

平成 12 年度

学士学位論文

適応アルゴリズムに適した共役勾配法

The conjugation slope method for having been
suitable for the Adaptive Algorithm

1010401 瀬尾光代

指導教員 福本昌弘

2001 年 2 月 5 日

高知工科大学 情報システム工学科

要 旨

適応アルゴリズムに適した共役勾配法

瀬尾光代

適応アルゴリズムの中にブロック直交射影アルゴリズムがある。ブロック直交射影アルゴリズムを実現する一方法として共役勾配法利用する方法がある。実際に適応アルゴリズムを利用するときには、雑音が存在する。雑音が存在することにより収束速度と収束精度を両立させることが困難になる。そのため、何種類かある共役勾配法から適応アルゴリズムで利用するのに適したものを選び出す。また、共役勾配法を利用する場合、繰返し回数が収束値を決める主な要因になる。このことから、適した繰返し回数を決めることが重要になる。

キーワード 適応アルゴリズム, 共役勾配法

Abstract

The conjugation slope method for having been suitable for
the Adaptive Algorithm

Mitsuyo SEO

A block orthogonality projection algorithm is in an Adaptive Algorithm. On the one hand, it uses as law, and there is a method conjugation-slope-method-used of realizing a block orthogonality projection algorithm. When actually using an Adaptive Algorithm, noise exists. When noise exists, it becomes difficult to reconcile convergence speed and convergence accuracy. Therefore, the thing suitable for using by the Adaptive Algorithm is selected out of several kinds or a certain conjugation slope method. Moreover, when using a conjugation slope method, a repeat count becomes the main factors which determine convergence value. It becomes important to decide from this the repeat count which was suitable.

key words Adaptive Algorithm, Conjugate-Slope- Method

目次

第 1 章	序論	1
1.1	背景と目的	1
1.2	本論文の概要	1
第 2 章	適応アルゴリズムと共役勾配法	2
2.1	適応アルゴリズムとは	2
2.1.1	FIR デジタルフィルタ	2
2.1.2	適応アルゴリズムに求められる課題	3
2.1.3	適応アルゴリズムの歴史と種類	4
2.2	学習同定法について	4
2.2.1	学習同定法の問題点	7
2.2.2	アフィン射影算法	7
2.3	ブロック適応アルゴリズムとは	8
2.3.1	ブロック直交射影アルゴリズム	9
2.3.2	アフィン射影算法との共通点	9
2.3.3	アフィン射影算法との相違点	9
2.3.4	共役勾配法の適用	10
2.4	共役勾配法	10
2.4.1	共役勾配法の種類	10
2.4.2	HS 版計算手順	10
2.4.3	高橋版の計算手順	13
2.4.4	2 階版の計算手順	15
2.4.5	RG 版の計算手順	19
2.4.6	単調版の計算手順	22

目次

第 3 章		29
3.1	まえがき	29
3.2	共役勾配法の選択	29
3.3	繰返し回数	30
3.4	HS 版シミュレーション結果	30
第 4 章	結論	31
4.1	結論	31
4.2	今後の課題	31
謝辞		32
参考文献		33

第 1 章

序論

1.1 背景と目的

従来の適応アルゴリズムは入力信号が有色性の信号であるとき、収束速度が著しく劣化する。1 回の係数修正に複数の過去の状態ベクトルを利用することで克服する手法がある。しかし、1 サンプルあたりの演算量が大幅に増大するため、実時間処理を必要とするシステムへの適用が困難である。この 2 点を改善し、アルゴリズムを高速化するために、ブロック処理の概念を用いたブロック直交射影アルゴリズムが提案された。その一実現法として共役勾配法を適用した BOP アルゴリズムがある。

実際、適応アルゴリズムで共役勾配法を使うとき雑音が存在する。共役勾配法を適応アルゴリズムで利用するとき、雑音が存在することにより、収束速度と収束精度を両立することが困難になる。このことから、雑音が存在しても安定性の高い共役勾配法を使う必要がある。また、繰返し回数が収束値を決める主要因であることから、適した繰返し回数を設定する。

1.2 本論文の概要

第 2 章では本研究を行う上で必要な適応アルゴリズムと共役勾配法についての説明をおこなう。第 3 章では安定性の高い共役勾配法を選び出すためのシミュレーションについて説明をする。

第 2 章

適応アルゴリズムと共役勾配法

2.1 適応アルゴリズムとは

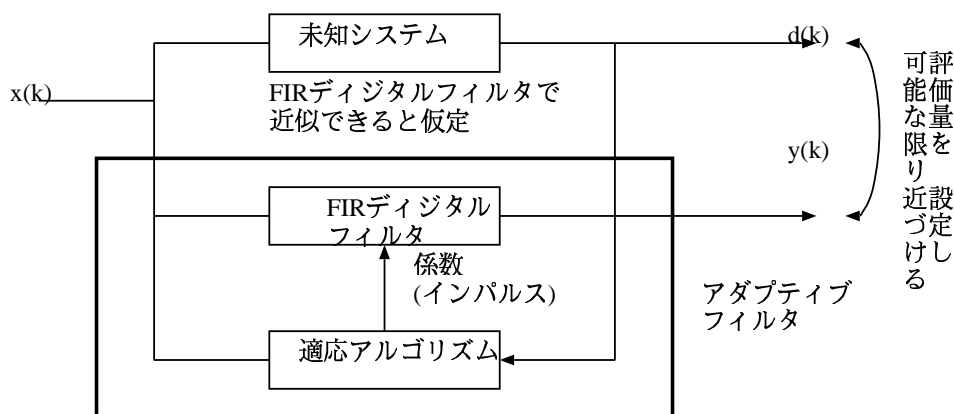


図 2.1 適応アルゴリズム

2.1.1 FIR デジタルフィルタ

離散的な入力信号に対して、アナログフィルタの周波数特性と同等なフィルタリングを行うように設計されたフィルタのことをデジタルフィルタという。

- フィルタのインパルス応答が有限項で表現される
- 常に安定性を満たす
- 完全な線形位相をもつ特性がある

2.1 適応アルゴリズムとは

FIR デジタルフィルタの入出力関係は次式のように表せる.

$$y = \sum_{i=0}^{N-1} h(i)x(k-i) \quad (2.1)$$

ただし,

$x(k)$: 時刻 kT (T はサンプリング周期) におけるフィルタの入力信号

$y(k)$: 時刻 kT (T はサンプリング周期) におけるフィルタの出力信号

N : フィルタ係数の個数

インパルス応答 $w(i)$ ($i = 0, 1, 2, \dots$) を有する時不変線形システムの入出力信号は, 次のような畳み込み演算を満足させる.

$$d(k) = \sum_{i=0}^{\infty} w(i)x(k-i) \quad (2.2)$$

このような、システムを未知システムとする. $(N-1)$ FIR デジタルフィルタの係数は, 入力信号 $x(k)$, フィルタの出力信号 $y(k)$ および未知システムの出力信号 $d(k)$ を使って修正する. この係数修正アルゴリズムを適応アルゴリズムという.

2.1.2 適応アルゴリズムに求められる課題

- 収束速度の高速化
- 実行速度の高速化
- ハードウェアの小規模化

2.2 学習同定法について

また、動作の安定性についても考慮する必要がある。

2.1.3 適応アルゴリズムの歴史と種類

1960年, Widora と whoff は適応スイッチング回路の研究において, LMS アルゴリズムと呼ばれる適応アルゴリズムを開発した. LMS アルゴリズムは, 広い意味で, 2乗平均誤差を最急降下法に基づいて最小にする一方式である. 演算量が少ないという理由で代表的なアルゴリズムとして利用されている.

1967年これとは独立に, 野田と南雲は学習同定法を発表した. 学習同定法は LMS アルゴリズムに比べやや複雑であるが, 高速な収束特性を有しており, 実用的にも優れた適応アルゴリズムといえることができる.

これらのアルゴリズムは, 与えられた信号の統計的性質が未知 (あるいはほとんど未知) の場合でも, この信号の統計量をもとに生成される Wiener-Hoff の方程式を解くことのできる繰り返し算法と見ることができる. また, 推定すべきパラメータが時間とともに比較的緩やかに変動しても, パラメータの変化にある程度追従できる特徴がある. 実際の応用では, このような状況はむしろ一般的と言えるので, この特徴は重要である. しかし, これらのアルゴリズムは入力信号が有色の場合, 収束速度 (特に, 推定すべきパラメータへの収束速度) が著しく劣化する欠点のあることが指摘されている.

この他にも, RLS アルゴリズム [1], BLMS アルゴリズム [1], 跳躍アルゴリズム [1] がある.

2.2 学習同定法について

学習同定法は 1967年野田, 南雲らによって発表された.

- 直交射影定理に基づく適応アルゴリズム

2.2 学習同定法について

- LMS アルゴリズムに比べてやや複雑
- 高速な収束性を有する
- 別名 Normalized LMS アルゴリズム
 - LMS アルゴリズムの係数修正項をフィルタの状態ベクトルノルムで正規化した形

アダプティブフィルタ：適応アルゴリズムを含む FIR デジタルフィルタ

(注意)

- 未知システムと既知システムの次数は等しい ($M = N$)
- 観測雑音 $v(t)$ は存在しない ($v(t) = 0$)

時刻 k でアダプティブフィルタの出力 $y(k)$ が未知システムの出力 $d(k)$ に等しい

$$d(k) = (x(k), x(k-1), \dots, x(k-N+1)) \begin{pmatrix} h(0) \\ h(1) \\ \vdots \\ h(N-1) \end{pmatrix} \quad (2.3)$$

$d(k)$: 未知システムにおける出力信号

$x(k)$: フィルタの入力信号

$h(k)$: k 番目のフィルタ係数

W_N : 未知システムのパラメータ

h_N : アダプティブフィルタのパラメータ

2.2 学習同定法について

式 (2.3) を満足する h_N は真値ベクトルを含むいわゆる解集合となる. 式 (2.3) を満足する h_N の代表ベクトルを $h_N(k+1)$ とし, そのベクトルを適当に定めた任意の点から解集合に下ろした垂線の足と考える. 解集合は式 (2.3) からわかるように状態ベクトル $x_N(k)$ に直交しているさらに, W_N はこの解集合に含まれるので, $h_N(k+1)$ はある点から $x_N(k)$ 方向に係数を修正したとき最も W_N に近い点である. このようなことを繰り返して $h_N(k+1)$ を W_N に接近させるためには, 適当に定めた点よりも W_N により近い $h_N(k)$ を次の係数修正の初期値とすればよい.

これらのことをまとめると

$$\begin{aligned}
 h_N(k+1) &= h_N(k) + \{h_N(k+1) - h_N(k)\} \\
 &= h_N(k) + \underbrace{\frac{\{W_N - h_N(k)^T\} \{h_N(k+1) - h_N(k)\}}{\|h_N(k+1) - h_N(k)\|}}_{\text{修正量}} \\
 &\quad \times \underbrace{\frac{h_N(k+1) - h_N(k)}{\|h_N(k+1) - h_N(k)\|}}_{\text{修正方向}}
 \end{aligned} \tag{2.4}$$

となる. ここで,

$$\frac{h_N(k+1) - h_N(k)}{\|h_N(k+1) - h_N(k)\|} = \frac{x_N(k)}{\|x_N(k)\|} \tag{2.5}$$

$$\{W_N - h_N(k)\}^T x_N(k) = d(k) - y(k) = e(k) \tag{2.6}$$

が成立するので,

$$h_N(k+1) = h_N(k) + \frac{x_N(k)}{\|x_N(k)\|^2} e(k) \tag{2.7}$$

のように変形できる. 学習同定法は上式の修正ベクトルにステップゲインを掛け

$$h_N(k+1) = h_N(k) + \alpha \frac{x_N(k)}{\|x_N(k)\|^2} e(k) \tag{2.8}$$

2.2 学習同定法について

で与えられる.

2.2.1 学習同定法の問題点

学習同定法による係数修正は

$$h_N(k+1) = h_N(k) + \alpha \frac{x_N(k)}{\|x_N(k)\|^2} e(k) \quad (2.9)$$

で与えられる. ただし, $x_N(k), e(k)$ はそれぞれ,

$$x_N(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T \quad (2.10)$$

$$e(k) = d(k) - x_N(k)^T h_N(k) \quad (2.11)$$

である. ここで, α はステップゲインと呼ばれる量である. このように学習同定法では, 1 本の入力信号ベクトルを用いて係数を修正するため, 有色入力信号に対して著しく収束速度が劣化することが知られている. 学習同定法の問題点を解決するために次のような解決策が提案された.

2.2.2 アフィン射影算法

実際の通信系や記録系で使用される多くの信号が有色性であることを考慮すると, この種の信号に対する収束速度の高速化は当然の要求といえる. そこで, 雛元らと尾関らはこのような要求を満足する方式として'拡張された学習同定法', 'アフィンの射影算法'をそれぞれ独立に提案した. これらの方式は, フィルタ係数を修正するために, ある時刻 k における入力信号ベクトル $x_N(k)$ のみならず, 適当な数の過去の入力信号ベクトルをも加えた, 複数個の入力ベクトルを用いている. これらの係数修正手順は,

2.3 ブロック適応アルゴリズムとは

$$h_N(k+1) = h_N(k) + A_{r,N}(k)^+ \cdot e_r(k) \quad (2.12)$$

$$\begin{aligned} e_r(k) &= d_r(k) - A_{r,N}(k) \cdot h_N(k) \\ &= A_{r,N}(k) [W_N - h_N(k)] \end{aligned} \quad (2.13)$$

で与えられる. ここで, $A_{r,N}(k)^+$ は $A_{r,N}(k)$ の Moore-Penrose 型一般逆行列であり, $A_{r,N}$ は

$$A_{r,N}(k) = [x_N(k-r+1), x_N(k-r+2), \dots, x_N(k)]^T \quad (2.14)$$

で定義される.

式より

$$h_N(k+1) - h_N(k) = A_{r,N}(k)^+ A_{r,N}(k) [W_N - h_N(k)] \quad (2.15)$$

が成り立つ. 式の右辺中の行列 $[A_{r,N}^+ A_{r,N}]$ は $A_{r,N}(k)$ の行ベクトルが張る部分空間 $S[A_{r,N}(k)^T]$ へ直行射影行列である. したがって式は $h_N(k+1)$ が W_N を $S[A_{r,N}(k)^T]$ への直行射影することによって得られる点であることを意味している.

このように, 係数修正を行うのに複数個の入力信号ベクトルを用いることにより, 有色信号入力時にも収束速度の高速化が可能になる. しかしながら, その代償としてサンプル当たりの演算量の増加を招き, ハードウェア構成の実現性に困難を生じる.

適応アルゴリズムには少ない演算量で高速な収束速度および処理速度が要求される. したがって, これらを満足するようなアルゴリズムの開発が必要である. それに対する一つの回答を与えてくれるのがブロック適応アルゴリズムである.

2.3 ブロック適応アルゴリズムとは

ブロック処理はフィルタリングを効率的に行う方式として, Burrus らによって提案され, この概念を適応信号処理に導入したブロック適応アルゴリズムが Clark らによって始められ

2.3 ブロック適応アルゴリズムとは

た. ブロック適応アルゴリズムは, 入力信号と所望出力信号を有限個ずつブロック化し, そのデータブロックごとに1回だけ係数修正を行う.

2.3.1 ブロック直交射影アルゴリズム

演算量の軽減を目的としてアフィン射影算法にブロック処理を導入したものが, ブロック直交射影アルゴリズムである.

2.3.2 アフィン射影算法との共通点

アフィン射影算法とブロック直交射影アルゴリズムは, 1回の係数修正に複数の入力信号ベクトルを用い, それぞれのベクトルが張る部分空間への直交射影演算に基礎を置いている. これは, 基本的に入力状態行列 (一般的にランク落ちしている場合が多い) を係数行列とする連立1次方程式の解の中で, 最小ノルム解を求めることに帰着する.

2.3.3 アフィン射影算法との相違点

アフィン射影算法では, 現サンプル時刻と次のサンプル時刻での処理の対象となるそれぞれの入力状態行列は, $(r - 1)$ 個の入力信号ベクトルを共に有している. 一方, BOP アルゴリズムの場合は連続するブロックにおける入力状態行列は, 共有している信号ベクトルを持つことはない.

2.4 共役勾配法

2.3.4 共役勾配法の適用

直交射影演算に基づくアルゴリズムは基本的に、非正則なデータ行列を有する連立方程式を解くことに帰着し、Moore-Penrose 型一般逆行列により表現される。係数行列が対象行列であるような方程式の一解法に共役勾配法がある。

2.4 共役勾配法

1952年 M.R.Hestenes と E.Stiefel によって発表された。大次元の問題や特殊な分野において利用されていた。1960年ごろから非線形最適化の問題の解法としてよりようされるようになる。反復法でありながら、有限回のステップで厳密解に到達するという性質を持っている。問題点として、少ない繰返し回数で答えが得られるが、多くの回数を繰り返しても十分な精度が得られないことがあげられる。

2.4.1 共役勾配法の種類

- HS 版
- 高橋版
- 2 階版
- RG 版
- 単調版

2.4.2 HS 版計算手順

Hestenes と Stiefel によって発表された、最も基本的なアルゴリズムである。他のアルゴリズムと区別するために、発表者の頭文字をとって、共役勾配法の HS 版とよぶ。この方法は、連立 1 次方程式

2.4 共役勾配法

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \quad (2.16)$$

を解くための公式で, 係数

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & & & \vdots \\ & & & a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

が対称行列の場合に適用できる. 理論的には行列ベクトルで表現する方が扱い易い. そこで, 連立 1 次方程式の係数行列, 定数項および解ベクトルを

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & & & \vdots \\ & & & a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

とすれば, 方程式は

$$Ax = b$$

と表すことができる. 次に, 第 k 近似解, 第 k 回の修正方向ベクトル, 第 k 回の残差ベクトルを,

$$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

2.4 共役勾配法

とすれば、計算手順は次のようになる

1. 第 0 近似解 (x_0) を適当に選ぶ
2. 第 0 近似解に対する残差を計算する

$$r_0 = b - Ax_0 \quad (2.17)$$

3. 補助変数 (p_0) の初期値を設定する

$$p_0 = r_0 \quad (2.18)$$

$$k = 0$$

4. 第 k 回の修正係数 α_k を

$$\alpha_k = \frac{(r_k, p_k)}{(p_k, Ap_k)} \quad (2.19)$$

によって求める

5. 第 $k + 1$ 近似値を

$$x_{k+1} = x_k + \alpha_k p_k \quad (2.20)$$

によって求める

6. 第 $k + 1$ 近似値に対する残差を

$$r_{k+1} = r_k - \alpha_k Ap_k \quad (2.21)$$

という形で計算する

7. β を次のように計算する

$$\beta = \frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)} \quad (2.22)$$

2.4 共役勾配法

8. 補助変数の新しい値を

$$p_{k+1} = r_{k+1} + \beta_k p_k \quad (2.23)$$

で計算する

9. 収束判定をし, まだ収束が十分でなければ手順4に戻る

2.4.3 高橋版の計算手順

HS 版の式 (2.23), すなわち

$$p_{k+1} = r_{k+1} + \frac{|r_{k+1}|^2}{|r_k|^2} p_k \quad (2.24)$$

の両辺を $|r_{k+1}|^2$ で割ると

$$\frac{p_{k+1}}{|r_{k+1}|^2} = \frac{r_{k+1}}{|r_{k+1}|^2} + \frac{p_k}{|r_k|^2} \quad (2.25)$$

という, きれいな形になる. そこで変数 p_k のかわりに

$$q_k = \frac{p_k}{|r_k|^2} \quad (2.26)$$

を用いることにすると, 上記に相当する計算式は

$$q_{k+1} = q_k + \frac{r_{k+1}}{|r_{k+1}|^2} \quad (2.27)$$

となる. これに合わせて, q_k の式に書きなおすと

$$r_{k+1} = r_k - \frac{|r_k|^2}{(p_k, Ap_k)} Ap_k$$

2.4 共役勾配法

$$\begin{aligned} &= r_k - \frac{|r_k|^2}{(|r_k|^2 q_k, |r_k|^2 Aq_k)} |r_k|^2 Aq_k \\ &= r_k - \frac{Aq_k}{(q_k, Aq_k)} \end{aligned} \quad (2.28)$$

となり, 式と対称的な形になる. 同様にして HS 版の式は

$$x_{k+1} = x_k + \frac{q_k}{(q_k, Aq_k)} \quad (2.29)$$

となる. HS 版よりも, はるかにすっきりすっきりしていて, 計算量も少ない.

計算手順は次のようになる.

1. 第 0 近似解 x_0 を適当にとる
2. 第 0 近似解に対する残差を計算する

$$r_0 = b - Ax_0$$

3. q を求める

$$q_0 = r_0 / |r_0|^2$$

$$k = 0$$

4. 第 $k + 1$ 近似値を

$$x_{k+1} = x_k + \frac{q_k}{(q_k, Aq_k)}$$

5. 第 $k + 1$ 近似値に対する残差を

$$r_{k+1} = r_k - \frac{Aq_k}{(q_k, Aq_k)}$$

2.4 共役勾配法

6.q の更新

$$q_{k+1} = q_k - \frac{r_{k+1}}{|r_{k+1}|^2}$$

7. 収束判定を行い収束が十分でなければ, 新 $k = k + 1$ として 4 に戻る

2.4.4 2 階版の計算手順

式 (2.27)(2.28) は, 移項すると,

$$q_{k+1} - q_k = \frac{r_{k+1}}{|r_{k+1}|^2} \quad (2.30)$$

$$r_{k+1} - r_k = -\frac{Aq_k}{(q_k, Aq_k)} \quad (2.31)$$

という具合に, 連立 1 階の差分方程式になる. こういう差分方程式は, 一方の変数を消去して, 単独 2 階の差分方程式になおすことができる.

まず式 (2.31) を変形して

$$(q_k, Aq_k)(r_{k+1} - r_k) = -Aq_k \quad (2.32)$$

これの添字を一つふやすと

$$(q_{k+1}, Aq_{k+1})(r_{k+2} - r_{k+1}) = -Aq_{k+1} \quad (2.33)$$

両式の差をとると

$$(q_k, Aq_k)(r_{k+1} - r_k) - (q_{k+1}, Aq_{k+1})(r_{k+2} - r_{k+1}) = A(q_{k+1} - q_k)$$

2.4 共役勾配法

ここで式を用いると

$$= \left(1/|r_{k+1}|^2\right) Ar_{k+1} \quad (2.34)$$

したがって

$$r_{k+2} = r_{k+1} + \frac{(q_k, Aq_k)}{(q_{k+1}, Aq_{k+1})} (r_{k+1} - r_k) - \frac{Ar_{k+1}}{(q_{k+1}, Aq_{k+1}) |r_{k+1}|^2} \quad (2.35)$$

を得る. しかし, このままだと, 係数に q が入っているから, ぐあいが悪い. そこで, これを q を使わないで計算することを考える. それには,

$$(q_{k+1}, Aq_{k+1}) = \frac{(r_{k+1}, Ar_{k+1})}{|r_{k+1}|^4} - (q_k, Aq_k) \quad (2.36)$$

という漸化式を用いればよい. その正当性は, 式 (2.30) を用いて, 次のようにして証明できる.

$$\begin{aligned} (r_{k+1}, Ar_{k+1}) &= \left(|r_{k+1}|^2 (q_{k+1} - q_k), A|r_{k+1}|^2 (q_{k+1} - q_k)\right) \\ &= |r_{k+1}|^4 \{(q_{k+1}, Aq_{k+1}) - 2(q_{k+1}, Aq_k) + (q_k, Aq_k)\} \end{aligned} \quad (2.37)$$

p_i の直交性 $(p_i, Ap_j) = 0 (i \neq j)$ および式 (2.26) から

$$(q_{k+1}, Aq_k) = 0 \quad (2.38)$$

したがって

$$(r_{k+1}, Ar_{k+1}) = |r_{k+1}|^4 \{(q_{k+1}, Aq_{k+1}) + (q_k, Aq_k)\} \quad (2.39)$$

これを移項したものが式 (2.36) である. なお, 出発時には

$$r_1 = p_1 = |r_1|^2 q_1 \quad (2.40)$$

2.4 共役勾配法

であるから

$$(q_1, Aq_1) = (r_1, Ar_1) / |r_1|^4 \quad (2.41)$$

とすればよい. これは式 (2.36) において

$$(q_0, Aq_0) = 0 \quad (2.42)$$

とすることに相当する. こうする方がプログラムは簡単になる.

一方, x の計算をするために式 (2.29) を変形すれば

$$(q_k, Aq_k)(x_{k+1} - x_k) = q_k \quad (2.43)$$

その添字を一つふやすと

$$(q_{k+1}, Aq_{k+1})(x_{k+2} - x_{k+1}) = q_{k+1} \quad (2.44)$$

両式の差を作ると

$$\begin{aligned} (q_{k+1}, Aq_{k+1})(x_{k+2} - x_{k+1}) - (q_k, Aq_k)(x_{k+1} - x_k) &= q_{k+1} - q_k \\ &= r_{k+1} |r_{k+1}|^2 q_k \\ &= r_{k+1} |r_{k+1}|^2 \end{aligned} \quad (2.45)$$

したがって

$$x_{k+2} = x_{k+1} + \frac{(q_k, Aq_k)}{(q_{k+1}, Aq_{k+1})} (x_{k+1} - x_k) + \frac{r_{k+1}}{(q_{k+1}, Aq_{k+1}) |r_{k+1}|^2} \quad (2.46)$$

2.4 共役勾配法

となる. ここで

$$a_k = (q_k, Aq_k) \quad (2.47)$$

$$b_k = |r_k|^2 \quad (2.48)$$

と書いて計算式をもとめると次のようになる.

1. x_0 を適当にとる
2. 第 0 近似解に対する残差を計算する

$$r_0 = b - Ax_0$$

3. 第 k 回の修正係数 α_k を

$$\alpha = \frac{(r_0, r_0)}{r_0, Ar_0}$$

4. x の更新

$$x_1 = x_0 + \alpha_0 r_0$$

5. r の更新

$$r_1 = r_0 - \alpha Ar_0$$

6. a の計算

$$(r_0, Ar_0) / |r_0|^4$$

7. b の計算

$$b_0 = |r_0|^2$$

2.4 共役勾配法

$$k = 0$$

8.a の更新

$$a_{k+1} = \frac{(r_{k+1}, Ar_{k+1})}{|r_{k+1}|^4} - a_k$$

9.b の更新

$$b_{k+1} = |r_{k+1}|^2$$

10.x の更新

$$x_{k+2} = \left(1 + \frac{a_k}{a_{k+1}}\right) x_{k+1} - \frac{a_k}{a_{k+1}} x_k + \frac{r_{k+1}}{a_{k+1} b_{k+1}} \quad (2.49)$$

11.r の更新

$$r_{k+2} = \left(1 + \frac{a_k}{a_{k+1}}\right) r_{k+1} - \frac{a_k}{a_{k+1}} r_k + \frac{Ar_{k+1}}{a_{k+1} b_{k+1}} \quad (2.50)$$

12. 収束判定を行い収束が十分でなければ, 新 $k = k + 1$ として 8 に戻る

2.4.5 RG 版の計算手順

式 (2.49) および (2.50) を移項すると

$$x_{k+2} - x_{k+1} = \frac{a_k}{a_{k+1}} (x_{k+1} - x_k) + \frac{r_{k+1}}{a_{k+1} b_{k+1}} \quad (2.51)$$

$$r_{k+2} - r_{k+1} = \frac{a_k}{a_{k+1}} (r_{k+1} - r_k) - \frac{Ar_{k+1}}{a_{k+1} b_{k+1}} \quad (2.52)$$

2.4 共役勾配法

となるから, x_k や r_k のかわりに, その増分

$$\Delta x_k = x_{k+1} - x_k \quad (2.53)$$

$$\Delta r_k = r_{k+1} - r_k \quad (2.54)$$

を主変数にとれば, もっと単純な形になるであろう. ついでに

$$\frac{1}{a_k b_k} = \frac{1}{\varepsilon} \quad (2.55)$$

$$\frac{a_{k-1}}{b_k} = \frac{\varepsilon_{k-1}}{\delta_k} \quad (2.56)$$

すなわち

$$\delta_k = a_k b_k \quad (2.57)$$

$$\varepsilon_{k-1} = a_{k-1} b_k \quad (2.58)$$

と置き, 添字を一つずらして書くと, 計算手順は次のようになる.

1. x_0 を適当にとる
2. 第 0 近似解に対する残差を計算する

$$r_0 = b - Ax_0$$

3. Δx_0 の定義

$$\Delta x_0 = 0$$

4. Δr_0 の定義

$$\Delta r_0 = 0$$

2.4 共役勾配法

$$k = 0$$

5. ε_{k-1} の定義

$k = 0$ のとき

$$\varepsilon_{k-1} = 0$$

$k > 0$ のとき

$$\varepsilon_{k-1} = \delta_{k-1} \frac{|r_k|^2}{|r_{k-1}|^2}$$

6. δ_k の計算

$$\delta_k = \frac{(r_k, Ar_k)}{|r_k|^2} - \varepsilon_{k-1}$$

7. Δx_k の更新

$$\Delta x_k = (1/\delta) (\varepsilon_{k-1} \Delta x_{k-1} + r_k)$$

8. Δr_k の更新

$$\Delta r_k = (1/\delta) (\varepsilon_{k-1} \Delta r_{k-1} + Ar_k)$$

9. x の更新

$$x_{k+1} = x_k + \Delta x_k$$

10. r の更新

$$r_{k+1} = r_k + \Delta r_k$$

2.4 共役勾配法

11. 収束判定を行い収束が十分でなければ, 新 $k = k + 1$ として 5 に戻る

2.4.6 単調版の計算手順

高橋版, 2 階版, RG 版は HS 版の変形であり, 本質的には同じ公式である. したがって, それらの内のどの公式で計算しても, 近似解の列は

$$x_0, x_1, \dots, x_m$$

は同じものが得られる. また, 残差

$$r_0, r_1, \dots, r_m$$

も共通である. その「大きさ」は,

$$(r_k, A^{-1}r_k)$$

すなわち, エネルギーを尺度として用いれば, 単調に減少するが,

$$(r_k, r_k)$$

すなわち, 残差 2 乗和の意味では単調に減少しない.

しかし, できるところならば, 残差 2 乗和 (r_k, r_k) が反復の毎回, 単調に減少する方が望ましい.

公式の骨子としてはこれまでの公式と同じ形を使用し, ただ, そこに現れる係数を少し変えることにより, 残差 2 乗和が単調に減少する公式を作る.

骨組となる公式としては, この場合, RG 版

$$\Delta x_k = (1/\delta_k) (\varepsilon_{k-1} \Delta x_{k-1} + r_k) \quad (2.59)$$

2.4 共役勾配法

$$x_{k+1} = x_k + \Delta x_k \quad (2.60)$$

を用いるのが便利である. この係数 $\varepsilon_{k-1}, \delta_k$ をどのように決めても

$$\Delta r_k = (1/\delta_k)(\varepsilon_{k-1}\Delta r_{k-1} - Ar_k) \quad (2.61)$$

$$r_{k+1} = r_k + \Delta r_k \quad (2.62)$$

となることは容易にわかる. いうまでもなく r_k は x_k に対する残差

$$r_k = b - Ax_k \quad (2.63)$$

である. これにより

$$r_{k+1} = \left(1 + \frac{\varepsilon_{k-1}}{\delta_k}\right)r_k - \frac{\varepsilon_{k-1}}{\delta_k}r_k - \frac{1}{\delta_k}Ar_k \quad (2.64)$$

を得る. 両辺に δ_k を掛けて, 移行すれば, 次のように書ける.

$$Ar_k = -\delta_k r_{k+1} + (\delta_k + \varepsilon_{k-1})r_k - \varepsilon_{k-1}r_{k-1} \quad (2.65)$$

行列 A を対称かつ正定値とする. その固有値を

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$$

対応する固有ベクトルを長さ 1 に正規化したものを

$$v_1, v_2, \dots, v_n$$

とする. 第 0 近似解に対する残差 r_0 をこの基底で表したものを

$$r_0 = \sum_{i=1}^n c_i v_i \quad (2.66)$$

2.4 共役勾配法

とする. 以後の残差は, ε や δ をどのようにとっても,

$$r_k = \varphi_k(A)r_0 \quad (2.67)$$

ただし φ_k は k 次多項式の形で表すことができる. この φ_k が残差多項式である. 式 (2.64) に式 (2.67) を入れれば, φ_k に関する漸化式

$$\varphi_{k+1}(t) = \left(1 + \frac{\varepsilon_{k-1}}{\delta_k}\right) \varphi_k(t) - \frac{\varepsilon_{k-1}}{\delta_k} \varphi_{k-1}(t) - \frac{1}{\delta_k} t \varphi_k(t) \quad (2.68)$$

が得られる.

$$\varphi_0(t) + 1 \quad (2.69)$$

$$\varphi_1(t) = 1 - (1/\delta_k)t \quad (2.70)$$

で出発して, 式で作られる残差多項式はすべての定数項が 1 となる.

定数項が 1 の, あらゆる k 次多項式の中でノルム $\sqrt{\langle \varphi_k, \varphi_k \rangle}$ が, 最小になるものを作るには, $\varphi_0, \varphi_1, \varphi_2, \dots$ が, 次式

$$f, g = \sum_{i=0}^n t_i w_i f(t_i) g(t_i) = \int_a^b f(t) g(t) t \rho(t) dt$$

の意味で直交関数系になるようにすればよい. 今の場合, 最小化したいものは

$$(r_k, r_k) = \langle \varphi_k, \varphi_k \rangle \quad (2.71)$$

ただし

$$\langle f, g \rangle = \sum_{i=1}^2 c_i^2 f(\lambda_i) g(\lambda_i) \quad (2.72)$$

2.4 共役勾配法

であるから, 残差多項式の方の重み

$$\lambda_1 c_1^2 \lambda_2 c_2^2 \cdots \lambda_n c_n^2$$

に関して直交, すなわち

$i \neq j$ のとき

$$\{\varphi_i, \varphi_j\} = \sum_{j=1}^n \lambda_i c_i^2 \varphi_i(\lambda_i) \varphi_j(\lambda_i) = 0 \quad (2.73)$$

となるように定めなければならない. それには漸化式

$$f_{k+1}(t) = t f_k(t) - \frac{\langle t f_k, f_k \rangle}{\langle f_k, f_k \rangle} f_k(t) - \frac{\langle t f_k, f_{k-1} \rangle}{\langle f_{k-1}, f_{k-1} \rangle} f_{k-1}(t)$$

を使えばよいはずであるが, 上式そのままでは式 (2.68) の形に合わない. 式は $t\varphi_k(t)$ に係数 $-1/\delta_k$ が掛かっている. よく考えてみると, この項は, いわば「 $k+1$ 次式の材料」であるからここに係数が掛かっても, それに合わせて

$$\varphi_{k+1}(t) = \frac{1}{\delta_k} t \varphi_k(t) - \frac{\{-t\varphi_k/\delta_k, \varphi_k\}}{\{\varphi_k, \varphi_k\}} \varphi_k(t) - \frac{\{-t\varphi_k/\delta_k, \varphi_{k-1}\}}{\{\varphi_{k-1}, \varphi_{k-1}\}} \varphi_{k-1}(t) \quad (2.74)$$

というように直交化してやればよい. この式の第2項の係数を G_k , 第3項の係数を h_k とすると

$$\delta_k G_k = \frac{\{t\varphi_k, \varphi_k\}}{\{\varphi_k, \varphi_k\}} = \frac{\sum_{i=1}^n \lambda_i^2 c_i^2 \varphi_k(\lambda_i)^2}{\sum_{i=1}^n \lambda_i c_i^2 \varphi_k(\lambda_i)^2} = \frac{(Ar_k, Ar_k)}{(r_k, Ar_k)} \quad (2.75)$$

これと式を比較すると, まず $\varphi_{k-1}(t)$ の係数から

$$\epsilon_{k-1} = -\frac{(Ar_k, At_{k-1})}{(r_{k-1}, Ar_{k-1})} \quad (2.76)$$

2.4 共役勾配法

また $\varphi_l(t)$ の係数から

$$\delta_k + \varepsilon_{k-1} = \frac{(Ar_k, Ar_k)}{(r_k, Ar_k)} \quad (2.77)$$

$$\delta_k = \frac{(Ar_k, Ar_k)}{(r_k, Ar_k)} - \varepsilon_{k-1} \quad (2.78)$$

これで計算式が確定した. なお, ε_{k-1} の計算式は次のように書くこともできる.

$$\varepsilon_{k-1} = \frac{(r_k, Ar_k)}{(r_{k-1}, Ar_{k-1})} \delta_{k-1} \quad (2.79)$$

実際, 式 (2.73) より, 一般に

$$i \neq j$$

ならば

$$(r_i, Ar_j) = 0$$

が成立するから, 式を移項した

$$\delta_k r_{k+1} = (\delta_k + \varepsilon_{k-1}) r_k - \varepsilon_{k-1} r_{k-1} + Ar_k \quad (2.80)$$

と Ar_{k+1} の内積を作ると

$$\delta_k (r_{k+1}, Ar_{k+1}) = -(Ar_k, Ar_{k+1}) \quad (2.81)$$

$$\delta_k = -\frac{(Ar_{k+1}, Ar_k)}{(r_{k+1}, Ar_{k+1})} \quad (2.82)$$

これと式から式を得る

計算手順は次のようになる

2.4 共役勾配法

1. x_0 を適当にとる

2. 第 0 近似解に対する残差を計算する

$$r_0 = b - Ax_0$$

3. Δx_0 の定義

$$\Delta x_0 = 0$$

4. Δr_0 の定義

$$\Delta r_0 = 0$$

$$k = 0$$

5. ε_{k-1} の定義

$k = 0$ のとき

$$\varepsilon_{k-1} = 0$$

$k > 0$ のとき

$$\varepsilon_{k-1} = \frac{(r_k, Ar_k)}{(r_{k-1}, Ar_{k-1})} \delta_{k-1}$$

6. δ_k の計算

$$\delta_k = \frac{(Ar_k, Ar_k)}{(r_k, Ar_k)} - \varepsilon_{k-1}$$

7. Δx_k の更新

$$\Delta x_k = (1/\delta_k)(\varepsilon_{k-1} \Delta x_{k-1} + r_k)$$

2.4 共役勾配法

8. Δr_k の更新

$$\Delta r_k = (1/\delta_k)(\varepsilon_{k-1}\Delta r_{k-1} - Ar_k)$$

9. x の更新

$$x_{k+1} = x_k + \Delta x_k$$

10. r の更新

$$r_{k+1} = r_k + \Delta r_k$$

11. 収束判定を行い収束が十分でなければ, 新 $k = k + 1$ として 5 に戻る

第 3 章

3.1 まえがき

共役勾配法を適応アルゴリズムで利用する。そのために、適応アルゴリズムで利用する上で適した共役勾配法を選ぶ必要がある。前章で述べた HS 版, 高橋版, 2 階版, RG 版, 単調版これらの方式から適したものを選ぶ。

適応アルゴリズムで共役勾配法を利用する場合, 雑音が存在することを考える必要がある。共役勾配法は雑音の影響を受けることにより, 収束速度と収束精度を両立させることが困難になる。このため, 適応アルゴリズムで利用する場合の共役勾配法には安定性が必要となる。

3.2 共役勾配法の選択

まず, シュミレーションを行うために, 前章の共役勾配法それぞれの計算手順をもとにプログラミングを行う。シュミレーションを行う場合の条件として次のことがあげられる。

共役勾配法は連立 1 次方程式の係数

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & & \vdots & \\ & & & a_{nn} \end{bmatrix}$$

が対称の場合に適用できる。そのことから, 本論文では係数 A は対称行列とする。適応アルゴリズムでの利用を考え, シュミレーションを行うときの元の数 n は 100 とする。最適な繰返し回数が決まるまで, 繰返し回数を元と同じ数にする。

3.3 繰返し回数

3.3 繰返し回数

共役勾配法の繰返し回数が収束値を決める主な要因になる。シミュレーション結果をもとに、共役勾配法の繰返し回数を決める。

3.4 HS 版シミュレーション結果

雑音が存在する場合と存在しない場合では誤差がどのように変化するかを調べる。雑音が存在しない場合は、 n 回の繰返しで収束する。しかし、雑音が存在する場合は、数回の繰返しで雑音が存在する場合とは逆の現象が起こる。

このことから、繰返し回数が収束値を決める主な要因であることがわかる。

第 4 章

結論

4.1 結論

共役勾配法を雑音の存在する状態で収束値が, どのような変化するかシミュレーションを行った. このことにより, 共役勾配法は雑音が存在する状態では収束速度が遅いことが判明した.

4.2 今後の課題

本研究では共役勾配法の HS 版のみのシミュレーションである.

- 共役勾配法の他の方式で同じシミュレーションを行った場合, どのような結果になるか
- 適応アルゴリズムで利用するのに適した共役勾配法を選ぶ
- 収束値を決める要因になる繰返し回数

以上のことが, 今後の課題になる.

謝辞

本研究を行うにあたり、御指導並びに御助言を頂いた高知工科大学情報システム工学科福本昌弘講師に深く感謝致します。

本論文を御審議して下さる高知工科大学情報システム工学科島村和典教授, 菊池豊助教授, 情報システム工学科の先生方に心より感謝致します。

また, 本論文の文書スタイルファイルを作ってくださった高知工科大学大学院後期過程の橋本学様, 高知工科大学大学院前期過程の井上富幸様, 中平拓司様に深く感謝致します。

最後に, 御協力を賜りました福本研究室の皆様方にも感謝致します。

参考文献

- [1] 辻井重男, 久保田一, 古川利博, 趙晋輝, 適応信号処理, 昭晃堂,1995.
- [2] 戸川隼人, 共役勾配法, 教育出版,1977.