Proceedings of the World Congress on Engineering 2012 Vol I
WCE 2012, July 4 - 6, 2012, London, U.K.

Minimizing a Function Using a Sequence of Interval Vectors

G Panda, Member IAENG, Md A T Ansary *

Abstract—**This paper develops a methodology to generate a sequence of interval vectors (boxes), which converges to a very small box of negligible volume containing the minimum point a function of several variable. Each point of that box is an approximate solution. Interval extension of real valued function is used to propose the method.**

Keywords: Interval valued function; Inverse stable interval matrix; Regular interval matrix; Newton's method.

1 Introduction

Interval analysis is used in optimization theory by Robinson [5], Ichida and Fujii [3], Hansen [1], [2] and others in various directions. Most of these algorithms search the optimal solution of the optimization problem $\min_{x\in R^n} f(x)$ by solving the system of nonlinear equations $\nabla f = 0$, using interval Newton's method. In this paper we consider an optimization problem $\min_{x\in D} f(x)$, where D is a rectangular parallelepiped in R^n. In stead of solving $\nabla f = 0$ by interval Newton's method we construct a sequence of interval vectors (sequence of boxes) using inverse of interval matrices and prove that this sequence converges to very small box of negligible volume which contains the solution of $\min_{x\in D} f(x)$. For computation of inverse of an interval matrix, we follow the concept developed by Rohn([6]).

Throughout this paper intervals and real numbers are denoted by capital letters and small letters respectively. In additions to these the following notations are used.

$I(R)$= Set of intervals. ($A \in I(R)$ is the set, $A = [\underline{a}, \overline{a}] = \{x \in R | \underline{a} \le x \le \overline{a}\}$.)

$(I(R))^k$= Product space, $\underbrace{I(R) \times I(R) \times ... \times I(R)}_{k\ times}$.

$\hat{A}$= Degenerate interval $[a, a]$.

$\overrightarrow{A}$= Column vector whose elements are intervals. $\overrightarrow{A} = (A_1, A_2, ..., A_n)^T$, $A_j = [\underline{a_j}, \overline{a_j}]$, $j = 1, 2, ..., n.(\overrightarrow{A} \in (I(R))^n)$

$m(A) = \frac{\overline{a}+\underline{a}}{2}$, $w(A) = \overline{a} - \underline{a}$, $w(\overrightarrow{A}) = \max_{1\le j\le n}\{w(A_j)\}$.

Two interval vectors of same dimension intersect if they are not componentwise disjoint and their intersection is computed componentwise. Let $* \in \{+, -, \cdot, /\}$ be a binary operation on the set of real numbers. For $A, B \in I(R)$, the algebraic operations in $I(R)$ is, $A \circledast B = \{a * b : a \in A, b \in B\}$. In the case of division, it is assumed that $0 \notin B$. The following pre requisites are necessary to develop the methodology of this paper.

Definition 1.1 *[4] For a given function $f : R^n \to R$, the set image of the interval vector $\vec{X} = (X_1, X_2, ..., X_n)^T$ under f is the set, $\mathbf{f}(X_1, X_2, ..., X_n) = \{f(x_1, x_2, ..., x_n) : x_1 \in X_1, ..., x_n \in X_n\}$.*

Inverse of an interval matrix:
An interval matrix of order $m\times n$ is defined as a set of real matrices of the form, $A^I = [\underline{A}, \overline{A}] = \{A, \underline{A} \le A \le \overline{A}\}$ for some real matrices $\underline{A}$ and $\overline{A}$ of order $m \times n$ satisfying $\underline{A} \le \overline{A}$. Matrix inequalities are to be understood componentwise. We may write $A^I = (A_{ij})$, where $A_{ij} = [\underline{a_{ij}}, \overline{a_{ij}}]$, $i = 1, 2, ..., m; j = 1, 2, ..., n$; $\underline{A} = (\underline{a_{ij}})$, $\overline{A} = (\overline{a_{ij}})$. The computation of inverse of an interval matrix is introduced by Rohn [6].

Definition 1.2 *([6]) A square interval matrix A^I is said to be regular if each $A \in A^I$ is non singular.*

Since the set $\{A^{-1} : A \in A^I\}$ need not be an interval matrix, Rohn redefined inverse of interval matrix of A^I as the narrowest interval containing $\{A^{-1} : A \in A^I\}$ which is an interval matrix $B^I = [\underline{B}, \overline{B}]$, where $\underline{B}_{ij} =$

*Manuscript received February 02, 2012. G.Panda and Md. A.T. Ansary are with the Department of Mathematics, Indian Institute of Technology, Kharagpur, WB, 721302 INDIA. e-mail: G.Panda (geetanjali@maths.iitkgp.ernet.in), Md. A.T. Ansary (md.abutalha2009@gmail.com).

$\min_{A\in A^I}(A^{-1})_{ij}$ and $\overline{B}_{ij} = \max_{A\in A^I}(A^{-1})_{ij}$.

Definition 1.3 *([6],[7]) An interval square matrix A^I is said to be*

1. *inverse stable if it is regular and either $(A^{-1})_{ij} < 0 \ \forall\ A \in A^I$ or $(A^{-1})_{ij} > 0 \ \forall\ A \in A^I$, for each i, j;*
2. *symmetric if both A_c^I and Δ are symmetric; ($A_c = (m(A_{ij}))$ and $\Delta = \frac{1}{2}(w(A_{ij}))$ are real matrices associated with the interval matrix A^I. Then A^I can be expressed as $A^I = [A_c - \Delta, A_c + \Delta]$.)*
3. *positive (semi) definite if each $A \in A^I$ is positive (semi) definite.*

Let $Y = \{y \in R^n : |y_j| = 1,\ j = 1, 2, ...n\}$. For $q \in R^n$, a diagonal matrix T_q is defined as

$$(T_q)_{ij} = \begin{cases} q_i, & i = j; \\ 0, & i \neq j. \end{cases}.$$

Define A_{yz} by $A_{yz} = A_c - T_y \Delta T_z$. Then

$$(A_{yz})_{ij} = (A_c)_{ij} - y_i(\Delta)_{ij} z_j = \begin{cases} \underline{a_{ij}}, & \text{if } y_i z_j = 1; \\ \overline{a_{ij}}, & \text{if } y_i z_j = -1. \end{cases}$$

Theorem 1.1 *([6])* Let A^I be regular. Then for each $A \in A^I$ there exists non negative diagonal matrices $L_{yz}, y, z \in Y$ satisfying $\sum_{y,z\in Y} L_{yz} = E$ such that $A^{-1} = \sum_{yz} A_{yz}^{-1} L_{yz}$ holds, where $E = ee^T$, $e = (1, 1, ..., 1)^T$.

For a vector $x \in R^n$, its sign vector $sgn\ x$ is,

$$(sgn\ x)_i = \begin{cases} 1, & x_i \geq 0; \\ -1, & x_i < 0. \end{cases}.$$

For an inverse stable interval matrix A^I, we denote $y(i)$ as the sign vector of i-th row of ${A_c}^{-1}$ and $z(j)$ as the sign vector of j-th column vector of ${A_c}^{-1}$.

Theorem 1.2 *([6])* If A^I is inverse stable then $\underline{B}_{ij} = (A^{-1}_{-y(i),z(j)})_{ij}$ and $\overline{B}_{ij} = (A^{-1}_{y(i),z(j)})_{ij}$ for all $i, j = 1, 2, ..., n$.

2 Generating a converging sequence of interval vectors

Theorem 2.1 Let f be a convex function on a rectangular parallelepiped D in R^n. (A rectangular parallelepiped in R^n can be treated as an interval vector.) Then the sequence of interval vectors $\{\vec{X}^k\}$ in $I(R)^n$, k=0,1,2,..., generated by the formula,

$$\vec{X}^0 = D, \quad \vec{X}^{k+1} = \vec{X}^k \bigcap \Big(\widehat{\vec{X}^k} - (\nabla f(\widehat{\vec{X}^k}))^T ({A^k}^I)^{-1} \Big),$$

where $\widehat{\vec{X}^k}$ is the degenerate interval $[x^k, x^k]$ for any $x^k \in \vec{X}^k$ and ${A^k}^I = (\nabla^2 \mathbf{f})(\vec{X}^k)$ is the set image of $\vec{X}^k$ under $(\nabla^2 f)x)$(See Definition 1.1) contains the solution of $(P):\ \min_{x\in D} f(x)$ for each k.

Proof: Let ξ be the solution of (P). We need to show that $\xi \in \vec{X}^k$ for each k. This may be done by method of induction. $\xi \in \vec{X}^0$ is true. Suppose $\xi \in \vec{X}^k$. For $x^k \in \vec{X}^k$, $f(x^k) \simeq f(\xi) + (x^k - \xi)^T \nabla f(\xi) + \frac{1}{2}(x^k - \xi)^T \nabla^2 f(\xi + \theta(x^k - \xi))(x^k - \xi)$, $\theta \in (0, 1)$. Since $\nabla f(\xi) = 0$, so x^k is the approximate solution of (P) if $x^k - \xi = (\nabla f(x^k))^T (\nabla^2 f(\xi + \theta(x^k - \xi))^{-1}$. Since f is convex on D, so it is convex on each set $\vec{X}^k$. ${A^k}^I = (\nabla^2 \mathbf{f})(\vec{X}^k) = \{A : A = \nabla^2 f(x^k), x^k \in X^k\}$ is the set image of $\vec{X}^k$ under $\nabla^2 f$. Since f is convex on $\vec{X}^k$ so $\nabla^2 f(x^k)$ is non-singular, for every $x^k \in \vec{X}^k$. Hence ${A^k}^I$ is regular by Definition 1.2. $({A^k}^I)^{-1}$ exists.
So for any $x^k \in \vec{X}^k$, $x^k - \xi \in (\nabla f(x^k))^T (A_k^I)^{-1}$. If $\widehat{\vec{X}^k}$ denotes the degenerate interval vector $[x^k, x^k]$, then $\xi \in \widehat{\vec{X}^k} - (\nabla f(\widehat{\vec{X}^k}))^T (A_k^I)^{-1}$.
This implies $\xi \in \vec{X}^{k+1}$. □

From the above result we conclude that $\xi \in \cap_k \vec{X}^k$. $\cap_k \vec{X}^k$ may be a large box. To obtain the approximate solution we need to show that $\cap_k \vec{X}^k$ is a box of negligible volume. Denote $N(X^k) = \widehat{\vec{X}^k} - (\nabla f(\widehat{\vec{X}^k}))^T (\nabla^2 \mathbf{f}(\vec{X}^k))^{-1}$, $\vec{X}^k = (X_1^k, X_2^k, ..., X_n^k)^T$, $({A^k}^I)^{-1} = {B^k}^I = ([{\underline{b_{ij}}}^k, \overline{b_{ij}^k}])$, $v_j = (\frac{\partial f}{\partial x_j})_{x=x^k}$ $j = 1, 2, ..., n$.

Theorem 2.2 *If $X^k \not\subseteq N(X^k)$ for all k then $w(\vec{X}^k) \to 0$ as $k \to \infty$.*

Proof: Let i^{th} component of $N(X^k)$ be $N(X_i^k)$. Then $N(X_i^k) = [x_i^k - \overline{m_i^k}, x_i^k - \underline{m_i^k}]$, where $\underline{m_i^k} = \sum_{i=1}^n c_{ij} v_j$ and $\overline{m_i^k} = \sum_{i=1}^n d_{ij} v_j$ with

$$c_{ij} = \begin{cases} \underline{b_{ij}}, & if\ v_j \geq 0; \\ \overline{b_{ij}}, & if\ v_j < 0 \end{cases}, \ and \quad d_{ij} = \begin{cases} \overline{b_{ij}}, & if\ v_j \geq 0; \\ \underline{b_{ij}}, & if\ v_j < 0. \end{cases}$$

ISBN: 978-988-19251-3-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$X_i^{k+1} = X_i^k \cap N(X_i^k) = [\underline{x_i^{k+1}}, \overline{x_i^{k+1}}]$ where,

$$\underline{x_i}^{k+1} = \begin{cases} max\{\underline{x_i}^k, x_i^k - \overline{m_i^k}\}, & if\ \overline{m_i^k} \geq 0 \\ x_i^k - \overline{m_i^k}, & if\ \overline{m_i^k} < 0, \end{cases}$$

$$\overline{x_i}^{k+1} = \begin{cases} x_i^k - \underline{m_i^k}, & if\ \underline{m_i^k} \geq 0 \\ min\{\overline{x_i^k}, x_i^k - \underline{m_i^k}\}, & if\ \underline{m_i^k} < 0. \end{cases}$$

Here arises three cases.
CASE-1:
let $\underline{m_i^k} > 0$. Then $\underline{x_i^{k+1}} = max\{\underline{x_i}^k, x_i^k - \overline{m_i^k}\}$ and $\overline{x_i^{k+1}} = x_i^k - \underline{m_i^k}$. Since $\underline{m_i^k} > 0$, there exist real numbers $c_i > 0$ and $d_i > 0$ such that $\underline{m_i^k} = c_i \ \sum_{j=1}^n |v_j|$ and $\overline{m_i^k} = d_i \ \sum_{j=1}^n |v_j|$. $\overline{m_i^k} > \underline{m_i^k}$ implies $d_i > c_i$.
If $\underline{x_i^k} \geq x_i^k - \overline{m_i^k}$, that is $\sum_{j=1}^n |v_j| \geq 1/d_i(x_i^k - \underline{x_i^k})$, then

$$\begin{aligned} w(X_i^{k+1}) = x_i^k - \underline{m_i^k} - \underline{x_i^k} &= x_i^k - \underline{x_i^k} - c_i \sum_{j=1}^n |v_j| \\ &\leq (1 - \frac{c_i}{d_i}) w(X_i^k). \end{aligned}$$

Similarly if $\underline{x_i^k} < x_i^k - \overline{m_i^k}$, then

$$w(X_i^{k+1}) < (1 - \frac{c_i}{d_i}) w(X_i^k)$$

CASE-2:
Let $\overline{m_i^k} < 0$. Then $\underline{x_i^{k+1}} = x_i^k - \overline{m_i^k}$ and $\overline{x_i^{k+1}} = min\{\overline{x_i^k}, x_i^k - \underline{m_i^k}\}$. Since $\overline{m_i^k} < 0$, so $\underline{m_i^k} < 0$. Hence there exist positive real numbers g_i and h_i such that $\underline{m_i^k} = -g_i \ \sum_{j=1}^n |v_j|$ and $\overline{m_i^k} = -h_i \ \sum_{j=1}^n |v_j|$. Further $g_i > h_i$ since $\underline{m_i^k} < \overline{m_i^k}$.
If $\overline{x_i^k} < x_i^k - \underline{m_i^k}$ then $\overline{x_i^k} - x_i^k < g_i \ \sum_{j=1}^n |v_j|$. So $\sum_{i=1}^n |v_i| > 1/g_i \ (\overline{x_i^k} - x_i^k)$. If $\overline{x_i^k} \geq x_i^k - \underline{m_i^k}$, that is $(\overline{x_i^k} - x_i^k) \geq g_i \ \sum_{j=1}^n |v_j|$ then $\sum_{j=1}^n |v_j| \leq 1/g_i \ (\overline{x_i^k} - x_i^k)$. Proceeding as in Case 1, we get

$$w(X_i^{k+1}) \leq (1 - \frac{h_i}{g_i}) w(X_i^k).$$

Case -3
Let $\underline{m_i^k} \leq 0 \leq \overline{m_i^k}$. Then $X_i^{k+1} = [\underline{x_i^{k+1}}, \overline{x_i^{k+1}}] = [max\{\underline{x_i}^k, x_i^k - \overline{m_i^k}\}, min\{\overline{x_i^k}, x_i^k - \underline{m_i^k}\}$. If $x_i^k - \overline{m_i^k} > \underline{x_i^k}$ and $x_i^k - \underline{m_i^k} < \overline{x_i^k}$, then $w(X_i^{k+1}) = \overline{x_i^k} - \overline{m_i^k} < \overline{x_i^k} - \underline{x_i^k} = w(X_i^k)$. If $x_i^k - \overline{m_i^k} > \underline{x_i^k}$ and $x_i^k - \underline{m_i^k} \geq \overline{x_i^k}$ then $w(X_i^{k+1}) = (x_i^k - \underline{m_i^k}) - (x_i^k - \overline{m_i^k}) < \overline{x_i^k} - \underline{x_i^k} = w(X_i^k)$.

If $x_i^k - \overline{m_i^k} \leq \underline{x_i^k}$ and $x_i^k - \underline{m_i^k} < \overline{x_i^k}$ then $w(X_i^{k+1}) = x_i^k - \underline{m_i^k} - \underline{x_i^k} < \overline{x_i^k} - \underline{x_i^k} = w(X_i^k)$.
In case $x_i^k - \overline{m_i^k} \leq \underline{x_i^k}$ and $x_i^k - \underline{m_i^k} \geq \overline{x_i^k}$ we get $X_i^k \subseteq N(X_i^k)$, which is not possible since $X_i^k \nsubseteq N(X_i^k)$ for all $i = 1, 2, ..., n$.
From the above derivations we conclude that $w(X_i^{k+1}) \leq \gamma_i \ w(X_i^k)$ for some $\gamma_i < 1$ and for all $i = 1, 2, ..., n$. So $w(X_i^{k+1}) \leq \gamma \ w(X_i^k)$, where $\gamma = max\{\gamma_i : i = 1, 2, ..., n\} < 1$. Hence $max_i w(X_i^{k+1}) \leq \gamma . max_i w(X_i^k)$. That is $w(\vec{X}^{k+1}) \leq \gamma . w(\vec{X}^k)$ with $\gamma < 1$. Proceeding in this manner we get $w(\vec{X^k}) \leq \gamma^k w(\vec{X^0}) \to 0$ as $k \to \infty$, since $\gamma < 1$.

Remark 2.1

1. If $\vec{X}^k \subseteq N(\vec{X}^k)$ for some k, then $w(\vec{X}^{k+1}) = w(\vec{X}^k)$. So the process terminates. In this situation we may select another $x^{k'} \in \vec{X}^k$ with $x^{k'} \neq x^k$ so that $\vec{X}^k \nsubseteq N(\vec{X}^k)$ and continue the process.

2. For each k if $x^k \in \vec{X^k}$ is replaced by $x^k = m(\vec{X^k})$ then $...w(\vec{X^{k+1}}) < \frac{1}{2} w(\vec{X^k}) < \frac{1}{2^2} w(\vec{X^{k-1}}) < ...$ Hence $\{\vec{X^k}\}$ converges more rapidly to ξ. For each k, the degenerate interval $\widehat{\vec{X^k}}$ is $[x^k, x^k]$ for any selected point x^k of $\vec{X^k}$. Hence without loss of generality we may select x^k as $m(\vec{X^k})$ to write a computer program.

3. In general $({A^k}^I)^{-1}$ can be found following Theorem 1.1 which needs 2^n number of computations. If ${A^k}^I$ is inverse stable then its inverse can be found following Theorem 1.2 which needs $2n^2$ computations.

We summarize the results of the above two theorems in the following algorithm.

2.1 Algorithm :

1. Input $f(x)$, $\vec{X^0}$ and ε, where $\vec{X^0}$ is initial interval vector and ε is tolerance limit.

2. Compute $\nabla f(x)$ and $\nabla^2 f(x)$, set k=0.

3. Compute interval matrix A^k as set image of $\vec{X^k}$ under $\nabla^2 f$.

4. Compute $N(\vec{X^k}) = \widehat{\vec{X^k}} - [\nabla f(\widehat{\vec{X^k}})]^T {B^k}^I$ where ${B^k}^I = ({A^k}^I)^{-1}$, $x^k = m(\vec{X^k})$.

5. Compute $\vec{X}^{k+1} = \vec{X}^k \bigcap N(\vec{X}^k)$.

6. If $\vec{X}^{k+1} = \vec{X}^k$ then the process terminates. While computing $N(\vec{X}^k)$ in stead of taking $m(\vec{X}^k)$ take any other point $x^k \in \vec{X}^k$ and continue the process.
7. If $w(\vec{X}^{k+1}) \geq \varepsilon$, set $k = k + 1$ then goto step 3.
8. If $w(\vec{X}^{k+1}) < \varepsilon$ then $\vec{X}^{k+1}$ gives the final result. Any point in $\vec{X}^{k+1}$ can be considered as a solution of the given problem.

This algorithm may be explained in the following examples.

Example 2.1 Let $f(x_1, x_2) = x_1^4 + 12x_1^2 - x_1x_2 + x_2^4 + 6x_2^2 - x_1 - x_2$ and $\vec{X}^{(0)} = \begin{pmatrix} [-2,6] \\ [-2,6] \end{pmatrix}$.
$\nabla^2 f(x) = \begin{pmatrix} 12x_1^2 + 24 & -1 \\ -1 & 12x_2^2 + 12 \end{pmatrix}$,
${A^0}^I = \nabla^2 \mathbf{f}(\vec{X}^0) = \begin{pmatrix} [24,456] & [-1,-1] \\ [-1,-1] & [12,444] \end{pmatrix}$.
Any element of ${A^0}^I$ is $A^0 = \begin{pmatrix} p & -1 \\ -1 & q \end{pmatrix} \in {A^0}^I$, where $24 \leq p \leq 456, 12 \leq q \leq 444$. $det(A^0) = pq - 1 \geq 287$.

By Definition 1.2, ${A^0}^I$ is regular.
${A^0}^{-1} = \begin{pmatrix} \frac{q}{pq-1} & \frac{1}{pq-1} \\ \frac{1}{pq-1} & \frac{p}{pq-1} \end{pmatrix}$, where $\frac{q}{pq-1} > 0, \frac{p}{pq-1} > 0$.
So by Definition 1.3, ${A^0}^I$ is inverse stable. Now we determine its inverse by Theorem 1.2.
$A_c^0 = \begin{pmatrix} 240 & -1 \\ -1 & 228 \end{pmatrix}$, ${A_c^0}^{-1} = \frac{1}{54719} \begin{pmatrix} 228 & 1 \\ 1 & 240 \end{pmatrix}$ and $\Delta^0 = \begin{pmatrix} 216 & 0 \\ 0 & 216 \end{pmatrix}$.
Also $y(1) = y(2) = z(1) = z(2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.
$A^0_{-y(1)z(1)} = A_c^0 - T_{-y(1)} \Delta^0 T_{z(1)} = \begin{pmatrix} 456 & -1 \\ -1 & 444 \end{pmatrix}$.
So $\underline{B^0}_{11} = ({A^0_{-y(1)z(1)}}^{-1})_{11} = .002193$.

Similarly $\underline{B^0}_{12}$, $\underline{B^0}_{21}$, $\underline{B^0}_{22}$ can be calculated. We see that $\underline{B^0} = \begin{pmatrix} .002193 & .000004 \\ .000004 & .0837 \end{pmatrix}$.
$A^0_{-y(1)z(1)} = A_c^0 - T_{-y(1)} \Delta^0 T_{z(1)} = \begin{pmatrix} 24 & -1 \\ -1 & 12 \end{pmatrix}$.
So $\overline{B^0}_{11} = ({A^0_{y(1)z(1)}}^{-1})_{11} = .0418$.
Similarly $\overline{B^0}_{12}$, $\overline{B^0}_{21}$, $\overline{B^0}_{22}$ can be calculated. We see that $\overline{B^0} = \begin{pmatrix} .0418 & .0035 \\ .0035 & .0837 \end{pmatrix}$.

Using Definition 1.1, we can calculate

$({A^0}^I)^{-1} = {B^0}^I =$
$\begin{pmatrix} [0.002193 ,0.0418] & [0.000004 ,0.0035] \\ [0.000004 ,0.0035] & [0.002252 ,0.0837] \end{pmatrix}$
and

$\nabla \mathbf{f}(m(\vec{X}_0)) = \begin{pmatrix} 77 \\ 53 \end{pmatrix}$.
So $N(\vec{X}^0) = \begin{pmatrix} [-1.4042 , 1.83088] \\ [-2.7004 , 1.88025] \end{pmatrix}$
and $\vec{X}^1 = \vec{X}^0 \bigcap N(\vec{X}^0) = \begin{pmatrix} [-1.4042 , 1.83088] \\ [-2 , 1.88025] \end{pmatrix}$.
Proceeding as above we find

$$\vec{X}^2 = \vec{X}^1 \bigcap N(\vec{X}^1) = \begin{pmatrix} [0.0374 , 0.1544] \\ [-0.0424 , 0.1007] \end{pmatrix}.$$

The sequence $\{\vec{X}^k\}$ can be generated in a similar way. Using the above algorithm, we generate this sequence through a MATLAB program (See Appendix for MATLAB program) with initial box $\vec{X}^0 = \begin{pmatrix} [-100,1000] \\ [-100,1000] \end{pmatrix}$, tolerance limit $\varepsilon = 10^{-7}$ and $x^k = m(\vec{X}^k)$ for each k and get it's solution,

$$\vec{X^*} = \begin{pmatrix} [\ 0.045271507, 0.045271508] \\ [0.086887301, 0.086887324] \end{pmatrix}.$$

Total number of iterations is 11, execution time is 0.09877717 seconds. Using Newton's Method with same tolerance limit, the solution is $x_1 = 0.0452715,\ x_2 = 0.0868873$.

Total number of iterations is 18 and execution time is 0.00041 seconds. In this example our algorithm searches four vertices in stead of a single point as in Newton's method. For this reason more time to execute the program. But this has an advantage that, any point in the box $\vec{X^*}$ is an approximate solution. Once this is established the user has the choice for selecting a convenient point in this box as solution.

Next, we consider another example where the process terminates after some iterations.

Example 2.2 For $f(x_1, x_2) = -12x_2 + x_1^3 + 3x_2^2 - 6x_1x_2$, if we start with $\vec{X}^{(0)} = \begin{pmatrix} [2,98] \\ [-10,110] \end{pmatrix}$, $\varepsilon = 10^{-7}$ and $x^k = m(\vec{X}^k)$, we see that the process terminates after

ISBN: 978-988-19251-3-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

44 iterations with solution $X_1 = [2, 11.92249939]$ and $X_2 = [-10, 13.92249939]$. Following Remark 2.1, if we change $x^k = m(\vec{X^k})$ to $x_i^k = (3\underline{x}_i^k + \overline{x}_i^k)/4$ then we get the final solution as $X_1 = [3.23606798, 3.23606798]$ and $X_2 = [5.23606798, 5.23606798]$ with total number of iterations as 15, total execution time as 0.12130 seconds.

3 Conclusion

In this paper we have addressed an unconstrained minimization problem on a rectangular parallelepiped in n dimension and developed an algorithm to find it's solution. In place of reaching at one solution by existing methods, we reach at a very small box of negligible volume, whose points are approximate solutions of the problem. So there is a flexibility for the decision maker to select a suitable solution. Our algorithm can be modified to generate a sequence for minimizing a function over a hyper-sphere.

4 Appendix:(Programming in MATLAB)

```
% Matlab programming for  min f(x)=x(1).^4+
12*x(1)^2-x(1)*x(2)-x(1)+x(2).^4
+6*x(2)^2-x(2) ,
int i;int j;int n;int iter;
double X0l;double X0u;
single eps;
 %define types of variables
%getting input
n=input('supply order');
eps=input('supply error of tolerance');
X0l=input('supply lower');
X0u=input('supplyupper');
tic
xk=[(X0l(1)+X0u(1))/2,(X0l(2)+X0u(2))/2];
iter=0;
wx=max(X0u-X0l);
while wx>=eps
 xk=[(X0l(1)+X0u(1))/2,(X0l(2)+X0u(2))/2];
  %taking inclusion isotonic extension
  % of grad^2 f
Al=[12*min(intvsquar([X0l(1),X0u(1)]))+24,-1;
-1,12*min(intvsquar([X0l(2),X0u(2)]))+12];
Au=[12*max(intvsquar([X0l(1),X0u(1)]))+24,-1;
-1,12*max(intvsquar([X0l(2),X0u(2)]))+12];
  %computing inverse of an interval matrix
  Ac=(Al+Au)/2;del=(Au-Al)/2;A=inv(Ac);
  for i=1:n
    for j=1:n
        if A(i,j)>=0 B(i,j)=1;
        else B(i,j)=-1;
        end
    end
  end
for i=1:1:n
    for j=1:n
       C=diag (B(i,:));E=diag(B(:,j));
       F=Ac+C*del*E;G=Ac-C*del*E;
       H=inv(F);invl(i,j)=H(i,j);
       K=inv(G);invu(i,j)=K(i,j);
    end
end
%[invl,invu] is the invrse matrix
gradf=[4*(xk(1)).^3+24*xk(1)-xk(2)-1,
4*(xk(2)).^3+12*xk(2)-xk(1)-1];
%computing N(X)
for i=1:n
    suml(i)=0;sumu(i)=0;
    for j=1:n
 suml(i)=suml(i)+min(intvmult([gradf(j),gradf(j)],
 [invl(j,i),invu(j,i)]));
 sumu(i)=sumu(i)+max(intvmult([gradf(j),gradf(j)],
 [invl(j,i),invu(j,i)]));
    end
    nxu(i)=xk(i)-suml(i);nxl(i)=xk(i)-sumu(i);
end
%taking intersection
for i=1:n
    Xnewl(i)=max([X0l(i),nxl(i)]);
    Xnewu(i)=min([X0u(i),nxu(i)]);
    if Xnewl(i)>Xnewu(i)
disp(sprintf(' No solution'));
break;
    end
end
 wxnew=max(Xnewu-Xnewl);
   if wxnew==wx
       disp(sprintf('the process terminates'))
       break
   else
      iter=iter+1;
   %  modify the value
      X0l=Xnewl;X0u=Xnewu;wx=wxnew;
   end
end
%printing final result
for i=1:n
    disp(sprintf('X(%d)=
    [%12.9f,%12.9f]',i,X0l(i),X0u(i)));
end
    disp(sprintf('No.of iterations=%d',iter));
```

ISBN: 978-988-19251-3-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

```
disp(sprintf('time taken=%10.8f seconds',toc));
```

References

[1] Hansen, E. R.: Global optimization using interval analysis-The one dimensional case., J. Optim.Theory Appl. 29, 331-344 (1979).

[2] Hansen, E. R., Walster, W. G.: Global Optimization Using Interval Analysis, Marcel Dekker, Inc. and Sun. Microsystems, Inc.(2004).

[3] Ichida, K., Fujii, Y.: Multicriterian optimization using interval analysis., Computing, 44, 47- 57 (1990).

[4] Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to interval analysis. Philadelphia, SIAM (2009).

[5] Robinson, S. M.:Computable error bounds for nonlinear programming, Math.Programming 5, 235-242 (1973).

[6] Rohn, J.: Inverse of an interval matrix. Siam J. Numer. Anal..30,3,864-870 (1993).

[7] Rohn, J.: Positive definiteness and stability of interval matrices. Siam J. Matrix Anal. Appl.., 15,1,175-184 (1994).

ISBN: 978-988-19251-3-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)