平成 11 年度

卒業論文

Affine Arithmetic を利用した全解探索

On All Solution Algorithms using Affine Arithmetic

平成 12 年 2 月 4 日

指導教授： 柏木 雅英 助教授

早稲田大学理工学部情報学科

G96P036-9 川上 修

目 次

図 目 次

表 目 次

第 1 章

序論

1.1 背景

n 次元非線形方程式

$$f(x) = 0$$

を解くために Krawczyk の区間写像による全解探索が研究されている. この手法は探索する領域を分割して行き, その分割された領域において解が存在するか否かを全ての領域を対象に解を探索していく手法である.

非線形方程式の全ての解を効率良く求めようと考えた場合, この Krawczyk 法による解の存在性判定を効率化することが重要である. 一方, 解の非存在性の判定法として $0 \notin f(x)$ が成立した場合に解の非存在性が示されるが, この判定法を用いて解が存在しない領域をいかにして速やかに削除して行くかということも効率化を考えると重要である. 文献 [2] には Affine Arithmetic を利用した非存在テストが提案されている.

1.2 本論文の目的

本論文では, 従来の非存在テストによる全解探索,Affine Arithmetic を利用した非存在テストによる全解探索のアルゴリズムをそれぞれ考え,Affine によってどれだけ成果, つまり, 時間を短縮させられたかをプログラムの数値結果を用いて確かめることを目的とする.

1.3 本論文の構成

本論文の構成は以下のとおりである.

- 第 2 章 本論文での準備として区間演算と Krawczyk 法についてそれぞれ解説する.
- 第 3 章 Affine Arithmetic についての基本的な概念を解説する.
- 第 4 章 Affine Arithmetic を利用した存在領域除去法を紹介し, それを使った全解探索のアルゴリズムを解説する.

- 第5章 実際の数値例にて本論文のアルゴリズムを検証する.

第 2 章

準備

2.1 はじめに

本章では, 準備として, 区間演算の基本的概念,Krawczyk 法について簡単に説明する.

2.2 区間演算の基本的概念

定義 2.2.1 (区間)

区間演算において区間とは

$$[\underline{x}, \overline{x}] = \{x \in R | \underline{x} \leq x \leq \overline{x}\}, \underline{x} \leq \overline{x} \in R$$

と表される閉区間である.

定義 2.2.2 (二項演算)

区間 $X = [p, q], Y = [v, w]$ に対して, 二項演算 $* \in \{+, -, \cdot, /\}$ を,

$$X * Y = \{x * y | x \in X, y \in Y\}$$

で定義する. ただし, 除算 $* = /$ の場合には $0 \notin Y$ とする. 実際の区間演算は,

$$X + Y = [p + v, q + w]$$

$$X - Y = [p - w, q - v]$$

$$X \cdot Y = [a, b](a = min(pv, pw, qv, qw), b = max(pv, pw, qv, qw)$$

$$X/Y = [p, q] \cdot [1/w, 1/v]$$

のように表現できる.

２つの区間 X, Y に関する二項演算 $*$ の結果は再び区間となる. すなわち, 区間全体は二項演算に関して閉じている.

2.3　Krawczyk の方法

まず準備として, 不動点定理のうちで最も基本的な**縮小写像原理**を示す.

定理 2.3.1 (縮小写像の原理)

X を $d(x,y)$ を距離とする完備距離空間とし,$g : X \to X$ を全ての $x, y \in X$ に対して

$$d(g(x), g(y)) \leq \alpha d(x,y) \tag{2.1}$$

を満たすような写像とする. ただし,$\alpha < 1$ とする (このような g を縮小写像と言い, 実数 α を縮小定数と言う.) このとき, 以下が成立する :

1. $g(x) = x$ を満たす点 x^*(不動点と言う) が X に唯一存在する.

2. x_0 を X 内の任意の点とし, 点列 $\{x_k\}$ を

$$x_{k+1} = g(x_x) \tag{2.2}$$

で定めると,x_k は x^* に収束し, その収束速度は

$$d(x_k, x^*) \leq \frac{\alpha^k}{1-\alpha} d(x_1, x_0) \tag{2.3}$$

で表される.

3. X 内の任意の点 x_0 に対して,

$$d(x_0, x^*) \leq \frac{d(g(x_0), x_0)}{1-\alpha} \tag{2.4}$$

□

上の定理は一般的な形で書いてあるが, 本稿では n 次元ユークリッド空間 R^n を扱っているので,

$$\begin{aligned} X &: R^n\text{内の閉集合} \\ d(x,y) &: \|x-y\|.\text{ただしノルムは } \|x\| = max|x_i| (\textbf{最大値ノルム}) \text{ で定める.} \end{aligned}$$

のように読み替えれば良い.

非線形方程式

$$f(x) = 0, f : R^n \to R^n \tag{2.5}$$

の解の存在を保証するために, これを同値な不動点形式に変換し, 不動点定理を適用することを考える. 単純に

$$x = x + f(x) \tag{2.6}$$

のようにしても良いが, これでは右辺の縮小性が成立するかどうかは f によることになってしまう. 近似解の近辺での f の傾きを近似する行列とし, これを用いて f の傾きを補正して,

$$x = x - L^{-1}f(x) \tag{2.7}$$

のようにする ($f(x) = 0$ を $-L^{-1}f(x) = 0$ だったと考える). 以下この右辺を $g(x)$ とおく：

$$g(x) = x - L^{-1}f(x) \tag{2.8}$$

L が逆行列をもつならば x が $f(x) = 0$ を満たすことと x が $x = g(x)$ を満たすことは同値なので, 以下 $x = g(x)$ の不動点の存在を示すことを考えれば良い. 以下, 区間 $I \subset R^n$ とし,I に g の不動点が存在することを示すことを考える. 示すべきことは,

1. $\{g(x)|x \in I\} \subset I$($g$ が I から I への写像であること)
2. g が I 上で縮小写像であること.

の 2 つである.

前者は, 区間演算を用いて

$$G(I) = I - L^{-1}F(I) \tag{2.9}$$

のように g に対する区間演算を用い,$G(I) \subset I$ を確認すれば容易に確認できるように見える. しかし, 区間演算の減算の定義の仕方では,$c = a - b$ ならば c の区間幅は a の区間幅と b の区

間幅の和となることを考えれば, 決して $G(I) \subset I$ は成立しない！写像 $g(x) = x - L^{-1}f(x)$ は, 傾きを 0 に近づけるためによく似た傾きの関数の減算を行なっている訳で, これは区間演算の苦手な計算の形そのものである. よって, 区間演算の過大評価を抑えるための何らかの手法を併用する必要がある.Krawczyk の方法では, 平均値形式を用いて,c を I の中心として

$$\begin{aligned} K(I) &= g(c) + G'(I)(I-c) \\ &= c - L^{-1}f(c) + (E - L^{-1}F'(I))(I-c) \end{aligned} \tag{2.10}$$

(E は単位行列) のような **Krawczyk 写像** K について,

$$K(I) \subset I \tag{2.11}$$

を確認する.

後者の条件については, 次のように確認する. 平均値の定理により,$x, y \in I$ ならば

$$\|g(x) - g(y)\| \leq \max_{x \in I} \|g'(x)\| \|x - y\| \tag{2.12}$$

が成立する. ここで行列 $g'(x)$ のノルムは,

$$\|A\| = \sup_{\|x\|=1} \|Ax\| \tag{2.13}$$

となるように定義されたもので,R^n のノルムが $\max |x_i|$(最大値ノルム) ならば

$$\|A\| = \max_i \sum_j |A_{ij}| \tag{2.14}$$

で計算できる. すなわち, 区間行列のノルムを

$$\|A\| = \max_i \sum_j \max_{x \in A_{ij}} |x| \tag{2.15}$$

で定義し,

$$\|G'(I)\| = \|E - L^{-1}F'(I)\| < 1 \tag{2.16}$$

が成立すれば,g は I 上で縮小写像であることが言える.

以上をまとめると, 次のようになる：

定理 2.3.2 (Krawczyk の方法)

$f : R^n \to R^n, I \in IR^n, c$ を I の中心,$L \simeq f'(c)$ を正則な行列,F' を f の区間包囲とする.

$$K(I) = c - L^{-1}f(c) + (E - L^{-1}F'(I))(I - c) \tag{2.17}$$

としたとき,

$$K(I) \subset I \tag{2.18}$$

$$\|E - L^{-1}F'(I)\| < 1 \tag{2.19}$$

が成立するならば,I に解が唯一存在することが保証される.

2.4 むすび

本章では, これから非線形方程式の解の非存在を検証するための準備として区間演算とKrawczyk 法について解説した.(文献 [1], 文献 [3], 文献 [4] 参照) 後に述べられる全解探索法について用いられ, 非線形の問題を考えていく上での基本となる事柄である.

第 3 章

Affine Arithmeticについて

3.1 はじめに

ここでは, 第 2 章で論じた区間演算の問題点を述べ, それを解決する方法としての Affine Arithmetic についての基本的な概要, そして, 実際例を使っての Affine Arithmetic による効果について, 論じていきたい.

3.2 区間演算の問題点

区間演算の問題点の一つとして, 確かに計算された区間は真の値を含むものの, 区間幅が極端に広くなってしまうことが多い点が挙げられる. これは特に関数の値域を評価する場合など, 比較的幅の広い区間を扱った場合に顕在化する. 例として, 関数

$$f(x) = x^2 - 2x \tag{3.1}$$

の値域を, 区間 $[0.9, 1.1]$ で評価する問題を考えよう. 区間演算の定義に従って計算すると,

$$\begin{aligned} [0.9, 1.1]^2 &- 2 \times [0.9, 1.1] \\ &= [0.81, 1.21] - [1.8, 2.2] \\ &= [-1.39, -0.59] \end{aligned} \tag{3.2}$$

のようになるが, 真の像は,

$$\{f(x) | x \in [0.9, 1.1]\} = [-1, -0.99] \tag{3.3}$$

であり, 実に 80 倍もの区間幅の開きがあることが分かる. この現象は 2 つの関数 x^2 と $2x$ が同じ x の関数で互いに相関があるにも関らず, 区間演算における減算ではそれを無視して独立な値として計算を行なってしまうことに原因がある.

この問題点を解消する方法として,Affine Arithmetic による方法がある. その Affine Arithmetic について, 説明したいと思う.

3.3 Affine Arithmeticの基本概念

3.3.1 Affine形式

Affine Arithmeticは, 変数間の相関性を考慮することにより区間演算の先程の問題を解決する方法の一つである.

Affine Arithmeticでは, 変動範囲が $-1 \leq \varepsilon_k \leq 1$ であるようなダミー変数 ε_k を用いて, その線形結合

$$a_0 + a_1\varepsilon_1 + \cdots + a_n\varepsilon_n \tag{3.4}$$

形 (Affine形式) で数 (区間) を表現する. 計算機にはその係数 $a_0 \cdots a_n$ 記憶する. なお, 後述するように ε_k の最大数 n は計算機の途中で変化する.

例えば,Affine形式 x, y があって, それが

$$x = 1 + 0.5\varepsilon_1 \tag{3.5}$$

$$y = 1 + 0.5\varepsilon_2 \tag{3.6}$$

であったとしよう. これは x と y に相関性がない状態で, このとき (x, y) が取り得る領域は図3.1のようになる.　これに対して,

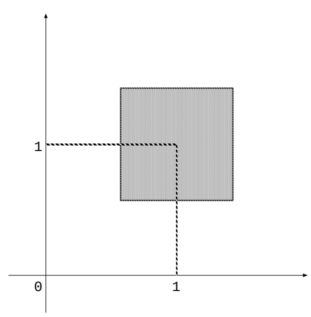

図 3.1: x と y に相関性が無い場合

$$\begin{aligned} x &= 1 + 0.5\varepsilon_1 && (3.7) \\ y &= 1 + 0.4\varepsilon_1 + 0.1\varepsilon_2 && (3.8) \end{aligned}$$

では,x, y それぞれが取り得る範囲は $[0.5, 1.5]$ で変わっていないが,ε_1 の係数を見ると分かるように両者には強い相関があり,(x, y) の取り得る領域は図 3.2 のようになる.

n 変数関数 $f(x_1, \cdots, x_n)$ を Affine Arithmetic で評価する場合, 最初に与えられる n 個の

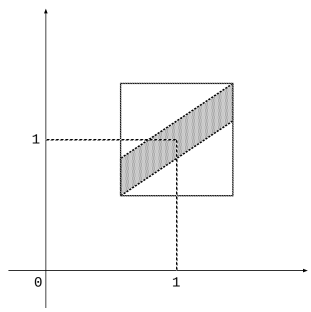

図 3.2: x と y に相関性がある場合

変数は一般に互いに無相関であるから,n 個の ε を用いて

$$
\begin{aligned}
x_1 &= \frac{\overline{x_1} + \underline{x_1}}{2} + \frac{\overline{x_1} - \underline{x_1}}{2}\varepsilon_1 \\
x_2 &= \frac{\overline{x_2} + \underline{x_2}}{2} + \frac{\overline{x_2} - \underline{x_2}}{2}\varepsilon_2 \\
&\vdots \\
x_n &= \frac{\overline{x_n} + \underline{x_n}}{2} + \frac{\overline{x_n} - \underline{x_n}}{2}\varepsilon_n
\end{aligned}
\tag{3.9}
$$

のような Affine 形式で初期化しておく. 但し, 入力変数 x_k の変域を $[\underline{x_k}, \overline{x_k}]$ とする.

なお,Affine 形式

$$x = a_0 + a_1\varepsilon_1 + \cdots + a_n\varepsilon_n \tag{3.10}$$

は,

$$[a_0 - \delta, a_0 + \delta], \delta = \sum_{i=1}^{n} |a_i| \tag{3.11}$$

によっていつでも通常の区間に戻すことができる.

3.3.2 線形演算

Affine 形式の変数における演算は, 加減算及び定数倍の場合は自明である.

Affine 多項式 x, y：

$$x = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \tag{3.12}$$

$$y = y_0 + y_1\varepsilon_1 + \cdots + y_n\varepsilon_n \tag{3.13}$$

に対して, 加算, 減算は次のように定義する.

$$x \pm y = (x_0 \pm y_0) + (x_1 \pm y_1)\varepsilon_1 + \cdots + (x_n \pm y_n)\varepsilon_n \tag{3.14}$$

$$x \pm \alpha = (x_0 \pm \alpha) + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \tag{3.15}$$

定数の乗算は次のように定義する.

$$\alpha x = (\alpha x_0) + (\alpha x_1)\varepsilon_1 + \cdots + (\alpha x_n)\varepsilon_n \tag{3.16}$$

3.3.3 非線形単項演算

Affine 多項式

$$x = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \tag{3.17}$$

に対する非線形な演算 f について,$z = f(x)$ は一般的に Affine 多項式で表すことはできない. そこで,f を線形演算で近似し, 近似誤差を新しいダミー変数 ε_{n+1} を導入することによって表すことを考える.

まず,x の変域 I を

$$I = [x_0 - \delta, x_0 + \delta], \delta = \sum_{i=1}^{n} |x_i| \tag{3.18}$$

で求める. 次に, この領域 I において f をなるべく良く近似するような一次関数 $ax + b$ を求める [図 3.3].

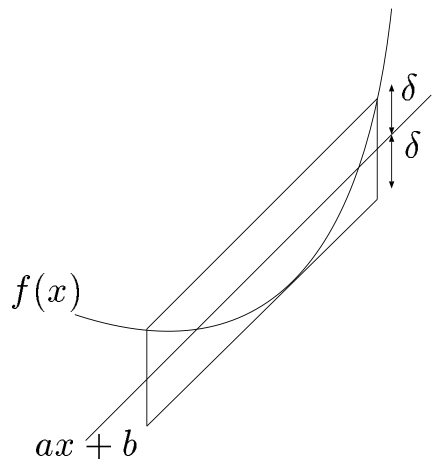

図 3.3: 非線形関数の線形近似

誤差の最大値

$$\delta = \max_{t\in I} |f(t) - (at+b)| \tag{3.19}$$

を求め, これをダミー変数 ε_{n+1} の係数とする. すなわち, 非線形関数 f は区間 I において

$$f(x) \in ax + b + \delta\varepsilon_{n+1} \tag{3.20}$$

であり, よって単項演算の結果は

$$a(x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n) + b + \delta\varepsilon_{n+1} \tag{3.21}$$

とすればよい. これはダミー変数が一つ増加した Affine 形式である.

単項演算の最適性は, 以下のように保証される. 最適な単項演算を, ダミー変数の係数が最も小さくなるものとする. すなわち,$\varepsilon_1, \cdots, \varepsilon_n \in [-1,1]\times\cdots\times[-1,1]$ において

$$\delta = \max_{-1\le\varepsilon_i\le1} |f(x) - (z_0 + z_1\varepsilon_1 + \cdots + z_n\varepsilon_n)| \tag{3.22}$$

を最小にするような Affine 多項式

$$z_0 + z_1\varepsilon_1 + \cdots + z_n\varepsilon_n \tag{3.23}$$

を用いれば,

$$z = z_0 + z_1\varepsilon_1 + \cdots + z_n\varepsilon_n + \delta\varepsilon_{n+1} \tag{3.24}$$

が最適な単項演算である. ここで, 先に述べた方法で一次関数 $ax+b$ を誤差が最小になるように選んだ場合は, 両者は同一の結果を与えることが証明できる.

3.3.4 非線形二項演算

Affine 多項式 x, y :

$$x = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \tag{3.25}$$

$$y = y_0 + y_1\varepsilon_1 + \cdots + y_n\varepsilon_n \tag{3.26}$$

に対する非線形な二項演算 $z = f(x, y)$ について考える. 例えば,$x \times y, x/y, x^y$ などである.

二項演算の場合も,$\varepsilon_1, \cdots, \varepsilon_n \in [-1, 1] \times \cdots \times [-1, 1]$ において

$$\delta = \max_{-1 \le \varepsilon_i \le 1} |f(x, y) - (z_0 + z_1\varepsilon_1 + \cdots + z_n\varepsilon_n)| \tag{3.27}$$

を最小にするような Affine 多項式

$$z_0 + z_1\varepsilon_1 + \cdots + z_n\varepsilon_n \tag{3.28}$$

を求めることができれば, これが最適である. しかし, これを直接解くことは極めて難しい.

単項演算の場合と同様に,x と y の変域 X, Y を求め,$(x, y) \in X \times Y$ において $f(x, y)$ を最良近似する一次式 $ax + by + c$ を求め,$ax + by + c + \delta\varepsilon_{n+1}$ を結果とする方法を使うことはできる. しかし, 単項演算の場合と違ってこの方法は必ずしも最良の近似を与えるとは限らない. これは,x と y に相関性がある場合, 点 (x, y) は一般に $X \times Y$ で与えられる長方形領域内の全ての点をとるとは限らないためである.

以上のように, 二項演算の場合の最良近似問題は未解決である. したがって, なるべくそれに近い方法を摸索することになる.

例えば乗算の場合は, 変域 X, Y の半径を δ_x, δ_y とすると $x \times y$ は

$$y_0x + x_0y - x_0y_0 \tag{3.29}$$

で最良近似され, 誤差は δ_x, δ_y となるので,

$$
\begin{aligned}
z &= y_0x + x_0y - x_0y_0 + \delta_x\delta_y\varepsilon_{n+1} \\
&= x_0y_0 + \sum_{i=1}^{n}(y_0x_i + x_0y_i)\varepsilon_i \\
&\quad +(\sum_{i=1}^{n}|x_i|)(\sum_{i=1}^{n}|y_i|)\varepsilon_{n+1}
\end{aligned} \tag{3.30}
$$

が用いられている. 前述したように, これは最適とは限らない.

3.4 Affine Arithmetic の効果

第 3.2 節で説明したとおり, 区間演算には区間幅が極端に広くなってしまうという弱点がある. 第 3.2 節の例を使ってどれ位その弱点が克服されたかを見る.

$$
\text{与える関数:} \quad f(x) = x^2 - 2x \tag{3.31}
$$

$$
\text{与える区間:} \quad [0.9, 1.1] \tag{3.32}
$$

であるので, まず,x を初期化すると,

$$
\begin{aligned}
x &= \frac{0.9 + 1.1}{2} + \frac{1.1 - 0.9}{2}\varepsilon_1 \\
&= 1 + 0.1\varepsilon_1
\end{aligned} \tag{3.33}
$$

ここで, この区間における x^2 の最良近似式は

$$
2x - 0.9995 + 0.005\varepsilon_2 \tag{3.34}
$$

なので,

$$
\begin{aligned}
x^2 &= 2(1 + 0.1\varepsilon_1) - 0.995 + 0.005\varepsilon_2 \\
&= 1.005 + 0.2\varepsilon_1 + 0.005\varepsilon_2
\end{aligned} \tag{3.35}
$$

また,

$$2x = 2 + 0.2\varepsilon_1 \tag{3.36}$$

なので,

$$\begin{aligned} x^2 - 2x &= 1.005 + 0.2\varepsilon_1 + 0.005\varepsilon_2 - (2 + 0.2\varepsilon_1) \\ &= -0.995 + 0.005\varepsilon_2 \end{aligned} \tag{3.37}$$

これを区間にもどすと $x^2 - 2x = [-1, -0.99]$ となり, 区間演算では $[-1.39, -0.59]$ となり 80 倍にもなってしまったのが, この Affine Arithmetic の方法では, 真の解そのものになることが分かる.

このように, Affine Arithmetic は非常に有用な方法であることが分かった.

3.5 むすび

本章では, 今回の論文の主題である Affine Arithmetic を利用した全解探索をするために必要とされる Affine Arithmetic の基本的概念について解説した. 次章より述べられる Affine Arithmetic を利用した非存在テストの方法を説明するのには, このことを知っているのを前提としている.

第 4 章

Affine Arithmetic を利用した存在領域除去法

4.1 はじめに

本章ではでまず, 非存在テストが f の区間包囲を利用したものである従来の全解探索法について述べ, それよりも早くなると予想される Affine Arithmetic を利用した方法 (文献 [2]) について説明したいと思う.

4.2 非線形の方程式の解の従来の全解探索

解の非存在を示すには, 以下のような方法がある.

- F を f の区間包囲とし,$0 \notin F(I)$ ならば I に $f(x)=0$ の解は存在しない.
- Krawczyk の方法で計算した $K(I)$ について,$K(I) \cap I = \emptyset$ ならば I に $f(x)=0$ の解は存在しない.

全解探索は, まず与えられた区間に対して

- Krawczyk の方法などによる解の存在テスト
- 上記の方法などによる解の非存在テスト

を行なう. どちらかが成立すればそれで探索は終了だが, どちらも成立しなければ区間を 2 分割し, それぞれについて同様のテストを再帰的に繰り返す. すなわち, 以下のようなアルゴリズムである :

アルゴリズム 4.2 (従来の全解探索アルゴリズム)

$f : R^n \to R^n, I_0 \in IR^n, F$ を f の区間包囲,F' を f の区間包囲とし, 方程式 $f(x)=0$ の I_0 内の全ての解を求めることを考える.

1. 区間のリスト $List$ を $List = \{I_0\}$ で初期化する.
2. $List$ が空なら終了. そうでなければ L から先頭の要素を取り出し I とする.
3. $F(I)$ を計算し,$0 \notin F(I)$ ならば I に解は存在しない.2 へ. そうでなければ次へ.

4. c を I の中心,$L \simeq f'(c)$ を正則な行列とし,

$$K(I) = c - L^{-1}f(c) + (E - L^{-1}F'(I))(I - c) \tag{4.1}$$

を計算し,

$$K(I) \subset I \tag{4.2}$$

$$\|E - L^{-1}F'(I)\| < 1 \tag{4.3}$$

をチェックする. もし成立すれば,I に唯一解が存在する.2 へ. そうでなければ次へ.

5. $K(I) \cap I = \emptyset$ ならば I に解は存在しない.2 へ. そうでなければ次へ.

6. I を I_1 と I_2 に分割し,$List$ の末尾に追加する.2 へ.

□

4.3 なぜ Affine Arithmetic を利用するか？

先程, 第 3.2 節で説明したのですが, 区間演算においては区間幅が広がってしまうと言う弱点がある. それ故, 第 4.2 節での非存在テストにおける $F(I)$ の計算において, 区間幅が広がってしまっているので, その無駄な領域に 0 が含まれてしまう, つまりこの非存在テストに引っかかる確率が低い. しかし, 図 4.1 のように領域を絞り込んだ Affine Arithmetic を利用したテストでは, 確率が上がり, 早い段階での領域除去ができることが予想できる.
これから, この Affine Arithmetic を利用した非存在テストについて論じて行く.

以下, 方程式 $f(x) = 0, f : R^n \to R^n$ と与えられた区間 I に対する非存在テストを考える. 区間 I における Affine Arithmetic による f の評価が

$$f_1 = a_{10} + a_{11}\varepsilon_1 + \cdots + a_{1m}\varepsilon_m$$

$$\vdots \tag{4.4}$$

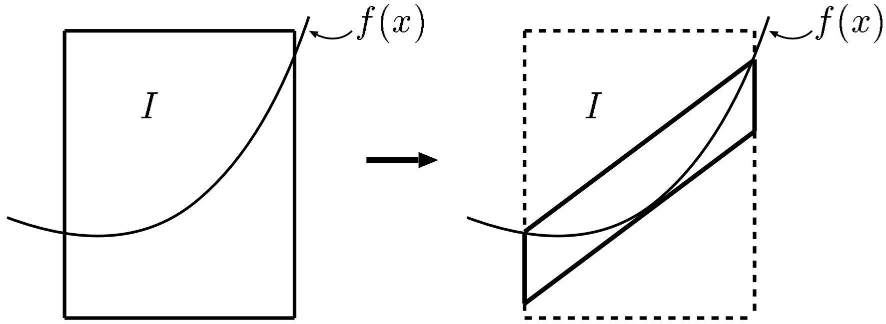

図 4.1: 領域の絞り込み

$$f_n = a_{n0} + a_{n1}\varepsilon_1 + \cdots + a_{nm}\varepsilon_m$$

$$(n \leq m)$$

のように得られたとする.

4.4 絶対値の和を利用する非存在テスト

$$R = \sum_{i=1}^{n} |f_i| \tag{4.5}$$

を Affine Arithmetic で評価し,$R > 0$ なら解は存在しない. その手順として,

- 区間包囲 $f(I)$ を求める.
- $|f_i|(i = 1, 2, \cdots, n)$ を Affine Arithmetic で表す.
- それの和を求め,R とする.
- $R > 0 \Rightarrow$, 非存在.

$|f(I)|$ は絶対値関数の単項演算であり, それを Affine 形式にしてその和が 0 より大きくなるならば,$0 \notin f_i$ となる i が少なくとも 1 つ存在する. 故に,I に解が存在しないのは明らかである.

4.5　一般逆行列を用いた非存在テスト

線形方程式

$$
\begin{array}{c}
a_{10} + a_{11}\varepsilon_1 + \cdots + a_{1m}\varepsilon_m = 0 \\
\vdots \\
a_{n0} + a_{n1}\varepsilon_1 + \cdots + a_{nm}\varepsilon_m = 0
\end{array}
\tag{4.6}
$$

の解集合を考える. もしこの解集合と超立方体

$$-1 \leq \varepsilon_i \leq 1 (i = 1, \cdots, m) \tag{4.7}$$

との交わりが存在しなければ, 解は存在しない. この交わりが空集合か否かは, 超立方体, 式 (4.7) の代わりに, それを包み込むような半径 $\sqrt{m}$ の超球を考える. もし式 (4.6) の最小ノルム解が得られてそのノルムが $\sqrt{m}$ より大きいならば, 式 (4.6) と式 (4.7) の交わりは空であり,$f(x) = 0$ の解が存在しないことが分かる. 式 (4.6) の最小ノルム解は一般逆行列を用いることにより計算できる. 行列 A とベクトル b を

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix} \tag{4.8}$$

$$b = -\begin{pmatrix} a_{10} \\ \vdots \\ a_{n0} \end{pmatrix} \tag{4.9}$$

とすると,$Ax = b$ の最小ノルム解は

$$A^t(AA^t)^{-1}b \tag{4.10}$$

で与えられる.

4.6 Affine Arithmeticを利用した全解探索

以下では,**アルゴリズム 4.2** で解説した全解探索アルゴリズムを Affine Arithmeticを利用したものに変えたものである.

アルゴリズム 4.6 (Affine Arithmetic を利用した全解探索アルゴリズム)

$f : R^n \to R^n, I_0 \in IR^n, F$ を f の区間包囲,F' を f の区間包囲とし, 方程式 $f(x) = 0$ の I_0 内の全ての解を求めることを考える.

1. 区間のリスト $List$ を $List = \{I_0\}$ で初期化する.
2. $List$ が空なら終了. そうでなければ L から先頭の要素を取り出し I とする.
3. 「絶対値の和を利用する非存在テスト」or「一般逆行列を用いた非存在テスト」を行なう. 存在しなければ 2 へ. そうでなければ次へ.
4. c を I の中心,$L \simeq f'(c)$ を正則な行列とし,

$$K(I) = c - L^{-1}f(c) + (E - L^{-1}F'(I))(I - c) \tag{4.11}$$

を計算し,

$$K(I) \subset I \tag{4.12}$$

$$\|E - L^{-1}F'(I)\| < 1 \tag{4.13}$$

をチェックする. もし成立すれば,I に唯一解が存在する.2 へ. そうでなければ次へ.

5. $K(I) \cap I = \emptyset$ ならば I に解は存在しない.2 へ. そうでなければ次へ.
6. I を I_1 と I_2 に分割し,$List$ の末尾に追加する.2 へ.

□

4.7 むすび

本章では,Affine Arithmetic を利用した非存在テストを 2 つ挙げて見た. 本来, 文献 [2] の中で提案されているのは 4 つあるのだが, 今回私は, その中の 2 つの方法についてのアルゴリズムを考え, プログラミングしている. 次章では, 実際に本章で挙げた 2 つの方法と従来の方法の実行結果を使って, 比較して行きたい.

第 5 章

数値例

5.1 はじめに

まず, 以下の 2 つの式を用いる.

$$\begin{cases} g(x_1) + x_1 + x_2 + \cdots + x_n = 1 \\ g(x_2) + x_1 + x_2 + \cdots + x_n = 2 \\ \qquad\qquad \vdots \\ g(x_n) + x_1 + x_2 + \cdots + x_n = n \end{cases} \tag{5.1}$$

$$\begin{aligned} I_1 : g(x_i) &= 10.5x_i^2 + 11.8x_i \\ I_2 : g(x_i) &= 10.5x_i^2 - 11.8x_i \end{aligned} \tag{5.2}$$

この 2 つの式における全解探索を行い, 区間リストの分割数, 実行時間を出力する. その結果から考察を考えていく.

なお, 私は以下の条件下で実験を行った.

- CPU : PentiumII 350Mhz
- メモリ : 128MB
- OS : FreeBSD 2.2.8
- gcc : version 2.7.2.1
- マウス : logitech(スーパーホイール付き)

5.2 数値例 1

以下の表 5.1, 表 5.2, 表 5.3 は, 式 (5.2) の I_1 についての出力結果である.

n	全解探索法の種類	与える区間	区間リスト分割数	実行時間 (s)	解
2	1. 従来の全解探索法	$[-4,5]$	157	0.117	4
	2. Affine の絶対値の和の方法	$[-4,5]$	83	0.141	
	3. 一般逆行列を用いた方法	$[-4,5]$	151	0.32	
3	1. 従来の全解探索法	$[-4,5]$	1011	1.54	8
	2. Affine の絶対値の和の方法	$[-4,5]$	359	1.25	
	3. 一般逆行列を用いた方法	$[-4,5]$	1095	5	
4	1. 従来の全解探索法	$[-4,5]$	3953	11	16
	2. Affine の絶対値の和の方法	$[-4,5]$	1813	10	
	3. 一般逆行列を用いた方法	$[-4,5]$	8079	72.6	
5	1. 従来の全解探索法	$[-4,5]$	19461	90	32
	2. Affine の絶対値の和の方法	$[-4,5]$	5203	48	
	3. 一般逆行列を用いた方法	$[-4,5]$	メモリオーバー	?	

表 5.1: I_1 の出力結果 1

n	全解探索法の種類	与える区間	区間リスト分割数	実行時間 (s)	解
6	1. 従来の全解探索法	$[-2, 0.1]$	3627	30	1
	2. Affine の絶対値の和の方法	$[-2, 0.1]$	77	1	
	3. 一般逆行列を用いた方法	$[-2, 0.1]$	375	10	
7	1. 従来の全解探索法	$[-2, 0.1]$	4991	54	1
	2. Affine の絶対値の和の方法	$[-2, 0.1]$	115	2	
	3. 一般逆行列を用いた方法	$[-2, 0.1]$	739	27	
8	1. 従来の全解探索法	$[-2, 0.5]$	7313	116	1
	2. Affine の絶対値の和の方法	$[-2, 0.5]$	71	1	
	3. 一般逆行列を用いた方法	$[-2, 0.5]$	2065	113	

表 5.2: I_1 の出力結果 2

n	全解探索法の種類	与える区間	区間リスト分割数	実行時間 (s)	解
6	1. 従来の全解探索法	$[-0.1, 2]$	183	1	1
	2. Affine の絶対値の和の方法	$[-0.1, 2]$	855	13	
	3. 一般逆行列を用いた方法	$[-0.1, 2]$	メモリオーバー	?	
7	1. 従来の全解探索法	$[-0.1, 2]$	271	3	1
	2. Affine の絶対値の和の方法	$[-0.1, 2]$	2241	51	
	3. 一般逆行列を用いた方法	$[-0.1, 2]$	メモリオーバー	?	
8	1. 従来の全解探索法	$[-0.1, 1]$	851	15	1
	2. Affine の絶対値の和の方法	$[-0.1, 1]$	メモリオーバー	?	
	3. 一般逆行列を用いた方法	$[-0.1, 1]$	メモリオーバー	?	

表 5.3: I_1 の出力結果 3

5.3　数值例 2

以下の表 5.4, 表 5.5, 表 5.6 は, 式 (5.2) の I_2 についての出力結果である.

n	全解探索法の種類	与える区間	区間リスト分割数	実行時間 (s)	解
2	1. 従来の全解探索法	$[-2, 3]$	293	0.211	4
	2. Affine の絶対値の和の方法	$[-2, 3]$	133	0.227	
	3. 一般逆行列を用いた方法	$[-2, 3]$	259	0.539	
3	1. 従来の全解探索法	$[-2, 3]$	2035	3	8
	2. Affine の絶対値の和の方法	$[-2, 3]$	591	2	
	3. 一般逆行列を用いた方法	$[-2, 3]$	1571	8	
4	1. 従来の全解探索法	$[-2, 3]$	26927	86	16
	2. Affine の絶対値の和の方法	$[-2, 3]$	3119	21	
	3. 一般逆行列を用いた方法	$[-2, 3]$	13359	137	

表 5.4: I_2 の出力結果 1

n	全解探索法の種類	与える区間	区間リスト分割数	実行時間 (s)	解
5	1. 従来の全解探索法	$[-0.5, 1]$	1311	6	1
	2. Affine の絶対値の和の方法	$[-0.5, 1]$	113	1	
	3. 一般逆行列を用いた方法	$[-0.5, 1]$	429	7	
6	1. 従来の全解探索法	$[-0.5, 1]$	3397	28	1
	2. Affine の絶対値の和の方法	$[-0.5, 1]$	153	2	
	3. 一般逆行列を用いた方法	$[-0.5, 1]$	307	8	
7	1. 従来の全解探索法	$[-0.5, 1]$	8689	108	1
	2. Affine の絶対値の和の方法	$[-0.5, 1]$	627	14	
	3. 一般逆行列を用いた方法	$[-0.5, 1]$	メモリオーバー	?	

表 5.5: I_2 の出力結果 2

n	全解探索法の種類	与える区間	区間リスト分割数	実行時間 (s)	解
5	1. 従来の全解探索法	$[-1, -0.1]$	13	0.0625	1
	2. Affine の絶対値の和の方法	$[-1, -0.1]$	21	0.195	
	3. 一般逆行列を用いた方法	$[-1, -0.1]$	343	6	
6	1. 従来の全解探索法	$[-1, -0.1]$	13	0.102	1
	2. Affine の絶対値の和の方法	$[-1, -0.1]$	37	0.516	
	3. 一般逆行列を用いた方法	$[-1, -0.1]$	1459	41	
7	1. 従来の全解探索法	$[-1, -0.1]$	11	0	1
	2. Affine の絶対値の和の方法	$[-1, -0.1]$	75	1	
	3. 一般逆行列を用いた方法	$[-1, -0.1]$	3635	156	

表 5.6: I_2 の出力結果 3

5.4 考察

数値例からの気付いた点

1. Affine の絶対値の和を利用した方法が一番効率が良い.

2. 一般逆行列を利用した方法は一番効率が悪い.

3. 減算を行う非線形方程式の場合は, 絶対値, 一般逆行列を用いた方法が良い.

4. 減算を行わない場合は, 従来の方法が良い.

まず 1 についてだが, 表 5.1, 表 5.4 は全部の解を出すように最初の区間を与えた時の結果だが, これを見た限りでは区間リストの分割数, プログラム実行時間ともに一番少ない数値が得られている. また, これは n の数が増えるほど, すなわち, 高次変数方程式になればなるほど顕著に見られる.

次に 2 についてだが, これは表 5.1, 表 5.3 を見ると分かるがとにかくメモリ不足になる恐れがある. その上, とにかくプログラムの実行時間が掛かり過ぎである. また, 区間リストの分割数から Affine Arithmetic の相関性による成果が見られない事が分かる.

この 1,2 の理由としては, プログラムの複雑さという事が挙げられる. 一般逆行列を用いた場合の方が Affine の絶対値の和を利用した場合よりも複雑で, 何行にも渡っているプログラムである. 故に, 実行時間は掛かる上にメモリも食うのである. 逆に, 絶対値の和を利用した方法では従来の方法よりもプログラムは複雑であるがそれほどの量でもないので,Affine Arithmetic の相関性により区間リストの分割数, 実行時間ともに早くなっている. また一般逆行列の方法において,Affine Arithmetic の相関性による成果が見られない理由としては, 第 4.5 節で説明した式 4.7 の超立方体を含むような超球を考えた事が挙げられる. 図 5.1 の影の部分の分だけ正確でなく, それが非存在領域除去に影響しているのである.

次に 3,4 についてだが, 表 5.2 と表 5.3 を比べると分かるように, 表 5.2 の方は絶対値の和, 一般逆行列の方法が非常に良い結果を出していて, 表 5.3 の方は従来の方法が良い結果を出している. これは, 表 5.2 の方は, 最初に与える区間がほぼマイナス, つまり ,$10.5x_i^2 + 11.8x_i$

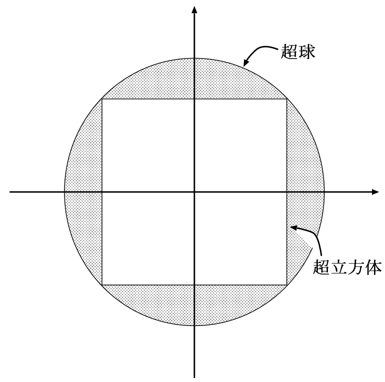

図 5.1: 超立方体を含む超球

は減算をやっている事になる. 故に, 第 3 章で述べたように,Affine Arithmetic の相関性が最も発揮されるのは減算をする時であるので, 表 5.2 のような結果が得られたと考えられる. 逆に表 5.3 の方は, 最初に与える区間がほぼプラス, つまり,$10.5x_i^2 + 11.8x_i$ は減算をやっていない事になる. 故に,Affine Arithmetic の相関性は発揮されないのである. このことは, 表 5.5 と表 5.6 でも同じ事がいえる. また他の例題でも同じ結果が出たので, この事は間違いはないであろう. ここで言えることは, 与える非線形方程式が Affine Arithmetic の相関性が利用できるようなときは, 当然絶対値の和を使った方法が有効的なのである. ここで,Affine Arithmetic の相関性が最も発揮する方程式というのは, 減算と傾きが似たような線形連立方程式などが挙げられる.

以上の事から, 次のような全解探索アルゴリズムが実現できると, 非常に効率が良いのではないか？

アルゴリズム 5.4 (新しい全解探索アルゴリズム)

$f : R^n \to R^n, I_0 \in IR^n, F$ を f の区間包囲,F' を f の区間包囲とし, 方程式 $f(x) = 0$ の I_0 内の全ての解を求めることを考える.

1. 非線形方程式が Affine Arithmetic の相関性が全く発揮しそうもない場合 4 を, それ以外は 5 を行う.

2. 区間のリスト $List$ を $List = \{I_0\}$ で初期化する.

3. $List$ が空なら終了. そうでなければ L から先頭の要素を取り出し I とする.

4. $F(I)$ を計算し,$0 \notin F(I)$ ならば I に存在しない.3 へ. そうでなければ 6 へ.

5. 絶対値の和を利用する非存在テストを行ない, 存在しなければ 3 へ. そうでなければ 6 へ.

6. c を I の中心,$L \simeq f'(c)$ を正則な行列とし,

$$K(I) = c - L^{-1}f(c) + (E - L^{-1}F'(I))(I - c) \tag{5.3}$$

を計算し,

$$K(I) \subset I \tag{5.4}$$

$$\|E - L^{-1}F'(I)\| < 1 \tag{5.5}$$

をチェックする. もし成立すれば,I に唯一解が存在する.3 へ. そうでなければ次へ.

7. $K(I) \cap I = \emptyset$ ならば I に解は存在しない.3 へ. そうでなければ次へ.

8. I を I_1 と I_2 に分割し,$List$ の末尾に追加する.3 へ.

□

ここで, このアルゴリズムを実現するにあたってはいろいろと問題が生じてくる. 与えられる区間によって減算か否かが決まるので, それをどのようにして場合分けをするかという点,sin, cos などの関数が入ってくると減算か否かを判定するのは難しくなるなどである. また, 傾きが似た線形連立方程式というのは,2 次元方程式だと分かりやすいのだが, 高次元方程式になるとどのようにして傾きを考えればいいのか検討もつかない. 故に, このアルゴリズムの実現については, 時間もなかったので, 今後の課題としておく.

5.5　むすび

絶対値の和を利用する全解探索ではかなり良い結果が出た. あとは, 課題として, メモリの解放がある. それを解決できれば, もっと複雑で, 高次元の非線形方程式を解く事ができるだろう. また, 先程も述べたが, 文献 [2] の中で Affine Arithmetic を利用した非存在テストは 4 つ提案されている. 本研究では, その内の 2 つの方法しか実現していない. 残り 2 つの方法は今後の課題としておく.

謝 辞

本論文の制作, 中間報告などにおいて非線形数値計算の分野にとどまらず, 度重なる御指導, 御鞭撻を賜わりました. 特に本論文の重要な要素である Affine Arithmetic については, 専門家ならではの非常に適切な御意見を頂きました, 柏木雅英 助教授に心より感謝致します.

また, 卒業論文中間報告の際など機会のあるごとに御指導, 御鞭撻を賜り, 研究に方向性を与えて下さった大石進一教授に深く感謝致します.

情報学科の先生方には, 講義にて 1 年次から熱心に御指導賜り, また, 勉学の面だけならずその他の面でも貴重な意見を頂き, 心より感謝致します.

そして何より, 休憩時間に私を楽しませてくれた, ちばてつや 氏には本当に感謝申し上げます.

柏木研において柏木研助手 相馬隆郎 氏 , ならびに博士課程 2 年 宮田孝富 氏には, プログラムの Gauss 消去法, 自動微分など非線形の分野について熱心に御指導してくださいました. また, 日頃の柏木研究室内でのあらゆる面において御教示頂いたことに, 心より感謝申し上げます.

柏木研修士課程 1 年高崎大輔 氏には, 中間発表の準備の際に適切なアドバイスを頂き, 心より感謝申し上げます.

柏木研修士課程 1 年岩折朱希嗣 氏には, 私のコンピュータに関する稚拙な疑問にも親身に御指導を賜り, 心より感謝申し上げます.

最後に, 非線形班と CG 班の隔てなく, 楽しい時間をともに過ごした柏木研究室学部 4 年の皆様, 金谷 卓充氏, 小泉 健氏, 櫻井幹夫氏, 白井 健一氏, 洲濱陽一氏, 長友 泰崇氏, 波多野伸哉氏, 深谷 光統氏, 村竹 範彦氏, 山田 浩之氏, 吉田 直史氏, 渡部 啓氏 に心より感謝致します.

参考文献

[1] 大石進一著,“応用解析セミナー数値計算”, 裳華房,1999

[2] 柏木雅英著,“Affine Arithmetic とその応用”, 平成 11 年度日本応用数理学会年会

[3] 柏木雅英著,“精度保証つきシミュレーション [1] -区間解析-”, 日本シミュレーション学会 “シミュレーション第 18 巻第 4 号 (平成 11 年 12 月)”

[4] 紫村仁史著,“非線形方程式の解の非存在性検証”, 平成 10 年度卒業論文 (大石研)