to calculate $\mathbf{G}(T)$, and based on their methods, the most simple method to calculate $\mathbf{G}(T)$ is a special case by (3.21), namely

$$\mathbf{G}(T) = \begin{bmatrix} I_n + \frac{(a-1)}{\|\mathbf{f}_0\|^2}\mathbf{f}_0\mathbf{f}_0^T & \frac{b\mathbf{f}_0}{\|\mathbf{f}_0\|} \\ \frac{b\mathbf{f}_0^T}{\|\mathbf{f}_0\|} & a \end{bmatrix},$$

where $\Delta t = T/K = T/1 = T$ so as

$$a := \cosh\left(\frac{T\|\mathbf{f}_0\|}{\|\mathbf{x}_0\|}\right),$$
$$b := \sinh\left(\frac{T\|\mathbf{f}_0\|}{\|\mathbf{x}_0\|}\right).$$

By employing (3.22) and (3.23), we get hold of an one-step GPS, specifically

$$\mathbf{x}_T = \mathbf{x}_0 + \eta\mathbf{f}_0,$$
$$\|\mathbf{x}_T\| = a\|\mathbf{x}_0\| + \frac{b}{\|\mathbf{f}_0\|}\mathbf{f}_0 \cdot \mathbf{x}_0,$$

where

$$\eta = \frac{(a-1)\mathbf{f}_0 \cdot \mathbf{x}_0 + b\|\mathbf{x}_0\|\|\mathbf{f}_0\|}{\|\mathbf{f}_0\|^2}.$$

Chapter 4

Numerical experiments

The simplest numerical method for the solution of initial value problems (IVPs) is the fictitious time integration method (FTIM). By combining (3.12), (3.17), and (3.22), we acquire

$$x_j^{r+1} = x_j^r + \eta_r f_i\big(x_j^r, \tau_r\big), \ [\mathbf{f} = f_i = (f_1, ..., f_N), \ \mathbf{x} = x_j^r = (x_1^r, ..., x_N^r) \in \mathbb{R}^N],$$

where $\tau_{r+1} = \tau_r + \eta_r$. The numerical procedures are able to be started out from an initial value of x_j^0 which can be arbitrarily chosen. In the numerical integration process, we can check out the convergence criterion of x_j at the r- and $(r+1)$- in the manner of

$$\sum_{j=1}^{N}\big(x_j^{r+1} - x_j^r\big)^2 \leq \epsilon^2 \Longleftrightarrow \big\|f_i\big(x_j^{r+1}\big) - f_i\big(x_j^r\big)\big\| \leq \epsilon, \tag{4.1}$$

where ϵ is a selected criterion. If at a time $\tau_0 \leq \tau_r$, then the equation (4.1) is satisfied, further the solution of x_j is obtained.

We report the numerical results of the explanation below for solving the ODEs in the equations (3.14)-(3.16). The numerical experiments are carried out in Matlab running on a PC with Intel i3 of 1.50 GHz CPU processor, 4.00 GB memory, and 32-bit operating system windows 7. Related to the computational experiments, the following parameters are used:

$$z_1 = 0.3, \ z_2 = 0.5, \ z_3 = 0.7, \ \Delta t = 0.001.$$

4.1 Example 1

Let us start from this problem defined by (Liu and Atluri, 2008a) i.e.

$$\begin{aligned} &\text{Minimize} \quad x^2 \\ &\text{subject to} \quad x \geq 1 \end{aligned}$$

After we did manipulate algebra at above problem its Lagrangian is

$$\begin{aligned} f(x) &= x^2, \\ g(x) &= x - 1, \\ L &= x^2 - \lambda(x-1). \end{aligned}$$

By using the KKT conditions and NCP, we get

$$\begin{aligned} &Q_1 : 2x - \lambda = 0, \\ &H_k : \lambda \geq 0,\ x - 1 \geq 0,\ \lambda(x-1) = 0 \Longleftrightarrow \phi_k(\lambda, x-1) = 0. \end{aligned}$$

The ordinary differential equations (ODEs) are

$$\begin{aligned}
Q_1 &= -\frac{z_1}{1+\tau}(2x-\lambda),\\
H_1 &= -\frac{z_3}{1+\tau}\Big[\sqrt[p]{|\lambda|^p+|x-1|^p}-\lambda-x+1\Big],\\
H_2 &= -\frac{z_3}{1+\tau}\Big[\Big(\sqrt{\lambda^2+(x-1)^2}\Big)^p-(\lambda+x-1)^p\Big],\\
H_3 &= -\frac{z_3}{1+\tau}\Big[\lambda^p-(\lambda-x+1)_+^p\Big],\\
H_4 &= -\frac{z_3}{1+\tau}\Big[\lambda^p-(\lambda-x+1)_+^p\Big],\ \text{if}\ \ \lambda>x-1,\\
H_4 &= -\frac{z_3}{1+\tau}\lambda^p,\ \text{if}\ \ \lambda=x-1,\\
H_4 &= -\frac{z_3}{1+\tau}(x-1)^p,\ \text{if}\ \ \lambda=x-1,\\
H_4 &= -\frac{z_3}{1+\tau}\Big[(x-1)^p-(x-1-\lambda)_+^p\Big],\ \text{if}\ \ \lambda<x-1,\\
H_5 &= -\frac{z_3}{1+\tau}\Big[\lambda^p(x-1)^p-(\lambda-x+1)^p(x-1)^p\Big],\ \text{if}\ \ \lambda>x-1,\\
H_5 &= -\frac{z_3}{1+\tau}\Big[\lambda^p(x-1)^p\Big],\ \text{if}\ \ \lambda=x-1,\\
H_5 &= -\frac{z_3}{1+\tau}\lambda^{2p},\ \text{if}\ \ \lambda=x-1,\\
H_5 &= -\frac{z_3}{1+\tau}\Big[\lambda^p(x-1)^p-(x-1-\lambda)^p\lambda^p\Big],\ \text{if}\ \ \lambda<x-1.
\end{aligned}$$

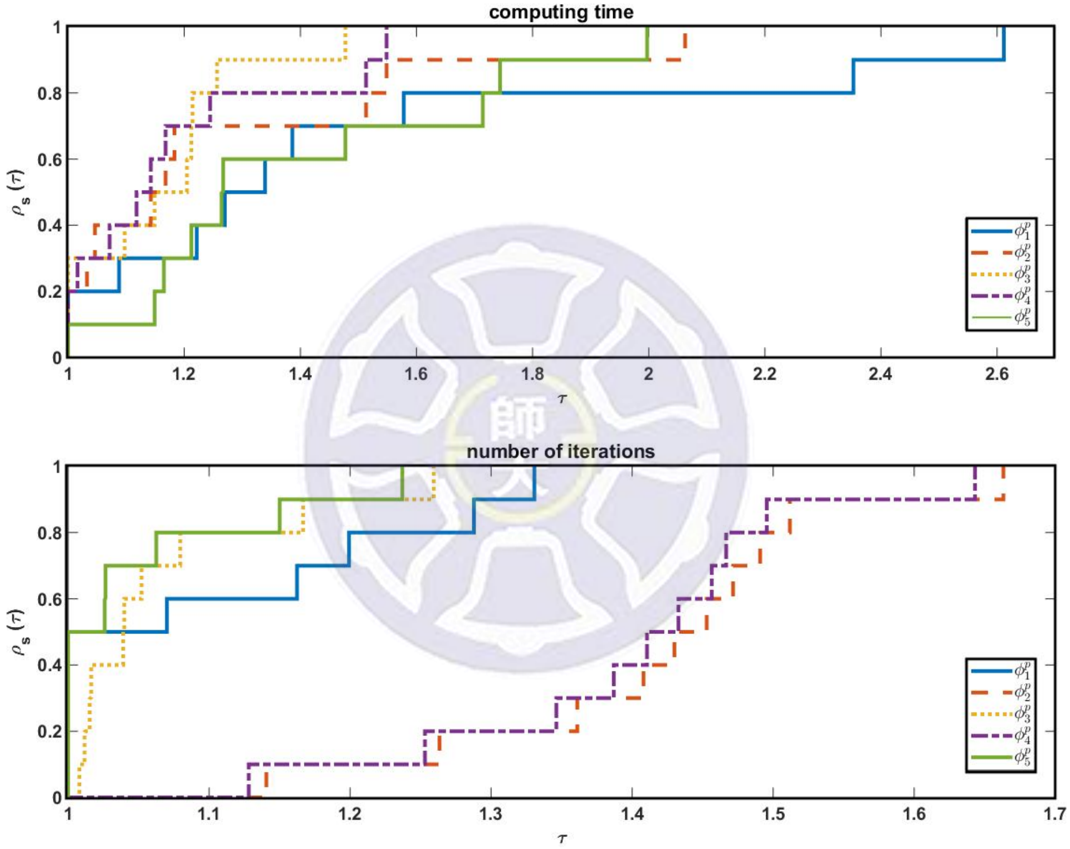


Figure 4.1: Performance profile of ϕ_k^p when $p = 50$, $x_j^0 = 1e - 8$, and $\epsilon = 10^{-5}$ in n dimensions

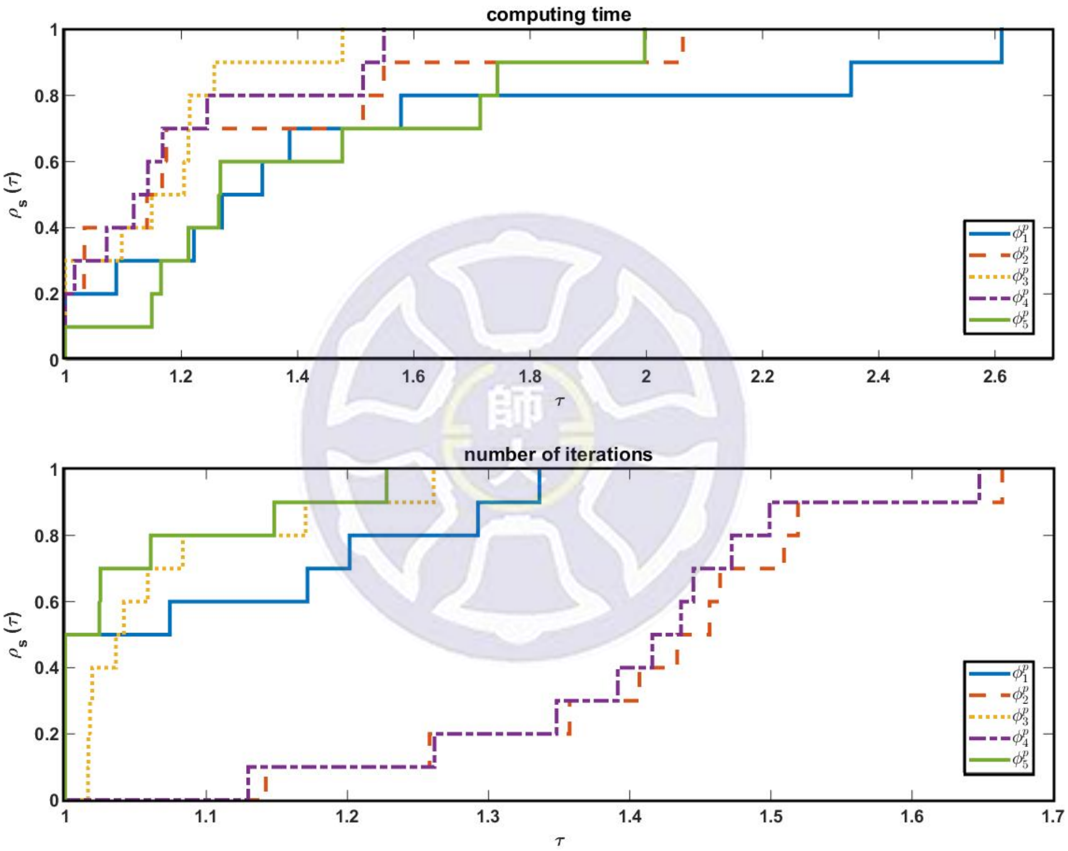


Figure 4.2: Performance profile of ϕ_k^p when $p = 100,\ x_j^0 = 1e{-5},\ \epsilon = 10^{-6}$ in n dimensions

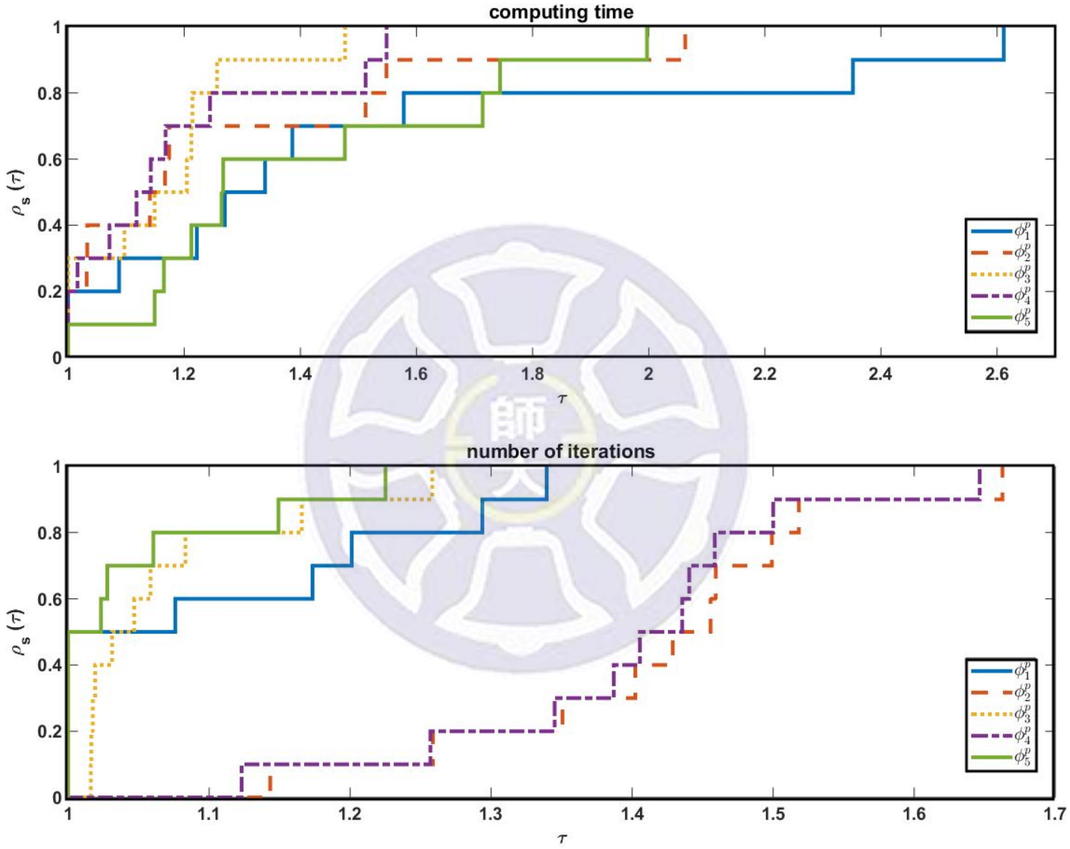


Figure 4.3: Performance profile of ϕ_k^p when $p > 1$, $x_j^0 = 1e-2$, and $\epsilon = 10^{-4}$ in n dimensions

Table 4.1: The solution of ϕ_1 and ϕ_2 in 2 dimensions by $p = 2, 4, 10$

$p = 2$			$p = 4$			$p = 10$		
time	ϕ_1	ϕ_2	time	ϕ_1	ϕ_2	time	ϕ_1	ϕ_2
0.0000	1.5000	2.0000	0.0000	1.5000	2.0000	0.0000	1.5000	2.0000
0.5804	1.5616	2.0756	0.4243	1.5643	2.0702	0.3817	1.5645	2.0650
1.1607	1.6044	2.1266	0.8486	1.6124	2.1215	0.7633	1.6129	2.1136
1.7411	1.6376	2.1656	1.2729	1.6511	2.1621	1.1450	1.6518	2.1526
2.3215	1.6654	2.1980	1.6973	1.6837	2.1962	1.5267	1.6847	2.1855
3.2935	1.7029	2.2412	2.5000	1.7337	2.2479	2.2676	1.7362	2.2370
4.2655	1.7331	2.2756	3.3027	1.7734	2.2888	3.0085	1.7770	2.2778
5.2375	1.7584	2.3043	4.1054	1.8064	2.3227	3.7493	1.8108	2.3116
6.2096	1.7803	2.3290	4.9081	1.8348	2.3518	4.4902	1.8399	2.3407
8.5977	1.8242	2.3782	6.7637	1.8883	2.4063	6.1733	1.8935	2.3943
10.9859	1.8585	2.4163	8.6193	1.9302	2.4488	7.8564	1.9356	2.4364
13.3740	1.8869	2.4475	10.4749	1.9647	2.4838	9.5394	1.9702	2.4711
15.7622	1.9111	2.4742	12.3305	1.9942	2.5137	11.2225	1.9999	2.5008
18.2622	1.9332	2.4983	14.8305	2.0281	2.5479	13.7225	2.0372	2.5380
20.7622	1.9526	2.5195	17.3305	2.0570	2.5771	16.2225	2.0685	2.5694
23.2622	1.9700	2.5385	19.8305	2.0822	2.6026	18.7225	2.0956	2.5965
25.7622	1.9858	2.5556	22.3305	2.1046	2.6251	21.2225	2.1195	2.6204
28.2622	2.0002	2.5712	24.8305	2.1247	2.6454	23.7225	2.1408	2.6417
30.7622	2.0134	2.5856	27.3305	2.1430	2.6638	26.2225	2.1601	2.6609
33.2622	2.0257	2.5988	29.8305	2.1597	2.6807	28.7225	2.1777	2.6785
35.7622	2.0372	2.6112	32.3305	2.1752	2.6962	31.2225	2.1938	2.6947
38.2622	2.0479	2.6228	34.8305	2.1895	2.7107	33.7225	2.2088	2.7096
40.7622	2.0580	2.6336	37.3305	2.2029	2.7241	36.2225	2.2227	2.7235
43.2622	2.0675	2.6438	39.8305	2.2154	2.7367	38.7225	2.2357	2.7365
45.7622	2.0765	2.6535	42.3305	2.2272	2.7486	41.2225	2.2479	2.7487
48.2622	2.0850	2.6627	44.8305	2.2383	2.7597	43.7225	2.2594	2.7602
50.7622	2.0932	2.6714	47.3305	2.2488	2.7703	46.2225	2.2703	2.7711
53.2622	2.1009	2.6797	49.8305	2.2588	2.7804	48.7225	2.2806	2.7814
55.7622	2.1084	2.6877	52.3305	2.2684	2.7900	51.2225	2.2904	2.7912
58.2622	2.1155	2.6953	54.8305	2.2775	2.7991	53.7225	2.2997	2.8006
60.7622	2.1223	2.7026	57.3305	2.2862	2.8078	56.2225	2.3087	2.8095
63.2622	2.1289	2.7096	59.8305	2.2945	2.8162	58.7225	2.3172	2.8181
65.7622	2.1352	2.7164	62.3305	2.3025	2.8242	61.2225	2.3254	2.8263
68.2622	2.1413	2.7229	64.8305	2.3102	2.8320	63.7225	2.3333	2.8342
70.7622	2.1472	2.7292	67.3305	2.3176	2.8394	66.2225	2.3409	2.8417

Table 4.2: The solution of ϕ_k in 5 dimensions by $p = 2$

$p = 2$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.0000	5.0000	3.0000	2.5000
0.5804	1.5606	2.0756	5.0884	3.0739	2.5710
1.1607	1.6024	2.1266	5.1473	3.1200	2.6254
1.7411	1.6316	2.1656	5.1919	3.1626	2.6667
2.3215	1.6624	2.1980	5.2288	3.2079	2.6910
3.2935	1.7009	2.2412	5.2779	3.2448	2.7466
4.2655	1.7311	2.2756	5.3168	3.2820	2.7828
5.2375	1.7514	2.3043	5.3490	3.3229	2.8129
6.2096	1.7813	2.3290	5.3767	3.3495	2.8418
8.5977	1.8202	2.3782	5.4315	3.4021	2.8951
10.9859	1.8505	2.4163	5.4739	3.4428	2.9388
13.3740	1.8809	2.4475	5.5084	3.4761	2.9714
15.7622	1.9011	2.4742	5.5379	3.5045	3.0012
18.2622	1.9132	2.4983	5.5645	3.5391	3.0453
20.7622	1.9226	2.5195	5.5878	3.5696	3.1473
23.2622	1.9200	2.5385	5.6086	3.5977	3.1669
25.7622	1.9158	2.5556	5.6274	3.6208	3.1671
28.2622	2.0000	2.5712	5.6445	3.6473	3.1689
30.7622	2.0034	2.5856	5.6612	3.6624	3.1691
33.2622	2.0157	2.5988	5.6787	3.6884	3.1796
35.7622	2.0272	2.6112	5.6982	3.6997	3.1962
38.2622	2.0379	2.6228	5.7098	3.7127	3.2252
40.7622	2.0480	2.6336	5.7326	3.7238	3.2254
43.2622	2.0575	2.6438	5.7437	3.7366	3.2369
46.9622	2.0665	2.6535	5.7543	3.7489	3.2489
48.2622	2.0750	2.6627	5.7742	3.7609	3.2609
50.7622	2.0832	2.6714	5.7837	3.7721	3.2721
54.3622	2.0909	2.6797	5.7858	3.7824	3.2824
57.7622	2.1014	2.6877	5.7924	3.7932	3.2932
58.9622	2.1105	2.6953	5.8097	3.8012	3.3012
62.7622	2.1203	2.7026	5.8106	3.8102	3.3103
63.2622	2.1229	2.7096	5.8192	3.8184	3.3184
65.7622	2.1312	2.7164	5.8265	3.8265	3.3265

Table 4.3: The solution of ϕ_k in 5 dimensions by $p = 4$

$p = 4$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.0000	5.0000	3.0000	2.5000
0.4804	1.5616	2.0656	5.0784	3.0839	2.5810
1.0607	1.6044	2.1166	5.1273	3.1400	2.6354
1.6411	1.6376	2.1556	5.1719	3.1826	2.6767
2.3015	1.6654	2.1780	5.1988	3.2179	2.7110
3.2535	1.7029	2.2212	5.2479	3.2648	2.7566
4.2455	1.7331	2.2456	5.3068	3.3020	2.7928
5.2275	1.7584	2.3001	5.3290	3.3329	2.8229
6.2196	1.7803	2.3190	5.3567	3.3595	2.8488
8.5377	1.8242	2.3582	5.4015	3.4121	2.9001
10.9059	1.8585	2.4063	5.4439	3.4528	2.9399
13.3140	1.8869	2.4275	5.4884	3.4861	2.9724
15.7222	1.9111	2.4342	5.5179	3.5145	3.0032
18.2322	1.9332	2.4683	5.5445	3.5401	3.0553
20.7222	1.9526	2.5095	5.5778	3.5726	3.1573
23.2222	1.9700	2.5285	5.5986	3.5987	3.1679
25.7122	1.9858	2.5446	5.6214	3.6209	3.1687
28.2222	2.0002	2.5612	5.6425	3.6483	3.1690
30.7522	2.0134	2.5756	5.6611	3.6634	3.1692
33.2422	2.0257	2.5888	5.6786	3.6864	3.1794
35.7222	2.0372	2.6012	5.6952	3.6995	3.1952
38.2022	2.0479	2.6128	5.7097	3.7117	3.2242
40.7122	2.0580	2.6226	5.7246	3.7241	3.2244
43.2222	2.0675	2.6328	5.7387	3.7368	3.2370
46.7322	2.0765	2.6425	5.7493	3.7490	3.2490
48.2122	2.0850	2.6517	5.7642	3.7612	3.2612
50.7322	2.0932	2.6614	5.7737	3.7724	3.2724
54.2222	2.1009	2.6697	5.7838	3.7826	3.2826
56.7222	2.1084	2.6772	5.7914	3.7934	3.2934
58.8122	2.1155	2.6852	5.8077	3.8014	3.3014
61.7322	2.1223	2.7016	5.8101	3.8106	3.3106
63.8222	2.1289	2.7086	5.8186	3.8186	3.3186
65.7122	2.1352	2.7144	5.8264	3.8268	3.3268

Table 4.4: The solution of ϕ_k in 5 dimensions by $p = 10$

$p = 10$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.0000	5.0000	3.0000	2.5000
0.3817	1.5645	2.0850	5.0650	3.0650	2.5650
0.7633	1.6129	2.1336	5.1136	3.1136	2.6136
1.1450	1.6518	2.1676	5.1526	3.1526	2.6526
1.5267	1.6847	2.1965	5.1855	3.1855	2.6855
2.2676	1.7362	2.2470	5.2370	3.2370	2.7370
3.0085	1.7770	2.2778	5.2778	3.2778	2.7778
3.0493	1.8108	2.3116	5.3116	3.3116	2.8116
4.4902	1.8399	2.3407	5.3407	3.3407	2.8407
5.1733	1.8935	2.3943	5.3943	3.3943	2.8943
6.8564	1.9356	2.4364	5.4364	3.4364	2.9364
8.5394	1.9702	2.4711	5.4711	3.4711	2.9711
10.2225	1.9999	2.5008	5.5008	3.5008	3.0008
12.7225	2.0372	2.5380	5.5381	3.5381	3.0381
14.2225	2.0685	2.5694	5.5694	3.5694	3.0694
17.7225	2.0956	2.5965	5.5965	3.5965	3.0965
19.2225	2.1195	2.6204	5.6204	3.6204	3.1204
22.7225	2.1408	2.6417	5.6417	3.6417	3.1417
24.2225	2.1601	2.6609	5.6610	3.6610	3.1610
28.1125	2.1777	2.6785	5.6785	3.6785	3.1785
30.1115	2.1938	2.6947	5.6947	3.6947	3.1947
33.1225	2.2088	2.7096	5.7096	3.7096	3.2096
34.2225	2.2227	2.7235	5.7235	3.7235	3.2235
38.0025	2.2357	2.7365	5.7365	3.7365	3.2365
40.2225	2.2479	2.7487	5.7487	3.7487	3.2487
43.2015	2.2594	2.7602	5.7602	3.7602	3.2602
46.2225	2.2703	2.7711	5.7711	3.7711	3.2711
47.7225	2.2806	2.7814	5.7814	3.7814	3.2814
50.2225	2.2904	2.7912	5.7912	3.7912	3.2912
53.7225	2.2997	2.8006	5.8006	3.8006	3.3006
56.2225	2.3087	2.8095	5.8095	3.8095	3.3095
58.7225	2.3172	2.8181	5.8181	3.8181	3.3181
61.2225	2.3254	2.8263	5.8263	3.8263	3.3263

Table 4.5: The solution of ϕ_k in 6 dimensions by $p = 2$

$p = 2$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.0000	5.0000	3.0000	2.5000
0.5804	1.5616	2.0656	5.0884	3.0639	2.5610
1.1607	1.6044	2.1166	5.1473	3.1200	2.6154
1.7411	1.6376	2.1556	5.1919	3.1626	2.6567
2.3215	1.6654	2.1880	5.2288	3.1879	2.6910
3.2935	1.7029	2.2412	5.2779	3.2501	2.7466
4.2655	1.7331	2.2756	5.3168	3.2900	2.7828
5.2375	1.7584	2.3043	5.3490	3.3229	2.8129
6.2096	1.7803	2.3290	5.3767	3.3515	2.8488
8.5977	1.8242	2.3782	5.4315	3.4021	2.9001
10.9859	1.8585	2.4163	5.4739	3.4428	2.9399
13.3740	1.8869	2.4475	5.5084	3.4861	2.9724
15.7622	1.9111	2.4742	5.5379	3.5145	3.0002
18.2622	1.9332	2.4983	5.5645	3.5401	3.0253
20.7622	1.9526	2.5195	5.5878	3.5626	3.0473
23.2622	1.9700	2.5385	5.6086	3.5827	3.0669
25.7622	1.9858	2.5556	5.6374	3.6008	3.0847
28.2622	2.0002	2.5712	5.6545	3.6173	3.1009
30.7622	2.0134	2.5856	5.6702	3.6324	3.1157
33.2622	2.0257	2.5988	5.6867	3.6464	3.1294
35.7622	2.0372	2.6112	5.7082	3.6595	3.1422
38.2622	2.0479	2.6228	5.7168	3.6717	3.1542
40.7622	2.0580	2.6336	5.7296	3.6831	3.1654
43.2622	2.0675	2.6438	5.7437	3.6938	3.1759
45.7622	2.0765	2.6535	5.7543	3.7040	3.1859
48.2622	2.0850	2.6627	5.7662	3.7137	3.1954
50.7622	2.0932	2.6714	5.7767	3.7229	3.2044
53.2622	2.1009	2.6797	5.7868	3.7316	3.2130
55.7622	2.1084	2.6877	5.7964	3.7400	3.2212
58.2622	2.1155	2.6953	5.8097	3.7479	3.2291
60.7622	2.1223	2.7026	5.8176	3.7556	3.2366
63.2622	2.1289	2.7096	5.8252	3.7630	3.2438
65.7622	2.1352	2.7164	5.8325	3.7701	3.2508

Table 4.6: The solution of ϕ_k in 6 dimensions by $p = 4$

$p = 4$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.0000	5.0000	3.0000	2.5000
0.4243	1.5623	2.0702	5.0712	3.0711	2.5709
0.8486	1.6114	2.1215	5.1231	3.1229	2.6226
1.2729	1.6411	2.1621	5.1641	3.1639	2.6635
1.6973	1.6737	2.1962	5.1985	3.1982	2.6978
2.5000	1.7137	2.2479	5.2508	3.2504	2.7499
3.3027	1.7534	2.2888	5.2920	3.2916	2.7910
4.1054	1.7764	2.3227	5.3261	3.3256	2.8250
4.9081	1.8048	2.3518	5.3553	3.3549	2.8542
6.7637	1.8383	2.4063	5.4101	3.4096	2.9089
8.6193	1.9002	2.4488	5.4529	3.4523	2.9515
10.4749	1.9147	2.4838	5.4880	3.4874	2.9866
12.3305	1.9242	2.5137	5.5181	3.5175	3.0166
14.8305	2.0081	2.5479	5.5525	3.5518	3.0509
17.3305	2.0170	2.5771	5.5818	3.5811	3.0802
19.8305	2.0222	2.6026	5.6073	3.6066	3.1057
22.3305	2.0946	2.6251	5.6300	3.6293	3.1284
24.8305	2.1047	2.6454	5.6504	3.6496	3.1487
27.3305	2.1130	2.6638	5.6688	3.6681	3.1671
29.8305	2.1297	2.6807	5.6858	3.6850	3.1840
32.3305	2.1352	2.6962	5.7013	3.7006	3.1996
34.8305	2.1495	2.7107	5.7158	3.7150	3.2140
37.3305	2.1529	2.7241	5.7293	3.7285	3.2275
39.8305	2.1754	2.7367	5.7419	3.7411	3.2401
42.3305	2.2072	2.7486	5.7538	3.7530	3.2520
44.8305	2.2183	2.7597	5.7650	3.7642	3.2632
47.3305	2.2288	2.7703	5.7757	3.7748	3.2738
49.8305	2.2388	2.7804	5.7857	3.7849	3.2839
52.3305	2.2484	2.7900	5.7953	3.7945	3.2935
54.8305	2.2575	2.7991	5.8045	3.8037	3.3026
57.3305	2.2662	2.8078	5.8133	3.8124	3.3114
59.8305	2.2745	2.8162	5.8217	3.8208	3.3197
62.3305	2.2825	2.8242	5.8297	3.8289	3.3278

Table 4.7: The solution of ϕ_k in 6 dimensions by $p = 10$

$p = 10$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.0000	5.0000	3.0000	2.5000
0.4143	1.5643	2.0802	5.0612	3.0811	2.5809
0.8386	1.6124	2.1315	5.1131	3.1329	2.6326
1.2529	1.6511	2.1721	5.1541	3.1739	2.6735
1.6773	1.6837	2.1982	5.1785	3.2082	2.7078
2.4000	1.7337	2.2579	5.2308	3.2604	2.7599
3.2027	1.7734	2.2988	5.2720	3.2936	2.7980
4.0054	1.8064	2.3427	5.3161	3.3286	2.8281
4.7081	1.8348	2.3528	5.3453	3.3569	2.8562
6.5637	1.8883	2.4163	5.4001	3.4196	2.9099
8.4193	1.9302	2.4588	5.4329	3.4583	2.9545
10.2749	1.9647	2.4848	5.4480	3.4894	2.9876
12.1305	1.9942	2.5237	5.5081	3.5185	3.0186
14.2305	2.0281	2.5579	5.5225	3.5538	3.0809
17.2305	2.0570	2.5871	5.5318	3.5841	3.0902
19.1305	2.0822	2.6126	5.5873	3.6086	3.1157
22.1305	2.1046	2.6351	5.6100	3.6393	3.1384
24.4305	2.1247	2.6554	5.6204	3.6596	3.1587
27.2305	2.1430	2.6738	5.6388	3.6781	3.1871
29.4305	2.1597	2.6907	5.6458	3.6950	3.1960
32.1305	2.1752	2.6982	5.6713	3.7106	3.2096
34.3305	2.1895	2.7117	5.7058	3.7250	3.2160
37.1305	2.2029	2.7261	5.7193	3.7385	3.2375
39.5305	2.2154	2.7377	5.7219	3.7521	3.2461
42.2305	2.2272	2.7496	5.7338	3.7620	3.2580
44.4305	2.2383	2.7697	5.7450	3.7742	3.2652
47.1305	2.2488	2.7713	5.7557	3.7838	3.2758
49.2305	2.2588	2.7824	5.7657	3.7919	3.2869
52.1305	2.2684	2.7910	5.7753	3.8045	3.2975
54.5305	2.2775	2.7998	5.7945	3.8137	3.3056
57.1305	2.2862	2.8088	5.8033	3.8224	3.3184
59.4305	2.2945	2.8262	5.8117	3.8308	3.3297
62.1305	2.3025	2.8342	5.8197	3.8419	3.3378

Table 4.8: The solution of ϕ_3, ϕ_4, and ϕ_5 in 3 dimensions by $p = 2, 4, 10$

$p = 2$				$p = 4$				$p = 10$			
time	ϕ_3	ϕ_4	ϕ_5	time	ϕ_3	ϕ_4	ϕ_5	time	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.0000	5.0000	0.0000	1.5000	2.0000	5.0000	0.0000	1.5000	2.0000	5.0000
0.5804	1.5616	2.0756	5.0884	0.4243	1.5643	2.0702	5.0712	0.3817	1.5645	2.0650	5.0650
1.1607	1.6044	2.1266	5.1473	0.8486	1.6124	2.1215	5.1231	0.7633	1.6129	2.1136	5.1136
1.7411	1.6376	2.1656	5.1919	1.2729	1.6511	2.1621	5.1641	1.1450	1.6518	2.1526	5.1526
2.3215	1.6654	2.2980	5.2288	1.6973	1.6837	2.1962	5.1985	1.5267	1.6847	2.1855	5.1855
3.2935	1.7029	2.3412	5.2779	2.5000	1.7337	2.2479	5.2508	2.2676	1.7362	2.2370	5.2370
4.2655	1.7331	2.3756	5.3168	3.3027	1.7734	2.2888	5.2920	3.0085	1.7770	2.2778	5.2778
5.2375	1.7584	2.4043	5.3490	4.1054	1.8064	2.3227	5.3261	3.7493	1.8108	2.3116	5.3116
6.2096	1.7803	2.4290	5.3767	4.9081	1.8348	2.3518	5.3553	4.4902	1.8399	2.3407	5.3407
8.5977	1.8242	2.4782	5.4315	6.7637	1.8883	2.4063	5.4101	6.1733	1.8935	2.3943	5.3943
10.9859	1.8585	2.5163	5.4739	8.6193	1.9302	2.4488	5.4529	7.8564	1.9356	2.4364	5.4364
13.3740	1.8869	2.5475	5.5084	10.4749	1.9647	2.4838	5.4880	9.5394	1.9702	2.4711	5.4711
15.7622	1.9111	2.5742	5.5379	12.3305	1.9942	2.5137	5.5181	11.2225	1.9999	2.5008	5.5008
18.2622	1.9332	2.5983	5.5645	14.8305	2.0281	2.5479	5.5525	13.7225	2.0372	2.5380	5.5381
20.7622	1.9526	2.6195	5.5878	17.3305	2.0570	2.5771	5.5818	16.2225	2.0685	2.5694	5.5694
23.2622	1.9700	2.6385	5.6086	19.8305	2.0822	2.6026	5.6073	18.7225	2.0956	2.5965	5.5965
25.7622	1.9858	2.6556	5.6274	22.3305	2.1046	2.6251	5.6230	21.2225	2.1195	2.6204	5.6204
28.2622	2.0002	2.6712	5.6445	24.8305	2.1247	2.6454	5.6434	23.7225	2.1408	2.6417	5.6417
30.7622	2.0134	2.6856	5.6692	27.3305	2.1430	2.6638	5.6688	26.2225	2.1601	2.6609	5.6610
33.2622	2.0257	2.6988	5.6847	29.8305	2.1597	2.6797	5.6858	28.7225	2.1777	2.6785	5.6785
35.7622	2.0372	2.7112	5.7082	32.3305	2.1752	2.6962	5.7013	31.2225	2.1938	2.6947	5.6947
38.2622	2.0479	2.7228	5.7118	34.8305	2.1895	2.7107	5.7158	33.7225	2.2088	2.7096	5.7096
40.7622	2.0580	2.7336	5.7286	37.3305	2.2029	2.7241	5.7293	36.2225	2.2227	2.7235	5.7235
43.2622	2.0675	2.7438	5.7387	39.8305	2.2154	2.7367	5.7419	38.7225	2.2357	2.7365	5.7365
45.7622	2.0765	2.7535	5.7493	42.3305	2.2272	2.7489	5.7538	41.2225	2.2479	2.7487	5.7487
48.2622	2.0850	2.7798	5.7801	44.8305	2.2383	2.7797	5.7650	43.7225	2.2594	2.7602	5.7602
50.7622	2.0932	2.7744	5.7937	47.3305	2.2488	2.7811	5.7757	46.2225	2.2703	2.7711	5.7711

4.2 Example 2

Suppose that the multivariate optimization problem with inequality constraints is

$$\begin{aligned} &\text{Minimize} \quad -xe^{-x^2-y^2} \\ &\text{subject to} \quad x, y \geq 0 \end{aligned}$$

Its Lagrangian is

$$L = -xe^{-x^2-y^2} - \lambda x - \lambda y.$$

By using the same ways with previous example

$$\begin{aligned} Q_1(x) &: \frac{\partial(-xe^{-x^2-y^2} - \lambda x - \lambda y)}{\partial x} = 0, \\ Q_1(y) &: \frac{\partial(-xe^{-x^2-y^2} - \lambda x - \lambda y)}{\partial y} = 0, \\ H_k(x) &: \lambda \geq 0,\ x \geq 0,\ \lambda x = 0 \Longleftrightarrow \phi_k(x)(\lambda, x) = 0, \\ H_k(y) &: \lambda \geq 0,\ y \geq 0,\ \lambda y = 0 \Longleftrightarrow \phi_k(y)(\lambda, y) = 0. \end{aligned}$$

Such the ODEs that is

$$\begin{aligned} H_k(x) &= -\frac{z_3}{1+\tau}\phi_k^p(x)(\lambda, x), \\ H_k(y) &= -\frac{z_3}{1+\tau}\phi_k^p(y)(\lambda, y). \end{aligned}$$

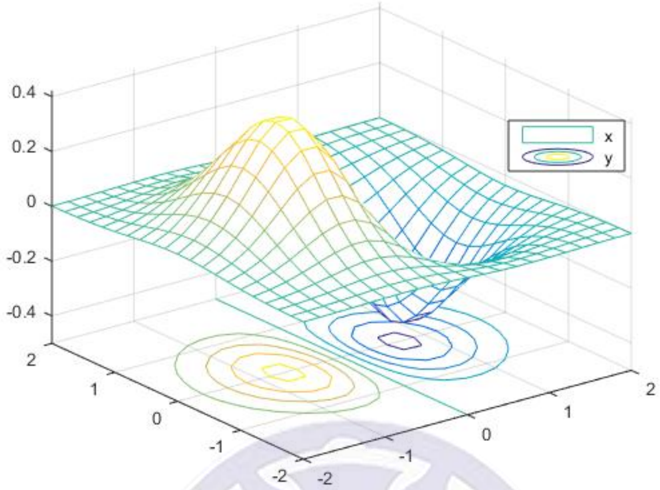


Figure 4.4: The graph of $f(x,y) = -xe^{-x^2-y^2}$ for $x = -2 : 0.2 : 2$ and $y = -2 : 0.2 : 2$

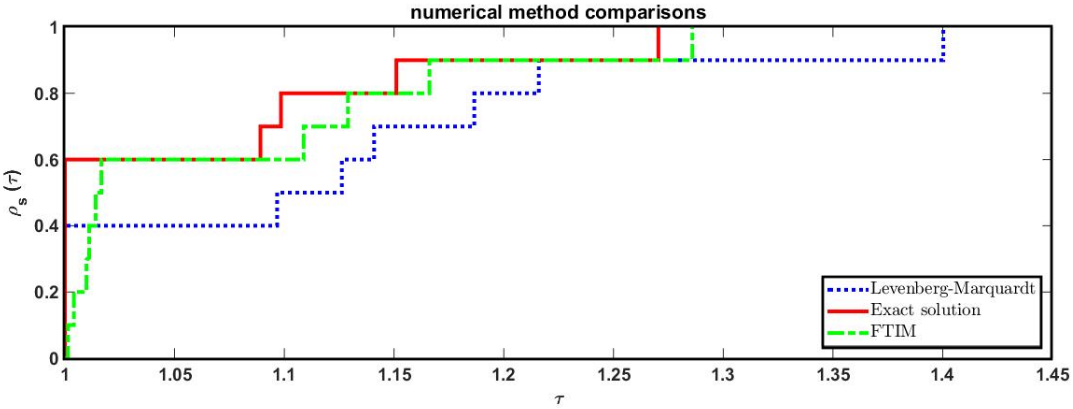


Figure 4.5: Performance profile of $H_k(x)$ and $H_k(y)$ in n dimensions solved by FTIM and they depend on ϕ_k^p by $p > 1,\ x_j^0 = 1e-4,\ \epsilon = 10^{-3}$

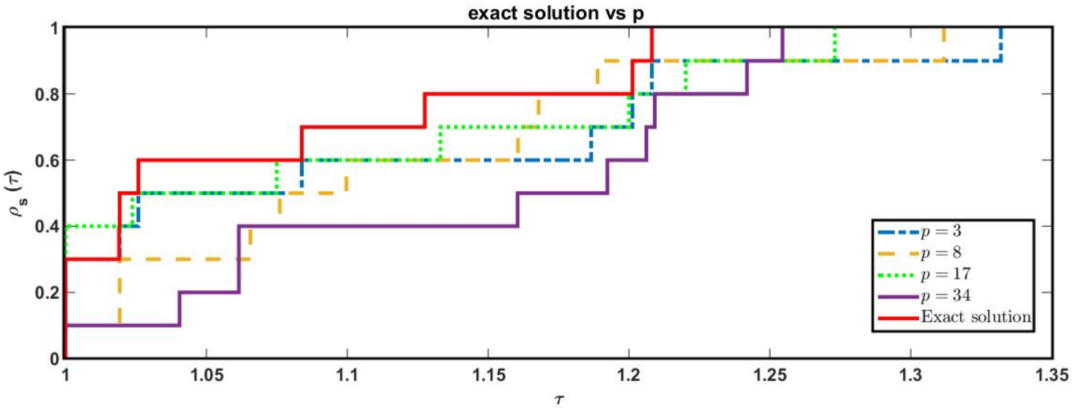


Figure 4.6: Performance profile of H_k in n dimensions solved by FTIM and every single p compared where $x_j^0 = 1e-4,\ \epsilon = 10^{-3}$

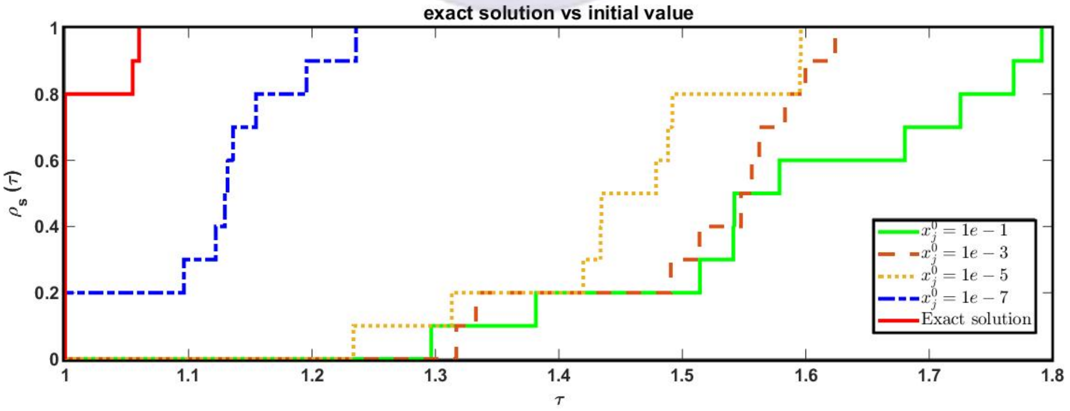


Figure 4.7: Performance profile of H_k in n dimensions solved by FTIM where x_j^0 is different while $p = 4$ and $\epsilon = 10^{-3}$

Table 4.9: The solution of ϕ_k^2 in 9 dimensions $[H_k(x)]$

$p = 2$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.1000	3.4000	4.1000	5.2000
0.4145	1.5650	2.1811	3.4898	4.1916	5.2932
0.8291	1.6147	2.2412	3.5558	4.2588	5.3615
1.2436	1.6552	2.2893	3.6082	4.3121	5.4157
1.6582	1.6897	2.3298	3.6523	4.3570	5.4613
2.4457	1.7435	2.3922	3.7198	4.4256	5.5309
3.2331	1.7870	2.4417	3.7731	4.4798	5.5859
4.0206	1.8234	2.4830	3.8174	4.5247	5.6315
4.8081	1.8550	2.5185	3.8554	4.5633	5.6706
6.6310	1.9152	2.5855	3.9268	4.6357	5.7440
8.4539	1.9628	2.6380	3.9826	4.6923	5.8014
10.2769	2.0024	2.6814	4.0286	4.7388	5.8485
12.0998	2.0365	2.7185	4.0679	4.7787	5.8889
14.5998	2.0765	2.7619	4.1137	4.8250	5.9358
17.0998	2.1107	2.7989	4.1526	4.8645	5.9757

Table 4.10: The solution of ϕ_k^4 in 9 dimensions $[H_k(x)]$

$p = 4$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.1000	3.4000	4.1000	5.2000
0.3031	1.5672	2.1735	3.4743	4.1743	5.2743
0.6062	1.6215	2.2314	3.5326	4.2327	5.3327
0.9092	1.6671	2.2793	3.5808	4.2809	5.3810
1.2123	1.7064	2.3203	3.6221	4.3223	5.4223
1.8760	1.7772	2.3935	3.6958	4.3959	5.4960
2.5396	1.8335	2.4512	3.7537	4.4539	5.5540
3.2033	1.8803	2.4990	3.8017	4.5019	5.6020
3.8669	1.9205	2.5399	3.8428	4.5430	5.6431
5.3165	1.9924	2.6128	3.9159	4.6162	5.7163
6.7661	2.0493	2.6703	3.9736	4.6739	5.7740
8.2156	2.0965	2.7180	4.0214	4.7216	5.8218
9.6652	2.1370	2.7588	4.0623	4.7626	5.8628
12.1652	2.1954	2.8177	4.1214	4.8216	5.9218
14.6652	2.2436	2.8662	4.1700	4.8703	5.9704

Table 4.11: The solution of ϕ_k^{10} in 9 dimensions $[H_k(x)]$

$p = 10$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.1000	3.4000	4.1000	5.2000
0.2726	1.5671	2.1677	3.4677	4.1677	5.2677
0.5452	1.6212	2.2219	3.5219	4.2219	5.3219
0.8179	1.6666	2.2673	3.5673	4.2673	5.3673
1.0905	1.7057	2.3065	3.6065	4.3065	5.4065
1.7177	1.7794	2.3802	3.6802	4.3802	5.4802
2.3450	1.8374	2.4382	3.7382	4.4382	5.5382
2.9722	1.8854	2.4862	3.7862	4.4862	5.5862
3.5995	1.9265	2.5273	3.8273	4.5273	5.6273
4.9244	1.9976	2.5984	3.8984	4.5984	5.6984
6.2492	2.0540	2.6548	3.9548	4.6548	5.7548
7.5741	2.1008	2.7017	4.0017	4.7017	5.8017
8.8989	2.1411	2.7420	4.0420	4.7420	5.8420
11.3989	2.2043	2.8051	4.1051	4.8051	5.9051
13.8989	2.2557	2.8565	4.1565	4.8565	5.9565

Table 4.12: The solution of ϕ_k^{2} in 11 dimensions $[H_k(y)]$

$p = 2$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	1.1000	1.3000	1.2000	1.4000
0.4145	1.5650	1.1234	1.3510	1.2397	1.4591
0.8291	1.6147	1.1434	1.3915	1.2722	1.5049
1.2436	1.6552	1.1612	1.4252	1.2999	1.5426
1.6582	1.6897	1.1772	1.4544	1.3242	1.5750
2.4608	1.7445	1.2044	1.5015	1.3642	1.6266
3.2635	1.7885	1.2280	1.5400	1.3976	1.6683
4.0661	1.8254	1.2488	1.5727	1.4262	1.7035
4.8688	1.8573	1.2676	1.6012	1.4515	1.7340
6.7052	1.9173	1.3046	1.6554	1.5000	1.7917
8.5416	1.9649	1.3357	1.6988	1.5394	1.8375
10.3781	2.0044	1.3625	1.7352	1.5728	1.8758
12.2145	2.0385	1.3862	1.7666	1.6018	1.9088
14.7145	2.0782	1.4147	1.8035	1.6360	1.9473
17.2145	2.1122	1.4397	1.8353	1.6656	1.9804

Table 4.13: The solution of ϕ_k^4 in 11 dimensions $[H_k(y)]$

$p = 4$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	1.1000	1.3000	1.2000	1.4000
0.3031	1.5672	1.1203	1.3533	1.2390	1.4622
0.6062	1.6215	1.1390	1.3992	1.2741	1.5137
0.9092	1.6671	1.1567	1.4393	1.3061	1.5575
1.2123	1.7064	1.1734	1.4748	1.3355	1.5956
1.8894	1.7785	1.2080	1.5416	1.3930	1.6660
2.5664	1.8356	1.2395	1.5957	1.4418	1.7222
3.2435	1.8829	1.2685	1.6412	1.4838	1.7689
3.9205	1.9236	1.2954	1.6805	1.5207	1.8091
5.3809	1.9952	1.3471	1.7504	1.5873	1.8802
6.8412	2.0520	1.3919	1.8061	1.6411	1.9365
8.3016	2.0990	1.4312	1.8525	1.6864	1.9834
9.7620	2.1395	1.4661	1.8924	1.7255	2.0236
12.2620	2.1975	1.5177	1.9498	1.7819	2.0814
14.7620	2.2453	1.5616	1.9973	1.8288	2.1292

Table 4.14: The solution of ϕ_k^{10} in 11 dimensions $[H_k(y)]$

$p = 10$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	1.1000	1.3000	1.2000	1.4000
0.2706	1.5666	1.1183	1.3539	1.2365	1.4640
0.5412	1.6205	1.1354	1.4024	1.2707	1.5173
0.8118	1.6656	1.1516	1.4454	1.3029	1.5623
1.0824	1.7046	1.1671	1.4834	1.3335	1.6011
1.6938	1.7769	1.2001	1.5549	1.3965	1.6733
2.3051	1.8340	1.2309	1.6119	1.4510	1.7304
2.9165	1.8814	1.2600	1.6593	1.4977	1.7778
3.5278	1.9221	1.2877	1.6999	1.5378	1.8184
4.8296	1.9930	1.3427	1.7708	1.6085	1.8894
6.1314	2.0494	1.3924	1.8271	1.6647	1.9457
7.4333	2.0962	1.4366	1.8739	1.7115	1.9925
8.7351	2.1365	1.4756	1.9142	1.7518	2.0328
11.2351	2.2006	1.5389	1.9783	1.8159	2.0969
13.7351	2.2526	1.5907	2.0303	1.8678	2.1489

4.3 Example 3

Let us define the multivariate optimization problem with inequality constraints that is

$$\begin{aligned} &\text{Minimize} \quad \cos xy \\ &\text{subject to} \quad x+a,\ y+a \geq 0,\ a \in \mathbb{R}^{+} \end{aligned}$$

Its Lagrangian is

$$L = \cos xy - \lambda(x+a) - \lambda(y+a).$$

By using the same ways with previous example

$$\begin{aligned} &Q_1(x):\ -y\sin xy - \lambda = 0, \\ &Q_1(y):\ -x\sin xy - \lambda = 0, \\ &H_k(x): \lambda \geq 0,\ (x+a) \geq 0,\ \lambda(x+a) = 0 \Longleftrightarrow \phi_k(x)(\lambda, x+a) = 0, \\ &H_k(y): \lambda \geq 0,\ (y+a) \geq 0,\ \lambda(y+a) = 0 \Longleftrightarrow \phi_k(y)(\lambda, y+a) = 0. \end{aligned}$$

Such the ODEs that is

$$\begin{aligned} Q_k(x) &= -\frac{z_1}{1+\tau}(-y\sin xy - \lambda), \\ Q_k(y) &= -\frac{z_1}{1+\tau}(-x\sin xy - \lambda), \\ H_k(x) &= -\frac{z_3}{1+\tau}\phi_k^p(x)(\lambda, x+a), \\ H_k(y) &= -\frac{z_3}{1+\tau}\phi_k^p(y)(\lambda, y+a). \end{aligned}$$

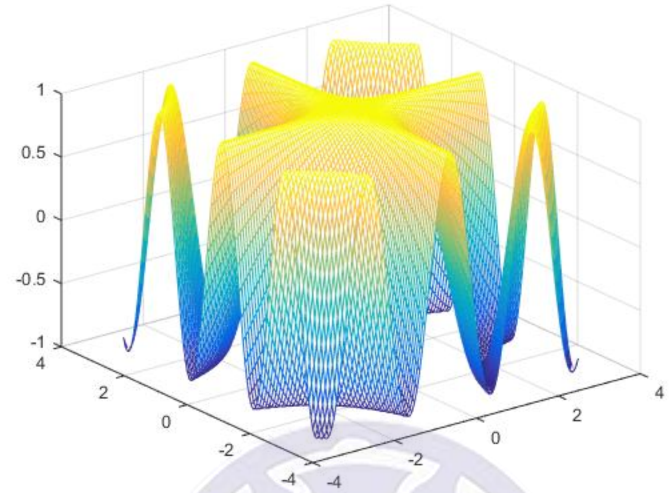


Figure 4.8: The graph of $f(x, y) = \cos xy$ for $x = [-\pi, \pi]$ and $y = [-\pi, \pi]$

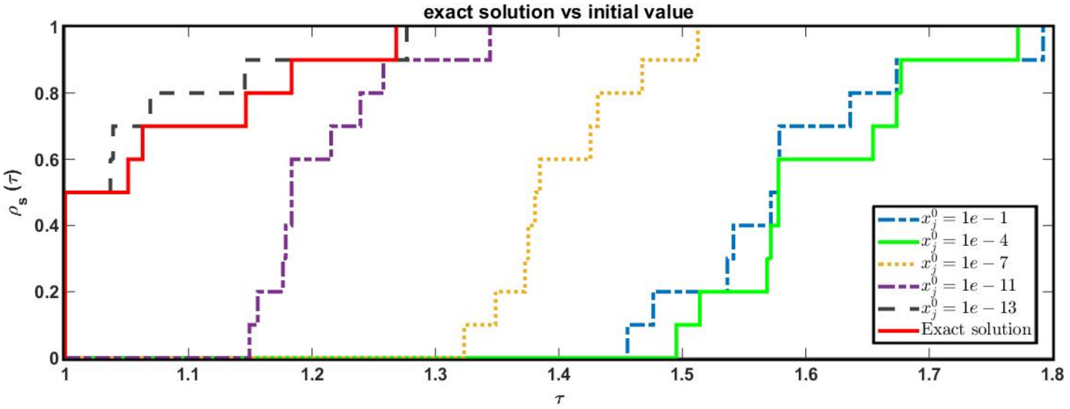


Figure 4.9: Performance profile of ϕ_k^p in n dimensions solved by FTIM where $p > 1$ and $\epsilon = 10^{-7}$

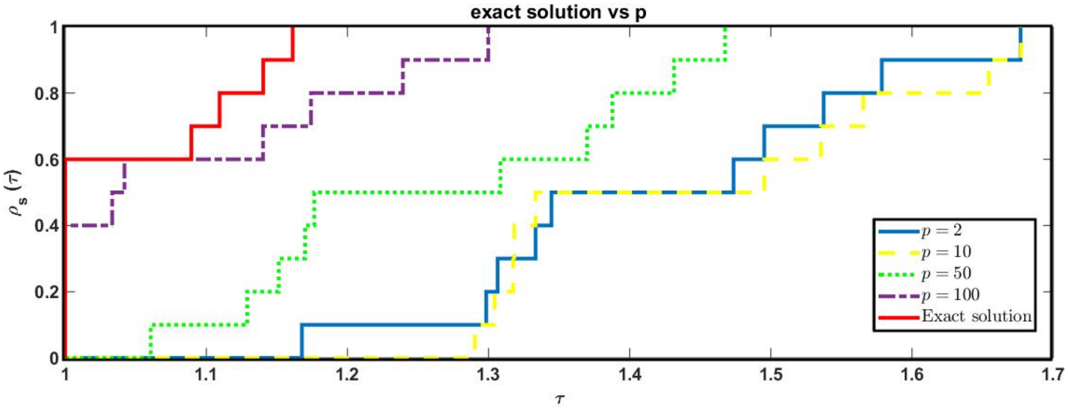


Figure 4.10: Performance profile of ϕ_k^p in n dimensions solved by FTIM wherever $x_j^0 = 1e-6$ and $\epsilon = 10^{-9}$

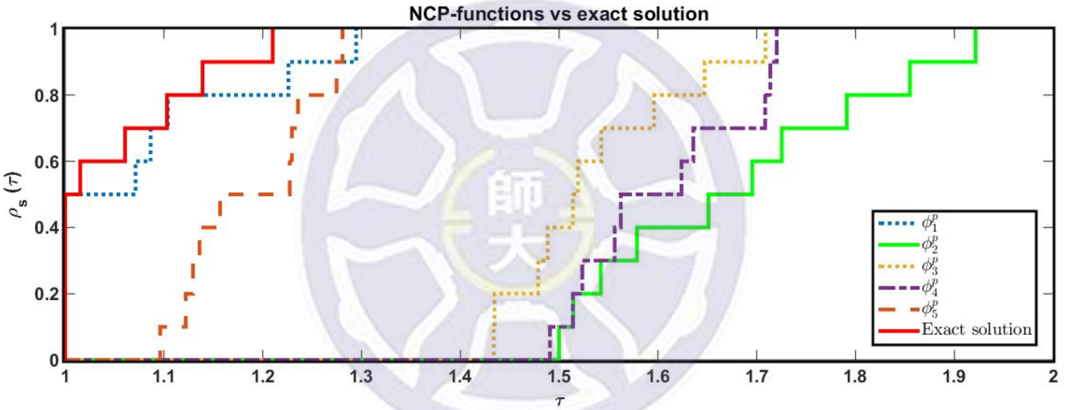


Figure 4.11: Performance profile of ϕ_k^p in n dimensions solved by FTIM when $p > 1,\ x_j^0 = 1e-4,$ and $\epsilon = 10^{-5}$

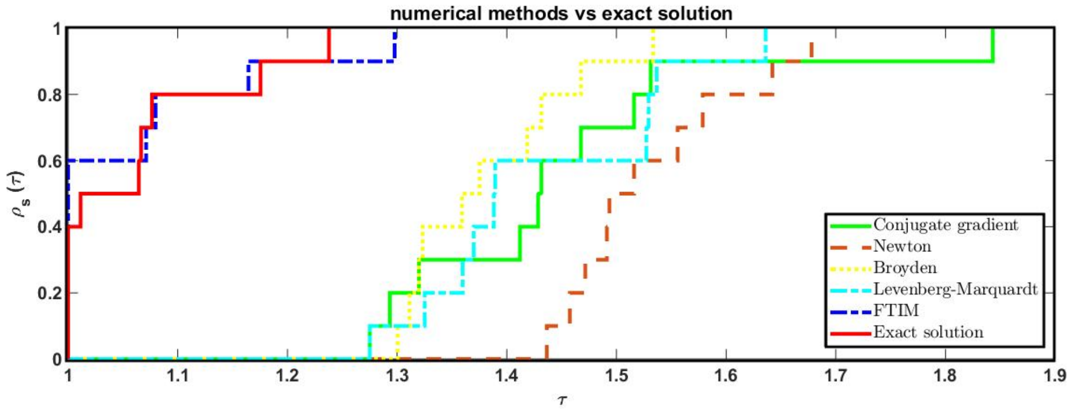


Figure 4.12: Performance profile of Q_k and H_k in n dimensions solved by several numerical methods when $p > 1,\ x_j^0 = 1e-2,$ and $\epsilon = 10^{-3}$

Table 4.15: The solution of ϕ_k^2 in 17 dimensions [$H_k(x)$]

$p = 2$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.1000	3.4000	4.1000	5.2000
0.3675	1.5664	2.1870	3.4990	4.2014	5.3037
0.7351	1.6191	2.2537	3.5739	4.2781	5.3819
1.1026	1.6631	2.3081	3.6344	4.3400	5.4450
1.4702	1.7011	2.3544	3.6858	4.3924	5.4985
2.2005	1.7639	2.4294	3.7684	4.4767	5.5844
2.9308	1.8149	2.4893	3.8338	4.5434	5.6523
3.6612	1.8580	2.5392	3.8882	4.5987	5.7086
4.3915	1.8955	2.5822	3.9349	4.6463	5.7570
6.0538	1.9660	2.6624	4.0213	4.7343	5.8464
7.7162	2.0224	2.7256	4.0892	4.8033	5.9166
9.3785	2.0695	2.7780	4.1453	4.8602	5.9744
11.0408	2.1103	2.8230	4.1933	4.9090	6.0239
13.5408	2.1624	2.8803	4.2543	4.9709	6.0868
16.0408	2.2067	2.9287	4.3056	5.0230	6.1395

Table 4.16: The solution of ϕ_k^4 in 17 dimensions [$H_k(x)$]

$p = 4$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.1000	3.4000	4.1000	5.2000
0.2655	1.5689	2.1809	3.4824	4.1825	5.2826
0.5311	1.6272	2.2463	3.5489	4.2490	5.3491
0.7966	1.6775	2.3013	3.6046	4.3048	5.4049
1.0622	1.7218	2.3490	3.6529	4.3531	5.4532
1.6751	1.8070	2.4393	3.7440	4.4444	5.5446
2.2879	1.8755	2.5106	3.8160	4.5164	5.6166
2.9008	1.9327	2.5698	3.8756	4.5760	5.6762
3.5137	1.9821	2.6206	3.9267	4.6271	5.7274
4.8459	2.0703	2.7107	4.0173	4.7179	5.8181
6.1782	2.1404	2.7820	4.0890	4.7895	5.8898
7.5104	2.1986	2.8411	4.1483	4.8489	5.9492
8.8427	2.2488	2.8919	4.1993	4.8999	6.0002
11.3427	2.3270	2.9709	4.2786	4.9792	6.0796
13.8427	2.3907	3.0351	4.3430	5.0436	6.1440

Table 4.17: The solution of ϕ_k^{10} in 17 dimensions $[H_k(x)]$

$p = 10$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	2.1000	3.4000	4.1000	5.2000
0.2320	1.5695	2.1731	3.4731	4.1731	5.2731
0.4639	1.6288	2.2334	3.5334	4.2334	5.3334
0.6959	1.6799	2.2848	3.5848	4.2848	5.3848
0.9278	1.7245	2.3297	3.6297	4.3297	5.4297
1.5040	1.8161	2.4215	3.7216	4.4216	5.5216
2.0803	1.8884	2.4939	3.7939	4.4939	5.5939
2.6565	1.9482	2.5537	3.8537	4.5537	5.6537
3.2327	1.9995	2.6050	3.9051	4.6051	5.7051
4.4184	2.0861	2.6917	3.9917	4.6917	5.7917
5.6041	2.1553	2.7608	4.0608	4.7608	5.8608
6.7899	2.2129	2.8185	4.1185	4.8185	5.9185
7.9756	2.2626	2.8682	4.1682	4.8682	5.9682
10.4756	2.3488	2.9544	4.2544	4.9544	6.0544
12.9756	2.4177	3.0232	4.3232	5.0232	6.1232

Table 4.18: The solution of ϕ_k^{2} in 25 dimensions $[H_k(y)]$

$p = 2$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	1.1000	1.3000	1.2000	1.4000
0.3675	1.5664	1.1216	1.3501	1.2379	1.4594
0.7351	1.6191	1.1406	1.3914	1.2701	1.5072
1.1026	1.6631	1.1577	1.4267	1.2981	1.5476
1.4702	1.7011	1.1733	1.4578	1.3232	1.5826
2.2221	1.7656	1.2019	1.5114	1.3674	1.6426
2.9740	1.8176	1.2268	1.5558	1.4047	1.6915
3.7259	1.8615	1.2492	1.5937	1.4370	1.7329
4.4779	1.8996	1.2694	1.6269	1.4658	1.7690
6.1626	1.9701	1.3092	1.6893	1.5204	1.8361
7.8474	2.0264	1.3430	1.7398	1.5655	1.8901
9.5322	2.0735	1.3726	1.7825	1.6039	1.9354
11.2170	2.1143	1.3990	1.8196	1.6375	1.9746
13.7170	2.1658	1.4337	1.8669	1.6808	2.0244
16.2170	2.2096	1.4641	1.9074	1.7181	2.0668

Table 4.19: The solution of ϕ_k^4 in 25 dimensions $[H_k(y)]$

$p = 4$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	1.1000	1.3000	1.2000	1.4000
0.2655	1.5689	1.1179	1.3500	1.2351	1.4614
0.5311	1.6272	1.1346	1.3950	1.2675	1.5147
0.7966	1.6775	1.1505	1.4358	1.2977	1.5617
1.0622	1.7218	1.1656	1.4729	1.3261	1.6035
1.6901	1.8089	1.1990	1.5491	1.3868	1.6870
2.3180	1.8785	1.2298	1.6126	1.4400	1.7546
2.9459	1.9366	1.2586	1.6668	1.4872	1.8115
3.5738	1.9866	1.2856	1.7140	1.5294	1.8606
4.9311	2.0752	1.3390	1.7989	1.6075	1.9480
6.2884	2.1456	1.3867	1.8672	1.6720	2.0177
7.6457	2.2041	1.4297	1.9244	1.7267	2.0757
9.0030	2.2543	1.4687	1.9737	1.7742	2.1257
11.5030	2.3314	1.5320	2.0496	1.8480	2.2023
14.0030	2.3944	1.5864	2.1117	1.9089	2.2650

Table 4.20: The solution of ϕ_k^{10} in 25 dimensions $[H_k(y)]$

$p = 10$					
time	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
0.0000	1.5000	1.1000	1.3000	1.2000	1.4000
0.2320	1.5695	1.1157	1.3471	1.2315	1.4613
0.4639	1.6288	1.1306	1.3913	1.2611	1.5166
0.6959	1.6799	1.1447	1.4330	1.2894	1.5659
0.9278	1.7245	1.1583	1.4721	1.3166	1.6098
1.4604	1.8099	1.1878	1.5519	1.3754	1.6944
1.9929	1.8783	1.2154	1.6189	1.4297	1.7627
2.5255	1.9354	1.2416	1.6758	1.4798	1.8198
3.0580	1.9847	1.2666	1.7248	1.5256	1.8691
4.1557	2.0687	1.3152	1.8085	1.6067	1.9530
5.2534	2.1361	1.3606	1.8759	1.6736	2.0205
6.3511	2.1926	1.4034	1.9324	1.7299	2.0769
7.4487	2.2414	1.4438	1.9812	1.7785	2.1257
9.7974	2.3275	1.5216	2.0672	1.8644	2.2118
12.1461	2.3962	1.5882	2.1360	1.9332	2.2805

Chapter 5

Conclusions

The FTIM gives the advantages a lot into numerical experiments and is giving an approximation solution at nonlinear optimization problem better than others. The way to reformulate from NAEs into ODEs by means of fictitious time function until obtained a new numerical equation is a long way to be conducted, anyway this way gives us more satisfying results such as a higher stability, an approximate accuracy, and also efficient loop in algorithm when used the performance profile to find computing times and number of iterations. The comparisons of numerical methods in optimization problems together with the conjugate gradient, Newton, Broyden, and Levenberg-Marquardt, as an evidencing to reveal the performance of the fictitious time integration method (FTIM), is better than others.

Bibliography

[1] Burden, R. L.; Faires, J. D. (2011), *Numerical analysis*, ninth edition, Boston, Brooks/Cole, Cengage Learning.

[2] Chen, J-S., *On some NCP-functions based on the generalized Fischer-Burmeister function*, Asia-Pacific J. Oper. Res., vol. 24, pp. 401-420, 2007.

[3] De Luca, T.; Facchinei, F.; Kanzow, C., *A semismooth equation approach to the solution of nonlinear complementarity problems*, Math. Program., vol. 75, pp. 407-439, 1996.

[4] Dolan, E. D.; More, J. J., *Benchmarking optimization software with performance profiles*, Math. Program., vol. 91, pp. 201-213, 2002.

[5] Ferris, M. C.; Pang, J-S., *Engineering and economic applications of complementarity problems*, SIAM Review, vol. 39, pp. 669-713, 1997.

[6] Gallier, J.; Quaintance, J. (2017), *Notes on differential geometry and Lie groups*, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, USA.

[7] Harker, PT.; Pang, J-S., *Finite dimensional variational inequality and nonlinear complementarity problem: A survey of theory, algorithms, and applications*, Mathematical Programming, vol. 48, pp. 161-220, 1990.

[8] Huang, C-H.; Weng, K-J.; Chen, J-S.; Chu, H-W.; Li, M-Y., *On four discrete-type families of NCP-functions*, Journal of Nonlinear and Convex Analysis, pp. 1-27, 2017.

[9] Kanzow, C.; Yamashita, N.; Fukushima, M., *New NCP-Functions and their properties*, Journal of Optimization Theory and Applications, vol. 94, pp. 115-135, 1997.

[10] Ku, C-Y.; Yeih, W.; Liu, C-S.; Chi, C-C., *Applications of the fictitious time integration method using a new time-like function*, CMES: Computer Modeling in Engineering and Sciences, vol. 43, no. 2, pp. 173-190, 2009.

[11] Liu, C-S.; Chang, J-R.; Chang, K-H.; Chen, Y-W., *Simultaneously estimating the time-dependent damping and stiffness coefficients with the aid of vibrational data*, CMC: Computer, Materials, and Continua, vol. 7, pp. 97-107, 2008.

[12] Liu, C-S.; Atluri, S. N., *A fictitious time integration method (FTIM) for solving mixed complementarity problems with applications to non-linear optimization*, CMES: Computer Modeling in Engineering and Sciences, vol. 34, no. 2, pp. 155-178, 2008a.

[13] Liu, C-S.; Atluri, S. N., *A novel time integration method for solving a large system of non-linear algebraic equations*, CMES: Computer Modeling in Engineering and Sciences, vol. 31, pp. 71-83, 2008b.

[14] Liu, C-S.; Atluri, S. N., *A fictitious time integration method for the numerical solution of the Fredholm integral equation and for numerical differentiation of noisy data, and its relation to the filter theory*, CMES: Computer Modeling in Engineering and Sciences, vol. 41, pp. 243-261, 2009.

[15] Liu, C-S., *Cone of non-linear dynamical system and group preserving schemes*, Int. J. Non-Linear Mech., vol. 36, pp. 1047-1068, 2001.

[16] Liu, C-S., *A Lie-group shooting method for simultaneously estimating the time-dependent damping and stiffness coefficients*, CMES: Computer Modeling in Engineering and Sciences, vol. 27, pp. 137-149, 2008a.

[17] Liu, C-S., *A fictitious time integration method for solving the discretized inverse Sturm-Liouville problems, for specified eigenvalues*, CMES: Computer Modeling in Engineering and Sciences, vol. 36, pp. 261-286, 2008b.

[18] Liu, C-S., *A fictitious time integration method for two-dimensional quasilinear elliptic boundary value problems*, CMES: Computer Modeling in Engineering and Sciences, vol. 33, pp. 179-198, 2008c.

[19] Liu, C-S., *A time-marching algorithm for solving nonlinear obstacle problems with the aid of an NCP-function*, CMC: Computer, Materials, and Continua, vol. 8, pp. 53-65, 2008d.

[20] Liu, C-S., *Solving an inverse Sturm-Liouville problem by a Lie-group method*, Boundary Value Problems, Article ID 749865, 2008e.

[21] Liu, C-S., *Identifying time-dependent damping and stiffness functions by a simple and yet accurate method*, J. Sound Vib., vol. 318, pp. 148-165, 2008f.

[22] Liu, C-S., *A fictitious time integration method for solving m-point boundary value problems*, CMES: Computer Modeling in Engineering and Sciences, vol. 39, pp. 125-154, 2009.

[23] Mangasarian, O. L., *Equivalence of the complementarity problem to a system of nonlinear equations*, SIAM Journal on Applied mathematics, vol. 31, pp. 89-92, 1976.

[24] Pang, J-S. (1994), *Complementarity problems*, Handbook of Global Optimization, R Horst and P Pardalos (eds.), pp. 271-338, Boston, MA: Kluwer Academic Publishers.

[25] Pourrajabian, A.; Ebrahimi, R.; Mirzaei, M.; Shams, M., *Applying genetic algorithms for solving nonlinear algebraic equations*, Applied Mathematics and Computation, vol. 219, pp. 11483-11494, 2013.

[26] Sun, D.; Qi, L., *On NCP-functions*, Computational Optimization and Applications, vol. 13, pp. 201-220, 1999.

[27] Tseng, P., *Global behaviour of a class of merit functions for the nonlinear complementarity problem*, Journal of Optimization Theory and Applications, vol. 89, pp. 17-37, 1996.

[28] Yamashita, N.; Fukushima, M., *Modified Newton methods for solving a semismooth reformulation of monotone complementarity problems*, Mathematical Programming, vol. 76, pp. 469-491, 1997.