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

Irfan Nurhidayat, Zijun Hao, Chu-Chin Hao, Jein-Shan Chen

Department of Mathematic, National Taiwan Normal University, Taipei 11677, Taiwan
School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China

*Corresponding Authors: irfannurhidayat09@gmail.com

1. Introduction

This section consists of two sections that is motivation which provides us with information to open up the reasons why we should undertake this research and what the goal will be achieved in this research. 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.

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 or other ways to find approximation solutions of numerical methods. Actually, Liu and Atluri (2008a) have been using this FTIM in various sciences such as engineering, even now continuously developed in plentiful areas. They found the 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 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 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 NAEs into an 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.

This research considers an ODE approach for solving NLP and NCP. The NLP is

\[
\text{Minimize } f(x) \\
\text{subject to } -g_i(x) \leq 0, [i=1, 2, 3, \ldots, l], \\
h_i(x) = 0, [i=1, 2, 3, \ldots, l], \\
x \in X,
\]

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

\[
L(x, \lambda_1, \ldots, \lambda_l, \mu_1, \ldots, \mu_k) = f(x) - \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{i=1}^{k} \mu_i h_i(x) \tag{1}
\]

under some constraint qualifications (e.g. linear independence constraint qualification (LICQ) and Mangasarian-Fromovitz constraint qualification (MFCQ)), there exist \( \lambda \) and \( \mu \) such that the KKT conditions of NLP are described as follows optimality condition:

Optimality condition: \( \nabla_x L(x, \lambda, \mu) = 0 \)

Feasible condition: \(-g_i(x) \leq 0, h_i(x) = 0 \)

Multiplier condition: \( \lambda_i \geq 0 \)

Complementarity condition: \( \langle \lambda_i, g_i(x) \rangle = 0 \)

In the above system, these constraints

\[
\lambda_i \geq 0, g_i(x) \geq 0, \langle \lambda_i, g_i(x) \rangle = 0 \Rightarrow \varnothing_k(\lambda_i, g_i(x)) = 0 \tag{3}
\]

consist of a nonlinear complementarity problem.

A nonlinear complementarity problem (NCP) aims to find a point \( x \in \mathbb{R}^n \) such that

\[
x \geq 0, F(x) \geq 0, \langle x, F(x) \rangle
\]

where \( \langle \cdot, \cdot \rangle \) denotes the Euclidean inner product and \( F := (F_1, \ldots, F_n)^T \) is a mapping from \( \mathbb{R}^n \) to \( \mathbb{R}^n \). Presently, there exist five primary means for solving these NCPs, which are the merit function approach, nonsmooth approach, smoothing function approach, regularization approach, and neural networks approach. 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 $\varnothing_k(a,b)$, $[k = 1, 2, 3, 4, 5]$, such as

$$\varnothing_1(a,b) := \sqrt{|a|^p + |b|^p} - (a + b),$$

$$\varnothing_2(a,b) := \left( a^2 + b^2 \right)^{p/2} - (a + b)^p,$$

$$\varnothing_3(a,b) := a^p - (a-b)^p,$$

$$\varnothing_3(a,b) := \begin{cases} \varnothing_3(a,b), & \text{if } a>b \\ a^p = b^p, & \text{if } a=b \\ \varnothing_3(b,a), & \text{if } a<b \end{cases}$$

$$\varnothing_3(a,b) := \begin{cases} \varnothing_3(a,b) b^p, & \text{if } a>b \\ a^p b^a = a^2 p, & \text{if } a=b \\ \varnothing_3(b,a) a^p, & \text{if } a<b \end{cases}$$

Many approaches that have been shown in (Burden & Faires, 2011; Yamashita & Fukushima, 1997), are to reformulate the NCP as a system of NAE. The definition of the system of nonlinear equations is given by

$$F_i(x_j) = 0, \ [i,j=1,\ldots,n]$$

where $x_i \in \mathbb{R}$ and each $F_i: \mathbb{R}^n \to \mathbb{R}$ is a nonlinear real function. Pourrajabian et al. (2013), solving of NAEs is a prominent problem in science and engineering. According to Liu & Atluri (2008a), we have a new method to reformulate of the complementarity problem (4) as the system of NAEs like as the equation below

$$F_i(x_1, \ldots, x_n) := \varnothing_k(P_i(x_1, \ldots, x_n), Q_i(x_1, \ldots, x_n)),$$

for some mapping $\varnothing_k: \mathbb{R}^n \to \mathbb{R}$. In other words, this property guarantees that a vector $(x_1, \ldots, x_n) \in \mathbb{R}^n$ is a solution of the complementarity problem (4) if and only if $(x_1, \ldots, x_n)$ solves the equations system (10) (Liu & Atluri, 2008a). The system of NAEs is able to be transformed into an ODE by force of a time-like function. We will bring off a transformation away from NAEs into ODEs by way of using a time-like function. A GPS is used to preserve ODEs such that the transformation of ODEs into numerical equation happens. In this paper, it recently will be alluded to step by step how to derive the system of ordinary differential equations into numerical equations by utilizing a GPS.

One decade ago, Liu & 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 FTIM has been proposed by Liu & Atluri (2008b). The FTIM was the first used to solve a nonlinear equation by introducing fictitious time (Liu & Atluri, 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 BVPs also has been proposed by Liu (2008b, 2009).
The goal of this paper is to offer the FTIM as an alternative simulation better in nonlinear optimization problems (NOPs). The contribution of this paper is a great advantage to easily extend to higher dimensional NOPs with nonlinear equality and inequality constraints. The organizations of this paper follow, section 1, we focus on introducing some references related to NLP, NCP, ODE, FTIM, and several stories from previous researchers, section 2, is the basic theories like an introduction of the original time-like or fictitious time function, the Lorentz group \( SO_0(n, 1) \) and the Lie algebra \( so(n, 1) \). Section 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. Section 4, focuses on the numerical experiments by using Matlab\(^\odot\) to compare \( \phi_k^p \) 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, section 5, is the conclusions of this research covering a tidy package of main contents in this paper.

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 \( SO_0(n, 1) \) and the Lie algebra \( so(n, 1) \) summarized and is adapted from Gallier & Quaintance (2017) while the performance profile theory in the numerical test adapted from Dolan & More (2002). These theories will become the fundamental concepts to construct the main topics of this paper in the next section.

2.1 Original time-like function

The fictitious time function that used in the transformation was named time-like function (Ku et al., 2009). In their studies, a more general form of the time-like function i.e.

\[
q^T = (1+\tau)^y, 0 \leq y \leq 1,
\]

with \( \tau \) is a fictitious time variable. Liu & Atluri (2008b), have the first shown that the time like function (12), has to be differentiable, that is \( q(0)=1 \) and \( q(\infty) \). In this paper, we used the original time-like function by \( y = 1 \) to transform an algebraic equation into the first order ordinary differential equation (FOODE).

2.2 The Lorentz group \( SO_0(n, 1) \) and the Lie algebra \( so(n, 1) \)

We presently would forward to see a little bit relating to the Lie algebra of the Lorentz group. Started by the Lorentz group \( SO_0(n, 1) \) is the proper orthochronous Lorentz group and its Lie algebra is \( so(n, 1) \). Both of them are able to be defined as below

\[
SO_0(n, 1) = \{ Y = (y_{ij}) \in SO(n, 1) | y_{n+1,n+1} > 0, \text{ for } y_{i=1,...,n+1,j=1,...,n+1} \in \mathbb{R} \},
\]

\[
so(n, 1) = \{ Y \in M_{(n+1) \times (n+1)}(\mathbb{R}) | Y^T J + JY = 0 \},
\]

for \( M_{(n+1) \times (n+1)}(\mathbb{R}) \) 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 \( so(n, 1) \) is a special orthogonal group and \( GL(n+1, \mathbb{R}) \) is a general linear group, namely

\[
SO(n, 1) = \{ G \in GL(n+1, \mathbb{R}) | G^T J G = J \text{ and } det G = 1 \},
\]
\( \text{GL}(n+1, \mathbb{R}) = \{ G \in M_{(n+1) \times (n+1)}(\mathbb{R}) | G \text{ invertible} \} \)

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

\( \text{so}(n, 1) = \begin{bmatrix} B & u \\ u^T & 0 \end{bmatrix} \in M_{(n+1) \times (n+1)}(\mathbb{R}) | u \in \mathbb{R}^n, B^T = -B \)

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

2.3 The Lorentz group \( \text{SO}_0(n, 1) \) and the Lie algebra \( \text{so}(n, 1) \)

The performance profile is the popular theory in the numerical test from Dolan & More (2002) where has been utilized by researchers to compare the performance of the functions \( \emptyset_k \). 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 \( \emptyset_k \) 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

\[ r_{p,s} := \frac{f_{p,s}}{\min\{f_{p,s} : s \in S\}} \]

where \( S \) 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 \( s \in S \), that is a performance ratio \( f_{p,s} \) inside a factor \( \tau \in \mathbb{R} \) of the best possible ratio.

3. ODE reformulation

This section is devoted to the main topic in this paper which covers a transformation into ODEs system and a 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 section is given in this section.

3.1 Transformation into an ODEs system

Let us consider the following NAEs system

\[ F_i(x_j) = 0, \ [i,j=1,...,n] \quad (13) \]

We define an original time-like function below
\[ y_i(\tau) = q(\tau) x_j \quad [i,j=1,\ldots,n] \]  
\[ q(\tau) = 1 + \tau \]

where \( \tau \) is a variable which is independent of \( x_j \). Taking the derivative of (14) with respect to \( \tau \), we have

\[ y'_i = \frac{dy_i}{d\tau} = x_j. \]  

If \( z \neq 0 \); then (13) is equivalent to

\[ 0 = -z F_i(x_j). \]  

By using (14), we obtain

\[ 0 = -z F_i \left( \frac{y_i}{1+\tau} \right). \]  

Adding (15), we get

\[ y_i = x_j - z F_i \left( \frac{y_i}{1+\tau} \right). \]  

By (14), we have \( x_j = y_i/(1+\tau) \) that means we can write (18) as an ODEs system for \( y_i \) like

\[ y_i = \frac{y_i}{1+\tau} - z F_i \left( \frac{y_i}{1+\tau} \right). \]  

Multiplying of each equation by the integrating factor \( (1+\tau) \) and used (19) once more so that

\[ \frac{y_i}{1+\tau} = \frac{y_i}{(1+\tau)^2} - z \frac{1}{1+\tau} F_i \left( \frac{y_i}{1+\tau} \right). \]  

Others were known that

\[ \frac{d}{d\tau} \left( \frac{y_i}{1+\tau} \right) = \frac{y_i}{1+\tau} - \frac{1}{1+\tau} \left( \frac{y_j}{1+\tau} \right). \]  

By utilizing (14) and (21), then (20) can be rewritten becoming

\[ \frac{d}{d\tau} x_j = -z \frac{1}{1+\tau} F_i(x_j) \quad [i,j=1,2,\ldots,n] \]  

Further (22) becomes

\[ x_j = -z \frac{1}{1+\tau} F_i(x_1,\ldots,x_n,\mu_1,\ldots,\mu_k,\lambda_1,\ldots,\lambda_n) \in \mathbb{R}^{n+k+l}, \tau>0 \]  

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 (23) to develop a more stable GPS where will be discussed in the next section, formerly we are going to rewrite (23) as a vector form that is

\[ x = f(x,\tau) = -z \frac{1}{1+\tau} F_i(x_1,\ldots,x_n,\mu_1,\ldots,\mu_k,\lambda_1,\ldots,\lambda_n) \in \mathbb{R}^{n+k+l}, \tau>0 \]  

where \( N = n+k+l \) is the dimensional state vector of algebraic equations. We currently will define the NAEs system of (13) as below
\[ F_i(x_j) := \emptyset_k \left( P_i(x_j), Q_i(x_j) \right), [i,j=1,\ldots,n \text{ and } k=1,2,3,4,5] \]  

In other words, we get

\[ F_i = \emptyset_k (P_i, Q_i) = 0 \]

We rewrite (23) as well as systems of ODE i.e.

\[ Q_1: x_i = -\frac{z_1}{1+t} F_1(x_i), [x_i \in \mathbb{R}^n] \]

\[ Q_2: \mu_j = -\frac{z_2}{1+t} F_2(\mu_j), [\mu_j \in \mathbb{R}^k] \]

\[ H_k: \lambda_j = -\frac{z_3}{1+t} F_3(\lambda_j), [\lambda_j \in \mathbb{R}^l] \]

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,

\[ Q_1 = -\frac{z_1}{1+t} \left[ \frac{\partial f}{\partial x_i} + \sum_{j=1}^{k} \mu_j \frac{\partial h_i}{\partial x_j} - \sum_{j=1}^{l} \mu_j \frac{\partial g_i}{\partial x_j} \right], \quad (26) \]

\[ Q_2 = -\frac{z_2}{1+t} h_i, \quad (27) \]

\[ H_k = -\frac{z_3}{1+t} \emptyset_k(\lambda_j g_i) \]

### 3.2 GPS for differential equations system

The above equations (26)-(28) can be combined together into the NDS as follows

\[ x = f(x, \tau) \Leftrightarrow x^T = [f(x, \tau)]^T, \in \mathbb{R}^n, \]

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

\[ X := (x^T, ||x||)^T = (x^T, ||x||) \in \mathbb{R}^{N+1}, \quad (30) \]

A GPS can preserve the internal symmetry group of the considered ODEs system (Liu & 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 \( \text{SO}_0(n, 1) \) and its Lie algebra \( \text{so}(n, 1) \) in the previous chapter, let us define again more detail that \( G(\tau) \in \text{SO}_0(n, 1) \) is the group value of \( G \) at a time \( \tau \). \( X \) denotes the numerical value of \( X \) at discrete time \( \tau \). Liu (2001) has been writing on his papers concerning to the connecting of \( G(\tau) \) and \( X \), i.e.

\[ X_{\tau+1} = G(\tau) X \]

Referring to Gallier & Quaintance (2017), specifically, we are able to redefine (29) as follows
\[ \omega := \|x\| = \|f_r(x, \tau)\|, r \in \mathbb{Z}^+ \cup \{0\}, \]

so that

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

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

\[ \exp: \mathfrak{so}(n, 1) \rightarrow SO_n(n, 1) \]

wherever

\[ e^{Y(r)} = \begin{bmatrix} I_n + \left( \frac{\cosh(\omega - 1)}{\omega^2} \right) f_r f_r^T & \frac{\sinh \omega}{\omega} f_r \\ \frac{\sinh \omega}{\omega} f_r^T & \cosh \omega \end{bmatrix} \]

\[ = \begin{bmatrix} I_n + \left( \frac{\cosh(\|f_r\| - 1)}{\|f_r\|^2} \right) f_r f_r^T & \frac{\sinh \|f_r\|}{\|f_r\|} f_r \\ \frac{\sinh \|f_r\|}{\|f_r\|} f_r^T & \cosh \|f_r\| \end{bmatrix} \] (32)

We are allowed to modify (32) into

\[ e^{\frac{\Delta t}{\|x_r\|} Y(r)} = \begin{bmatrix} I_n + \left( \frac{\cosh(\Delta t \|f_r\|\|x_r\| - 1)}{\|f_r\|^2} \right) f_r f_r^T & \frac{\sinh(\Delta t \|f_r\|\|x_r\|)}{\|f_r\|} f_r \\ \frac{\sinh(\Delta t \|f_r\|\|x_r\|)}{\|f_r\|} f_r^T & \cosh(\Delta t \|f_r\|\|x_r\|) \end{bmatrix} \]

For some \( \Omega := \frac{\Delta t}{\|x_r\|} \), we have

\[ e^{\Omega Y(r)} = \begin{bmatrix} I_n + \left( \frac{\cosh(\Omega \|f_r\|)}{\|f_r\|^2} - 1 \right) f_r f_r^T & \frac{\sinh(\Omega \|f_r\|)}{\|f_r\|} f_r \\ \frac{\sinh(\Omega \|f_r\|)}{\|f_r\|} f_r^T & \cosh(\Omega \|f_r\|) \end{bmatrix} \]

More far away,

\[ G(r) = e^{\Omega Y(r)} = \begin{bmatrix} I_n + \left( \frac{a_r - 1}{\|f_r\|^2} \right) f_r f_r^T & \frac{b_r f_r}{\|f_r\|} \\ \frac{b_r f_r}{\|f_r\|} & a_r \end{bmatrix} \in SO_0(n, 1) \] (33)

where

\[ a_r := \cosh(\Omega \|f_r\|) \]
\[ b_r := \sinh(\Omega \|f_r\|) \]

Substituting (33) into (31) and then exploiting (30) as well, are obtained
\[
\begin{bmatrix}
\frac{(a_r - 1)}{\|f_r\|} f_r \nabla f_r \\
\frac{b_r}{\|f_r\|} \nabla f_r \\
\frac{b_r}{\|f_r\|} \nabla f_r \\
\end{bmatrix}
\begin{bmatrix}
x_r + \frac{x_r(a_r - 1)}{\|f_r\|^2} f_r \nabla f_r + \frac{b_r}{\|f_r\|} \nabla f_r \\
x_r + \frac{b_r}{\|f_r\|} \nabla f_r \\
x_r + \frac{b_r}{\|f_r\|} \nabla f_r \\
\end{bmatrix}
\begin{bmatrix}
x_r \\
\|x_r\| \\
\|x_r\| \\
\end{bmatrix}
\]

\[
\Rightarrow \begin{bmatrix}
\|x_{r+1}\| = \frac{b_r}{\|f_r\|} x_r + a_r \|x_r\| \\
\|x_{r+1}\| = \frac{b_r}{\|f_r\|} x_r + a_r \|x_r\| \\
\|x_{r+1}\| = \frac{b_r}{\|f_r\|} x_r + a_r \|x_r\| \\
\end{bmatrix}
\]

Since we know that \( f_r = f_r \), accordingly

\[
x_r = x_r + \frac{x_r(a_r - 1)}{\|f_r\|^2} f_r \nabla f_r + \frac{b_r}{\|f_r\|} \nabla f_r \|x_r\| \\
= x_r + \left( \frac{(a_r - 1)f_r x_r + b_r \|x_r\| \|f_r\|}{\|f_r\|^2} \right) f_r \\
= x_r + \frac{\eta_r}{\|f_r\|} f_r \\
\]

whereupon

\[
\eta_r := \frac{(a_r - 1)f_r x_r + b_r \|x_r\| \|f_r\|}{\|f_r\|^2} \\
\]

Besides that, we have possession of

\[
\|x_{r+1}\| = \frac{b_r}{\|f_r\|} x_r + a_r \|x_r\| \\
= a_r \|x_r\| + \frac{b_r}{\|f_r\|} f_r x_r \\
\]

As well as we have been knowing previously that \( \eta_r \) is an adaptive factor. Next, we are going to give a proof for \( \eta_r > 0 \) started by the definition of Schwartz inequality that is \( f_r x_r \geq \|f_r\| \|x_r\| \) so as

\[
(a_r - 1)f_r x_r \geq -(a_r - 1)\|f_r\| \|x_r\| \\
\Rightarrow \eta_r \geq \left[ 1 - \left\{ \cosh \left( \frac{\Delta t \|f_r\|}{x_r} \right) - \sinh \left( \frac{\Delta t \|f_r\|}{x_r} \right) \right\} \right] \|x_r\| \|f_r\| \\
\Rightarrow \eta_r \geq \left[ 1 - \exp \left( \frac{\Delta t \|f_r\|}{x_r} \right) \right] \|x_r\| \|f_r\| > 0, [\forall \Delta t > 0]
\]

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

4. Numerical experiments

The simplest numerical method for the solution of initial value problems (IVPs) is the FTIM. By combining (24), (29), and (34), we acquire

$$x_{j}^{r+1} = x_{j}^{r} + \eta_{r} f_{i}(x_{j}^{r}, \tau_{r}), \quad \left[ f = f_{i} = (f_{1}, ..., f_{N}), x = x_{j} = (x_{1}^{r}, ..., x_{N}^{r}) \right. \in \mathbb{R}^{N}],$$

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

$$\sum_{j=1}^{N} (x_{j}^{r+1} - x_{j}^{r}) \leq \varepsilon^{2} \iff \| f_{i}(x_{j}^{r+1}) - f_{i}(x_{j}^{r}) \| \leq \varepsilon$$

(36)

where $\varepsilon$ is a selected criterion. If at a time $\tau_{0} \leq \tau_{r}$, then the equation (36) 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 (26)-(28). 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. The examples used to get the performance proles for the numerical experiments at this paper, refer to (Liu & Atluri, 2008a; Hock & Schittkowski, 1981; Schittkowski, 1987). The following parameters are used:

$$z_{1} = 0.3, z_{2} = 0.5, z_{3} = 0.7, \Delta t = 0.001$$

Figure 1. Performance profile of $\varphi_{p}$ in $n$ dimensions solved by FTIM where $p > 1$ and $\varepsilon = 10^{-7}$
5. Conclusion

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 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 nd 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 FTIM, is better than others. This research can be used to solve a few applications in engineering, economics, even still developed as a tool to support smart city research as well.

References


Prochazka, A. Numerical analysis in Matlab, Department of Computing and Control Engineering. Institute of Chemical Technology, Prague, Czech Republic.

