JAIST Repository
https://dspace.jaist.ac.jp/

Title	WAVELETを用いたナビエ・ストークス方程式の並列解法
Author(s)	福井，直人
Citation	
Issue Date	1998-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1115
Rights	
Description	Supervisor:松澤 照男，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

WAVELET を用いたナビエ・ストークス方程式の並列解法

指導教官　松澤 照男 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

福井直人

1998 年 2 月 13 日

Copyright © 1998 by Naoto Fukui

要 旨

目 次

1 はじめに **1**

2 不完全離散ウェーブレット変換と共役勾配法の前処理への応用 **3**

2.1 ウェーブレット 3

2.1.1 ウェーブレット 3

2.1.2 Daubechies ウェーブレット 6

2.2 前処理つき共役勾配法 11

2.3 離散ウェーブレット変換の微分作用素への応用 12

2.4 不完全離散ウェーブレット変換 15

2.5 対角スケーリング行列 17

3 基礎方程式 **21**

3.1 ポアソン方程式 21

3.1.1 差分法による離散化 21

3.2 拡散方程式 22

3.2.1 差分法による離散化 22

3.3 １次元バーガース方程式 22

3.3.1 差分法による離散化 23

4 実験 **24**

4.1 ポアソン方程式 24

4.1.1 １次元問題 24

4.1.2 ２次元問題 25

4.2 １次元拡散方程式 28

4.3　１次元バーガース方程式 34

5　考察 **36**

5.1　実験結果より 36

5.2　並列化への展望 37

6　謝辞 **39**

第 1 章

はじめに

数値流体解析においての数値シミュレーションは、ほとんどが大きな自由度を必要とする大規模計算である。このようなシミュレーションでは、対象の現象を支配する偏微分方程式は、有限要素法や有限差分法といった方法で離散化することにより計算される。その計算の計算時間の大部分を占めるのが、連立一次方程式を解く部分である。一般に、その方程式の規模が大きい場合は反復解法が用いられるが、問題が大規模になれば、その係数行列の固有値分布が悪性化し、計算時間の増大を招くことになる。これは係数行列の固有値分布が悪性化し、反復数が増大するためである。このような計算時間の増大は詳細な数値解析を困難にしている要因のひとつである。

一方、近年フーリエ解析に代わる手法としてウェーブレット解析が信号解析、画像処理等で用いられるようになった。更に、コンパクトなサポートを持つウェーブレットが I.Daubechies により提案されたことにより [5]、その適用範囲も広がり、数値解析の分野でもその応用が可能となった。このウェーブレットの数値解析への応用で注目されるのは、連立一次方程式の解法に、ウェーブレットを利用すると、その条件数は格子点数に依存しなくなることである [7]。その性質を利用した行列解法も提案されているが、アルゴリズムが非常に複雑で実用的でないない [6]。そこで、このようなウェーブレットの特長を活かしつつ、容易に実行可能な近似手法が提案された。それは、行列解法そのものではなく、その前処理に適用するものである。これは、行列解法自体には厳密さが要求されるが、行列解法の前処理としては、厳密で複雑なものより、処理手順が簡単で、計算効率の良い近似手法のほうが適しているからである。その近似方法は、ウェーブレットの問題である境界での値の取り扱いを簡略化したもので、不完全ウェーブレット変換 (iDWT) と

呼ぶ。フーリエ変換のように周期的な関数を用いた場合、こうした近似は、全ての展開係数に影響を与える。これに対し、ウェーブレット変換は、その局所性に特長があり、このような近似における影響は境界付近のみで抑えられる。ここにウェーブレットを用いる理由の一つがある。また、ウェーブレットを用いる、もう一つの理由も、その局所性に基づく。離散ウェーブレット変換の計算機上での処理が、局所的なデータ操作だけで良いため、アルゴリズムがベクトル化や並列化に向いているという点である。

流れの運動を表すナビエ・ストークス方程式は、放物型と楕円型の偏微分方程式は組み合わされた形となる。楕円型は最終的には、ポアソン方程式の形となり、放物型は非線形の形となる。本研究ではナビエ・ストークス方程式を解くために、まず、楕円型問題としてポアソン方程式放物型問題として拡散方程式、非線形問題であるバーガーズ方程式をiDWTを前処理に用いた共役勾配法を使って解き、その有効性を確認する。また、その並列化についての展望を述べる。

第 2 章

不完全離散ウェーブレット変換と共役勾配法の前処理への応用

2.1 ウェーブレット

2.1.1 ウェーブレット

ウェーブレットとは局在する波を表す関数の総称で、そのウェーブレット関数をスケール（伸縮）や平行移動することにより、任意の波形の局在的な部分を表すことができる。ウェーブレット関数を$\psi(x)$ とすると、ある波形をこのウェーブレット関数で表そうとすることをウェーブレット変換という。

ウェーブレット変換と同じような変換にフーリエ変換がある。しかしフーリエ変換は、局所的に激しく変化する波形に対して振る舞いが悪いことが知られており、ウェーブレット変換はこのような問題を解決するために考えだされた。

ウェーブレット関数を$\psi(x)$ とすると、ウェーブレット変換は次のように定義される。

$$(W_\psi f)(b,a) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{|a|}} \overline{\psi\Big(\frac{x-b}{a}\Big)} f(x)dx \tag{2.1}$$

ウェーブレット逆変換は次のように定義される。

$$f(x) = \frac{1}{C_\psi} \int\int (W_\psi f)(b,a) \frac{1}{\sqrt{|a|}} \overline{\psi\Big(\frac{x-b}{a}\Big)} \frac{dadb}{a^2} \tag{2.2}$$

(2.1) 式での a, b は、それぞれ、スケール（伸縮）、平行移動を表すパラメータである。(2.2) 式の右辺が定義されるためには、次の条件が満たされなけならない。

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} dw < \infty \tag{2.3}$$

ただし、$\hat{\psi}(x)$ は$\psi(x)$ のフーリエ変換。(2.3) 式の代わりに、ふつう次の条件式が使われる。

$$\int_{-\infty}^{\infty} \psi(x) dx = 0 \tag{2.4}$$

これは、$\psi(x)$ が振動的であることを意味する。

離散ウェーブレット変換 (DWT) は $(b, 1/a)$ を $(b, 1/a) = (2^{-j}k, 2^j)$ と置いて離散化される。$(W_\psi f)(2^{-j}k, 2^{-j})$ を d_k^jとすると、DWT とその逆変換は次のように書ける。

$$d_k^j = 2^j \int_{-\infty}^{\infty} \overline{\psi(2^j x - k)} f(x) dx \tag{2.5}$$

$$f(x) = \sum_j \sum_k d_k^j \psi(2^j x - k) \tag{2.6}$$

(2.6) 式での右辺の 2 重和の一方を、次のように表す。

$$\hat{f}_j(x) = \sum_k d_k^j \psi(2^j x - k) \tag{2.7}$$

また、

$$f_j(x) = \hat{f}_{j+1}(x) + \hat{f}_{j+2}(x) + \cdots \tag{2.8}$$

と表すことにする。ここで整数 jはレベルと呼ばれる。信号 $f(x)$ を基底解像度レベルの信号 $f_0(x)$ と見なすと、(2.6) 式は

$$f_0(x) = \hat{f}_1(x) + \hat{f}_2(x) + \cdots \tag{2.9}$$

と書ける。これは信号 $f_0(x)$ をウェーブレット成分 $g_1(x), g_2(x), ...,$ に分解したことに対応する。この分解は、ウェーブレット関数ψを基底関数として一意的に得られる。

ウェーブレット関数$\psi(x)$ は多重解像度解析と呼ばれる関数空間の階層構造を利用して作られる。与えられた係数列 $\{h_k\}$ に対して、トゥースケール関係と呼ばれる関係式

$$\phi(x) = \sum_{k \in Z} h_k \phi(2x - k) \tag{2.10}$$

を満たす関数$\phi(x)$ をスケーリング関数という。レベル jを固定してすべての整数 kについての$\phi(2^{-j}x-k)$ が張る空間を V_jとすると、トゥースケール関係から $V_j \subset V_{j-1}$が導かれる。スケーリング関数を使ってウェーブレット関数を、係数列 $\{g_k\}$ を用いて、

$$\psi(x) = \sum_{k \in Z} g_k \phi(2x-k) \tag{2.11}$$

と定義することができる。このように定義されたウェーブレット関数は補空間 $W_0 = V_1 \backslash V_0$ の基底関数となることが示される。

(2.8) 式は $f_j(x)$ についての再帰的な形

$$f_j(x) = \hat{f}_{j+1}(x) + f_{j+1}(x) \tag{2.12}$$

と書き直すことができる。このように (2.12) 式を使って、レベルを 1 ずつ上げることができ、解像度をその度に半分して、関数 $f(x)$ を分解することが出来る。

レベル j での $f_j(x)$ は次のような再帰的な形で表される。

$$f_j(x) \cong \sum_k s_k^j \phi(2^j x - k) \tag{2.13}$$

つまり、(2.7) 式、(2.13) 式より関数 $f(x)$ はウェーブレット関数$\psi(x)$、スケーリング関数 $\phi(x)$ により、(2.12) 式のような解像度レベル j で再帰的な形に分解される。

(2.13) 式で、右辺の関数ϕはレベル j によらず同一である。(2.7) 式における関数$\hat{f}_j(x)$ も同一の関数ψにより表される。これは次のような意味を持つ。$f_j(x)$ の分解は (2.12) 式にしたがって行なわれるが、実際には (2.13) 式に現れる係数 s_k^jから s_k^{j+1}と (2.7) 式に現れる d_k^{j+1}を求める。これは次の分解アルゴリズムが使われる。

$$s_k^{j+1} = \frac{1}{2} \sum_{l \in Z} g_{2k-l} s_l^j \tag{2.14}$$

$$d_k^{j+1} = \frac{1}{2} \sum_{l \in Z} h_{2k-l} s_l^j \tag{2.15}$$

分解係数列 $\{g_k\}\{h_k\}$ はトゥースケール関係より決定される。ここで用いられる $\{g_k\}\{h_k\}$ はどのレベルからの分解でも共通であり、これは、f_jまたは g_jがそれぞれ同一のϕまたはψによって表されることに対応している。こうして関数 $f(x)$ は各解像度レベルに分解される。

2.1.2 Daubechies ウェーブレット

ここではDaubechiesが考案したDaubechiesウェーブレットを考える。Daubechiesウェーブレットは、直交ウェーブレットであり、直交基底を作る連続かつコンパクトサポートなウェーブレットである。スケーリング関数ϕは多重解像度解析を生成し、ウェーブレット関数ψはその直交補空間を張るというのが基本的な性質である。自然数 M によって変わるスケーリング関数ϕと対応するウェーブレット関数ψは、M とともに滑らかさが増大する。サポートとは、関数 $f(x)$ の値がゼロでない区間を $f(x)$ のサポートという。このサポートが有限の区間であるとき、コンパクトサポートという。

ψとϕは以下のトゥースケール関係を満たす。

$$\psi(x) = \sqrt{2}\sum_{k=0}^{2M-1} g_k \phi(2x-k) \tag{2.16}$$

$$\phi(x) = \sqrt{2}\sum_{k=0}^{2M-1} h_k \phi(2x-k) \tag{2.17}$$

但し、係数列 $\{h_k\}$ と $\{g_k\}$ は次の関係を満たす。

$$g_k = (-1)^{k-1} h_{2M-k+1} \tag{2.18}$$

但し、M(自然数) は消失モーメントである。

Daubechiesは有限個の 0 でない $\{h_k\}$ から、連続かつコンパクトサポートなスケーリング関数を構成することを示した。Daubechies ウェーブレットのフィルタ係数列 $\{h_i\}$ は、ウェーブレット固有の係数であり、用いるウェーブレットを決めると一意に定まるものである。Daubechies ウェーブレットのフィルタ係数列 $\{h_i\}$ の例を表2.1に示す。

２つの関数 u_1, u_2の内積を、

$$\langle u_1 \,|\, u_2 \rangle = \int_\Omega u_1(x) u_2(x) dx$$

と定義する。スケーリング関数$\phi(x)$ とウェーブレット関数$\psi(x)$ は、次の直交関係を表す内積の式を満たす。

$$\begin{aligned}
\langle \phi(x-l) \,|\, \phi(x-m) \rangle &= \delta_{l,m} \\
\langle \psi(x-l) \,|\, \psi(x-m) \rangle &= \delta_{l,m} \\
\langle \psi(x-l) \,|\, \phi(x-m) \rangle &= 0, \quad l, m \in Z
\end{aligned}$$

表 2.1: Daubechies ウェーブレットフィルタ係数 $\{h_k\}$

i	M=1	M=2	M=3	M=4	M=5
0	$2^{-1/2}$	0.482962913145	0.332670552950	0.230377813309	0.160102397974
1	$2^{-1/2}$	0.836516303738	0.806891570553	0.714846570553	0.603829269797
2	0	0.224143868042	0.459877502118	0.680880767930	0.724308528438
3	0	-0.129409522551	-0.135011020010	0.030841381836	0.138428145901
4	0	0	-0.085441273882	-0.187034811719	-0.242294887066
5	0	0	0.035226291882	0.030841381836	-0.032244869585
6	0	0	0	0.032883011667	0.077571493840
7	0	0	0	-0.010597401785	-0.0006241490213
8	0	0	0	0	-0.012580751999
9	0	0	0	0	0.003335725285

これよりスケーリング関数$\phi(x)$ は次の式を満たす。

$$\int_{-\infty}^{\infty} \phi(x)dx = 1 \tag{2.19}$$

また、ウェーブレット関数$\psi(x)$ は (2.4) 式を満たす。

以上の条件だけでは、必ずしも連続な関数にはならない。そこで、Daubechies ウェーブレットは、モーメントの条件を持つ。

$$\int_{-\infty}^{+\infty} x^m \psi(x)dx = 0, \quad m = 0, ..., M-1 \tag{2.20}$$

この条件は Daubechies のウェーブレットが連続で滑らかな関数であることを意味する。

スケーリング関数$\phi(x)$ とウェーブレット関数$\psi(x)$ は、直交系を形成するように次のように定義する。

$$\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k) \tag{2.21}$$

$$\phi_{j,k}(x) = 2^{-j/2}\phi(2^{-j}x - k) \tag{2.22}$$

レベル jのスケーリング関数 $\{\phi(2^{-j}x - k)\}_{k\in Z}$の張る空間を $\boldsymbol{V}_j$とすれば、(2.17) 式のトゥースケール関係より、$\boldsymbol{V}_j \subset \boldsymbol{V}_{j-1}$となる。こうして、スケーリング関数$\phi(x)$ が与えられ

ると、それに対応して関数空間の階層構造

$$\cdots \quad \subset \boldsymbol{V}_2 \subset \boldsymbol{V}_1 \subset \boldsymbol{V}_0 \subset \boldsymbol{V}_{-1} \subset \boldsymbol{V}_{-2} \quad \cdots$$

が決まる。これをスケーリング関数$\phi(x)$ によって生成される多重解像度解析という。

一方、レベル jのウェーブレット関数 $\{\psi(2^{-j}x-k)\}_{k\in Z}$の張る空間を $\boldsymbol{W}_j$とする。(2.16) 式のトゥースケール関係より、

$$\boldsymbol{V}_{j-1} = \boldsymbol{V}_j \oplus \boldsymbol{W}_j$$

と書ける。$\boldsymbol{W}_j$は $\boldsymbol{V}_{j-1}$の $\boldsymbol{V}_j$に関する直交補空間である。

(2.7) 式、(2.12) 式、(2.13) 式より、関数 $f(x)$ を解像度レベル Jまで展開すると、次にように表される。

$$f(x) \cong \sum_{k\in Z} s_k^J \psi_{J,k} + \sum_{j=1}^{J}\sum_{k\in Z} d_k^j \phi_{j,k} \tag{2.23}$$

フーリエ変換同様、展開式 (2.23) の展開係数が意味を持ち、その展開係数は、

$$s_k^j = \int_\Omega f(x)\phi_{j,k}(x)dx, \quad d_k^j = \int_\Omega f(x)\psi_{j,k}(x)dx \tag{2.24}$$

で定義される。これらの展開係数は、次の関係により (2.24) 式の積分を行なうことなく計算される。

$$s_k^j = \sum_{i=0}^{2M-1} h_i s_{i+2k}^{j-1} \tag{2.25}$$

$$d_k^j = \sum_{i=0}^{2M-1} g_i s_{i+2k}^{j-1} \tag{2.26}$$

(2.25) 式、(2.26) 式の分解アルゴリズムにより、関数 $f(x)$ は各解像度レベルに分解される。

ここで、(2.22) 式、(2.24) 式より基底解像度の展開係数 $\{s_k^0\}$ は

$$s_k^0 = 2^{n/2}\int_\Omega f(x)\psi(2^n x-k)dx \tag{2.27}$$

で定義される。(2.27) 式を台形公式で近似すると、次の関係が得られる。

$$s_k^0 \cong 2^{-n/2}\sum_{i=0}^{2M-1} f_{k+i}\psi(i) \tag{2.28}$$

(2.28) 式により、$\{s_k^0\}$ を求めてもよいが、n が十分大きく、$f(x)$ が滑らかであれば、

$$\sum_{i=0}^{2M-1} \psi(i) = 1 \tag{2.29}$$

の関係より、さらに、

$$s_k^0 \cong 2^{-n/2} f_k \tag{2.30}$$

と単純化した近似も考えられる。この方法を用いると、格子データ $\{f_i\}$ の $2^{-n/2}$倍したものを基底解像度のデータとみなし、ピラミッドアルゴリズムにより、順次、各解像度レベルのデータに分解することができる。これにより、離散ウェーブレット変換 (DWT) することが可能となる。しかし、有限体系問題のウェーブレット展開では、図 2.1に示すように体系外のデータが必要となる。一般に、この点を解決するために、問題を周期化し展開される。周期化を施した DWT は正規直交性を完全に満足することとなる。

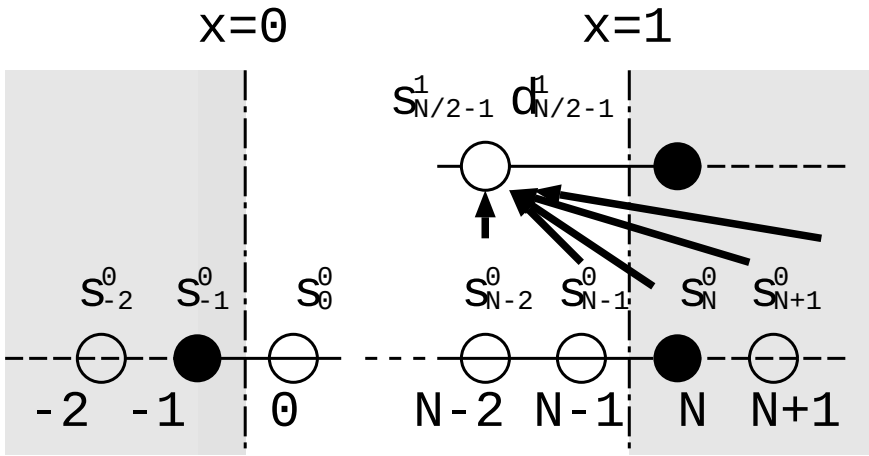

図 2.1: 1 次元の DWT

得られた展開係数 $\{s_k^j\}, \{d_k^j\}$ より $\{s_k^{j-1}\}$ を求めることができる。これは次の再構成アルゴリズムにより得られる。

$$s_{2n}^{j-1} = \sum_{i=0}^{M-1} h_{2i} s_{n-i}^j + \sum_{i=0}^{M-1} g_{2i} d_{n-i}^j \tag{2.31}$$

$$s_{2n+1}^{j-1} = \sum_{i=0}^{M-1} h_{2i+1} s_{n-i}^j + \sum_{i=0}^{M-1} g_{2i+1} d_{n-i}^j \tag{2.32}$$

この分解、再構成アルゴリズムをピラミッドアルゴリズムと呼ぶ。

このピラミッドアルゴリズムは、以下のような行列表現することができる。DWT によ

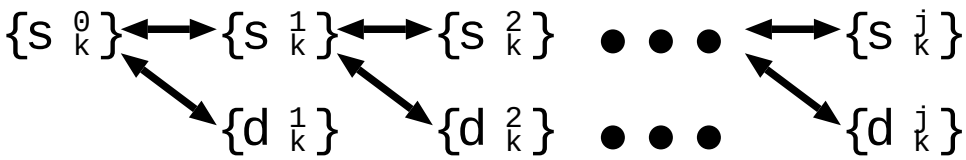

図 2.2: ピラミッドアルゴリズム

るレベル j-1 から j への展開係数を変換する行列$\tilde{\boldsymbol{W}}_j$とすると、

$$\tilde{\boldsymbol{W}}_j = \begin{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{H}}_j \\ \tilde{\boldsymbol{G}}_j \end{pmatrix} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{N-2^{n-j}} \end{pmatrix} \tag{2.33}$$

と表される。ただし、$\tilde{\boldsymbol{H}}_j$,$\tilde{\boldsymbol{G}}_j$は、それぞれ $\{h_k\}, \{g_k\}$ を用いて次のように定義する。

$$\tilde{\boldsymbol{H}}_j = \underbrace{\begin{pmatrix} h_0 & h_1 & h_2 & \cdots & h_{2M-1} & 0 & & \cdots & & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & \cdots & h_{2M-1} & 0 & \cdots & 0 \\ & & & & \ddots & & \ddots & & & \\ h_4 & \cdots & h_{2M-1} & 0 & \cdots & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & h_4 & \cdots & h_{2M-1} & 0 & \cdots & 0 & h_0 & h_1 \end{pmatrix}}_{2^{n-j}} \Bigg\} 2^{n-j-1} \tag{2.34}$$

$$\tilde{\boldsymbol{G}}_j = \underbrace{\begin{pmatrix} g_{2M-2} & g_{2M-1} & 0 & \cdots & 0 & g_0 & \cdots & g_{2M-3} \\ g_0 & g_1 & g_2 & \cdots & g_{2M-1} & 0 & \cdots & 0 \\ & & & \ddots & & \ddots & & \\ 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & g_{2M-1} & 0 & 0 \\ 0 & & \cdots & & 0 & g_0 & g_1 & g_2 & \cdots & g_{2M-1} \end{pmatrix}}_{2^{n-j}} \Bigg\} 2^{n-j-1} \tag{2.35}$$

この行列$\tilde{\boldsymbol{W}}_j$は、

$$\tilde{\boldsymbol{W}}_j \tilde{\boldsymbol{W}}_j^T = \boldsymbol{I} \tag{2.36}$$

となり、正規直交性を満たす。この行列$\tilde{\boldsymbol{W}}_j$は分解アルゴリズムにおけるフィルタ係数列$\{\mathrm{h}_k\},\{g_k\}$を行列表現したものである。また、この転置行列$\tilde{\boldsymbol{W}}_j^T$が再構成アルゴリズムのフィルタ係数列$\{\mathrm{h}_k\},\{g_k\}$を行列表現したものとなる。この行列$\tilde{\boldsymbol{W}}_j$により、解像度レベル0から$J(0 \leq j \leq J)$への展開係数の変換を行なう行列$\tilde{\boldsymbol{W}}_{(J)}$が得られる。

$$\tilde{\boldsymbol{W}}_{(J)} = \tilde{\boldsymbol{W}}_{J-1}\tilde{\boldsymbol{W}}_{J-2}\cdots\tilde{\boldsymbol{W}}_1\tilde{\boldsymbol{W}}_0 \tag{2.37}$$

2.2 前処理つき共役勾配法

共役勾配法（conjugate gradient method）は、行列の前処理と併用すれば、その収束性を格段に高めることが出来る。正定値対称行列の解法としては、最も汎用な解法である。

解くべき連立一次方程式を以下のようにする。

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.38}$$

とする。$\boldsymbol{K}$が正定値であれば、

$$\boldsymbol{K} = \boldsymbol{V}^T\boldsymbol{V} \tag{2.39}$$

となる正則行列$\boldsymbol{V}$が存在する。前処理つき共役勾配法では、この$\boldsymbol{V}$を用いて、(2.38) 式を以下のように書き換える。

$$\boldsymbol{V}^T\boldsymbol{A}\boldsymbol{V}\ \hat{\boldsymbol{x}} = \hat{\boldsymbol{b}} \tag{2.40}$$

ただし、$\hat{\boldsymbol{x}}$,$\hat{\boldsymbol{b}}$は以下のように与えられる。

$$\hat{\boldsymbol{x}} = \boldsymbol{V}^{-T}\boldsymbol{x},\ \hat{\boldsymbol{b}} = \boldsymbol{V}\boldsymbol{b} \tag{2.41}$$

前処理つき共役勾配法では、(2.38) 式の代わりに (2.40) 式を解くが、そのアルゴリズムは次のようになる。

手順１ (初期化) (A) 初期ベクトル $\boldsymbol{x}_0$

(B) $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$

(C) $\boldsymbol{p}_0 = \tilde{\boldsymbol{r}}_0 = \boldsymbol{K}\boldsymbol{r}_0$

手順２ (反復) (D) $\alpha_k = \frac{(\tilde{\boldsymbol{r}}_k, \boldsymbol{r}_k)}{(\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)}$

(E) $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$

(F) $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k$

(G) 収束判定

(H) $\tilde{\boldsymbol{r}}_{k+1} = \boldsymbol{K}\boldsymbol{r}_{k+1}$

(I) $\beta_k = \frac{(\tilde{\boldsymbol{r}}_{k+1}, \boldsymbol{p}_k)}{(\tilde{\boldsymbol{r}}_k, \boldsymbol{r}_k)}$

(J) $\boldsymbol{p}_{k+1} = \tilde{\boldsymbol{r}}_k + \beta_k \boldsymbol{p}_k$

このアルゴリズムでは、前処理は、手順１ (C) と手順２ (H) で$\tilde{\boldsymbol{r}} = \boldsymbol{K}\boldsymbol{r}$を計算するだけである。この行列$\boldsymbol{K}$は前処理行列と呼び、前処理つき共役勾配法では非常に重要な役割を果たす。前処理に必要とされる条件の主なものは、

(a) 正定値であること。

(b) 前処理後の行列の条件数（最大固有値と最小固有値の比）が改善されていること。

(c) $\tilde{\boldsymbol{r}} = \boldsymbol{K}\boldsymbol{r}$の計算が早くでき、更には、ベクトル化や並列化に適していること。

である。このような前処理の代表的なものとして不完全コレスキー分解に基づく手法(ICCG 法) がある。しかしながら、ICCG 法も、後述の大規模化に伴う計算時間の増大を防ぐことはできない。また、その不完全コレスキー分解を用いた前処理部、本来、ベクトル化、並列化に向かず、逆に、前処理をしない CG 法よりも計算時間が増大してしまうと言うこともある。ここでは、ウェーブレットを、この前処理部に用いることによりそうした問題を解決することも可能である。

2.3 離散ウェーブレット変換の微分作用素への応用

微分方程式、

$$\frac{d^m u(x)}{dx^m} = f(x) \quad in \quad \Omega \tag{2.42}$$

を考える。以下では、簡単のため、$d^m u/dx^m$を $u^{(m)}$と記す。

$u(x), f(x)$ を基底解像度のスケーリング関数により展開すると、

$$u = \sum_k s_k^0 \phi_{0,k} \tag{2.43}$$

$$f = \sum_k p_k^0 \phi_{0,k} \tag{2.44}$$

と展開できる。ここで、

$$u^{(m)} = \sum_k s_k^0 (\phi_{0,k})^{(m)} \tag{2.45}$$

となり、

$$\begin{aligned} (\phi_{0,k})^{(m)} &= \sum_{k'} \left\langle (\phi_{0.k})^{(m)} \;\middle|\; \phi_{0.k'} \right\rangle \phi_{0,k'} & (2.46) \\ &= 2^{nm} \sum_{k'} \left\langle (\phi^{(m)})_{0,k} \;\middle|\; \phi_{0,k'} \right\rangle \phi_{0,k'} & (2.47) \end{aligned}$$

であるから（以下では、$(\phi_{0,k}^{(m)})$ を$\phi_{0,k}^{(m)}$と書くことにする）、$u^{(m)}$は以下のように展開できる。

$$u^{(m)} = 2^{(nm)} \sum_{k,k'} \left\langle \phi_{0,k}^{(m)} \;\middle|\; \phi_{0,k'} \right\rangle s_k^0 \phi_{0,k'} \tag{2.48}$$

ここで、$m = 2$ の場合を考える。(2.43) 式より、

$$\frac{d^2 u}{dx^2} = \sum_{j,k} 2^{2n-2j} d_k^j \psi_{j,k}^{(2)} \tag{2.49}$$

となる。(2.49) 式より、

$$\left\langle \frac{d^2 u}{dx^2} \;\middle|\; \psi_{j',k'} \right\rangle = \langle f \mid \psi_{j',k'} \rangle, \quad j', k' = 1, 2, 3, \cdots \tag{2.50}$$

$$\Leftrightarrow 2^{2n} \sum_{j'k'} 2^{-2j} \left\langle \psi_{j,k}^{(2)} \;\middle|\; \psi_{j',k'} \right\rangle d_k^j = q_k^{j'}, \quad j', k' = 1, 2, 3, \cdots \tag{2.51}$$

となる。これは、行列・ベクトル表示するならば、

$$2^{2n} \boldsymbol{D} \boldsymbol{M} \boldsymbol{D} \hat{\boldsymbol{u}}_\infty = \hat{\boldsymbol{f}}_\infty \tag{2.52}$$

となる。ただし、

$$\boldsymbol{D} = \left[\left\{ 2^{-j} \; \boldsymbol{I}_{\;2^{n-j}} \right\} \right], \quad \boldsymbol{M} = \left[\left\langle \psi_{j,k}^{(2)} \;\middle|\; \psi_{j',k'} \right\rangle \right] = \left[\left\langle \psi_{j,k}^{(1)} \;\middle|\; \psi_{j',k'}^{(1)} \right\rangle \right] \tag{2.53}$$

$$\hat{\boldsymbol{u}}_\infty = \left[d_k^j \right], \quad \hat{\boldsymbol{f}}_\infty = \left[q_{k'}^{j'} \right] \tag{2.54}$$

である。

一方、

$$\frac{du}{dx} = \sum_{j,k} 2^{n-j} d_k^j \psi_{j,k}^{(1)} \tag{2.55}$$

であるから、

$$\hat{\boldsymbol{u}}_\infty^T \boldsymbol{D}\boldsymbol{M}\boldsymbol{D}\hat{\boldsymbol{u}}_\infty = \left\langle \frac{d^2u}{dx^2} \;\middle|\; u \right\rangle \tag{2.56}$$

$$= \left\| \frac{du}{dx} \right\|^2 \tag{2.57}$$

となる。ここで、以下の式が成り立つことが証明されている [7]。

$$C \sum_{j,k} (2^{n-j} d_k^j)^2 \leq \left\| \frac{du}{dx} \right\|^2 \leq C' \sum_{j,k} (2^{n-j} d_k^j)^2 \tag{2.58}$$

従って、(2.57) より、

$$C \sum_{j,k} (2^{n-j} d_k^j)^2 \leq \hat{\boldsymbol{u}}_\infty^T \boldsymbol{D}\boldsymbol{M}\boldsymbol{D}\hat{\boldsymbol{u}}_\infty \leq C' \sum_{j,k} (2^{n-j} d_k^j)^2 \tag{2.59}$$

となり、

$$C \sum_{j,k} (d_k^j)^2 \leq \hat{\boldsymbol{u}}_\infty^T \boldsymbol{M}\hat{\boldsymbol{u}}_\infty \leq C' \sum_{j,k} (d_k^j)^2 \tag{2.60}$$

最終的に、

$$C \left\| \hat{\boldsymbol{u}}_\infty \right\|^2 \leq \hat{\boldsymbol{u}}_\infty^T \boldsymbol{M}\hat{\boldsymbol{u}}_\infty \leq C' \left\| \hat{\boldsymbol{u}}_\infty \right\|^2 \tag{2.61}$$

となる。この式を書き換えると、

$$C \leq \frac{\hat{\boldsymbol{u}}_\infty^T \boldsymbol{M}\hat{\boldsymbol{u}}_\infty}{\left\| \hat{\boldsymbol{u}}_\infty \right\|^2} \leq C' \tag{2.62}$$

となるから、行列$\boldsymbol{M}$の固有値の最小値はCに、最大値はC'によって抑えられる。従って、行列$\boldsymbol{M}$の条件数はC'/Cにより抑えられることがわかる。

微分作用素d^2/dx^2の離散化行列を$\boldsymbol{R}$とし、離散ウェーブレット変換 (DWT) を、

$$\tilde{\boldsymbol{W}}_J = \tilde{\boldsymbol{W}}_{J-1} \tilde{\boldsymbol{W}}_{J-2} \cdots \tilde{\boldsymbol{W}}_0 \tag{2.63}$$

で定義する。この行列を用いると、

$$\hat{\boldsymbol{u}}_\infty = \tilde{\boldsymbol{W}}_\infty \hat{\boldsymbol{u}}_0, \quad \hat{\boldsymbol{f}}_\infty = \tilde{\boldsymbol{W}}_\infty \hat{\boldsymbol{f}}_0 \tag{2.64}$$

の関係より、

$$2^{2n}\boldsymbol{R}\hat{\boldsymbol{u}}_0 = \hat{\boldsymbol{f}}_0 \tag{2.65}$$

$$\Leftrightarrow 2^{2n}\tilde{\boldsymbol{W}}_\infty \boldsymbol{R}\tilde{\boldsymbol{W}}_\infty^T \hat{\boldsymbol{u}}_\infty = \hat{\boldsymbol{f}}_\infty \tag{2.66}$$

となる。従って、

$$\boldsymbol{DMD} = \tilde{\boldsymbol{W}}_\infty \boldsymbol{R}\tilde{\boldsymbol{W}}_\infty^T \Leftrightarrow \boldsymbol{M} = \boldsymbol{D}^{-1}\tilde{\boldsymbol{W}}_\infty \boldsymbol{R}\tilde{\boldsymbol{W}}_\infty^T \boldsymbol{D}^{-1} \tag{2.67}$$

となる。これは微分作用素 d^2/dx^2の離散化行列を離散ウェーブレット変換し、(2.53) の $\boldsymbol{D}^{-1}$で定義される対角行列によりスケーリングを施すと、その行列 $(\boldsymbol{D}^{-1}\tilde{\boldsymbol{W}}_\infty \boldsymbol{R}\tilde{\boldsymbol{W}}_\infty^T \boldsymbol{D}^{-1})$ の条件数は、格子分割数によらず、C'/C に抑えられることを示している。

これは、周期化微分作用素を離散化してできる行列に、DWT を行ない、適切なスケーリングを行なうと、「格子点数に関わらず、その行列の条件数はあるよりも小さく抑えられる。」という特長を持つということである。しかし、その特長を生かした微分方程式の解法では、以下のような問題点がある。

(a) 問題を周期化したものに変換しなければならない。

(b) DWT を周期化しなければならない。

(c) 変換後の行列がランク落ちする。（０となる固有値が現れる。）

(d) 解法が非常に複雑になる。

2.4 不完全離散ウェーブレット変換

前節で説明した離散化された微分作用素にウェーブレットを応用した際に起きる問題点を抑えるために、ここでは周期化を行なわない不完全な DWT、不完全ウェーブレット変換 (iDWT:imcomplete discrete wavelet transform) を用いる。

ここでは、説明を簡単にするために、１次元楕円型方程式の境界値問題を対象として考える。境界条件はディレクレ境界条件とする。

$$-\Delta u = f \qquad in \quad \Omega = [0,1] \tag{2.68}$$

(2.68) 式を $h = 1/2^n$ の等格子間隔で差分化すると（以下、$N = 2^n$ とする）、次の連立一次方程式が得られる。

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.69}$$

ただし、

$$\boldsymbol{x} = (u_i), \quad \boldsymbol{b} = h^2(f_i) \tag{2.70}$$

係数行列$\boldsymbol{A}$は、次のようになる。

$$\boldsymbol{A} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \vdots \\ 0 & -1 & 2 & \ddots & \\ \vdots & & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \tag{2.71}$$

ここでは、(2.69) 式の連立一次方程式を対象として、その解法に DWT を応用すると前節で述べたようにいくつかの問題点が起きる。ここでは、その問題点を解決するため、DWT を近似的に表現して、周期化は行なわずに体系外のデータを無視する方法を適用する。そして、これを行列解法の前処理に用いる。これは、行列解法での、コレスキー分解に対する、不完全コレスキー分解のように、行列解法の前処理としては、厳密なものよりも、処理手順が簡単で計算効率の良い不完全な方が適しているからである。

この方法では、周期化された DWT によるレベル j-1 から j への展開係数を変換する行列$\tilde{\boldsymbol{W}}_j$の近似として、行列成分の末端部を省略した行列、

$$\boldsymbol{W}_j = \begin{pmatrix} \begin{pmatrix} \boldsymbol{H}_j \\ \boldsymbol{G}_j \end{pmatrix} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{N-2^{n-j}} \end{pmatrix} \tag{2.72}$$

を用いる。ただし、$\boldsymbol{H}_j$,$\boldsymbol{G}_j$は、それぞれ $\{h_k\}, \{g_k\}$ を用いて、

$$\boldsymbol{H}_j = \underbrace{\left.\begin{pmatrix} h_0 & h_1 & h_2 & \cdots & h_{2M-1} & & & & & 0 \\ & & h_0 & h_1 & h_2 & \cdots & h_{2M-1} & & & \\ & & & & \ddots & & \ddots & & & \\ & & & & & & h_0 & h_1 & h_2 & h_3 \\ 0 & & & & & & & & h_0 & h_1 \end{pmatrix}\right\} 2^{n-j-1}}_{2^{n-j}} \tag{2.73}$$

$$
\boldsymbol{G}_j = \underbrace{\left.\begin{pmatrix}
h_1 & -h_0 & 0 & \cdots & & & & & 0 \\
h_3 & -h_2 & h_1 & h_0 & 0 & \cdots & & & \\
 & & & \ddots & \ddots & & & & \\
 & & & h_{2M-1} & \cdots & -h_2 & h_1 & -h_0 & \\
0 & & & & & h_{2M-1} & \cdots & -h_2 & h_1 & -h_0
\end{pmatrix}\right\} 2^{n-j-1}}_{2^{n-j}} \tag{2.74}
$$

と定義する。

この行列$\boldsymbol{W}_j$は、

$$
\boldsymbol{W}_j \boldsymbol{W}_j^T \cong \boldsymbol{I} \tag{2.75}
$$

となり、近似的に正規直交性を満たす。この行列$\boldsymbol{W}_j$を不完全離散ウェーブレット変換 (iDWT) と呼ぶ。この行列$\boldsymbol{W}_j$ を用いると、解像度レベル 0 から J $(0 \leq j \leq n)$ への展開係数の変換を行なう行列$\boldsymbol{W}_{(J)}$は次のように表される。

$$
\boldsymbol{W}_{(J)} = \boldsymbol{W}_{J-1} \boldsymbol{W}_{J-2} \cdots \boldsymbol{W}_1 \boldsymbol{W}_0 \tag{2.76}
$$

この行列$\boldsymbol{W}_{(J)}$を用いて、(2.69) 式を以下の近似式に変換される。

$$
\boldsymbol{W}_{(J)} \boldsymbol{A} \boldsymbol{W}_{(J)}^T \tilde{\boldsymbol{x}}_J \cong \tilde{\boldsymbol{b}}_J \tag{2.77}
$$

ただし、$\tilde{\boldsymbol{x}}_J, \tilde{\boldsymbol{b}}_J$は、$u(x), f(x)$ の展開係数より成る。

$$
\tilde{\boldsymbol{x}}_J = 2^{-n/2} \boldsymbol{W}_{(J)} \boldsymbol{x}\ ,\ \tilde{\boldsymbol{b}}_J = 2^{-n/2} \boldsymbol{W}_{(J)} \boldsymbol{b} \tag{2.78}
$$

2.5 対角スケーリング行列

先に述べた DWT を行なった行列の条件数に関する特長を引き出すために、対角スケーリングを行なうわけだが、具体的に、この行列の行列成分を考える。

スケーリング行列$\boldsymbol{P}$によって、(2.77) 式を変換すると次のよう表される。

$$
\boldsymbol{P} \boldsymbol{W}_{(J)} \boldsymbol{A} \boldsymbol{W}_{(J)}^T \boldsymbol{P} \hat{\boldsymbol{x}}_J \cong \hat{\boldsymbol{b}}_J \tag{2.79}
$$

ただし、$\hat{\boldsymbol{x}}, \hat{\boldsymbol{b}}$は、次のようになる。

$$
\hat{\boldsymbol{x}} = \boldsymbol{P} \boldsymbol{W}_{(J)} \boldsymbol{x}, \quad \hat{\boldsymbol{b}} = \boldsymbol{P} \boldsymbol{W}_{(J)} \boldsymbol{b} \tag{2.80}
$$

このスケーリング行列と iDWT を前処理に用いた CG 法での前処理行列$\boldsymbol{K}$は次のようになる。

$$\boldsymbol{K} = \boldsymbol{W}_{(J)}^T \boldsymbol{P}^2 \boldsymbol{W}_{(J)} \tag{2.81}$$

このスケーリング行列には次の 2 種類が考えられる。

(a) スケーリング手法１
行列の対角成分をとる一般的な手法。

$$\boldsymbol{W}_{(J)} \boldsymbol{A} \boldsymbol{W}_{(J)}^T = (\omega_{i,l}) \tag{2.82}$$

スケーリング行列は次のようになる。

$$\boldsymbol{P} = (\delta_{i,l}/\sqrt{\omega_{i,l}}) \tag{2.83}$$

(b) スケーリング手法２
対象とする偏微分方程式がポアソン方程式の場合、つまり、微分作用素 $\mathcal{L}$ が $\mathcal{L}$=Δ の場合。そのスケーリングに用いる対角行列$\boldsymbol{P}_{(J)}$は、次のようになる。

- $J \geq 2$ の場合

$$\boldsymbol{P}_{(J)} = \begin{pmatrix} 2^J \boldsymbol{I}_{2^{n-J+1}} & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{O} & 2^{J-1} \boldsymbol{I}_{2^{n-J+1}} & \boldsymbol{O} & & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & 2^{J-2} \boldsymbol{I}_{2^{n-J+2}} & & \vdots \\ \vdots & & & \ddots & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{O} & 2\boldsymbol{I}_{2^{n-1}} \end{pmatrix} \tag{2.84}$$

- J=1 の場合

$$\boldsymbol{P}_{(J)} = 2\boldsymbol{I}_{2^n} \tag{2.85}$$

(2.84) 式、(2.85) 式より、対角スケーリング行列は、次のように表される。

$$\boldsymbol{P}_{i,l} = \delta_{i,l} 2^j \tag{2.86}$$

ただし、$1 \leq j \leq J, \quad N - N/2^{j-1} + 1 \leq i, l \leq N - N/2^j$である。

この対角行列を用いて、(2.77) 式は次のように変換される。

$$\boldsymbol{P}_{(J)}\boldsymbol{W}_{(J)}\boldsymbol{A}\boldsymbol{W}_{(J)}^{T}\boldsymbol{P}_{(J)}\hat{\boldsymbol{x}}_J \cong \hat{\boldsymbol{b}}_J \tag{2.87}$$

ただし、

$$\hat{\boldsymbol{x}}_J = 2^{-n/2}\boldsymbol{P}_{(J)}^{-1}\boldsymbol{W}_{(J)}\boldsymbol{x}, \quad \hat{\boldsymbol{b}}_J = 2^{-n/2}\boldsymbol{P}_{(J)}\boldsymbol{W}_{(J)}\boldsymbol{b} \tag{2.88}$$

(2.87) 式を前処理つき共役勾配法で解く場合、(2.40) 式と比較して、$\boldsymbol{V} = \boldsymbol{P}_{(J)}\boldsymbol{W}_{(J)}$ と考えれば良いから、共役勾配法に現れる前処理因子は

$$\boldsymbol{K} = \boldsymbol{V}_T\boldsymbol{V} = \boldsymbol{W}_{(J)}^{T}\boldsymbol{P}_{(J)}^{2}\boldsymbol{W}_{(J)} \tag{2.89}$$

となる。上式の$\boldsymbol{P}_{(J)}^{2}$の処理は、対象がポアソン方程式の場合、不完全離散ウェーブレット変換の場合、そのフィルタ係数列 $\{h_k, g_k\}$ を、あらかじめ、１次元では、

$$\{h'_k, g'_k\} = 2\{h_k, g_k\} \tag{2.90}$$

と変換しておき、そのフィルタ係数を用いて、不完全ウェーブレット変換を行なえば省略できる。また、２次元のポアソン方程式の場合、２次元のウェーブレット展開は以下の式となり、

$$u(x,y) \cong \sum_{l,m} s_{l,m}^{J}\psi_{J,l}\psi_{J,m} + \sum_{j=1}^{J}\sum_{m,l}\{d_{l,m}^{(x),j}\phi_{j,l}\psi_{j,m} + d_{l,m}^{(y),j}\psi_{j,l}\phi_{j,m} + d_{l,m}^{(xy),j}\phi_{j,l}\phi_{j,m}\} \tag{2.91}$$

そして、その展開係数は１次元の場合と同様に、ピラミッドアルゴリズムにより、基底解像度のデータより求めることができる。

$$s_{l,m}^{j} = \sum_{i,k} h_i h_k s_{2l+i,2m+k}^{j-1} \tag{2.92}$$

$$d_{l,m}^{(x),j} = \sum_{i,k} g_i h_k s_{2l+i,2m+k}^{j-1} \tag{2.93}$$

$$d_{l,m}^{(y),j} = \sum_{i,k} h_i g_k s_{2l+i,2m+k}^{j-1} \tag{2.94}$$

$$s_{l,m}^{(xy),j} = \sum_{i,k} g_i g_k s_{2l+i,2m+k}^{j-1} \tag{2.95}$$

２次元のポアソン方程式の場合の対角スケーリング行列は、１次元の場合と同様に、あらかじめ、フィルタ係数列 $\{h_k, g_k\}$ を以下の式で、

$$\{h'_k, g'_k\} = \sqrt{2}\{h_k, g_k\} \tag{2.96}$$

と、変換しておけば、$\boldsymbol{P}^2_{(J)}$の計算の省略ができる。

以上をまとめると、共役勾配法の前処理に iDWT を適用する場合、前処理つき共役勾配法の手順 (C)(H) において、

$$\tilde{\boldsymbol{r}} = \boldsymbol{W}^T_{(j)} \boldsymbol{P}^2 \boldsymbol{W}_{(j)} \boldsymbol{r} \tag{2.97}$$

を処理するようになるが、これは、フィルタ係数列 $\{h_k\}, \{g_k\}$ を前もって (2.90) 式、(2.96) 式で変換しておき、そのフィルタ係数列を用いた iDWT を $\boldsymbol{r}$に施し、次に、その結果に iDWT の逆変換を施す、という手順になる。

これにより、前述の微分方程式を離散化して得られた (2.69) 式のような連立一次方程式の解法にウェーブレットを適用する際の問題点を解決し、完全ではないが、ウェーブレットを適用した際の特長が活かされ、連立一次方程式の解法に反復解法を用いた際の反復数の増加を抑える効果が得られる。

第 3 章

基礎方程式

3.1 ポアソン方程式

次のような２次元ポアソン方程式を考える。基礎方程式は以下のようになる。

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = -f(x,y) \tag{3.1}$$

ポアソン方程式は、代表的な楕円型問題である。

ポアソン方程式は定常熱伝導問題でよく用いられる方程式である。熱伝導問題の場合、u は温度、fはデカルト座標系での発熱量となる。時間項は含まれないので、必ず定常解が求められる。ポアソン方程式の解は定常状態の温度分布を表す。

3.1.1 差分法による離散化

ポアソン方程式を中心差分により図 3.1のように離散化すると次のようになる。

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = -f_{i,j} \tag{3.2}$$

格子点間隔$\Delta x = \Delta y = h$ とすると、

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = -hf_{i,j} \tag{3.3}$$

(3.3) 式より得られる連立一次方程式は、格子点間隔を $1/N$とすると、格子点数は $N \times N$ 個となり、連立一次方程式の係数行列は $N^2 \times N^2$の正定値対称行列となる。

この連立一次方程式を解くことにより、領域内での温度分布が得られる。

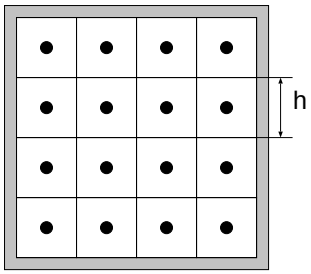

図 3.1: ポアソン方程式の離散化

3.2 拡散方程式

次のような１次元拡散方程式を考える。

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \tag{3.4}$$

ただし、νは拡散係数。

拡散方程式は、代表的な放物型問題で、熱伝導方程式とも呼ばれる。金属のような、一様な媒質中を高温から低温へと熱流が発生して、熱エネルギーが移動する。この現象を熱伝導という。拡散方程式は、この物理を表している。

3.2.1 差分法による離散化

拡散方程式をクランク・ニコルソン法により時間離散化すると次のようになる。

$$(1 - \frac{\nu \delta t}{2} \frac{\partial^2}{\partial x^2}) u^{n+1} = (1 + \frac{\nu \delta t}{2} \frac{\partial^2}{\partial x^2}) u^n + \delta t f \tag{3.5}$$

中心差分により空間離散化し、$\alpha = \frac{\nu \Delta t}{2 \Delta x^2}$とすると次のようになる。

$$-\alpha u_{i+1}^{n+1} + (1 + 2\alpha) u_i^{n+1} - \alpha u_{i-1}^{n+1} = \alpha u_{i+1}^n + (1 - 2\alpha) u_i^n + \alpha u_{i-1}^n \tag{3.6}$$

(3.6) 式より得られる連立一次方程式を解くことにより、領域内での拡散が得られる。

3.3 １次元バーガース方程式

次のような１次元バーガース方程式を考える。

バーガーズ方程式は非線形の移流・拡散方程式である。バーガーズ方程式はナビエ・ストークス方程式のもつ非線形性の一面をもつ。この非線形性ゆえに乱流、衝撃波といった複雑な物理に対するモデル問題として頻繁に用いられる。

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0 \tag{3.7}$$

3.3.1 差分法による離散化

１次元バーガース方程式を中心差分により次のように離散化する。

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} = -u_i^n\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \tag{3.8}$$

$\alpha = \frac{\nu\Delta t}{\Delta x^2}$、$\beta = \frac{\Delta t}{2\Delta x}$とすると、次のようになる。

$$-\alpha u_{i+1}^{n+1} + (1+2\alpha)u_i^{n+1} - \alpha u_{i-1}^{n+1} = u_i^n - u_i^n\beta(u_{i+1}^n - u_{i-1}^n) \tag{3.9}$$

(3.9) 式より得られる連立一次方程式を解くことにより、領域内での移流、衝撃波といった物理現象をみることができる。

第 4 章

実験

4.1 ポアソン方程式

共役勾配法 (CG 法)、不完全ウェーブレット変換による前処理つき共役勾配法 (iDWTCG 法) の反復回数を確認するため、楕円型問題として、ポアソン方程式について計算を行なった。実験は全て SGI・CRAY 社の Onyx 上において行なった。

4.1.1 1 次元問題

まず、以下の簡単な 1 次元問題を考える。

$$-\Delta u(x) = f(x) \quad in \quad \Omega = [0, 1] \tag{4.1}$$

境界条件は、以下の Dirichlet 条件を与える。

$$u(0) = u(1) = 0.0$$

$f(x)$ は、

$$f(x) = 1.0$$

とする。また、格子点数 $N = 2^n$, 格子点間隔 $1/N$, 収束条件を $\|\boldsymbol{Ax}_k - \boldsymbol{b}\|_2/\|\boldsymbol{b}\|_2 < 10^{-8}$ として、反復回数を調べた。ただし、ウェーブレットのモーメントは $M = 3$, 解像度レベル $J = n - 2$ とした。得られた結果を表 4.1に示す。

表 4.1の結果より、CG 法の反復回数は、格子点数 Nに比例して増加していることが確認される。一方、iDWTCG 法は、格子点数を増加しても、反復回数の増加が抑えられて

表 4.1: 1 次元のポアソン方程式の反復回数

N	CG	iDWTCG
16	8	13
32	16	17
64	32	21
128	64	26
256	130	33
512	269	39
1024	549	47

おり、iDWT による前処理が、反復解法の反復数増大を抑えるのに有効であることがわかる。

4.1.2 2 次元問題

次に、以下の 2 次元ポアソン方程式に対して、実験を行なう。

$$-\Delta u(x,y) = f(x,y) \quad in \quad \Omega = [0,1] \times [0,1] \tag{4.2}$$

境界条件は、以下の Dirichlet 条件を与える。

$$u(0,y) = u(1,y) = u(x,0) = u(x,1) = 0$$

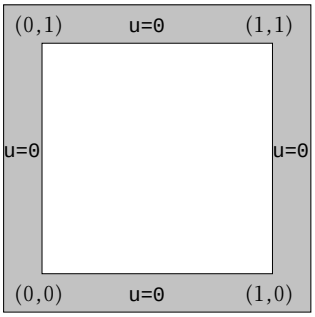

図 4.1: 計算領域

$f(x,y)$ は、

$$f(x,y) = 1.0$$

とする。これは厳密解が次のように求められる。

$$u(x,y) = \frac{4}{\pi^2}\sum_{n,m=1}^{\infty}(\frac{2}{(2m-1)\pi})(\frac{2}{(2n-1)\pi})\sin(2m-1)\pi x\cos(2n-1)\pi y \tag{4.3}$$

格子点数 $N = 2^n$, 格子点間隔 $1/N$, その結果、得られる連立一次方程式は、$N^2 \times N^2$の係数行列となる。収束条件を $\|\boldsymbol{Ax}_k - \boldsymbol{b}\|_2/\|\boldsymbol{b}\|_2 < 10^{-8}$として、反復回数を調べた。ただし、ウェーブレットのモーメントは $M = 3$, 解像度レベル $J = n-2$ とした。反復回数を示した結果を表 4.2に示す。厳密解と得られた数値解との誤差を表 4.3に示す。図 4.2に厳密解を、また、図 4.3に得られた数値解を示す。

表 4.2: 2 次元ポアソン方程式の反復回数

$N^2 \times N^2$	CG	iDWT-CG
$16^2 \times 16^2$	21	20
$32^2 \times 32^2$	42	27
$64^2 \times 64^2$	85	37

表 4.3: 2 次元ポアソン方程式の誤差

$N^2 \times N^2$	CG	iDWT-CG
$16^2 \times 16^2$	$6.52\ \times 10^{-4}$	$6.52\ \times 10^{-4}$
$32^2 \times 32^2$	$1.72\ \times 10^{-4}$	$1.72\ \times 10^{-4}$
$64^2 \times 64^2$	$4.44\ \times 10^{-4}$	$4.44\ \times 10^{-4}$

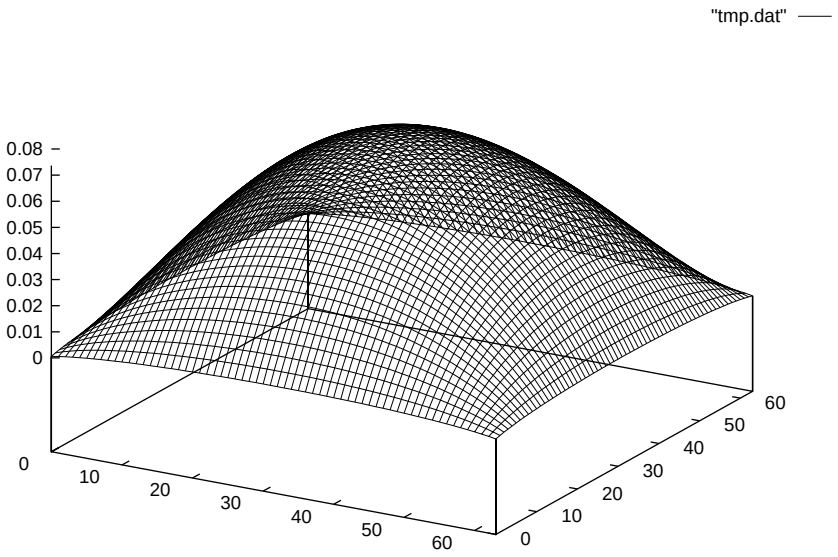

図 4.2: 厳密解

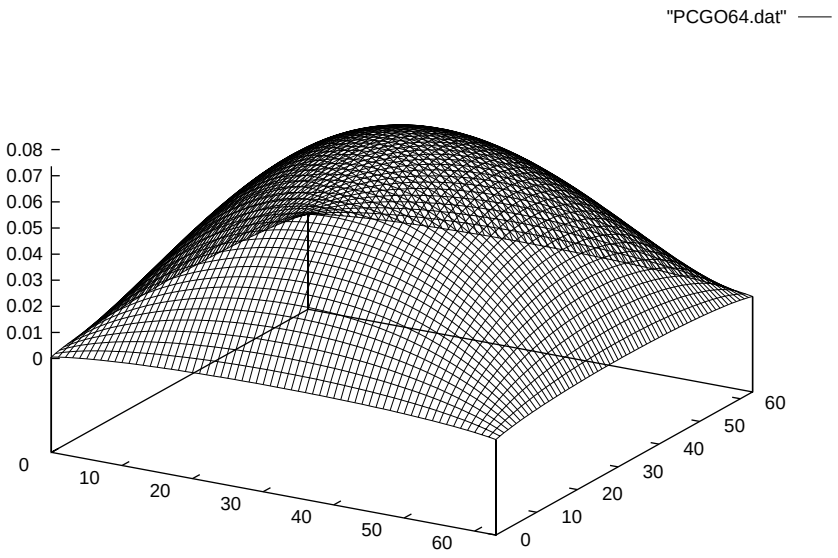

図 4.3: iDWT-CG 法による計算結果

4.2 １次元拡散方程式

放物型問題として、1 次元拡散方程式について、以下の初期条件、境界条件のもとで、計算を試みた。

$$u(x,0) = \sin \pi x \tag{4.4}$$

$$u(0,t) = u(1,t) = 0 \tag{4.5}$$

この場合の厳密解は、次式のようになる。

$$u(x,t) = e^{-\pi^2 t} \sin \pi x \tag{4.6}$$

また、格子点数 $N = 512$, 格子点間隔 $1/N$, 時間間隔$\Delta t = 10^{-6}$ 拡散係数$\nu = 1.0$ 収束条件を $\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\|_2/\|\boldsymbol{b}\|_2 < 10^{-8}$として、反復回数を調べた。ただし、ウェーブレットのモーメントは $M = 3$, 解像度レベル $J = 7$ とした。図 4.4にその計算結果を、図 4.5に、iDWT-CG 法での反復回数を、図 4.6に iDWT-CG 法での誤差を示す。また、図 4.7に CG 法での反復回数、図 4.8に CG 法での誤差を示す。

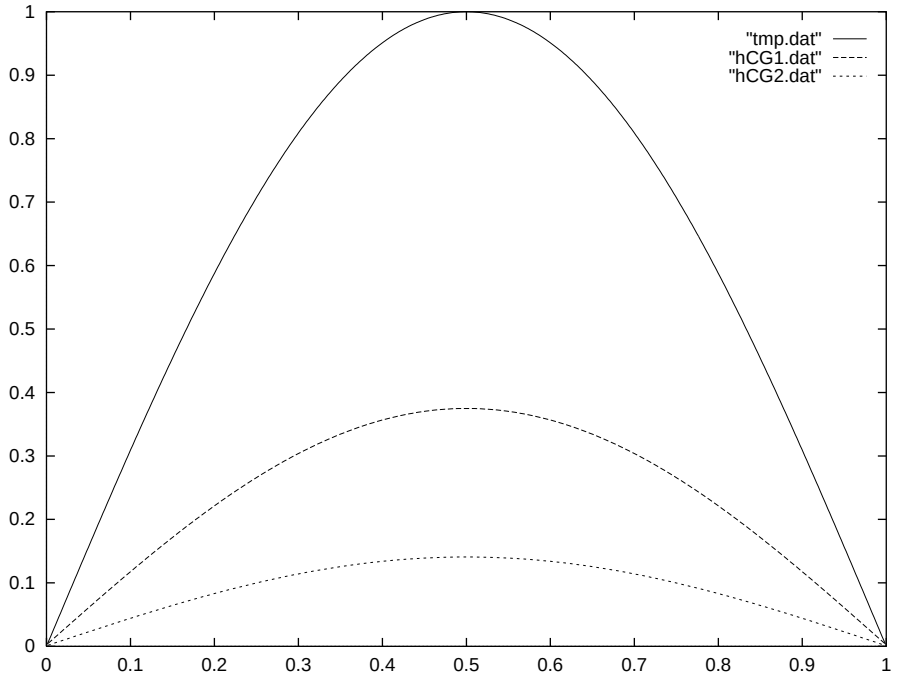

図 4.4: iDWT-CG 法による計算結果 (初期値、時間 t=0.1,0.2)

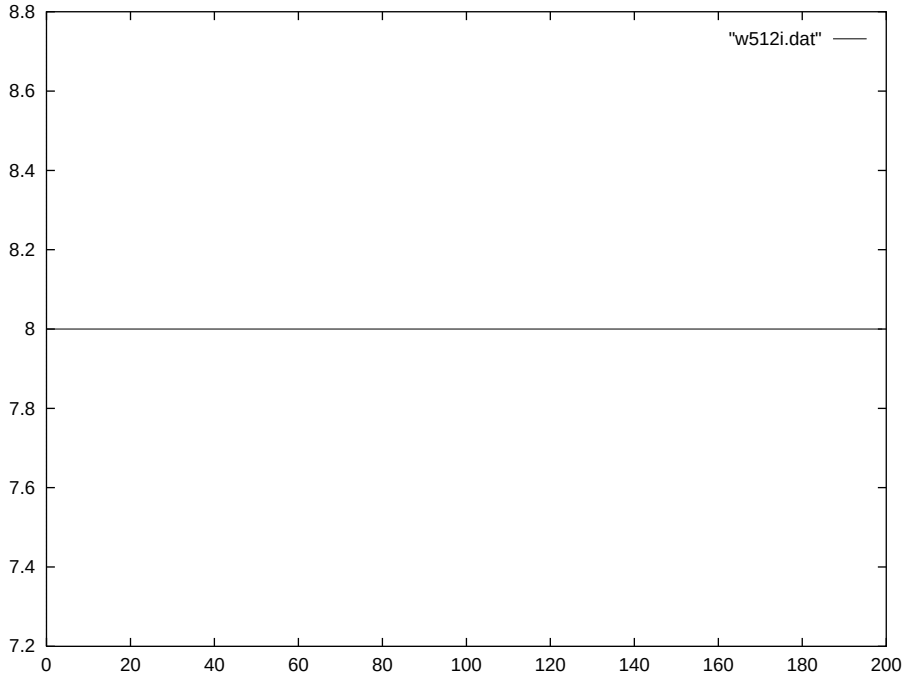

図 4.5: タイムステップ 200 までの反復回数 (iDWT-CG)

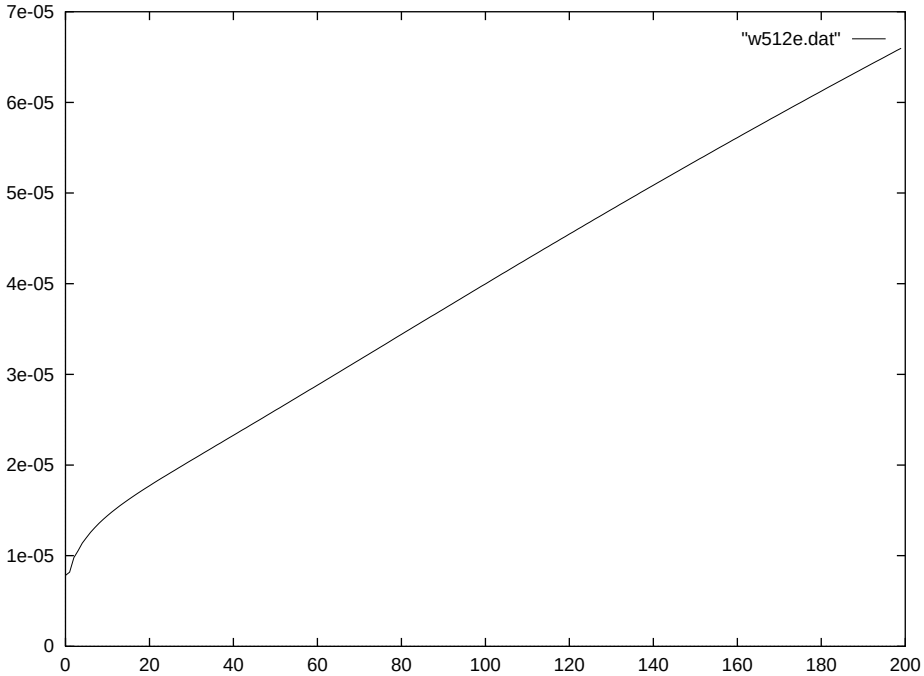

図 4.6: タイムステップ 200 までの誤差 (iDWT-CG)

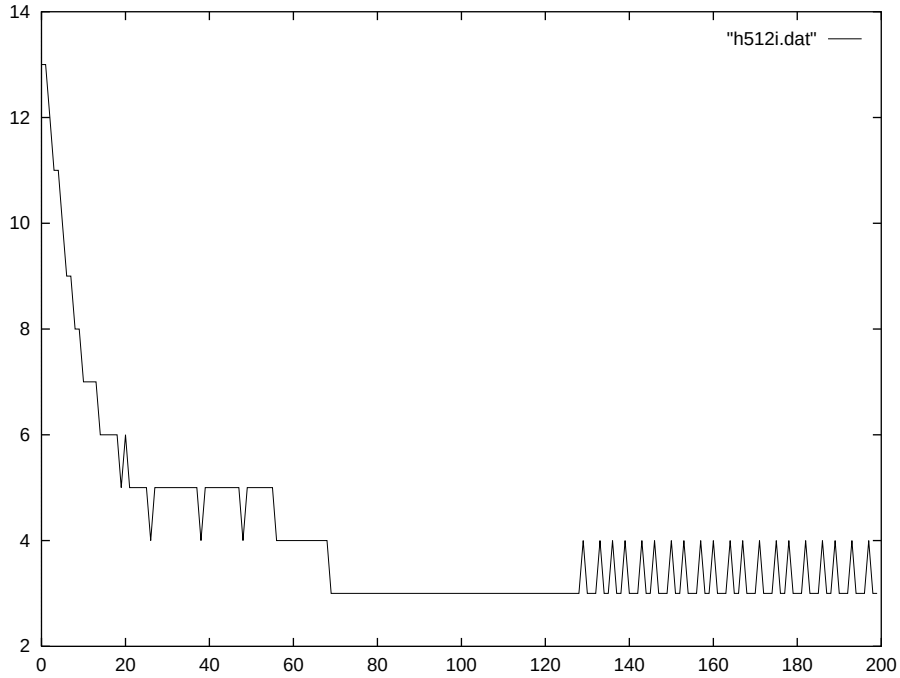

図 4.7: タイムステップ 200 までの反復回数 (CG)

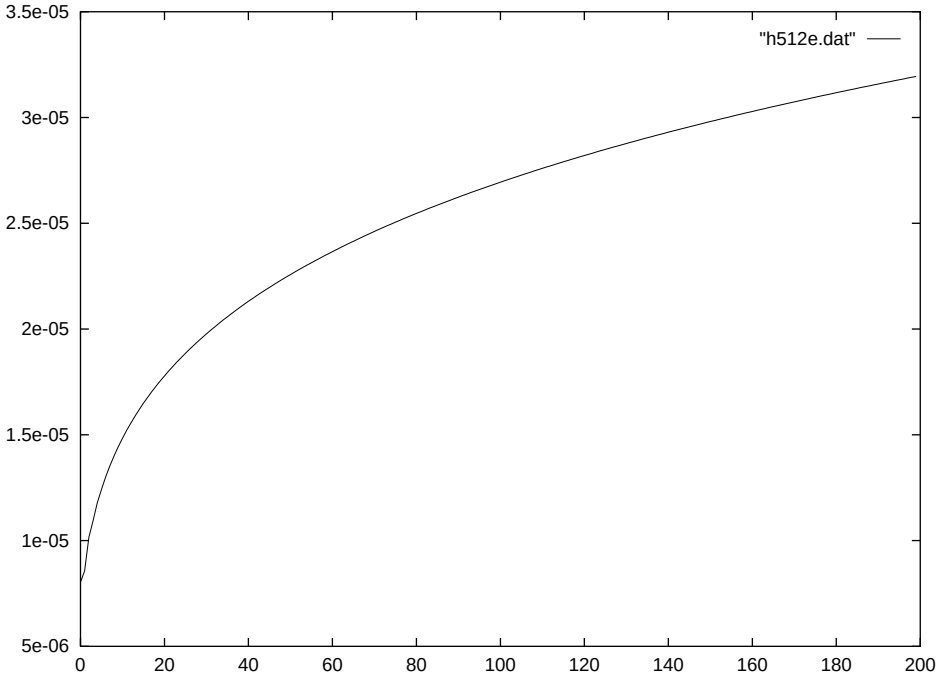

図 4.8: タイムステップ 200 までの誤差 (CG)

4.3 １次元バーガース方程式

非線形問題として、1 次元バーガーズ方程式について、以下の初期条件、境界条件のもとで、計算を試みた。

$$u(x,0) = \sin \pi x \tag{4.7}$$

$$u(0,t) = u(2,t) = 0 \tag{4.8}$$

また、格子点数 $N = 512$, 格子点間隔 $1/N$, 時間間隔$\Delta t = 10^{-6}$ 拡散係数$\nu = 1.0$ 収束条件を $\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\|_2/\|\boldsymbol{b}\|_2 < 10^{-8}$として、反復回数を調べた。ただし、ウェーブレットのモーメントは $M = 3$, 解像度レベル $J = 7$ とした。図 4.9に iDWT-CG 法により得られた計算結果を示す。

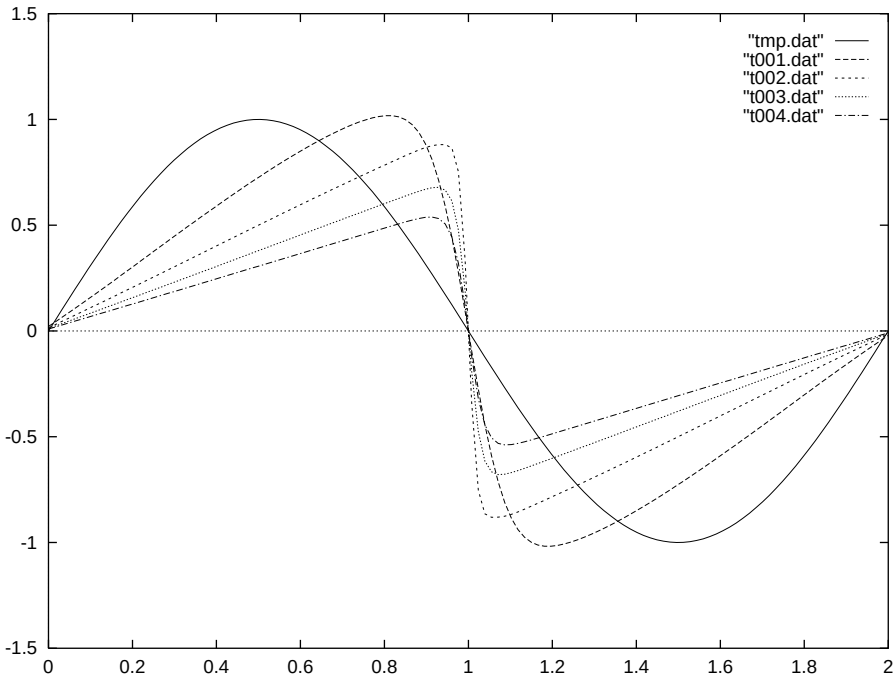

図 4.9: バーガーズ方程式の計算結果

第 5 章

考察

5.1 実験結果より

ポアソン方程式については、実験結果より以下のような知見を得た。

- 反復回数については、1 次元、2 次元ともに、格子点数を増やしても、iDWT-CG 法において、反復数の増加は緩やかなものとなった。また、CPU 時間においても、2 次元ポアソン方程式 (N=64) の場合で、CG 法は 15 分 6 秒、iDWT-CG 法は 6 分 26 秒となり、計算時間の減少も確かめられた。

- 誤差については、実験結果からは、CG 法、iDWT-CG 法ともに、変わらない誤差となった。この誤差を少数点以下を 18 桁まで取り出したところ、2 次元ポアソン方程式 (N=64) の場合で、少数点第 9 桁まで同じ誤差であった。これより、ポアソン方程式の場合、精度の面において CG-法と iDWTG 法は同等のものと考えられる。

1 次元拡散方程式については、実験結果より以下のような知見を得た。

- 図 4.6と図 4.8より、どちらも誤差はタイムステップが増すにつれて増加しているが、iDWT-CG 法の方が増加の度合いが大きい。これは、フィルタ係数列で iDWT を行なっているために、境界付近での誤差がタイムステップを進めるにつれて、次のタイムステップに影響を及ぼしているのではないかと考える。

- 反復回数は、iDWT-CG 法では、全てのタイムステップが 8 回の反復で収束している。一方、CG 法では、最初の数ステップで、10 回以上の反復を繰り返しているが、

タイムステップが進むにつれ、2,3 回の反復で計算が終了している。これは、拡散方程式がタイムステップを進めるにあたって、解が一様に近づき、定常状態になろうとしているためである。また、iDWT-CG 法では、解くべき連立一次方程式を iDWT により、ある関数空間へ射影して CG 法を行なっているために、タイムステップが進めても、反復回数は一定である。

5.2 並列化への展望

先にも述べたように、ウェーブレットのように局所的なデータだけを取り扱うために並列化が比較的容易に可能なのではないかと考える。図 5.1と図 5.2で説明する [4]。図 5.1は、1 次元 iDWT の一般的な並列化である。しかし、(図 5.1は、モーメント M=2 の Daubechies ウェーブレットので場合である。) ここで、問題となるのは、プロセッサの負荷に不均衡が生じていることである。この例では一つのプロセッサが計算量が多くなり、もう一つのプロセッサは、アイドル状態となっている。この負荷の不均衡は計算時間の増大を招くことなる。

図 5.2は、この問題点を解決している。このようにデータを配置することにより、負荷の不均衡は緩和され、アイドル状態となっているプロセッサ数が減少し、そして、通信量も減少していることになる。例えば、図 5.1の場合では、通信量 16 に対し、図 5.2の場合では、通信量 10 となっている。このように並列化することにより、さらに計算効率は上がるものと考える。

1

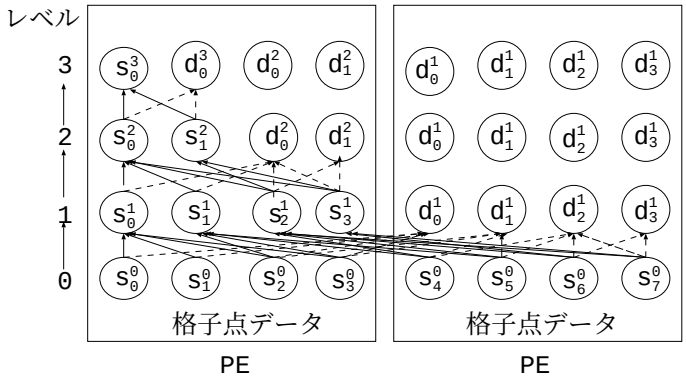

図 5.1: 1 次元の iDWT の並列化 1

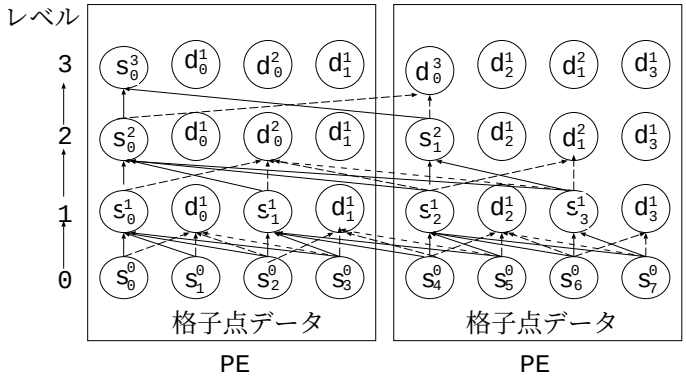

図 5.2: 1 次元の iDWT の並列化 2

第 6 章

謝辞

本研究を進めるにあたり、貴重な御助言、御指導を賜わりました松澤照男教授に深く感謝致します。最後に 2 年間お世話になった研究室の皆様、その他御迷惑をお掛けした皆様に深く感謝致します。

参考文献

[1] 田中伸厚, 寺坂晴夫 不完全ウェーブレット変換とその２次元ポアソン方程式解法第8回数値流体力学シンポジウム講演論文集, pp479-482, 1994

[2] 田中伸厚, 清水武司, 瀧川 幸夫 ウェーブレットを前処理に用いた共役勾配法の並列処理第8回計算力学会講演会講演論文集, pp295-296, 1995

[3] N.TANAKA, H.TERASAKA, T.SHIMIZU, Y.TAKIGAWA Imcomplete Discrete Wavelet Transform and Its Application to Poisson Equation Solver Journal of NUCLEAR SCIENCE and TECHNOLOGY, Vol.33,No.7,pp555-561,1996

[4] 田中伸厚 不完全離散ウェーブレット変換のポアソン方程式解法への応用と並列処理 日本応用数理学会論文誌, Vol.7,No.4,pp373-388,1997

[5] I.Daubechies Orthonomal Bases of Compactly Supported Wavelets Comm.Pure Appl.Math.,41,pp909-996,1998

[6] G.Beylkin On wavelet based algorithms for solving differential equations ウェーブレット 理論と応用, Springer-verlag Tokyo,pp450-465,1995

[7] S.Jafford Wavelet methods for fast resolution of elliptic problems SIAM J. Numer.Anal.,29-4,pp965-986,1992

[8] A.Averbuch, G.Beylkin, R.Coifman, M.Israeli Multiscale Inversion of Elliptic Operators Signal and Image Representation in Combined Spaces, Academic Press,inc,pp1-16,1995

[9] 榊原進 ウェーブレットビギナーズガイド 東京電機大学出版, 1995