

波動方程式に対する差分法

桂田 祐史

2000年6月15日, 2008年5月3日, 2014年5月27日

目次

第1章	卒業研究における波動方程式の研究	5
1.1	円盤領域、楕円領域における波動方程式の差分近似	5
1.2	波の性質を調べる	5
1.2.1	ホイヘンスの原理	5
1.2.2	楕円形の酒場	6
1.3	Javaによるシミュレーションプログラム	6
1.4	Friedrichsの差分法	6
第2章	1次元波動方程式の初期値境界値問題の数値実験	8
2.1	波動方程式 — 簡単な説明	8
2.2	差分法方程式 (Dirichlet 境界条件の場合)	9
2.3	差分法方程式 (Neumann 境界条件の場合)	11
2.3.1	1次近似	11
2.3.2	2次近似	11
2.4	実験してみよう	12
2.5	wave1d-glsc.c	13
2.6	wave1n-glsc.c	18
2.7	wave1d.BAS	23
2.8	wave1n.BAS	26
2.9	misc.	28
2.9.1	コンパクトな台を持つ滑らかな関数 (プログラム中の <code>h_ab()</code>)	28
2.9.2	コンパクトな台を持つ C^2 級の関数の簡単な例	30
2.10	その他	31
第3章	波動方程式に対する差分法の解析	32
3.1	Galerkin 法による微分方程式の解法	32
3.1.1	熱方程式の場合	32
3.1.2	波動方程式の場合	34
3.2	波動方程式の初期値問題の差分解	34

3.2.1	$e^{i\alpha x}$ が固有関数であること	34
3.2.2	差分方程式の形式解の導出	35
3.2.3	形式解が well-defined であり、厳密解に収束すること	39
3.3	波動方程式の初期値境界値問題の差分解	39
3.3.1	初期値境界値問題	40
3.3.2	差分方程式	40
3.3.3	$\sin n\pi x$ が $(\delta_x)^2$ の固有関数であること	41
3.3.4	差分方程式の形式解の導出	41
3.3.5	差分解の存在	43
3.3.6	いくつかの補題	43
3.3.7	差分解の厳密解への収束	47
3.4	付録 1: 線形差分方程式	47
3.5	付録 2: 三角関数についてのメモ	49
第 4 章	波動方程式に対する差分法 — 4 年 1 6 組 6 8 番 藤沼 祐一 (1999 年 3 月 16 日)	52
4.1	はじめに	52
4.2	問題	52
4.2.1	Dirichlet 境界条件の問題	52
4.2.2	新しい関数を用いた解法	53
4.2.3	まとめて	56
4.3	境界条件が変わると	57
4.3.1	Neumann 境界条件での解法	57
4.3.2	まとめて	60
4.4	エネルギーの保存則について	61
4.4.1	エネルギーの定義	61
4.4.2	λ の値を変えた解析	63
4.4.3	数値解析 (N の値を変えて)	74
4.5	安定性	78
4.5.1	λ の値と安定性	78
4.6	まとめ	78
4.7	プログラム	79
4.8	参考文献	84
第 5 章	桂田メモ	85
5.1	対称双曲型方程式の問題への変換	85

5.1.1	対称双曲型微分方程式への変換	85
5.1.2	初期条件	85
5.1.3	境界条件	85
5.2	Friedrichs の差分法	87
5.3	エネルギー保存則	89
第 6 章	2次元長方形領域における波動方程式の初期値境界値問題	91
6.1	Dirichlet 境界条件	91
6.1.1	差分法のプログラム	91
6.2	Neumann 境界条件	99
6.2.1	差分法のプログラム	100
6.3	misc.	106
6.3.1	CFL 条件	106
第 7 章	3次元直方体領域における波動方程式の初期値境界値問題	108
7.1	コメント	108
第 8 章	円盤領域における波動方程式の初期値境界値問題	110
8.1	Dirichlet 境界値問題の厳密解	110
8.1.1	$-\Delta$ の固有値問題	110
8.1.2	変数分離解	112
付 録 A	物理メモ	114
A.1	1次元波動方程式	114
A.1.1	管の中の気体の振動	114
A.1.2	弾性体の棒の縦振動	114
A.2	2次元波動方程式	115
A.2.1	膜の振動	115
A.3	棒の振動	115
A.4	板の振動	116
A.5	水の波	116
A.6	Schrödinger の波動方程式	116

Changelog

- (2012年12月10日) 久しぶりの新作。十進BASIC用のプログラム `wave1d.BAS`, `wave1n.BAS` を用意した。

第1章 卒業研究における波動方程式の研究

1.1 円盤領域、楕円領域における波動方程式の差分近似

1996年度卒研の松本英久 [1] において、円盤領域における熱方程式について詳しい研究がなされた。それが契機になったと思われるが、1997年度卒研で養田孝 [2] は、円盤領域における波動方程式の差分法による数値シミュレーションに取り組んだ。極座標変換を用いて、長方形領域における微分方程式に直してから、差分法により離散化したものであるが、安定性の問題が生じて、まともにシミュレートするのが難しい。それに対して陰解法などを考えている。

2006年度金子裕司 [3], 2007年度久保田祥史 [4] で、円盤領域における熱方程式に対する Shortley-Weller 近似が取り上げられたが、波動方程式に対してこれを適用するとどうなるか興味深い。Shortley-Weller 近似を用いる場合、円盤以外の領域も簡単に扱えるので、例えば「楕円形の酒場」のシミュレーションなどにチャレンジ出来る可能性が出て来た。

随分と間が空いて、2011年度 濱勇樹 [5] で、それが実行された。プログラムが大分クリーンになった。楕円体でのシミュレーションをやってみたくなった (その理由は後述)。

1.2 波の性質を調べる

1.2.1 ホイヘンスの原理

2001年度卒研で、二人の学生がホイヘンスの原理を取り上げた。ホイヘンスの原理は、3以上の奇数の次元における波動方程式についてのみ成り立つことは、有名である。坪井泰洋 [6] は、ネットで探した「証明」を解読することにチャレンジした。関口洋正 [7] は、実際に1,2,3次元で数値実験して、ホイヘンスの原理が成り立つことを確めた。

溝畑 [8] にある n 次元波動方程式の基本解の話をきちんと咀嚼して、この問題に対する完璧なレポートを誰かが書いてくれることを望んでいる。

1.2.2 楕円形の酒場

元ネタをどこで知ったのか忘れてしまったが、シャーロックホームズの小説にあるということなのだが、どの小説か分かっていない。

ウィキペディアにも「ささやきの回廊 (whispering gallery)」という項目がある。曰く「人の囁き声が距離の離れたところで聞こえる建築物、またその現象自体を指す」とか。楕円形ではないけれど、現実にあるということで調べてみるのは面白そうだ。

2004 年度卒研でも中西謙太君に勧めてみた。(そのときは変数変換を使うという方針で、桂田 [9] というメモがある)、歯が立たず、円盤の場合に戻ってレポート [10] を書いてもらった。

2011 年度 濱勇樹 [5] 君の卒研で、Shortley-Weller 近似を用いて、楕円領域での数値シミュレーションが行なわれた。その結果を見るとあまりはつきりしない。2次元ではホイヘンスの原理が成り立たないからかもしれない。3次元で「楕円体の酒場」をやらないといけないのかもしれない。

1.3 Java によるシミュレーションプログラム

熱方程式に比べると、波動方程式の計算は「軽め」なので、Java で十分なスピードのプログラムを書くことが出来る。

2002 年度卒研の三井康之 [11], 2005 年度卒研の伊藤秀範 [12] などのレポートがある。これらに含まれるプログラムは力作であるが、もっとうまいデモ問題を用意することが良い卒研の課題になりそうな気がしている。

なお、波動方程式の数値計算そのものを卒研テーマにしたわけではないが、可視化の工夫 (3次元グラフィックス) について論じた 2007 年度卒研中村圭佑 [13] も読むに値する。

1.4 Friedrichs の差分法

1998 年度卒研で、藤沼祐一 [14] は、Friedrichs の差分法のエネルギー保存について、小さいながらも鮮やかな定理 ($\lambda = 1$ ならばエネルギーは保存される) を得た。

$\lambda < 1$ の場合は、数値実験的にはエネルギーは減少する。そのことの証明と二次元への一般化が出来れば嬉しいと思って、2001 年度卒研で伊藤雄一 [15] というチャレンジがあったが、成功には至っていない。

第2章 1次元波動方程式の初期値境界値問題の数値実験

(明治大学数学科計算機室のWS、またはCygwinをインストールした、貸し出し用ノートパソコンを用いた数値実験をするためのメモ)

<http://nalab.mind.meiji.ac.jp/~mk/program/>

から wave1d-glsc.c を入手してください。

```
oyabun% ccmg wave1d-glsc.c
```

でコンパイルできるはずですが。

2.1 波動方程式 — 簡単な説明

2つの独立変数 t, x についての関数 $u = u(x, t)$ についての方程式

$$(2.1) \quad \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \quad (x \in (0, L), t > 0)$$

は1次元波動方程式 (wave equation) と呼ばれます¹(ここで c は与えられた正の定数です)。それは、この方程式が、一様な弦の微小振動や、細い管の中の空気の振動などの、1次元的な振動・波動現象を表わすものであると解釈出来るからです。弦の振動の場合は $u(x, t)$ は時刻 t における、弦上の点 x の釣り合いの位置からの変位を表わします。

実は定数 c は波の伝播の速さになりますが、以下では時刻の単位を適当に取り替える (数学的には、 ct を新たに t とする変数変換を行う) ことによって、 $c = 1$ であるとして扱うことにしても一般性は失いません²。同時に空間方向についても同様の変数変換を施すことによって、 $L = 1$ と仮定することも出来ます。

¹熱方程式の場合と同様に、波動方程式にも空間2次元や3次元のものが考えられます。

²このような x, t の単位の変換 (数学的には変数の1次変換) は「スケーリング」と呼ばれ、かなり重要な話題なのですが、ここでは注意するだけに留めておきます。

この方程式は、時刻 $t = 0$ での各部分の変位と「速度」を指定することに相当する初期条件

$$(2.2) \quad u(x, 0) = \phi(x) \quad (0 \leq x \leq L)$$

$$(2.3) \quad \frac{\partial u}{\partial t}(x, 0) = \psi(x) \quad (0 \leq x \leq L)$$

と (ここで ϕ と ψ は与えられた関数です)、各時刻での弦の両端の状態を指定する境界条件を課すことにより、解 u が一意に決定される問題となります。境界条件としては、両端が固定されていて (管の中の空気の振動の場合では「端が閉じられていて」) 変位が常に 0 であるという

$$(2.4) \quad u(0, t) = u(L, t) = 0 \quad (t > 0),$$

あるいは、両端で自由に動ける (管の中の空気の振動の場合では「端が開放されている」) という

$$(2.5) \quad \frac{\partial u}{\partial x}(0, t) = \frac{\partial u}{\partial x}(L, t) = 0 \quad (t > 0)$$

を考えることにしましょう。熱伝導方程式の場合と同様に、(2.4) を Dirichlet 境界条件、(2.5) を Neumann 境界条件と呼びます。

2.2 差分法方程式 (Dirichlet 境界条件の場合)

計算手続きの基本的な考え方は熱方程式の場合と同様です。

「空間変数」 x については、区間 $[0, L]$ を N 等分します：

$$h = \frac{L}{N}, \quad x_i = ih \quad (i = 0, 1, 2, \dots, N).$$

「時間変数」 t については、刻み幅 (間隔) を $\tau > 0$ としましょう。 t_n を

$$t_n = n\tau \quad (n = 0, 1, \dots)$$

で定めます。

方程式 (2.1) に現れる 2 つの微分、 t に関する 2 階偏微分 $\frac{\partial u}{\partial t}(x, t)$ と x に関する 2 階偏微分 $\frac{\partial^2 u}{\partial x^2}(x, t)$ の双方を、ともに「2 階中心差分商」で近似すると次の近似方程式が得られます：

$$\frac{1}{c^2} \frac{u(x, t + \tau) - 2u(x, t) + u(x, t - \tau)}{\tau^2} = \frac{u(x + h, t) - 2u(x, t) + u(x - h, t)}{h^2}.$$

そこで格子点上 (x_i, t_n) での u の近似値 U_i^n を決定する方程式としては次のものが考えられます。

$$\frac{1}{c^2} \frac{U_i^{n+1} - 2U_i^n + U_i^{n-1}}{\tau^2} = \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{h^2} \quad (i = 1, 2, \dots, N-1; n = 1, 2, 3, \dots),$$

これを变形すると

(2.6)

$$U_i^{n+1} = 2(1 - \lambda^2)U_i^n + \lambda^2(U_{i-1}^n + U_{i+1}^n) - U_i^{n-1} \quad (i = 1, 2, \dots, N-1; n = 1, 2, 3, \dots)$$

となります。ただし $\lambda = c\tau/h$ と置きました。

一方 (2.2) からは、ごく自然に

$$(2.7) \quad U_i^0 = \phi(x_i) \quad (i = 0, 1, 2, \dots, N)$$

が得られます。一方 (2.3) からは、例えば (他にも色々なやり方が考えられますが)

$$(2.8) \quad U_i^1 = (1 - \lambda^2)\phi(x_i) + \frac{\lambda^2}{2}(\phi(x_{i-1}) + \phi(x_{i+1})) + \tau\psi(x_i) \quad (1 \leq i \leq N-1)$$

が得られます³。(2.4) からは

$$(2.9) \quad U_0^n = U_N^n = 0 \quad (n = 0, 1, 2, \dots),$$

数列 $\{U_i^n\}$ に関する方程式 (2.6), (2.7), (2.8), (2.9) は二つの添字 i, n を含んでいますが、(2.6) を漸化式として、時刻に関する方の添字 n の小さい方から順に計算していくことが出来ます。熱方程式の場合は、二番目の添字のところには、 $n, n+1$ しか現れませんでした。が、(2.6) では $n-1, n, n+1$ と3つのものが現れています。 $n+1$ での値を求めるために、一段前の n での値のみならず、もう一段前の $n-1$ での値が必要になっているわけです。このことは、もとの方程式が時刻 t に関して2階であることに対応しています。そのため計算を出発させるためには、 $n=0$ での値だけでなく、 $n=1$ での値も必要になりますが、それは時刻に関する1階の微分を指定している初期条件 (2.3) に由来する (2.8) で与えられています。

熱方程式の場合と同様に、 h, τ と性質の異なる刻み幅が2つあります。前回と同様に、安定に計算するためには両者を全く勝手なやり方で0に持っていくだけでは不十分です。どうすればいいか、とりあえず結論だけ述べておくと「 λ を $0 < \lambda \leq 1$ と選んで固定したまま、 $h, \tau \rightarrow 0$ 」でOKです。

³(ちょっと難しいですが) $t \mapsto u(x, t)$ の、 $t=0$ における展開 $u(x, \tau) \doteq u(x, 0) + \tau \frac{\partial u}{\partial t}(x, 0) + \frac{\tau^2}{2} \frac{\partial^2 u}{\partial t^2}(x, 0)$ において、関係式 $\frac{1}{c^2} \frac{\partial^2 u}{\partial t^2}(x, 0) = \frac{\partial^2 u}{\partial x^2}(x, 0) = \phi''(x)$ が成立するとして良いだろう。すると $u(x, \tau) \doteq \phi(x) + \tau\psi(x) + \frac{c^2\tau^2}{2}\phi''(x)$ 。この $\phi''(x)$ を2階中心差分商で近似すると、(2.8) が得られます。

2.3 差分法方程式 (Neumann 境界条件の場合)

(2.5) の近似については、複数の近似法が考えられます。

2.3.1 1次近似

一つは、 $u_x(0, t)$ については前進差分近似、 $u_x(L, t)$ については後退差分近似を
すると考えて

$$(2.10) \quad U_0^{n+1} = U_1^{n+1}, \quad U_N^{n+1} = U_{N-1}^{n+1} \quad (n = 0, 1, 2, \dots)$$

とするもの。

wave1d-gls.c を Neumann 境界値問題のプログラムに直す

76 行目 (`u2[0] = u2[N] = 0;`) を次のように書き換える。

```
u2[0] = u2[1]; u2[N] = u2[N-1];
```

127 行目 (`u3[0] = u3[N] = 0;`) を次のように書き換える。

```
u3[0] = u3[1]; u3[N] = u3[N-1];
```

2.3.2 2次近似

もう一つは、仮想格子点 x_{-1}, x_{N+1} を導入して、 $u_x(0, t)$ と $u_x(L, t)$ をともに1
階中心差分近似をするというもので、

$$(2.11) \quad U_{-1}^{n+1} = U_1^{n+1}, \quad U_{N+1}^{n+1} = U_{N-1}^{n+1} \quad (n = 0, 1, 2, \dots)$$

が得られます。 $i = 0, N$ についても (2.6) が成立するとして、

$$U_0^{n+1} = 2(1 - \lambda^2)U_0^n + \lambda^2(U_{-1}^n + U_{+1}^n) - U_0^{n-1} = 2(1 - \lambda^2)U_0^n + 2\lambda^2U_{+1}^n - U_0^{n-1},$$

$$U_N^{n+1} = 2(1 - \lambda^2)U_N^n + \lambda^2(U_{N-1}^n + U_{N+1}^n) - U_N^{n-1} = 2(1 - \lambda^2)U_N^n + 2\lambda^2U_{N-1}^n - U_N^{n-1} \\ (n = 1, 2, \dots)$$

という方程式が得られます。

U_0^1, U_N^1 については、(2.8) が、 $i = 0, i = N$ でも成立するとして、

$$U_0^1 = (1 - \lambda^2)\phi(x_0) + \frac{\lambda^2}{2}(\phi(x_{-1}) + \phi(x_1)) + \tau\psi(x_0),$$
$$U_N^1 = (1 - \lambda^2)\phi(x_N) + \frac{\lambda^2}{2}(\phi(x_{N-1}) + \phi(x_{N+1})) + \tau\psi(x_N).$$

ϕ が $[0, L]$ の外では定義されていない場合に、 $\phi(x_{-1}), \phi(x_{N+1})$ をどうするか、若干問題は残るが(目をつむって、それぞれ $\phi(x_1), \phi(x_{N-1})$ に等しいとするのかもしれない? 初期条件と境界条件が整合している $\phi'(x_0) = \phi'(x_N) = 0$ の場合であれば納得できるだろうか)。

2.4 実験してみよう

熱方程式は(時間が経つと温度が場所に依らず一様になるという)現象としては少々地味なものでしたが、波動方程式は工夫次第で色々な現象が観察できてなかなか面白いものです。

今回例題プログラムとして用意した `wave1d-glsc.c` を元に色々実験してみてください。

- λ を変えて実験してみる。 $\lambda \leq 1$ が安定性の条件で、精度を考えると $\lambda = 1$ が良い、ということになっているのだが。
- 初速が 0 でないもの。
- Neumann 境界条件にしてみる。端での波の反射の状況はどう変わるか?
- 波の速さは本当に c か? (サンプル・プログラムの場合 1 か?)

上の二つの例では、いずれも $\psi = 0$ としてあります。これは時刻 $t = 0$ で弦をそっと放した(=速度が 0)場合で、波の山が左右に二つに割れて、それぞれ同じ速さで反対向きに進んでいくことになりますが、 g を適当に設定すると、「山」の動きをコントロール出来ます。

注意 波と波がぶつかると、一見「はね返る」ようにも感じられますが、それは間違いで、むしろ「通り抜ける」と見る方が正しいことに注意して下さい。これは異なる高さの波が衝突する実験を試みれば納得できるでしょう。

2.5 wave1d-glsc.c

```
1 /*
2  * wave1d-glsc.c --- 1次元波動方程式 (Dirichlet 境界条件)
3  *
4  *   ccmg wave1d-glsc.c
5  * ノートパソコンでは
6  *   ccmg wave1d-glsc.c
7  *
8  *   入力を目安:
9  *     n は適度に大きめに (500 とか 1000)
10 *     λ は 1 (安定性のために ≤ 1, 精度のために =1 が望ましいので)
11 *     Tmax は適当に (10 とか 100 とか 1000 とか)
12 *
13 *   アルゴリズムについては以下を参照せよ
14 *     http://www.math.meiji.ac.jp/~mk/labo/text/wave.pdf
15 */
16
17 #include <stdio.h>
18 #include <math.h>
19 #define G_DOUBLE
20 #include <glsc.h>
21 #include <matrix.h>
22
23 void copy_vector(int, vector, vector);
24 int msleep(int);
25
26 double pi;
27
28 int main()
29 {
30     /* N は区間の分割数 */
31     int N;
32     int i, n, nmax, nfunc;
33     /* lambda は λ, tau は τ */
34     double lambda, Tmax, h, tau, lambda2;
35     /* u1[i]=u_{i,j-1}
36      * u2[i]=u_{i,j}
37      * u3[i]=u_{i,j+1}
38      */
39     vector u1, u2, u3;
40     double phi(double, int), psi(double, int);
41     double win_width, win_height, w_margin, h_margin;
42     char message[100];
43
44     pi = 4.0 * atan(1.0);
45
46     /* n は 100 とか 1000 とか */
47     printf("N=");
```

```

48  scanf("%d", &N);
49  u1 = new_vector(N + 1);
50  u2 = new_vector(N + 1);
51  u3 = new_vector(N + 1);
52
53  /* λについては本に詳しいが、λ=1 が良い結果を出す */
54  printf("λ=");
55  scanf("%lf", &lambda);
56  /* Tmax は最終時刻だから、長く計算したければ大きな値を入れる */
57  printf("Tmax=");
58  scanf("%lf", &Tmax);
59
60  lambda2 = lambda * lambda;
61  h = 1.0 / N;
62  tau = lambda * h;
63
64  /* 初期値を選ぶ */
65  printf("1: 定常波, 2: 割れる山, 3: 滑かな山, 4: ホイヘンスの原理反例\n");
66  printf("nfunc(初期値の種類 1..5)=");
67  scanf("%d", &nfunc);
68  /* 初期値 */
69  /* u_{i,0}=φ i */
70  for (i = 0; i <= N; i++)
71      u1[i] = phi(i * h, nfunc);
72  /* u_{i,1}=(1-λ 2)... */
73  for (i = 1; i < N; i++)
74      u2[i] = (1 - lambda2) * u1[i] + 0.5 * lambda2 * (u1[i - 1] + u1[i + 1])
75          + tau * psi(i * h, nfunc);
76  u2[0] = u2[N] = 0.0;
77
78  /* ***** グラフィックスの準備 ***** */
79  /* メタファイル名は "WAVE",
80   * ウィンドウのサイズは、
81   * 横 win_width + 2 * w_margin, 縦 win_height + 2 * h_margin */
82  win_width = 200.0; win_height = 200.0; w_margin = 10.0; h_margin = 10.0;
83  g_init("WAVE", win_width + 2 * w_margin, win_height + 2 * h_margin);
84  /* 画面とメタファイルの両方に記録する */
85  g_device(G_BOTH);
86  /* 座標系の定義: [-0.1,1.1] × [-1.1,1.1] という閉領域を表示する */
87  g_def_scale(0,
88             -0.1, 1.1, -1.1, 1.1,
89             w_margin, h_margin, win_width, win_height);
90  /* 線を二種類用意する */
91  g_def_line(0, G_BLACK, 0, G_LINE_SOLID);
92  g_def_line(1, G_BLACK, 0, G_LINE_DOTS);
93  /* 表示するための文字列の属性を定義する */
94  g_def_text(0, G_BLACK, 3);
95  /* 定義したものを選択する */
96  g_sel_scale(0); g_sel_line(0); g_sel_text(0);

```

```

97
98  /* タイトルと入力パラメータを表示する */
99  g_text(30.0, 30.0,
100         "wave equation, homogeneous Dirichlet boundary condition");
101  sprintf(message, "N=%d, lambda=%g, Tmax=%g", N, lambda, Tmax);
102  g_text(30.0, 60.0, message);
103
104  /* 座標軸を表示する */
105  g_sel_line(1);
106  g_move(-0.1, 0.0); g_plot(1.1, 0.0);
107  g_move(0.0, -0.1); g_plot(0.0, 1.1);
108  g_sel_line(0);
109
110  /* t=0 でのグラフ */
111  g_move(0.0, u1[0]);
112  for (i = 1; i <= N; i++)
113      g_plot(i * h, u1[i]);
114
115  /* t=t1=τ でのグラフ */
116  g_cls();
117  g_move(0.0, u2[0]);
118  for (i = 1; i <= N; i++)
119      g_plot(i * h, u2[i]);
120
121  nmax = rint(Tmax / tau);
122  /* 時間に関するループ */
123  for (n = 1; n <= nmax; n++) {
124      for (i = 1; i < N; i++)
125          u3[i] = 2 * (1.0 - lambda2) * u2[i] + lambda2 * (u2[i - 1] + u2[i + 1])
126  - u1[i];
127      u3[0] = u3[N] = 0.0;
128
129  #ifdef USESLEEP
130      /* 5 倍の秒数待つ */
131      msleep(5 * (int) (tau * 1000));
132  #endif
133      /* t=t_{n+1}=(n+1) τ でのグラフ */
134      g_cls();
135      g_move(0.0, u3[0]);
136      for (i = 1; i <= N; i++)
137          g_plot(i * h, u3[i]);
138
139      /* u1 <- u2, u2 <- u3 */
140      copy_vector(N + 1, u1, u2);
141      copy_vector(N + 1, u2, u3);
142  }
143  printf("終わりました。X の場合はウィンドウをクリックして下さい。 \n");
144  g_sleep(-1.0);
145  /* ウィンドウを閉じる */

```



```

146     g_term();
147     return 0;
148 }
149
150 /* 初期値 (nfunc=3,4) を作るための道具 */
151 double sqr(double x)
152 {
153     return x * x;
154 }
155
156 double mountain(double x)
157 {
158     double a = 0.4, b = 0.6;
159     if (x <= a || x >= b)
160         return 0;
161     else {
162         double c = sqr(sqr(2 / (b - a)));
163         return c * sqr((x - a) * (x - b));
164     }
165 }
166
167 /* 初期値 (nfunc=5) を作るための道具 */
168
169 /* これは初期値ではない! */
170 double f(double x)
171 {
172     if (x <= 0)
173         return 0;
174     else
175         return exp(- 1.0 / x);
176 }
177
178 /* これは初期値ではない! */
179 double g(double x)
180 {
181     return f(x) / (f(x) + f(1 - x));
182 }
183
184 /*
185 *           1   (|x|<a)
186 *  h_ab(x) ={
187 *           0   (|x|>b)
188 *
189 */
190 double h_ab(double x, double a, double b)
191 {
192     return g((x + a) / (a - b)) * g((- x + a) / (a - b));
193 }
194

```

```

195 /* 初期値  $\phi = u(\cdot, 0)$  */
196 double phi(double x, int nfunc)
197 {
198     switch (nfunc) {
199     case 1:
200         return sin(pi * x);
201         break;
202     case 2:
203         if (x > 0.375 && x <= 0.5)
204             return 8.0 * x - 3.0;
205         else if (x > 0.5 && x < 0.625)
206             return -8.0 * x + 5.0;
207         else
208             return 0.0;
209         break;
210     case 3:
211         return mountain(x);
212         break;
213     case 4:
214         return mountain(x);
215         break;
216     case 5:
217         {
218             double a = 0.3, b = 0.2;
219             return h_ab(x - 0.5, a, b);
220         }
221         break;
222     default:
223         return 0.0;
224     }
225 }
226
227 /* 初期値  $\psi = ut(\cdot, 0)$  */
228 double psi(double x, int nfunc)
229 {
230     if (nfunc == 1)
231         return 0.0;
232     else if (nfunc == 2)
233         return 0.0;
234     else if (nfunc == 3)
235         return 0.0;
236     else if (nfunc == 4)
237         return 5 * mountain(x);
238     else if (nfunc == 5)
239         return 0.0;
240     else
241         return 0.0;
242 }
243

```

```

244 /*
245  * ベクトル b をベクトル a にコピー
246  */
247 void copy_vector(int N, vector a, vector b)
248 {
249     int i;
250     for (i = 0; i < N; i++)
251         a[i] = b[i];
252     return;
253 }

```

2.6 wave1n-glsc.c

```

/*
 * wave1n-glsc.c --- 1次元波動方程式 (Neumann 境界条件)
 *
 * ccmg wave1n-glsc.c
 * ノートパソコンでは
 * ccmg wave1n-glsc.c
 *
 * 入力目安:
 *   n は適度に大きめに (500 とか 1000)
 *   λ は 1 (安定性のために ≤ 1, 精度のために =1 が望ましいので)
 *   Tmax は適当に (10 とか 100 とか 1000 とか)
 *
 * アルゴリズムについては以下を参照せよ
 *   http://www.math.meiji.ac.jp/~mk/labo/text/wave.pdf
 */

#include <stdio.h>
#include <math.h>
#define G_DOUBLE
#include <glsc.h>
#include <matrix.h>

void copy_vector(int, vector, vector);
int msleep(int);

double pi;

int main()
{
    /* N は区間の分割数 */
    int N;
    int i, n, nmax, nfunc;
    /* lambda は λ, tau は τ */
    double lambda, Tmax, h, tau, lambda2;
    /* u1[i]=u_{i,j-1}
     * u2[i]=u_{i,j}

```

```

    * u3[i]=u_{i,j+1}
    */
vector u1, u2, u3;
double phi(double, int), psi(double, int);
double win_width, win_height, w_margin, h_margin;
char message[100];

pi = 4.0 * atan(1.0);

/* n は 100 とか 1000 とか */
printf("N=");
scanf("%d", &N);
u1 = new_vector(N + 1);
u2 = new_vector(N + 1);
u3 = new_vector(N + 1);

/* λについては本に詳しいが、λ=1 が良い結果を出す */
printf("λ=");
scanf("%lf", &lambda);
/* Tmax は最終時刻だから、長く計算したければ大きな値を入れる */
printf("Tmax=");
scanf("%lf", &Tmax);

lambda2 = lambda * lambda;
h = 1.0 / N;
tau = lambda * h;

/* 初期値を選ぶ */
printf("1: 定常波, 2: 割れる山, 3: 滑かな山, 4: ホイヘンスの原理反例\n");
printf("nfunc(初期値の種類 1..5)=");
scanf("%d", &nfunc);
/* 初期値 */
/* u_{i,0}=φ i */
for (i = 0; i <= N; i++)
    u1[i] = phi(i * h, nfunc);
/* u_{i,1}=(1-λ 2)... */
for (i = 1; i < N; i++)
    u2[i] = (1 - lambda2) * u1[i]
            + 0.5 * lambda2 * (u1[i - 1] + u1[i + 1]) + tau * psi(i * h, nfunc);
u2[0] = (1 - lambda2) * u1[0] + lambda2 * u1[1] + tau * psi(0.0, nfunc);
u2[N] = (1 - lambda2) * u1[N] + lambda2 * u1[N - 1] + tau * psi(1.0, nfunc);

/* ***** グラフィックスの準備 ***** */
/* メタファイル名は "WAVE",
 * ウィンドウのサイズは、
 * 横 win_width + 2 * w_margin, 縦 win_height + 2 * h_margin */
win_width = 200.0; win_height = 200.0; w_margin = 10.0; h_margin = 10.0;
g_init("WAVE", win_width + 2 * w_margin, win_height + 2 * h_margin);
/* 画面とメタファイルの両方に記録する */

```

```

g_device(G_BOTH);
/* 座標系の定義: [-0.1,1.1] × [-1.1,1.1] という閉領域を表示する */
g_def_scale(0,
            -0.1, 1.1, -1.1, 1.1,
            w_margin, h_margin, win_width, win_height);
/* 線を二種類用意する */
g_def_line(0, G_BLACK, 0, G_LINE_SOLID);
g_def_line(1, G_BLACK, 0, G_LINE_DOTS);
/* 表示するための文字列の属性を定義する */
g_def_text(0, G_BLACK, 3);
/* 定義したものを選択する */
g_sel_scale(0); g_sel_line(0); g_sel_text(0);

/* タイトルと入力パラメーターを表示する */
g_text(30.0, 30.0,
       "wave equation, homogeneous Dirichlet boundary condition");
sprintf(message, "N=%d, lambda=%g, Tmax=%g", N, lambda, Tmax);
g_text(30.0, 60.0, message);

/* 座標軸を表示する */
g_sel_line(1);
g_move(-0.1, 0.0); g_plot(1.1, 0.0);
g_move(0.0, -0.1); g_plot(0.0, 1.1);
g_sel_line(0);

/* t=0 でのグラフ */
g_move(0.0, u1[0]);
for (i = 1; i <= N; i++)
    g_plot(i * h, u1[i]);

/* t=t1=τ でのグラフ */
g_cls();
g_move(0.0, u2[0]);
for (i = 1; i <= N; i++)
    g_plot(i * h, u2[i]);

nmax = rint(Tmax / tau);
/* 時間に関するループ */
for (n = 1; n <= nmax; n++) {
    for (i = 1; i < N; i++)
        u3[i] = 2 * (1.0 - lambda2) * u2[i]
+ lambda2 * (u2[i - 1] + u2[i + 1]) - u1[i];
    u3[0] = 2 * (1.0 - lambda2) * u2[0] + 2 * lambda2 * u2[1] - u1[0];
    u3[N] = 2 * (1.0 - lambda2) * u2[N] + 2 * lambda2 * u2[N - 1] - u1[N];
}

#ifdef USESLEEP
    /* 5 倍の秒数待つ */
    msleep(5 * (int) (tau * 1000));
#endif

```

```

    /* t=t_{n+1}=(n+1)  $\tau$  でのグラフ */
    g_cls();
    g_move(0.0, u3[0]);
    for (i = 1; i <= N; i++)
        g_plot(i * h, u3[i]);

    /* u1 <- u2, u2 <- u3 */
    copy_vector(N + 1, u1, u2);
    copy_vector(N + 1, u2, u3);
}
printf("終わりました。X の場合はウィンドウをクリックして下さい。 \n");
g_sleep(-1.0);
/* ウィンドウを閉じる */
g_term();
return 0;
}

/* 初期値 (nfunc=3,4) を作るための道具 */
double sqr(double x)
{
    return x * x;
}

double mountain(double x)
{
    double a = 0.4, b = 0.6;
    if (x <= a || x >= b)
        return 0;
    else {
        double c = sqr(sqr(2 / (b - a)));
        return c * sqr((x - a) * (x - b));
    }
}

/* 初期値 (nfunc=5) を作るための道具 */

/* これは初期値ではない! */
double f(double x)
{
    if (x <= 0)
        return 0;
    else
        return exp(- 1.0 / x);
}

/* これは初期値ではない! */
double g(double x)
{
    return f(x) / (f(x) + f(1 - x));
}

```

```

}

/*
 *          1   (|x|<a)
 * h_ab(x) ={
 *          0   (|x|>b)
 *
 */
double h_ab(double x, double a, double b)
{
    return g((x + a) / (a - b)) * g((- x + a) / (a - b));
}

/* 初期値  $\phi=u(\cdot,0)$  */
double phi(double x, int nfunc)
{
    if (nfunc == 1)
        return sin(pi * x);
    else if (nfunc == 2) {
        if (x > 0.375 && x <= 0.5)
            return 8.0 * x - 3.0;
        else if (x > 0.5 && x < 0.625)
            return -8.0 * x + 5.0;
        else
            return 0.0;
    }
    else if (nfunc == 3)
        return mountain(x);
    else if (nfunc == 4)
        return mountain(x);
    else if (nfunc == 5) {
        double a = 0.3, b = 0.2;
        return h_ab(x - 0.5, a, b);
    }
    else
        return 0.0;
}

/* 初期値  $\psi=ut(\cdot,0)$  */
double psi(double x, int nfunc)
{
    if (nfunc == 1)
        return 0.0;
    else if (nfunc == 2)
        return 0.0;
    else if (nfunc == 3)
        return 0.0;
    else if (nfunc == 4)
        return 5 * mountain(x);
}

```

```

    else if (nfunc == 5)
        return 0.0;
    else
        return 0.0;
}

/*
 * ベクトル b をベクトル a にコピー
 */
void copy_vector(int N, vector a, vector b)
{
    int i;
    for (i = 0; i < N; i++)
        a[i] = b[i];
    return;
}

```

2.7 wave1d.BAS

```

REM wave1d.bas --- 1次元波動方程式の初期値境界値問題, Dirichlet 境界条件
REM u_t(x,t)=u_xx(x,t)
REM u(0,t)=u(1,t)=0
REM u(x,0)= $\phi(x)$ 
REM u_t(x,0)= $\psi(x)$ 
REM ----- 準備 -----
REM 滑らかな山 (0 に山頂、標高 1, 幅 1)
FUNCTION f(x)
    IF ABS(x)<= 1 THEN
        LET f=(x*x-1)^4
    ELSE
        LET f=0
    END IF
END FUNCTION
REM f(x) の導関数
FUNCTION df(x)
    IF ABS(x)<=1 THEN
        LET df=8*x*(x*x-1)^3
    ELSE
        LET df=0
    END IF
END FUNCTION
REM a にある標高 1, 幅 r の山
FUNCTION p(x,a,r)
    LET p=f((x-a)/r)
END FUNCTION
FUNCTION dp(x,a,r)
    LET dp=df((x-a)/r)/r

```



```

END FUNCTION
REM ----- 初期値 -----
REM 初期値  $\phi(x)=u(x,0)$ 
FUNCTION phi(x,n)
  SELECT CASE n
  CASE 0
    LET phi=SIN(PI*x)
  CASE 1
    LET phi=SIN(3*PI*x)
  CASE 2
    LET phi=p(x, 0.5, 0.2)
  CASE 3
    LET phi=p(x, 0.5, 0.2)
  CASE 4
    LET phi=0.8*p(x, 0.25, 0.15)+0.4*p(x, 0.7, 0.1)
  END SELECT
END FUNCTION
REM 初期値  $\psi(x)=u_t(x,0)$ 
FUNCTION psi(x,n)
  SELECT CASE n
  CASE 0
    LET psi=0
  CASE 1
    LET psi=1
  CASE 2
    LET psi=0
  CASE 3
    LET psi=-c*dp(x,0.5,0.2)
  CASE 4
    LET psi=c*(-0.8*dp(x, 0.25, 0.15)+0.4*dp(x, 0.7, 0.1))
  END SELECT
END FUNCTION
REM ----- ここからがメイン -----
DECLARE EXTERNAL SUB DRAW
LET C=2
INPUT PROMPT "N(0以下はお任せモード): ": N
IF n<=0 THEN
  LET n=1000
  LET lambda=1
  LET tmax=10
  LET dt=0.01
ELSE
  INPUT PROMPT " $\lambda$  ": lambda
  INPUT PROMPT "Tmax ": Tmax
  INPUT PROMPT " $\Delta t$  ": dt
END IF
DIM u(0 TO n),up1(0 TO n),up2(0 TO n)
LET h=1/N
LET tau=lambda*h/c

```

```

LET lambda2=lambda*lambda
LET Nmax=Tmax/tau
LET SKIP=dt/tau
PRINT h,tau,lambda2,Nmax
REM
INPUT PROMPT "0: sin(π x), 1: sin(3 π), 2: 割れる山, 3: 右に進行, 4: ぶつかる": nfunc
FOR i=0 TO n
  LET u(i)=phi(i*h,nfunc)
NEXT i
SET WINDOW -0.2,1.2,-1.2,1.2
PLOT TEXT ,AT 0.3,0.8: "波動方程式"
CALL DRAW(n,u)
FOR i=1 TO n-1
  LET up1(i)=(1-lambda2)*u(i)+0.5*lambda2*(u(i-1)+u(i+1))+tau*psi(i*h,nfunc)
NEXT i
LET up1(0)=0
LET up1(n)=0
CALL DRAW(n,up1)
FOR k=2 TO Nmax
  FOR i=1 TO n-1
    LET up2(i)=2*(1-lambda2)*up1(i)+lambda2*(up1(i-1)+up1(i+1))-u(i)
  NEXT i
  LET up2(0)=0
  LET up2(n)=0
  IF MOD(k,SKIP)=0 THEN
    CALL DRAW(n,up2)
  END IF
  FOR i=0 TO n
    LET u(i)=up1(i)
    LET up1(i)=up2(i)
  next i
NEXT k
END
REM *****
EXTERNAL SUB DRAW(n,u())
REM WAIT delay 0.01
SET DRAW mode hidden
clear
LET h=1/N
FOR i=0 TO n
  PLOT LINES: i*h,u(i);
NEXT i
PLOT LINES
SET DRAW mode explicit
END SUB

```

2.8 wave1n.BAS

```
REM wave1n.bas --- 1次元波動方程式の初期値境界値問題, Neumann 境界条件
REM (1/c^2)u_t(x,t)=u_xx(x,t)
REM u_x(0,t)=u_x(1,t)=0
REM u(x,0)=φ(x)
REM u_t(x,0)=ψ(x)
REM ----- 準備 -----
REM 滑らかな山 (0に山頂、標高1, 幅1)
FUNCTION f(x)
  IF ABS(x)<= 1 THEN
    LET f=(x*x-1)^4
  ELSE
    LET f=0
  END IF
END FUNCTION
REM f(x)の導関数
FUNCTION df(x)
  IF ABS(x)<=1 THEN
    LET df=8*x*(x*x-1)^3
  ELSE
    LET df=0
  END IF
END FUNCTION
REM aにある標高1, 幅rの山
FUNCTION p(x,a,r)
  LET p=f((x-a)/r)
END FUNCTION
FUNCTION dp(x,a,r)
  LET dp=df((x-a)/r)/r
END FUNCTION
REM ----- 初期値 -----
REM 初期値 φ(x)=u(x,0)
FUNCTION phi(x,n)
  SELECT CASE n
  CASE 0
    LET phi=cos(PI*x)
  CASE 1
    LET phi=cos(3*PI*x)
  CASE 2
    LET phi=p(x, 0.5, 0.2)
  CASE 3
    LET phi=0.5*p(x, 0.5, 0.2)
  CASE 4
    LET phi=0.6*p(x, 0.25, 0.15)+0.3*p(x, 0.7, 0.1)
  END SELECT
END FUNCTION
REM 初期値 ψ(x)=u_t(x,0)
FUNCTION psi(x,n)
```

```

SELECT CASE n
CASE 0
  LET psi=0
CASE 1
  LET psi=0
CASE 2
  LET psi=0
CASE 3
  LET psi=-c*0.5*dp(x,0.5,0.2)
CASE 4
  LET psi=c*(-0.6*dp(x, 0.25, 0.15)+0.3*dp(x, 0.7, 0.1))
END SELECT
END FUNCTION
REM ----- ここからがメイン -----
DECLARE EXTERNAL SUB DRAW
LET C=1
INPUT PROMPT "N(0 以下はお任せモード): ": N
IF n<=0 THEN
  LET n=1000
  LET lambda=1
  LET tmax=10
  LET dt=0.01
ELSE
  INPUT PROMPT "λ:": lambda
  INPUT PROMPT "Tmax:": Tmax
  INPUT PROMPT "Δ t:": dt
END IF
DIM u(0 TO n),up1(0 TO n),up2(0 TO n)
LET h=1/N
LET tau=lambda*h/c
LET lambda2=lambda*lambda
LET Nmax=Tmax/tau
LET SKIP=dt/tau
PRINT h,tau,lambda2,Nmax
REM
INPUT PROMPT "0: cos(π x), 1: cos(3 π), 2: 割れる山, 3: 右に進行, 4: ぶつかる": nfunc
FOR i=0 TO n
  LET u(i)=phi(i*h,nfunc)
NEXT i
SET WINDOW -0.2,1.2,-1.2,1.2
PLOT TEXT ,AT 0.3,0.8: "波動方程式"
CALL DRAW(n,u)
FOR i=1 TO n-1
  LET up1(i)=(1-lambda2)*u(i)+0.5*lambda2*(u(i-1)+u(i+1))+tau*psi(i*h,nfunc)
NEXT i
LET up1(0)=(1-lambda2)*u(0)+lambda2*u(1)+tau*psi(0.0, nfunc)
LET up1(n)=(1-lambda2)*u(n)+lambda2*u(n-1)+tau*psi(1.0, nfunc)
CALL DRAW(n,up1)
FOR k=2 TO Nmax

```

```

FOR i=1 TO n-1
  LET up2(i)=2*(1-lambda2)*up1(i)+lambda2*(up1(i-1)+up1(i+1))-u(i)
NEXT i
LET up2(0)=2*(1-lambda2)*up1(0)+2*lambda2*up1(1)-u(0)
LET up2(n)=2*(1-lambda2)*up1(n)+2*lambda2*up1(n-1)-u(n)
IF MOD(k,SKIP)=0 THEN
  CALL DRAW(n,up2)
END IF
FOR i=0 TO n
  LET u(i)=up1(i)
  LET up1(i)=up2(i)
next i
NEXT k
END
REM *****
EXTERNAL SUB DRAW(n,u())
REM WAIT delay 0.01
SET DRAW mode hidden
clear
LET h=1/N
FOR i=0 TO n
  PLOT LINES: i*h,u(i);
NEXT i
PLOT LINES
SET DRAW mode explicit
END SUB

```

2.9 misc.

2.9.1 コンパクトな台を持つ滑らかな関数(プログラム中の h_ab())

コンパクトな台を持つ滑らかな関数の例として用意した。多様体や超関数でおなじみのはずだが、以下の説明は

横田一郎, 多様体とモース理論, 現代数学社 (1978)

から採った。

$$(2.12) \quad f(x) := \begin{cases} 0 & (x \leq 0) \\ \exp\left(-\frac{1}{x}\right) & (x > 0) \end{cases}$$

で定義される $f: \mathbf{R} \rightarrow \mathbf{R}$ は C^∞ -級である。

$$(2.13) \quad g(x) := \frac{f(x)}{f(x) + f(1-x)}$$

とおくと、 g は C^∞ -級で

$$0 \leq g \leq 1, \quad g(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x \geq 1). \end{cases}$$

$a > b > 0$ を満たす任意の a, b に対して、 $h_{a,b}: \mathbf{R} \rightarrow \mathbf{R}$ を

$$(2.14) \quad h_{a,b}(x) := g\left(\frac{x+a}{a-b}\right) g\left(\frac{-x+a}{a-b}\right)$$

で定めると、 $h_{a,b}$ は C^∞ -級で、

$$h_{a,b}(x) = \begin{cases} 1 & (|x| \leq b) \\ 0 & (|x| \geq a) \end{cases}$$

```
/* ./h | graph -b -g 1 | xplot */

#include <math.h>

int main()
{
    double h_ab(double, double, double);
    int i, n;
    double xmin, xmax, x, h;
    n = 100;
    xmin = - 3.0; xmax = 3.0;
    h = (xmax - xmin) / n;
    for (i = 0; i <= n; i++) {
        x = xmin + i * h;
        printf("%f %f\n", x, h_ab(x, 2.0, 1.0));
    }
}

double f(double x)
{
    if (x <= 0)
        return 0;
    else
        return exp(- 1.0 / x);
}

double g(double x)
```

```

{
  return f(x) / (f(x) + f(1 - x));
}

double h_ab(double x, double a, double b)
{
  return g((x + a) / (a - b)) * g((- x + a) / (a - b));
}

gnuplot> plot [-3:3] [-0.5:1.5] "h.data" with lines
gnuplot> set output "h.eps"
gnuplot> set term postscript eps color

```

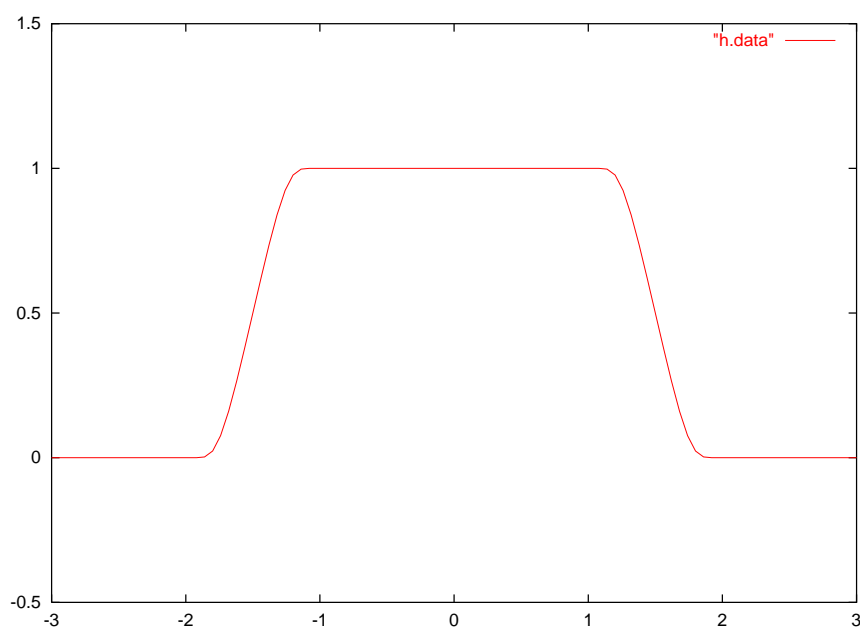


図 2.1: h_{ab} のグラフ ($a = 1, b = 2$ の場合)

2.9.2 コンパクトな台を持つ C^2 級の関数の簡単な例

前項の関数は C^∞ であるが、正直覚えにくい。 C^2 級で満足できる場合には、以下の関数が便利である。

$$f(x) := \begin{cases} (x^2 - 1)^4 & (|x| \leq 1) \\ 0 & (|x| > 1). \end{cases}$$

gnuplot で描く

```
f(x)=(abs(x)<1)? (x**2-1)**4: 0  
plot [-2:2] [-0.1:1.1] f(x)
```

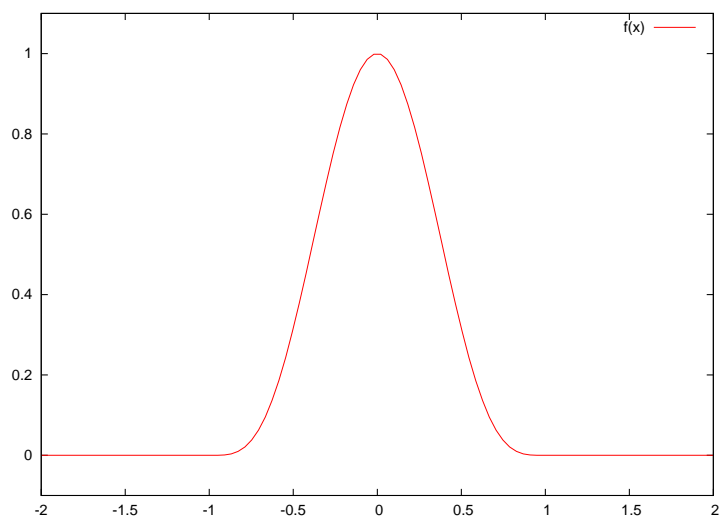


図 2.2: f のグラフ

2.10 その他

2001 年度桂田研の卒業研究で、三井康之君は波動方程式の数値シミュレーション用の Java アプレットを作成した (三井 [11])。次の URL にアクセスすると、試してみることが出来る。

<http://nalab.mind.meiji.ac.jp/~ee88010/>

また 2004 年度の卒業研究で伊藤秀範君はそれを改良した。

<http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2004-itou-prog/>

第3章 波動方程式に対する差分法の解析

3.1 Galerkin 法による微分方程式の解法

3.1.1 熱方程式の場合

Fourier 解析で、次の二つのことを学んだ。

- 周期 2π の任意の (適度に滑らかな) 関数 f は $\{e^{inx}\}_{n \in \mathbf{Z}}$ で表現できる:

$$f(x) = \sum_{n \in \mathbf{Z}} f_n e^{inx} dx, \quad f_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-inx} dx.$$

- 任意の $f \in L^2(\mathbf{R})$ は $e^{i\alpha x}$ ($\alpha \in \mathbf{R}$) で表現できる:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} \hat{f}(\alpha) e^{i\alpha x} d\alpha, \quad \hat{f}(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} f(x) e^{-i\alpha x} dx.$$

e^{inx} や $e^{i\alpha x}$ は $\partial^2/\partial x^2$ の固有関数であることもあって (つまり $(\partial/\partial x)^2 e^{i\alpha x} = -\alpha^2 e^{i\alpha x}$)、例えば次のような応用がある。

例 3.1.1 (熱方程式の初期値境界値問題の解) 応用解析 II で扱った、熱方程式の初期値 (Dirichlet) 境界値問題 (H-IBP) の解を求めてみよう。各時刻 t を止めたとき、 $u(\cdot, t)$ は $\sin n\pi x$ ($n \in \mathbf{N}$) で表現できるはずであるから ($\sin n\pi x$ を選んだのは、境界条件を考慮したものである)、

$$u(x, t) = \sum_{n=1}^{\infty} B_n(t) \sin n\pi x$$

とおけるはずである。熱方程式に代入すると (形式的な計算ではあるが)

$$\sum_{n=1}^{\infty} B'_n(t) \sin n\pi x = \sum_{n=1}^{\infty} -n^2 \pi^2 B_n(t) \sin n\pi x.$$

これから

$$B'_n(t) = -n^2\pi^2 B_n(t) \quad (n \in \mathbf{N}).$$

一方、初期条件に代入して、

$$B_n(0) = b_n := 2 \int_0^1 f(x) \sin n\pi x \quad (n \in \mathbf{N}).$$

すると、 $B_n(t)$ は常微分方程式の初期値問題の解として次のように求まる:

$$B_n(t) = b_n e^{-n^2\pi^2 t} \quad (n \in \mathbf{N}).$$

ゆえに

$$u(x, t) = \sum_{n=1}^{\infty} b_n e^{-n^2\pi^2 t} \sin n\pi x. \blacksquare$$

例 3.1.2 (熱方程式の初期値問題の解) 熱方程式の初期値問題

$$u_t = u_{xx} \quad ((x, t) \in \mathbf{R} \times (0, \infty)), \quad u(x, 0) = f(x) \quad (x \in \mathbf{R})$$

の解を求めてみよう。各時刻 t を止めたとき、 $u(\cdot, t)$ は $e^{i\alpha x}$ で表現できるはずであるから、

$$u(x, t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} b(\alpha, t) e^{i\alpha x} dx$$

とおけるはずである。熱方程式に代入すると

$$\frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} \frac{\partial}{\partial t} b(\alpha, t) e^{i\alpha x} dx = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} \left(\frac{\partial}{\partial x} \right)^2 b(\alpha, t) e^{i\alpha x} dx = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} -\alpha^2 b(\alpha, t) e^{i\alpha x} dx.$$

これから

$$\frac{\partial}{\partial t} b(\alpha, t) = -\alpha^2 b(\alpha, t) \quad (\alpha \in \mathbf{R}).$$

一方初期条件に代入して、

$$b(\alpha, 0) = \widehat{f}(\alpha) \stackrel{\text{def.}}{=} \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} f(x) e^{-\alpha x} dx \quad (\alpha \in \mathbf{R}).$$

すると、 $b(\alpha, t)$ は常微分方程式の初期値問題の解として

$$b(\alpha, t) = \widehat{f}(\alpha) e^{-\alpha^2 t}.$$

これから

$$u(x, t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} \widehat{f}(\alpha) e^{-\alpha^2 t} e^{i\alpha x} dx.$$

この後、Fourier 変換の各種公式を使って変形すると

$$u(x, t) = \int_{\mathbf{R}} H(x-y, t) f(y) dy, \quad H(x, t) := \frac{1}{\sqrt{4\pi t}} e^{-\frac{x^2}{4t}}. \blacksquare$$

注意 3.1.3 (普通の本に書いてある説明) 熱方程式の初期値問題については、次のように書いてある本が多い。

$$u_t = u_{xx}$$

の両辺を x について Fourier 変換して、

$$(3.1) \quad \frac{d}{dt} \hat{u}(\alpha, t) = (-i\alpha)^2 \hat{u}(\alpha, t) = -\alpha^2 \hat{u}(\alpha, t),$$

ただし

$$\hat{u}(\alpha, t) := \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} u(x, t) e^{-\alpha x} dx.$$

一方で初期条件

$$u(x, 0) = f(x)$$

を x について Fourier 変換して

$$(3.2) \quad \hat{u}(\alpha, 0) = \hat{f}(\alpha).$$

(3.1), (3.2) を (パラメーター α を含む) 常微分方程式の初期値問題として解いて

$$\hat{u}(\alpha, t) = \hat{f}(\alpha) e^{-\alpha^2 t}.$$

ここからは上の例と同じである。■

3.1.2 波動方程式の場合

(準備中 — 自分で調べてみましょう。)

3.2 波動方程式の初期値問題の差分解

この節の内容は F. John [16] による。

3.2.1 $e^{i\alpha x}$ が固有関数であること

x についての差分に関しても $e^{i\alpha x}$ は固有関数になっている。例えば

$$\delta_x v(x) = v(x + h/2) - v(x - h/2)$$

により、一階の中心差分作用素 δ_x を定義すると、

$$\delta_x e^{i\alpha x} = e^{i\alpha(x+h/2)} - e^{i\alpha(x-h/2)} = (e^{i\alpha h/2} - e^{-i\alpha h/2})e^{i\alpha x} = 2i \sin\left(\frac{\alpha h}{2}\right) e^{i\alpha x}.$$

これから

$$(\delta_x)^2 e^{i\alpha x} = -4 \sin^2\left(\frac{\alpha h}{2}\right) e^{i\alpha x}.$$

つまり $e^{i\alpha x}$ は $(\delta_x)^2$ の固有関数でもある。後のために

$$(\delta_x)^2 v(x) = v(x+h) - 2v(x) + v(x-h)$$

であることを注意しておく。

3.2.2 差分方程式の形式解の導出

$e^{i\alpha x} b(\alpha, t)$ の形の一般解

t の値を固定するごとに、 x の関数として、 $e^{i\alpha x}$ で展開できる、つまり

$$u(x, t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} b(\alpha, t) e^{i\alpha x} d\alpha$$

と表わすことができるであろう。そして任意の α に対して、 $v(x, t) := b(\alpha, t) e^{i\alpha x}$ 自身が差分方程式

$$(3.3) \quad v(x, t+k) - 2v(x, t) + v(x, t-k) = \lambda^2 (v(x+h, t) - 2v(x, t) + v(x-h, t))$$

を満たさねばならない。代入すると

$$[b(\alpha, t+k) - 2b(\alpha, t) + b(\alpha, t-k)] e^{i\alpha x} = \lambda^2 \left[-4 \sin^2\left(\frac{\alpha h}{2}\right) e^{i\alpha x} \right] b(\alpha, t)$$

であるから、

$$(3.4) \quad b(\alpha, t+k) - 2b(\alpha, t) + b(\alpha, t-k) = -4\lambda^2 \sin^2\left(\frac{\alpha h}{2}\right) b(\alpha, t).$$

これは α をパラメーターに持つ 2 階の線形差分方程式である。

この解空間は 2 次元の線型空間であるが、一般解を求めるには $b(\alpha, t) = e^{i\beta t}$ とおいて、代入してみれば良いのであった。

$$b(\alpha, t+k) - 2b(\alpha, t) + b(\alpha, t-k) = e^{i\beta(t+k)} - 2e^{i\beta t} + e^{i\beta(t-k)} = -4 \sin^2\left(\frac{\beta k}{2}\right) e^{i\beta t}$$

であるから (3.4) は

$$-4 \sin^2 \left(\frac{\beta k}{2} \right) e^{i\beta t} = -4\lambda^2 \sin^2 \left(\frac{\alpha h}{2} \right) e^{i\beta t},$$

すなわち

$$(3.5) \quad \sin^2 \left(\frac{\beta k}{2} \right) = \lambda^2 \sin^2 \left(\frac{\alpha h}{2} \right)$$

となる。

$0 < \lambda \leq 1$ と仮定すると、任意の α に対して (3.5) の右辺は 0 以上 1 以下の数であるから、 β は実数である。また (3.5) を満たす一つの $\beta = \beta_\alpha$ が得られた場合、他の β は

$$\exists n \in \mathbf{Z} \quad \text{s.t.} \quad \frac{\beta k}{2} = \pm \left(\frac{\beta_\alpha k}{2} \right) + n\pi$$

を満たす。すると $e^{i\beta k}$ の値としては

$$e^{\pm i\beta_\alpha k}$$

の二通りしかないことが分かる。ゆえに (3.4) の一般解は、

$$\begin{aligned} b(\alpha, t) &= C_1 e^{i\beta_\alpha t} + C_2 e^{-i\beta_\alpha t} \quad (C_1, C_2 \text{ は任意定数}) \\ &= (A \cos \beta_\alpha t + B \sin \beta_\alpha t) \quad (A, B \text{ は任意定数}). \end{aligned}$$

ゆえに (3.3) の一般解は

$$v(x, t) = e^{i\alpha x} (A \cos \beta_\alpha t + B \sin \beta_\alpha t) \quad (A, B \text{ は任意定数}).$$

初期条件に対応する差分方程式

微分方程式の初期値問題

$$(3.6) \quad u_{tt}(x, t) = u_{xx}(x, t) \quad ((x, t) \in \mathbf{R} \times \mathbf{R}),$$

$$(3.7) \quad u(x, 0) = f(x) \quad (x \in \mathbf{R}),$$

$$(3.8) \quad u_t(x, 0) = g(x) \quad (x \in \mathbf{R})$$

を解くことを目標としているが、(3.7) に対応する差分方程式として、

$$v(x, 0) = f(x)$$

を採用するのは当然として、(3.8) に対応する差分方程式として、ここでは

$$(3.9) \quad \frac{v(x, k) - v(x, -k)}{2k} = g(x)$$

を採用する。

素朴に考えると

$$u(x, k) \doteq u(x, 0) + ku_t(x, 0) = f(x) + k\psi(x)$$

であるから、

$$v(x, k) = f(x) + k\psi(x)$$

なども考えられるところであるが、これは誤差が $O(k)$ であり、好ましくない。

また、菊地・山本 [17] では、 $t = 0$ でも波動方程式が成り立つと仮定して得られる

$$\begin{aligned} u(x, k) &\doteq u(x, 0) + ku_t(x, 0) + \frac{k^2}{2}u_{tt}(x, 0) \\ &= u(x, 0) + ku_t(x, 0) + \frac{k^2}{2}u_{xx}(x, 0) \\ &= f(x) + kg(x) + \frac{k^2}{2}f''(x) \\ &\doteq f(x) + kg(x) + \frac{k^2}{2} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \end{aligned}$$

から、

$$(3.10) \quad v(x, k) = f(x) + kg(x) + \frac{\lambda^2}{2}(f(x+h) - 2f(x) + f(x-h))$$

を推奨していた。

(3.9) は、1 階中心差分近似という意味は明白だが、一体 $v(x, -k)$ はどうやって求めるのか？という疑問が生じるかもしれない。これについて、以下一つの回答を与える。それは上の菊地・山本の仮定と同様に $t = 0$ でも波動方程式に対応する差分方程式が成り立つ、つまり $t = 0$ で $Lv = 0$ が成り立つと仮定することである：

$$\frac{v(x, k) - 2v(x, 0) + v(x, -k)}{k^2} = \frac{v(x+h, 0) - 2v(x, 0) + v(x-h, 0)}{h^2}.$$

これから

$$v(x, k) + v(x, -k) = \lambda^2(f(x+k) - 2f(x) + f(x-k)) + 2f(x).$$

この方程式と、(3.9) から得られる

$$v(x, k) - v(x, -k) = 2kg(x)$$

を連立すると、

$$\begin{aligned} v(x, k) &= \frac{1}{2} [\lambda^2(f(x+h) - 2f(x) + f(x-h)) + 2f(x) + 2kg(x)] \\ &= f(x) + kg(x) + \frac{\lambda^2}{2}(f(x+h) - 2f(x) + f(x-h)). \end{aligned}$$

これは (3.10) と一致している。

特別な初期値に対する初期値問題の解

$$f(x) = e^{i\alpha x}, \quad g(x) = 0 \quad (x \in \mathbf{R})$$

の場合は、容易に $A = 1, B = 0$ となることが分かり、

$$v(x, t) = e^{i\alpha x} \cos \beta_\alpha t$$

が解となる。

$$f(x) = 0, \quad g(x) = e^{i\alpha x} \quad (x \in \mathbf{R})$$

の場合は、

$$v(x, t) = e^{i\alpha x} k \frac{\sin \beta_\alpha t}{\sin \beta_\alpha k}$$

が解となる。

初期値問題の形式解

Fourier 変換

$$\hat{f}(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} f(x) e^{-i\alpha x} dx, \quad \hat{g}(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} g(x) e^{-i\alpha x} dx,$$

を用いると

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} \hat{f}(\alpha) e^{i\alpha x} d\alpha, \quad g(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} \hat{g}(\alpha) e^{i\alpha x} d\alpha,$$

となるので、解の候補として

$$(3.11) \quad v(x, t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbf{R}} e^{i\alpha x} \left(\cos \beta_\alpha t \hat{f}(\alpha) + k \frac{\sin \beta_\alpha t}{\sin \beta_\alpha k} \hat{g}(\alpha) \right) d\alpha,$$

が考えられる。

3.2.3 形式解が well-defined であり、厳密解に収束すること

(準備中 — とりあえず、引用した。)

差分解 $v(x, t)$ が厳密解 $u(x, t)$ に収束することを証明する。

$$u(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\alpha x} \left(\cos \alpha t \hat{f}(\alpha) + k \frac{\sin \alpha t}{\alpha} \hat{g}(\alpha) \right) d\alpha.$$

いま、 $\int |f| d\alpha$ と $\int |\hat{g}| d\alpha$ は収束するので、十分大きな A を選んで、

$$\left| \int_A^{\infty} e^{i\alpha x} \left(\cos \beta t \hat{f}(\alpha) + k \frac{\sin \beta t}{\sin \beta k} \hat{g}(\alpha) \right) d\alpha \right| \leq \varepsilon,$$

$$\left| \int_{-\infty}^{-A} e^{i\alpha x} \left(\cos \beta t \hat{f}(\alpha) + k \frac{\sin \beta t}{\sin \beta k} \hat{g}(\alpha) \right) d\alpha \right| \leq \varepsilon$$

が成り立つようにできる。その上で、 h, k を小さくとり、

$$\left| \int_{-A}^A e^{i\alpha x} \left(\cos \alpha t \hat{f}(\alpha) + k \frac{\sin \alpha t}{\alpha} \hat{g}(\alpha) \right) d\alpha - \int_{-A}^A e^{i\alpha x} \left(\cos \beta t \hat{f}(\alpha) + k \frac{\sin \beta t}{\sin \beta k} \hat{g}(\alpha) \right) d\alpha \right| \leq \varepsilon$$

とできる。補題 1.1, 1.2, 1.3 と以上の事より、

$$\begin{aligned} & |u(x, t) - v(x, t)| \\ &= \left| \int_{-\infty}^{\infty} e^{i\alpha x} \left(\cos \alpha t \hat{f}(\alpha) + k \frac{\sin \alpha t}{\alpha} \hat{g}(\alpha) \right) d\alpha - \int_{-A}^A e^{i\alpha x} \left(\cos \beta t \hat{f}(\alpha) + k \frac{\sin \beta t}{\sin \beta k} \hat{g}(\alpha) \right) d\alpha \right| \\ &\leq \left| \int_{-\infty}^{-A} e^{i\alpha x} \left(\cos \alpha t \hat{f}(\alpha) + k \frac{\sin \alpha t}{\alpha} \hat{g}(\alpha) \right) d\alpha \right| + \left| \int_{-A}^A e^{i\alpha x} \left(\cos \alpha t \hat{f}(\alpha) + k \frac{\sin \alpha t}{\alpha} \hat{g}(\alpha) \right) d\alpha \right| \\ &+ \left| \int_A^{\infty} e^{i\alpha x} \left(\cos \alpha t \hat{f}(\alpha) + k \frac{\sin \alpha t}{\alpha} \hat{g}(\alpha) \right) d\alpha \right| + \left| \int_{-\infty}^{-A} e^{i\alpha x} \left(\cos \beta t \hat{f}(\alpha) + k \frac{\sin \beta t}{\sin \beta k} \hat{g}(\alpha) \right) d\alpha \right| \\ &+ \left| \int_{-A}^A e^{i\alpha x} \left(\cos \beta t \hat{f}(\alpha) + k \frac{\sin \beta t}{\sin \beta k} \hat{g}(\alpha) \right) d\alpha \right| + \left| \int_A^{\infty} e^{i\alpha x} \left(\cos \beta t \hat{f}(\alpha) + k \frac{\sin \beta t}{\sin \beta k} \hat{g}(\alpha) \right) d\alpha \right| \\ &\leq 5\varepsilon. \end{aligned}$$

よって、差分解が厳密解に収束することが証明できた。

3.3 波動方程式の初期値境界値問題の差分解

この節の結果は、笠井 [18] による。

3.3.1 初期値境界値問題

$$(3.12) \quad u_{tt}(x, t) = u_{xx}(x, t) \quad ((x, t) \in (0, 1) \times \mathbf{R}),$$

$$(3.13) \quad u(0, t) = u(1, t) = 0 \quad (t \in \mathbf{R}),$$

$$(3.14) \quad u(x, 0) = f(x) \quad (x \in [0, 1]),$$

$$(3.15) \quad u_t(x, 0) = g(x) \quad (x \in [0, 1]).$$

3.3.2 差分方程式

$u_{tt} = u_{xx}$ に対応する差分方程式として、

$$(3.16) \quad L[v] = 0, \quad \text{ただし} \quad L[v] := \frac{v(x, t+k) - 2v(x, t) + v(x, t-k)}{k^2} - \frac{v(x+h, t) - 2v(x, t) + v(x-h, t)}{h^2}$$

を採用する。境界条件としては、

$$(3.17) \quad v(0, t) = v(1, t) = 0,$$

初期条件としては

$$(3.18) \quad v(x, 0) = f(x), \quad \frac{v(x, k) - v(x, -k)}{2k} = g(x)$$

を採用する。

$$\begin{aligned} \delta_t F(t) &= F\left(\frac{t+k}{2}\right) - F\left(\frac{t-k}{2}\right), \\ \delta_x G(x) &= G\left(\frac{x+h}{2}\right) - G\left(\frac{x-h}{2}\right) \end{aligned}$$

によって、1階中心差分作用素 δ_t, δ_x を定義すると、

$$(\delta_t)^2 F(t) = F(t+k) - 2F(t) + F(t-k), \quad (\delta_x)^2 G(x) = G(x+h) - 2G(x) + G(x-h)$$

となるので、

$$L[v] = \frac{1}{k^2}(\delta_t)^2 - \frac{1}{h^2}(\delta_x)^2$$

となる。

3.3.3 $\sin n\pi x$ が $(\delta_x)^2$ の固有関数であること

x についての差分に関して、固有関数が $\sin n\pi x$ ($n \in \mathbf{N}$) である。実際

$$\begin{aligned} (\delta_x)^2 \sin n\pi x &= \sin n\pi(x+h) - 2\sin n\pi x + \sin n\pi(x-h) \\ &= 2\sin n\pi x \cos n\pi h - 2\sin n\pi x \\ &= 2\sin n\pi x (\cos n\pi h - 1) \\ &= -4\sin n\pi x \sin^2 \frac{n\pi h}{2}. \end{aligned}$$

3.3.4 差分方程式の形式解の導出

$\sin n\pi x b_n(t)$ の形の一般解

t の値を固定することに、 $v(x, t)$ を x の関数として、 $\sin n\pi x$ で展開できる。つまり

$$v(x, t) = \sum_{n=1} b_n(t) \sin n\pi x$$

と表わすことができる。そして任意の $n \in \mathbf{N}$ について、 $v(x, t) = b_n(t) \sin n\pi x$ 自身が差分方程式 $L[v] = 0$ を満たさなければならない。実際に $L[v] = 0$ に代入すると

$$(3.19) \quad b_n(t+k) - 2b_n(t) - b_n(t-k) = \lambda \left[-4\sin^2 \frac{n\pi h}{2} \right] b_n(t)$$

が得られる。

これは n をパラメーターにもつ、2階の線型差分方程式である。

一般解を求めるには、 $b_n(t) = e^{i\beta t}$ とおいて代入すればよい。

$$e^{i\beta(t+k)} - 2e^{i\beta t} + e^{i\beta(t-k)} = -4\sin^2 \left(\frac{\beta k}{2} \right) e^{i\beta t}.$$

これから

$$(3.20) \quad \sin^2 \left(\frac{\beta k}{2} \right) = \lambda^2 \sin^2 \left(\frac{n\pi h}{2} \right).$$

$|\lambda| \leq 1$ とするとき、右辺は 0 以上 1 以下の数となるので、この方程式の解 β は実数となる。最小の正の解 β を β_n と書こう。すると他の β については、

$$\exists \ell \in \mathbf{Z} \quad \text{s.t.} \quad \frac{\beta k}{2} = \pm \frac{\beta_n k}{2} + \ell\pi.$$

すると、 $e^{i\beta k}$ の値としては

$$e^{\pm i\beta_n k}$$

の二通りしかない。

(3.19) の一般解は

$$\begin{aligned} b_n(t) &= C_1 e^{i\beta_n t} + C_2 e^{-i\beta_n t} \quad (C_1, C_2 \text{ は任意定数}) \\ &= (A \cos \beta_n t + B \sin \beta_n t) \quad (A, B \text{ は任意定数}). \end{aligned}$$

ゆえに $L[v] = 0$ の、 $v(x, t) = b_n \sin n\pi x$ の形をした解は

$$v(x, t) = \sin n\pi x (A \cos \beta_n t + B \sin \beta_n t) \quad (A, B \text{ は任意定数}).$$

特別な初期値に対する解

$$L[v] = 0, \quad v(0, t) = v(1, t) = 0, \quad v(x, 0) = f(x), \quad \frac{v(x, k) - v(x, -k)}{2k} = g(x)$$

を考えているわけだが、 $f(x) = \sin n\pi x$, $g(x) = 0$ のときは、

$$v(x, t) = \sin n\pi x \cos \beta_n t,$$

$f(x) = 0$, $g(x) = \sin n\pi x$ のときは、

$$v(x, t) = \sin n\pi x \frac{k \sin \beta_n t}{\sin \beta_n k}.$$

初期値境界値問題の形式解

Fourier 係数

$$f_n = 2 \int_0^1 f(x) \sin n\pi x \, dx, \quad g_n = 2 \int_0^1 g(x) \sin n\pi x \, dx$$

を用いると、

$$f(x) = \sum_{n=1}^{\infty} f_n \sin n\pi x, \quad g(x) = \sum_{n=1}^{\infty} g_n \sin n\pi x$$

であるから、

$$v(x, t) = \sum_{n=1}^{\infty} f_n \sin n\pi x \cos \beta_n t + \sum_{n=1}^{\infty} g_n \sin n\pi x \frac{k \sin \beta_n t}{\sin \beta_n k}$$

が解になると期待される。

3.3.5 差分解の存在

前節までに求めた形式解は実は収束し、差分解を与える。

定理 3.3.1 (形式解の収束) 初期値 f, g の Fourier 係数 f_n, g_n について、

$$\sum_{n=1}^{\infty} |f_n| < \infty, \quad \sum_{n=1}^{\infty} |g_n| < \infty$$

が成り立つならば

$$v(x, t) \stackrel{\text{def.}}{=} \sum_{n=1}^{\infty} f_n \sin n\pi x \cos \beta_n t + \sum_{n=1}^{\infty} g_n \sin n\pi x \frac{k \sin \beta_n t}{\sin \beta_n k}$$

は、任意の $T > 0$ に対して、 $[0, 1] \times [0, T]$ で一様に絶対収束し、差分方程式 (3.16), (3.17), (3.18) を満たす。

証明 後述の補題 3.3.5 より、 $t = jk$ に対して、

$$\left| \frac{\sin \beta_n t}{\sin \beta_n k} \right| = \left| \frac{\sin \beta_n jk}{\sin \beta_n k} \right| \leq j$$

となることと、 $jk \leq T$ により、一様絶対収束することが分かる。(3.16) は線型同次方程式であり、差分作用素と $\sum_{n=1}^{\infty}$ は交換可能性であるから、 v は (3.16) を満たす。 v が境界条件 (3.17) を満たすことは明らか。 $\sum |f_n| < \infty, \sum |g_n| < \infty$ が成り立つとき、 f と g の Fourier 級数は一様収束して、それぞれ f, g に等しいことから、 v が初期条件 (3.18) を満たすことが分かる。■

3.3.6 いくつかの補題

(3.20) を満たす最小の正数として β_n を定義したが、以下では $nh \leq 1$ として考える。すると (3.20) は

$$\sin \frac{\beta_n k}{2} = \frac{k}{h} \sin \frac{n\pi h}{2}$$

となる。

まず β_n が存在することと、その範囲を確認しておこう。

補題 3.3.2 (β の存在とその範囲) $\forall \lambda \in (0, 1], \forall n \in \mathbf{N}, \forall h \in (0, 1/n], \exists! \beta = \beta_{n,\lambda,h} \in [0, \pi/k]$ s.t.

$$\sin \frac{\beta k}{2} = \frac{k}{h} \sin \frac{n\pi h}{2}.$$

ただし $k = \lambda h$ である。さらに β は

$$\beta \leq \frac{n\pi h}{k} = \frac{n\pi}{\lambda}$$

を満たす。

証明 方程式の右辺が 0 以上 1 以下であることに注意すれば、 β の存在が分かる。

$0 < k/h = \lambda \leq 1$ より、

$$\sin \frac{\beta k}{2} \leq \sin \frac{n\pi h}{2}$$

であるから、

$$\frac{\beta k}{2} \leq \frac{n\pi h}{2} \quad \therefore \quad \beta \leq n\pi \frac{h}{k} = \frac{n\pi}{\lambda}$$

となる。■

補題 3.3.3 (β_n は $n\pi$ に近い)

$(\forall \lambda \in (0, 1]) \quad (\forall N \in \mathbf{N}) \quad (\exists C > 0) \quad (\exists \delta > 0) \quad (\forall n \in \mathbf{N} : n \leq N) \quad (\forall h \in (0, \delta])$

$$|\beta_n - n\pi| \leq C|n^3 h^2|.$$

証明 $f(x) = \sin x - x$ を Taylor 展開する。

$$\sin x - x = -\frac{1}{3!}x^3 + \dots = -\frac{f^{(3)}(\theta x)}{3!}x^3 \quad (0 < \theta < 1)$$

両辺の絶対値をとると、

$$(3.21) \quad |\sin x - x| \leq \frac{1}{6}|x|^3.$$

この不等式に $x = \beta_n k/2, x = n\pi/2$ を代入して、

$$(3.22) \quad \left| \sin \frac{\beta_n k}{2} - \frac{\beta_n k}{2} \right| \leq \frac{1}{6} \left(\frac{\beta_n k}{2} \right)^3,$$

$$\left| \sin \frac{n\pi h}{2} - \frac{n\pi h}{2} \right| \leq \frac{1}{6} \left(\frac{n\pi h}{2} \right)^3.$$

この式の両辺に λ をかけて、

$$(3.23) \quad \left| \lambda \sin \frac{n\pi h}{2} - \lambda \frac{n\pi h}{2} \right| \leq \frac{\lambda}{6} \left(\frac{n\pi h}{2} \right)^3$$

$\sin(\beta_n k/2) = \lambda \sin(n\pi h/2)$ に注意すると、(3.22) – (3.23) より、

$$\left| \frac{\beta_n k}{2} - \lambda \frac{n\pi h}{2} \right| \leq \frac{1}{6} \left[\left(\frac{\beta_n k}{2} \right)^3 + \lambda \left(\frac{n\pi h}{2} \right)^3 \right].$$

左辺に $\lambda = k/h$ を代入して、

$$\left| \frac{\beta_n k}{2} - \frac{n\pi k}{2} \right| \leq \frac{1}{6} \left[\left(\frac{\beta_n k}{2} \right)^3 + \frac{k}{h} \left(\frac{n\pi h}{2} \right)^3 \right].$$

ゆえに

$$|\beta_n - n\pi| \leq \frac{1}{3} \left(\frac{\beta_n^3 k^2}{8} + \frac{n^3 \pi^3 h^2}{8} \right) = \frac{1}{24} (\beta_n^3 k^2 + n^3 \pi^3 h^2)$$

$\delta = 1/N$ とおき、 $0 < h \leq \delta$ とすると、補題 3.3.2 より $\beta_n \leq n\pi/\lambda$ となるので、

$$|\beta_n - n\pi| \leq \frac{1}{24} (k^2 (n\pi/\lambda)^3 + n^3 \pi^3 h^2) = \frac{n^3 \pi^3}{24} \left(h^2 + \frac{k^2}{\lambda^3} \right) = \frac{n^3 \pi^3}{24} h^2 \left(1 + \frac{1}{\lambda} \right).$$

ゆえに

$$C = \frac{\pi^3}{24} \left(1 + \frac{1}{\lambda} \right)$$

とおけばよい。■

補題 3.3.4 (収束を証明するための一様評価) ($\forall \lambda \in (0, 1]$) ($\forall N \in \mathbf{N}$) ($\forall T > 0$)

$$\lim_{h \rightarrow 0} \sup_{\substack{1 \leq n \leq N \\ 0 \leq j \leq T/k}} |\cos \beta_n j k - \cos n\pi j k| = 0,$$

$$\lim_{h \rightarrow 0} \sup_{\substack{1 \leq n \leq N \\ 0 \leq j \leq T/k}} \left| k \frac{\sin \beta_n j k}{\sin \beta_n k} - \frac{\sin n\pi j k}{n\pi} \right| = 0.$$

証明 $t = jk$ とおく。補題 3.3.3 より、

$$\begin{aligned} |\cos \beta_n jk - \cos n\pi jk| &= |\cos((n\pi + O(n^3 h^2))t) - \cos n\pi t| \\ &= |O(n^3 h^2)t| \\ &= N^3 T \cdot O(h^2) \\ &\rightarrow 0 \quad (h \rightarrow 0) \end{aligned}$$

(ここで $\cos(x + \Delta x) - \cos x = -\sin(x + \theta\Delta x)\Delta x$ より導かれる $|\cos(x + \Delta x) - \cos x| \leq |\Delta x|$ を用いた。)

補題 3.3.2, 補題 3.3.3 から、

$$\begin{aligned} |\sin \beta_n k - \beta_n k| &\leq \frac{1}{6} |\beta_n k|^3 = O(n^3 h^3), \\ |\beta_n - n\pi| &= O(n^3 h^2) \end{aligned}$$

であるから、

$$|\sin \beta_n k - n\pi k| = O(n^3 h^2) + O(n^3 h^2 k) = O(n^3 h^3).$$

$h \rightarrow 0$ ならば、 $k \rightarrow 0$ となるので、

$$\begin{aligned} \left| k \frac{\sin \beta_n jk}{\sin \beta_n k} - \frac{\sin n\pi t}{n\pi} \right| &= \frac{1}{n\pi \sin \beta_n k} |kn\pi \sin \beta_n t - \sin \beta_n k \sin n\pi t| \\ &= \frac{1}{n\pi \sin \beta_n k} |kn\pi [\sin(n\pi t) + O(n^3 h^2 t)] - [n\pi k + O(n^3 h^3)] \sin n\pi t| \\ &= \frac{1}{n\pi \sin \beta_n k} (O(n^4 h^2 kt) + O(n^3 h^3)) \\ &= \frac{1}{\sin \beta_n k} TN^3 \cdot O(h^3) \rightarrow 0 \quad (h \rightarrow 0). \blacksquare \end{aligned}$$

補題 3.3.5 $\forall \lambda \in (0, 1], \forall T > 0$ に対して

$$\sup_{\substack{n \in \mathbf{N} \\ 0 \leq j \leq T/k}} \left| k \frac{\sin \beta_n jk}{\sin \beta_n k} \right| \leq T$$

証明 命題 3.5.2 より

$$\left| \frac{\sin \beta_n jk}{\sin \beta_n k} \right| \leq j$$

であり、 $jk \leq T$ であるから。 ■

3.3.7 差分解の厳密解への収束

さて、

$$\sum_{n=1}^{\infty} |f_n| \stackrel{\text{def.}}{=} M_f < \infty, \quad \sum_{n=1}^{\infty} |g_n| \stackrel{\text{def.}}{=} M_g < \infty.$$

を仮定する。すると $\forall \varepsilon > 0$ に対して、十分大きな $N \in \mathbf{N}$ を取ると

$$\sum_{n=N+1}^{\infty} |f_n| < \varepsilon, \quad \sum_{n=N+1}^{\infty} |g_n| < \varepsilon.$$

$$\begin{aligned} |u(x, t) - v(x, t)| &= \left| \sum_{n=1}^{\infty} \sin n\pi x \left(f_n \cos n\pi t + g_n \frac{\sin n\pi t}{n\pi} \right) - \sum_{n=1}^{\infty} \sin n\pi x \left(f_n \cos \beta_n t + g_n \frac{k \sin \beta_n t}{\sin \beta_n k} \right) \right| \\ &\leq \sum_{n=1}^{\infty} \left(|f_n| |\cos n\pi t - \cos \beta_n t| + |g_n| \left| \frac{\sin n\pi t}{n\pi} - \frac{k \sin \beta_n t}{\sin \beta_n k} \right| \right) \\ &\leq \sum_{n=1}^N (|f_n| |\cos n\pi t - \cos \beta_n t| + |g_n| \left| \frac{\sin n\pi t}{n\pi} - \frac{k \sin \beta_n t}{\sin \beta_n k} \right|) \\ &\quad + \sum_{n>N} |f_n| (|\cos n\pi t| + |\cos \beta_n t|) + \sum_{n>N} |g_n| \left(\left| \frac{\sin n\pi t}{n\pi} \right| + \left| \frac{k \sin \beta_n t}{\sin \beta_n k} \right| \right) \\ &\leq \sup_{\substack{1 \leq n \leq N \\ |t| \leq T}} |\cos n\pi t - \cos \beta_n t| \sum_{n=1}^N |f_n| + \sup_{\substack{1 \leq n \leq N \\ |t| \leq T}} \left| \frac{\sin n\pi t}{n\pi} - \frac{k \sin \beta_n t}{\sin \beta_n k} \right| \sum_{n=1}^N |g_n| \\ &\quad + \sum_{n>N} |f_n| (|\cos n\pi t| + |\cos \beta_n t|) + \sum_{n>N} |g_n| \left(\left| \frac{\sin n\pi t}{n\pi} \right| + \left| \frac{k \sin \beta_n t}{\sin \beta_n k} \right| \right) \\ &\leq M_f \sup_{\substack{1 \leq n \leq N \\ |t| \leq T}} |\cos n\pi t - \cos \beta_n t| + M_g \sup_{\substack{1 \leq n \leq N \\ |t| \leq T}} \left| \frac{\sin n\pi t}{n\pi} - \frac{k \sin \beta_n t}{\sin \beta_n k} \right| \\ &\quad + \left(2 + \frac{1}{\pi} + T \right) \varepsilon \end{aligned}$$

h を十分小さくすれば第1項, 第2項の和は ε で押さえられる。

$$|u(x, t) - v(x, t)| \leq \varepsilon + \left(2 + \frac{1}{\pi} + T \right) \varepsilon \leq \left(3 + \frac{1}{\pi} + T \right) \varepsilon. \blacksquare$$

3.4 付録 1: 線形差分方程式

(ここに述べることは線形代数学の教科書に書いてあることが多い。探してみること。)

$\mathbf{N}_0 \stackrel{\text{def.}}{=} \mathbf{N} \cup \{0\} = \{0, 1, 2, \dots\}$ とおく。

複素数列の全体 $\mathbf{C}^{\mathbf{N}_0}$ は自然な線型空間の構造を持つ。つまり、数列 $\{a_n\}_{n=0}^{\infty}$, $\{b_n\}_{n=0}^{\infty} \in \mathbf{C}^{\mathbf{N}_0}$, $\mu \in \mathbf{C}$ に対して、

$$\{a_n\}_{n=0}^{\infty} + \{b_n\}_{n=0}^{\infty} = \{a_n + b_n\}_{n=0}^{\infty}, \quad \mu\{a_n\}_{n=0}^{\infty} = \{\mu a_n\}_{n=0}^{\infty}$$

を和、スカラー乗法の定義とすることにより線型空間となる。

$m+1$ 個の定数 $\{c_i\}_{i=0}^m$ を定めて、数列 $\{a_n\}_{n=0}^{\infty}$ に対して、

$$(3.24) \quad c_0 a_n + c_1 a_{n+1} + \dots + c_m a_{n+m} = 0 \quad (n = 0, 1, 2, \dots)$$

という条件を考える。(3.24) のことを線形差分方程式と呼ぶ。

線形差分方程式 (3.24) の解全体は、 $\mathbf{C}^{\mathbf{N}_0}$ の線形部分空間となることはすぐに分かるが、その次元は m であり、その基底は以下説明するようにして求めることができる。

代数方程式

$$c_0 + c_1 \mu + c_2 \mu^2 + \dots + c_m \mu^m = 0$$

の根 μ に対して、数列

$$\{\mu^n\}_{n=0}^{\infty}$$

は線形差分方程式 (3.24) の解となること、さらには相異なる根 μ_1, \dots, μ_r があるとき、 r 個の数列

$$\{(\mu_i)^n\}_{n=0}^{\infty} \quad (i = 1, 2, \dots, r)$$

は線型独立であることも容易に確かめられる。

$a_n = \mu^n$ とおいて、(3.24) に代入すると

$$c_0 \mu^n + c_1 \mu^{n+1} + \dots + c_m \mu^{n+m} = 0 \quad (n = 0, 1, 2, \dots)$$

となるが、これはただ一つの代数方程式

$$c_0 + c_1 \mu + \dots + c_m \mu^m = 0$$

と同値であることは明らかである。

若干面倒なのは重根を持つ場合であるが、その場合は (以下略)

例 3.4.1 (フィボナッチ数列) 「漸化式」

$$(3.25) \quad a_{n+2} = a_{n+1} + a_n \quad (n = 0, 1, \dots)$$

を満たす数列 $\{a_n\}_{n=0}^{\infty}$ をフィボナッチ数列というが、(3.25) は線形差分方程式に他ならない。

$$\mu^2 - \mu - 1 = 0$$

を解くと

$$\mu = \frac{1 \pm \sqrt{5}}{2}$$

となるので、一般解は

$$a_n = C_1 \left(\frac{1 + \sqrt{5}}{2} \right)^n + C_2 \left(\frac{1 - \sqrt{5}}{2} \right)^n \quad (C_1, C_2 \text{ は任意定数}). \blacksquare$$

3.5 付録2: 三角関数についてのメモ

$z = x + iy$ ($x, y \in \mathbf{R}$) とするとき、

$$\begin{aligned} \sin z &= \frac{e^{iz} - e^{-iz}}{2i} = \frac{e^{i(x+iy)} - e^{-i(x+iy)}}{2i} = \frac{1}{2i}(e^{-y}e^{ix} - e^ye^{-ix}) \\ &= \frac{1}{2i}[e^{-y}(\cos x + i \sin x) - e^y(\cos x - i \sin x)] \\ &= \frac{-i}{2}[(e^{-y} - e^y) \cos x + i(e^{-y} + e^y) \sin x] \\ &= \frac{e^y + e^{-y}}{2} \sin x + i \frac{e^y - e^{-y}}{2} \cos x \\ &= \cosh y \sin x + i \sinh y \cos x. \end{aligned}$$

これから

$$|\sin z| = \text{ごしよごしよ} = \sqrt{\sinh^2 y + \sin^2 x}.$$

ゆえに

$$\sin z = 0 \Leftrightarrow \sinh y = \sin x = 0 \Leftrightarrow y = 0, \quad x \in \pi\mathbf{Z} \Leftrightarrow z \in \pi\mathbf{Z}.$$

もつとも $\sin z = 0$ の解だけならば、次のようにする方が簡単である。

$$\sin z = 0 \Leftrightarrow \frac{e^{iz} - e^{-iz}}{2i} = 0 \Leftrightarrow e^{2zi} = 1 \Leftrightarrow 2z \in 2\pi\mathbf{Z} \Leftrightarrow z \in \pi\mathbf{Z}.$$

同様に

$$\cos z = 0 \Leftrightarrow \frac{e^{iz} + e^{-iz}}{2} = 0 \Leftrightarrow e^{2zi} = -1 \Leftrightarrow 2z + \pi \in 2\pi\mathbf{Z} \Leftrightarrow z \in \frac{\pi}{2} + \pi\mathbf{Z}.$$

与えられた複素数 c に対して、方程式

$$\sin z = c$$

の解集合はどうなるか？

まず解が存在するかどうか考えよう。 $e^{iz} = \zeta$ とおくと、

$$\sin z = \frac{e^{iz} - e^{-iz}}{2i} = \frac{\zeta - \zeta^{-1}}{2i}$$

であるから、 $\sin z = c$ より

$$\frac{\zeta - \zeta^{-1}}{2i} = c.$$

これから

$$\zeta^2 - 2ic\zeta - 1 = 0.$$

これは ζ に関する 2 次方程式であり、確かに解 $\zeta (\neq 0)$ が存在する。

すると、 z についての方程式

$$e^{iz} = \zeta$$

は $\zeta \neq 0$ であるから、確かに解が存在する。実際

$$e^{x+iy} = re^{i\theta} \Leftrightarrow e^x = r \text{ かつ } e^{iy} = e^{i\theta} \Leftrightarrow x = \log r \text{ かつ } y \in \theta + 2\pi\mathbf{Z}.$$

以上から $\sin z = c$ は必ず解を持つことが分かった。

次に解がどれくらいたくさんあるか調べよう。

$$\sin z = \sin z' = c$$

より

$$0 = \sin z - \sin z' = 2 \cos \frac{z+z'}{2} \sin \frac{z-z'}{2}$$

となるから、

$$\frac{z+z'}{2} \in \frac{\pi}{2} + \pi\mathbf{Z} \quad \text{or} \quad \frac{z-z'}{2} \in \pi\mathbf{Z}.$$

すなわち

$$\exists n \in \mathbf{Z} \text{ s.t. } z' = (\pi - z) + 2n\pi \quad \text{or} \quad z' = z + 2n\pi.$$

つまり、形だけは高校生も知っている恒等式

$$\sin(\pi - z) = \sin z, \quad \sin(z + 2n\pi) = \sin z$$

から得られる解がすべてだと言うことが分かる。

以上分かったことをまとめておこう。

命題 3.5.1 複素変数の \sin, \cos について

(1) $z = x + iy$ ($x, y \in \mathbf{R}$) とするとき $|\sin z| = \sqrt{\sinh^2 y + \sin^2 x}$.

(2) $\sin z = 0 \Leftrightarrow z \in \pi\mathbf{Z}$.

(3) $\cos z = 0 \Leftrightarrow z \in \frac{\pi}{2} + \pi\mathbf{Z}$.

(4) 任意の複素数 c に対して $\sin z = c$ は無限個の解を持つ。その一つを z_0 とするとき、解集合は

$$\{z_0 + 2n\pi; n \in \mathbf{Z}\} \cup \{(\pi - z_0) + 2n\pi; n \in \mathbf{Z}\}.$$

問 $\cos z = c, \exp z = c$ という方程式の解についても同様のことを行なってまとめよ。

命題 3.5.2 任意の自然数 n について、

$$\left| \frac{\sin nx}{\sin x} \right| \leq n \quad (x \in \mathbf{R})$$

証明 帰納法による、 $n = 1$ のときは明らか。 $n - 1$ まで成り立つとすると、

$$\begin{aligned} \left| \frac{\sin nx}{\sin x} \right| &= \left| \frac{\sin x \cos(n-1)x + \cos x \sin(n-1)x}{\sin x} \right| \\ &= \left| \cos(n-1)x + \cos x \frac{\sin(n-1)x}{\sin x} \right| \\ &\leq 1 + \left| \frac{\sin(n-1)x}{\sin x} \right| \end{aligned}$$

帰納法の仮定により、右辺は $1 + (n - 1) = n$ 以下である。すなわち n のときも成り立つ。■

第4章 波動方程式に対する差分法 — 4年16組68番 藤沼 祐一 (1999年3月16日)

4.1 はじめに

一次元の波動方程式は

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

である。この方程式を差分法で解く方法として、両辺を二階中心差分商で置き換えて、

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\tau^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}$$

(h, τ 及び u_i^n の定義は、後に出てくるものと同じ) として解く方法がよく使われる。

この卒業研究では、これ以外の方法で解くことを試みた。又、その際エネルギーの保存則が成り立つのかを調べた。

4.2 問題

4.2.1 Dirichlet 境界条件の問題

まず、 $0 \leq x \leq 1, t \geq 0$ で定義された $u = u(x, t)$ に対する波動方程式

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \quad (0 < x < 1, t > 0)$$

を初期条件

$$\left. \begin{array}{l} u(x, 0) = \varphi(x) \\ \frac{\partial u}{\partial t} = \psi(x) \end{array} \right\} \quad (0 \leq x \leq 1)$$

で Dirichlet 境界条件

$$u(0, t) = 0, \quad u(1, t) = 0 \quad (t > 0)$$

の下で解く。この問題を (A) とする。

物理的イメージとしては、 $x = 0$ から $x = 1$ に張られた弦が、両端の点 $x = 0$, $x = 1$ で固定されて振動している。

4.2.2 新しい関数を用いた解法

ここで、 $v(x, t)$ 、 $w(x, t)$ という関数を次のように定義する。

$$\begin{aligned} \frac{\partial u}{\partial t}(x, t) &= v(x, t) \\ \frac{\partial u}{\partial x}(x, t) &= w(x, t) \quad (0 \leq x \leq 1, t \geq 0) \end{aligned}$$

この2つの関数を用いて次の初期値境界値問題 (1) を導く。

$$(4.1) \quad \begin{cases} \frac{\partial}{\partial t} \begin{pmatrix} v \\ w \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{\partial}{\partial x} \begin{pmatrix} v \\ w \end{pmatrix} & (0 < x < 1, t > 0) \quad \dots(\star) \\ v(x, 0) = \psi(x) & (0 \leq x \leq 1) \\ w(x, 0) = \varphi'(x) & (0 \leq x \leq 1) \\ v(0, t) = v(1, t) = 0 & (t > 0) \\ \frac{\partial}{\partial x} w(0, t) = \frac{\partial}{\partial x} w(1, t) = 0 & (t > 0). \end{cases}$$

問題 (1) を差分法で解く。

区間 $[0, 1]$ を N 等分する。 $h = \frac{1}{N}$, 時間ステップを τ とし、各格子点 $x_i = ih$ 時刻 $t_j = j\tau$ とする。また、

$$U = \begin{pmatrix} v \\ w \end{pmatrix} \quad U_{i,j} = \begin{pmatrix} v_{i,j} \\ w_{i,j} \end{pmatrix} \quad A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

とする。ここで、フリードリクス (Friedrichs) の差分法を採用する。

フリードリクス (Friedrichs) の差分法とは、

$$\frac{\partial U}{\partial t}(x_i, t_j) \doteq \frac{U_{i,j+1} - \frac{1}{2}(U_{i+1,j} + U_{i-1,j})}{\tau}$$

と近似し、 $\frac{\partial U}{\partial x}(x_i, t_j)$ を中心差分商 $\frac{U_{i+1,j} - U_{i-1,j}}{2h}$ で近似する方法である。

(☆) の式を、フリードリクス (Friedrichs) の差分法で離散化すると、

$$\frac{U_{i,j+1} - \frac{1}{2}(U_{i+1,j} + U_{i-1,j})}{\tau} = A \frac{U_{i+1,j} - U_{i-1,j}}{2h} \quad (0 < i < N, j = 0, 1, 2, \dots).$$

ここで $\lambda = \frac{\tau}{h}$, I を 2 次の単位行列とすると、

$$U_{i,j+1} = \frac{1}{2}(I + \lambda A)U_{i+1,j} + \frac{1}{2}(I - \lambda A)U_{i-1,j} \quad (0 < i < N, j = 0, 1, 2, \dots)$$

これを成分表示で書くと、

$$(4.2) \quad v_{i,j+1} = \frac{1}{2}\{(v_{i+1,j} + v_{i-1,j}) + \lambda(w_{i+1,j} - w_{i-1,j})\},$$

$$(4.3) \quad w_{i,j+1} = \frac{1}{2}\{\lambda(v_{i+1,j} - v_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})\} \\ (0 < i < N, j = 0, 1, 2, \dots).$$

初期条件、境界条件については次のように行なう。

$$\begin{aligned} v_{i,0} &= \psi(ih) & (0 \leq i \leq N) \\ w_{i,0} &= \varphi'(ih) & (0 \leq i \leq N) \\ v_{0,j} &= v_{N,j} = 0 & (j = 1, 2, \dots) \end{aligned}$$

. 次に $w_{0,j+1}$ ($j = 0, 1, 2, \dots$) について考察する。

$$\frac{\partial}{\partial x} w(0, t) = \frac{\partial^2}{\partial x^2} u(0, t) = \frac{\partial^2}{\partial t^2} u(0, t) = \frac{\partial}{\partial t} v(0, t) = 0$$

なので、仮想格子点 x_{-1} を導入し $\frac{\partial}{\partial x} w(0, t_j)$ を中心差分商で近似する。

$$\frac{\partial}{\partial x} w(0, t_j) \doteq \frac{w_{1,j} - w_{-1,j}}{2h} = 0.$$

より、

$$w_{-1,j} = w_{1,j} \quad (j = 0, 1, 2, \dots).$$

そこで、(2), (3) の式に $i = 0$ で用いると

$$\begin{aligned} v_{0,j+1} &= \frac{1}{2}\{(v_{1,j} + v_{-1,j}) + \lambda(w_{1,j} - w_{-1,j})\} \\ w_{0,j+1} &= \frac{1}{2}\{\lambda(v_{1,j} - v_{-1,j}) + (w_{1,j} + w_{-1,j})\} \\ & \quad (j = 0, 1, 2, \dots). \end{aligned}$$

上の式に $w_{-1,j} = w_{1,j}$, $v_{0,j+1} = 0$ を代入すると、

$$0 = \frac{1}{2}\{(v_{1,j} + v_{-1,j}) + \lambda \cdot 0\}$$

より、

$$v_{-1,j} = -v_{1,j} \quad (j = 0, 1, 2, \dots).$$

ゆえに、

$$w_{0,j+1} = \frac{1}{2}\{\lambda(v_{1,j} - v_{-1,j}) + (w_{1,j} + w_{-1,j})\} = \lambda v_{1,j} + w_{1,j} \quad (j = 0, 1, 2, \dots)$$

同様に $w_{N,j+1}$ ($j = 0, 1, 2, \dots$) についても行なう。

$$\frac{\partial}{\partial x} w(1, t) = \frac{\partial^2}{\partial x^2} u(1, t) = \frac{\partial^2}{\partial t^2} u(1, t) = \frac{\partial}{\partial t} v(1, t) = 0$$

なので、仮想格子点 x_{N+1} を導入し $\frac{\partial}{\partial x} w(1, t_j)$ を中心差分商で近似する。

$$\frac{\partial}{\partial x} w(1, t_j) \doteq \frac{w_{N+1,j} - w_{N-1,j}}{2h} = 0$$

より、

$$w_{N-1,j} = w_{N+1,j} \quad (j = 0, 1, 2, \dots).$$

そこで、(2), (3) の式に $i = N$ で用いると

$$\begin{aligned} v_{N,j+1} &= \frac{1}{2}\{(v_{N+1,j} + v_{N-1,j}) + \lambda(w_{N+1,j} - w_{N-1,j})\} \\ w_{N,j+1} &= \frac{1}{2}\{\lambda(v_{N+1,j} - v_{N-1,j}) + (w_{N+1,j} + w_{N-1,j})\} \\ &\quad (j = 0, 1, 2, \dots). \end{aligned}$$

上の式に $w_{N-1,j} = w_{N+1,j}$, $v_{N,j+1} = 0$ を代入すると、

$$0 = \frac{1}{2}\{(v_{N+1,j} + v_{N-1,j}) + \lambda \cdot 0\}$$

より、

$$v_{N+1,j} = -v_{N-1,j} \quad (j = 0, 1, 2, \dots)$$

ゆえに、

$$w_{N,j+1} = \frac{1}{2}\{\lambda(v_{N+1,j} - v_{N-1,j}) + (w_{N+1,j} + w_{N-1,j})\} = -\lambda v_{N-1,j} + w_{N-1,j} \quad (j = 0, 1, 2, \dots).$$

と表せる。

(1)の問題で求めたのは、 $v_{i,j}, w_{i,j}$ ($0 \leq i \leq N, j = 0, 1, 2, \dots$)である。そこで (A) で求めたい解を $u_{i,j}$ ($0 \leq i \leq N, j = 0, 1, 2, \dots$) とすると、 $u_{i,j}$ は、次の式で求められる。

$$\begin{cases} u_{i,j+1} = \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \tau v_{i,j} & (0 < i < N, j = 0, 1, 2, \dots) \\ u_{i,0} = \varphi(ih) & (0 \leq i \leq N) \\ u_{0,j} = u_{N,j} = 0 & (j = 0, 1, 2, \dots). \end{cases}$$

4.2.3 まとめ

問題 (A) の差分解 $u_{i,j}$ を求めるために必要な式をまとめる。
まず、 $v_{i,j}, w_{i,j}$ を求める式は、

$$\begin{cases} v_{i,j+1} = \frac{1}{2}\{(v_{i+1,j} + v_{i-1,j}) + \lambda(w_{i+1,j} - w_{i-1,j})\} & (0 < i < N, j = 0, 1, 2, \dots) \\ w_{i,j+1} = \frac{1}{2}\{\lambda(v_{i+1,j} - v_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})\} & (0 < i < N, j = 0, 1, 2, \dots) \\ v_{i,0} = \psi(ih) & (0 \leq i \leq N) \\ w_{i,0} = \varphi'(ih) & (0 \leq i \leq N) \\ v_{0,j} = v_{N,j} = 0 & (j = 1, 2, \dots) \\ w_{0,j+1} = \lambda v_{1,j} + w_{1,j} & (j = 0, 1, 2, \dots) \\ w_{N,j+1} = -\lambda v_{N-1,j} + w_{N-1,j} & (j = 0, 1, 2, \dots) \end{cases}$$

となり、ここで求めた $v_{i,j}$ を用いて $u_{i,j}$ を求める式は、

$$\begin{cases} u_{i,j+1} = \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \tau v_{i,j} & (0 < i < N, j = 0, 1, 2, \dots) \\ u_{i,0} = \varphi(ih) & (0 \leq i \leq N) \\ u_{0,j} = u_{N,j} = 0 & (j = 0, 1, 2, \dots) \end{cases}$$

となる。

4.3 境界条件が変わると

4.3.1 Neumann 境界条件での解法

次に、Neumann 境界条件で考察する。 $0 \leq x \leq 1, t \geq 0$ で定義された $u = u(x, t)$ に対する波動方程式

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \quad (0 < x < 1, t > 0)$$

を初期条件

$$\left. \begin{aligned} u(x, 0) &= \varphi(x) \\ \frac{\partial u}{\partial t} &= \psi(x) \end{aligned} \right\} \quad (0 \leq x \leq 1)$$

で Neumann 境界条件

$$\frac{\partial}{\partial x} u(0, t) = 0, \quad \frac{\partial}{\partial x} u(1, t) = 0 \quad (t > 0)$$

の下で解く。この問題を (B) とする。

Dirichlet 境界条件の時と同様に、 v, w を用いて次の初期値境界値問題 (4) を導く。

$$(4.4) \quad \left\{ \begin{aligned} \frac{\partial}{\partial t} \begin{pmatrix} v \\ w \end{pmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{\partial}{\partial x} \begin{pmatrix} v \\ w \end{pmatrix} & (0 < x < 1, t > 0) & \dots(\star) \\ v(x, 0) &= \psi(x) & (0 \leq x \leq 1) \\ w(x, 0) &= \varphi'(x) & (0 \leq x \leq 1) \\ w(0, t) &= w(1, t) = 0 & (t > 0) \\ \frac{\partial}{\partial x} v(0, t) &= \frac{\partial}{\partial x} v(1, t) = 0 & (t > 0). \end{aligned} \right.$$

(★) の式を、フリードリクス (Friedrichs) の差分法を採用して離散化し、Dirichlet 境界条件の時と同様に、 τ, λ を用い、成分で書くと、

$$v_{(4.5)} = \frac{1}{2} \{ (v_{i+1,j} + v_{i-1,j}) + \lambda (w_{i+1,j} - w_{i-1,j}) \}$$

$$w_{(4.6)} = \frac{1}{2} \{ \lambda (v_{i+1,j} - v_{i-1,j}) + (w_{i+1,j} + w_{i-1,j}) \} \quad (0 < i < N, j = 0, 1, 2, \dots).$$

となる。初期条件、境界条件については次のように行なう。

$$\begin{aligned} v_{i,0} &= \psi(ih) & (0 \leq i \leq N) \\ w_{i,0} &= \varphi'(ih) & (0 \leq i \leq N) \\ w_{0,j} &= w_{N,j} = 0 & (j = 1, 2, \dots). \end{aligned}$$

ここで、 $v_{0,j+1}$ ($j = 0, 1, 2, \dots$) について考察する。

$$\frac{\partial}{\partial x}v(0, t) = \frac{\partial^2}{\partial x \partial t}u(0, t) = \frac{\partial^2}{\partial t \partial x}u(0, t) = \frac{\partial}{\partial t}w(0, t) = 0$$

なので、仮想格子点 x_{-1} を導入し $\frac{\partial}{\partial x}v(0, t_j)$ を中心差分商で近似する。

$$\frac{\partial}{\partial x}v(0, t_j) \doteq \frac{v_{1,j} - v_{-1,j}}{2h} = 0$$

より、

$$v_{-1,j} = v_{1,j} \quad (j = 0, 1, 2, \dots)$$

そこで、(5), (6) の式に $i = 0$ で用いると

$$\begin{aligned} v_{0,j+1} &= \frac{1}{2}\{(v_{1,j} + v_{-1,j}) + \lambda(w_{1,j} - w_{-1,j})\} \\ w_{0,j+1} &= \frac{1}{2}\{\lambda(v_{1,j} - v_{-1,j}) + (w_{1,j} + w_{-1,j})\} \\ &\quad (j = 0, 1, 2, \dots). \end{aligned}$$

上の式に $v_{-1,j} = v_{1,j}$ 、 $w_{0,j+1} = 0$ を代入すると、

$$0 = \frac{1}{2}\{\lambda \cdot 0 + (w_{-1,j} + w_{1,j})\}$$

より、

$$w_{-1,j} = -w_{1,j} \quad (j = 0, 1, 2, \dots).$$

ゆえに、

$$v_{0,j+1} = \frac{1}{2}\{(v_{1,j} + v_{-1,j}) + \lambda(w_{1,j} - w_{-1,j})\} = v_{1,j} + \lambda w_{1,j} \quad (j = 0, 1, 2, \dots).$$

同様に $v_{N,j+1}$ ($j = 0, 1, 2, \dots$) についても行なう。

$$\frac{\partial}{\partial x}v(1, t) = \frac{\partial^2}{\partial x \partial t}u(1, t) = \frac{\partial^2}{\partial t \partial x}u(1, t) = \frac{\partial}{\partial t}w(1, t) = 0$$

なので、仮想格子点 x_{N+1} を導入し $\frac{\partial}{\partial x}v(1, t_j)$ を中心差分商で近似する。

$$\frac{\partial}{\partial x}v(1, t_j) \doteq \frac{v_{N+1,j} - v_{N-1,j}}{2h} = 0$$

より、

$$v_{N-1,j} = v_{N+1,j} \quad (j = 0, 1, 2, \dots).$$

そこで、(5), (6) の式に $i = N$ で用いると

$$\begin{aligned} v_{N,j+1} &= \frac{1}{2}\{(v_{N+1,j} + v_{N-1,j}) + \lambda(w_{N+1,j} - w_{N-1,j})\} \\ w_{N,j+1} &= \frac{1}{2}\{\lambda(v_{N+1,j} - v_{N-1,j}) + (w_{N+1,j} + w_{N-1,j})\} \\ &\quad (j = 0, 1, 2, \dots). \end{aligned}$$

上の式に $v_{N-1,j} = v_{N+1,j}$ 、 $w_{N,j+1} = 0$ を代入すると、

$$0 = \frac{1}{2}\{\lambda \cdot 0 + (v_{N+1,j} + v_{N-1,j})\}$$

より、

$$w_{N+1,j} = -w_{N-1,j} \quad (j = 0, 1, 2, \dots).$$

ゆえに、

$$v_{N,j+1} = \frac{1}{2}\{(v_{N+1,j} + v_{N-1,j}) + \lambda(w_{N+1,j} - w_{N-1,j})\} = v_{N-1,j} - \lambda w_{N-1,j} \quad (j = 0, 1, 2, \dots)$$

と表せる。

(4) の問題で求めたのは、 $v_{i,j}$ 、 $w_{i,j}$ ($0 \leq i \leq N$, $j = 0, 1, 2, \dots$) である。そこで問題(B)で求めたい解を $u_{i,j}$ ($0 \leq i \leq N$, $j = 0, 1, 2, \dots$) とすると、 $u_{i,j}$ は、次の式

$$(4.7) \quad u_{i,j+1} = \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \tau v_{i,j} \quad (0 < i < N, j = 0, 1, 2, \dots)$$

$$(4.8) \quad u_{i,0} = \varphi(ih) \quad (0 \leq i \leq N)$$

と、境界条件で求まる。境界条件は、次のようにする。

$$\frac{\partial}{\partial x} u(x, t) = w(x, t)$$

だから、仮想格子点 x_{-1} 、を導入し、 $\frac{\partial}{\partial x} u(0, t_j)$ を中心差分商で近似する。

$$\frac{\partial}{\partial x} u(0, t_j) \doteq \frac{u_{1,j} - u_{-1,j}}{2h} = w_{0,j} = 0$$

より、

$$u_{-1,j} = u_{1,j} \quad (j = 0, 1, 2, \dots).$$

(7) の式を、 $i = 0$ で用いて、

$$u_{0,j+1} = \frac{1}{2}(u_{1,j} + u_{-1,j}) + \tau v_{0,j} = u_{1,j} + \tau v_{0,j} \quad (j = 0, 1, 2, \dots)$$

同様に仮想格子点 x_{N+1} を導入し、 $\frac{\partial}{\partial x}u(1, t_j)$ を中心差分商で近似する。

$$\frac{\partial}{\partial x}u(1, t_j) \doteq \frac{u_{N+1,j} - u_{N-1,j}}{2h} = w_{N,j} = 0$$

より,

$$u_{N+1,j} = u_{N-1,j} \quad (j = 0, 1, 2, \dots).$$

(7) の式を、 $i = N$ で用いて、

$$u_{N,j+1} = \frac{1}{2}(u_{N+1,j} + u_{N-1,j}) + \tau v_{N,j} = u_{N-1,j} + \tau v_{N,j} \quad (j = 0, 1, 2, \dots).$$

4.3.2 まとめて

問題(B)をまとめる。

まず、 $v_{i,j}$, $w_{i,j}$ を求める式は、

$$\left\{ \begin{array}{l} v_{i,j+1} = \frac{1}{2}\{(v_{i+1,j} + v_{i-1,j}) + \lambda(w_{i+1,j} - w_{i-1,j})\} \quad (0 < i < N, j = 0, 1, 2, \dots) \\ w_{i,j+1} = \frac{1}{2}\{\lambda(v_{i+1,j} - v_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})\} \quad (0 < i < N, j = 0, 1, 2, \dots) \\ v_{i,0} = \psi(ih) \quad (0 \leq i \leq N) \\ w_{i,0} = \varphi'(ih) \quad (0 \leq i \leq N) \\ w_{0,j} = w_{N,j} = 0 \quad (j = 1, 2, \dots) \\ v_{0,j+1} = v_{1,j} + \lambda w_{1,j} \quad (j = 0, 1, 2, \dots) \\ v_{N,j+1} = v_{N-1,j} - \lambda w_{N-1,j} \quad (j = 0, 1, 2, \dots) \end{array} \right.$$

となり、ここで求めた $v_{i,j}$ を用いて $u_{i,j}$ を求める式は、

$$\left\{ \begin{array}{l} u_{i,j+1} = \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \tau v_{i,j} \quad (0 < i < N, j = 0, 1, 2, \dots) \\ u_{i,0} = \varphi(ih) \quad (0 \leq i \leq N) \\ u_{0,j+1} = u_{1,j} + \tau v_{0,j} \quad (j = 0, 1, 2, \dots) \\ u_{N,j+1} = u_{N-1,j} + \tau v_{N,j} \quad (j = 0, 1, 2, \dots) \end{array} \right.$$

となる。

もし、境界条件が、 $\frac{\partial}{\partial x}u(0, t) = 0$ 、 $u(1, t) = 0$ 、の時は、問題 (A), (B) を組合せ、 $v_{i,j}$, $w_{i,j}$ を求める式は、

$$\left\{ \begin{array}{l} v_{i,j+1} = \frac{1}{2}\{(v_{i+1,j} + v_{i-1,j}) + \lambda(w_{i+1,j} - w_{i-1,j})\} \quad (0 < i < N, \quad j = 0, 1, 2, \dots) \\ w_{i,j+1} = \frac{1}{2}\{\lambda(v_{i+1,j} - v_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})\} \quad (0 < i < N, \quad j = 0, 1, 2, \dots) \\ v_{i,0} = \psi(ih) \quad (0 \leq i \leq N) \\ w_{i,0} = \varphi'(ih) \quad (0 \leq i \leq N) \\ w_{0,j} = v_{N,j} = 0 \quad (j = 1, 2, \dots) \\ v_{0,j+1} = v_{1,j} + \lambda w_{1,j} \quad (j = 0, 1, 2, \dots) \\ w_{N,j+1} = -\lambda v_{N-1,j} + w_{N-1,j} \quad (j = 0, 1, 2, \dots) \end{array} \right.$$

$u_{i,j}$ を求める式は、

$$\left\{ \begin{array}{l} u_{i,j+1} = \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \tau v_{i,j} \quad (0 < i < N, \quad j = 0, 1, 2, \dots) \\ u_{i,0} = \varphi(ih) \quad (0 \leq i \leq N) \\ u_{0,j+1} = u_{1,j} + \tau v_{0,j} \quad (j = 0, 1, 2, \dots) \\ u_{N,j} = 0 \quad (j = 0, 1, 2, \dots) \end{array} \right.$$

となる。

4.4 エネルギーの保存則について

4.4.1 エネルギーの定義

波動方程式の解、 u の時刻 t におけるエネルギー $E(t)$ は次のように定義される。

$$E(t) = \frac{1}{2} \int_0^1 \left(\frac{\partial u}{\partial t} \right)^2 + \left(\frac{\partial u}{\partial x} \right)^2 dx \quad \dots(\ast)$$

そこで、離散化された問題の解 $\begin{pmatrix} v_{i,j} \\ w_{i,j} \end{pmatrix}$ で時刻 $j\tau$ におけるエネルギー E_j を考察

する。

まず、

$$(4.9) \quad \int_0^1 f(x) dx$$

を、今まで使っていた h 、 x_i を用いて近似してみる。

(9) の式は、 $x = 0$ 、 $x = 1$ 、 $y = f(x)$ 及び、 x 軸で囲まれた面積である。それなので、図 1 の様に面積を区切る。すると、求める面積は上底が $f(x_i)$ 、下底が $f(x_{i+1})$ 、高さ h の台形的面積の和で近似できる ($0 \leq i \leq N - 1$)。この近似は N が大きいほど誤差は少なくなる。

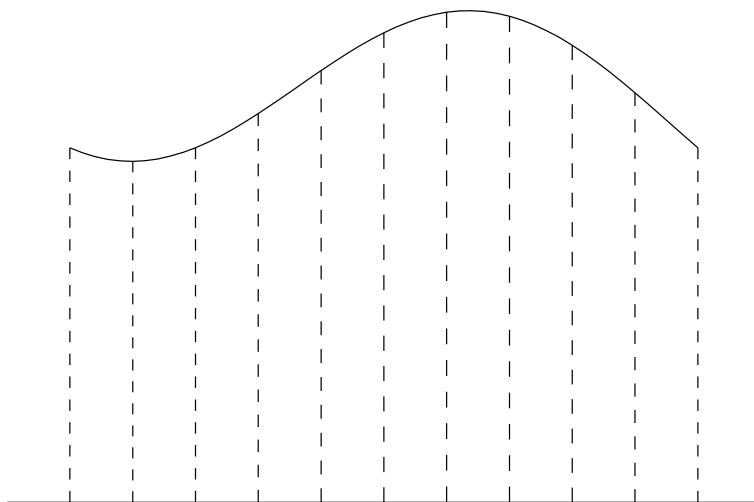


図 4.1: 面積の区切り方

よって、

$$\begin{aligned} \int_0^1 f(x) dx &\doteq \frac{f(x_0) + f(x_1)}{2} h + \frac{f(x_1) + f(x_2)}{2} h \\ &+ \dots + \frac{f(x_{N-2}) + f(x_{N-1})}{2} h + \frac{f(x_{N-1}) + f(x_N)}{2} h \\ &= \left(\frac{1}{2} (f(x_0) + f(x_N)) + \sum_{i=1}^{N-1} f(x_i) \right) h \end{aligned}$$

と近似できる。だから (※) は、

$$\begin{aligned}
E_j &= \frac{h}{2} \left(\frac{1}{2}(v_{0,j}^2 + v_{N,j}^2) + \sum_{i=1}^{N-1} v_{i,j}^2 + \frac{1}{2}(w_{0,j}^2 + w_{N,j}^2) + \sum_{i=1}^{N-1} w_{i,j}^2 \right) \\
&= \frac{h}{2} \left(\frac{1}{2}(v_{0,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{N,j}^2) + \sum_{i=1}^{N-1} (v_{i,j}^2 + w_{i,j}^2) \right)
\end{aligned}$$

と近似できる。

4.4.2 λ の値を変えた解析

具体的にコンピューターで E_j を計算してみると、次の様な結果を得る。
Dirichlet 境界条件の問題 (A) の時で、図 2、3、4 のグラフは、縦軸にエネルギー E_j を取り、横軸に時間 $t = j\tau$ を取ったものである。

- (a) $\lambda = 1$
一定値を取る。

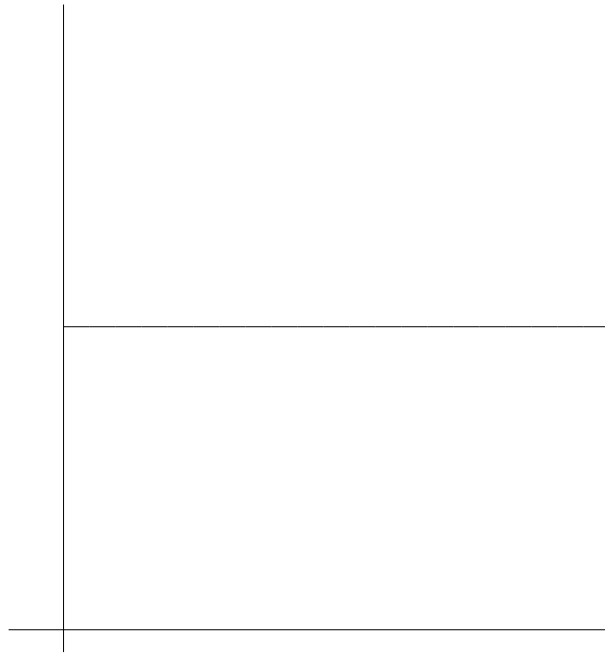


図 4.2: $N = 100, \lambda = 0.5, t = 100$ まで

(b) $0 < \lambda < 1$
減少する。

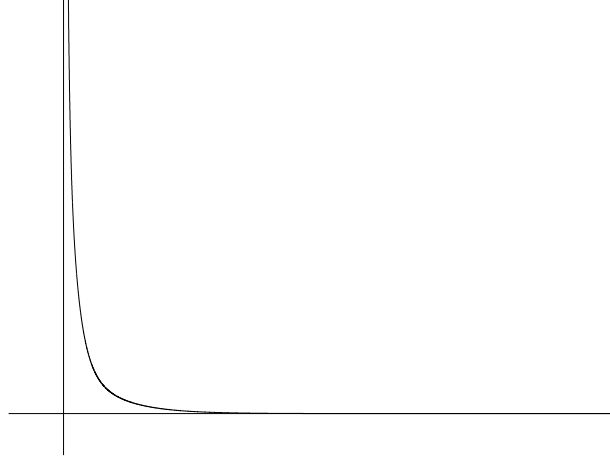


図 4.3: $N = 100, \lambda = 0.5, t = 100$ まで

$\lambda = 1$ のときの解析

(a) $\lambda = 1$ の時

E_{j+1} を、解析的に計算してみる。まず、(2), (3) (5), (6) でも同じ) を用いて、

$$\begin{aligned}
 v_{i,j+1}^2 + w_{i,j+1}^2 &= \frac{1}{4} \{ (v_{i+1,j} + v_{i-1,j})^2 + 2\lambda(v_{i+1,j} + v_{i-1,j})(w_{i+1,j} - w_{i-1,j}) + \lambda^2(w_{i+1,j} - w_{i-1,j})^2 \} \\
 &+ \frac{1}{4} \{ \lambda^2(v_{i+1,j} - v_{i-1,j})^2 + 2\lambda(v_{i+1,j} - v_{i-1,j})(w_{i+1,j} + w_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})^2 \} \\
 &= \frac{1}{4} \{ (1 + \lambda^2)(v_{i+1,j}^2 + v_{i-1,j}^2) + (1 + \lambda^2)(w_{i+1,j}^2 + w_{i-1,j}^2) \} \\
 &+ \frac{1}{4} \{ 2(1 - \lambda^2)(v_{i+1,j}v_{i-1,j} + w_{i+1,j}w_{i-1,j}) + 4\lambda(v_{i+1,j}w_{i+1,j} - v_{i-1,j}w_{i-1,j}) \}.
 \end{aligned}$$

(c) $\lambda > 1$
発散する。

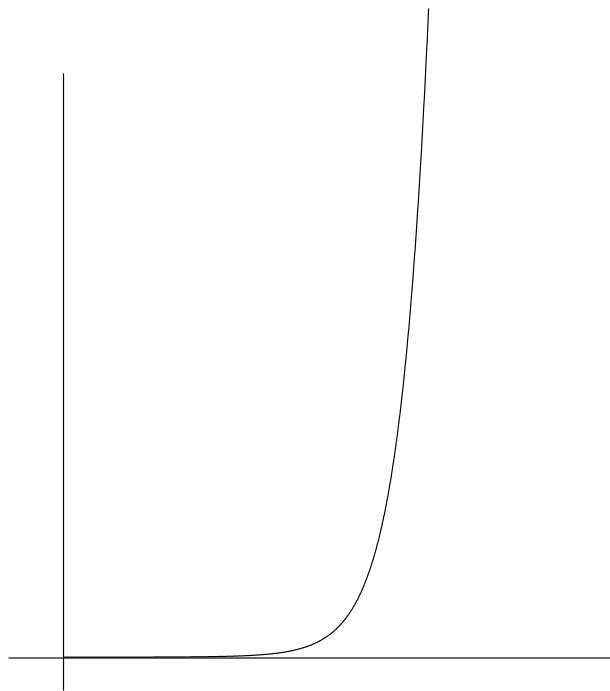


図 4.4: $N = 1000$, $\lambda = 1.01$, $t = 1$ まで

そこで、

$$\begin{aligned}
4 \sum_{i=1}^{N-1} (v_{i,j+1}^2 + w_{i,j+1}^2) &= \sum_{i=1}^{N-1} \{(1 + \lambda^2)(v_{i+1,j}^2 + v_{i-1,j}^2 + w_{i+1,j}^2 + w_{i-1,j}^2)\} \\
&+ \sum_{i=1}^{N-1} \{2(1 - \lambda^2)(v_{i+1,j}v_{i-1,j} + w_{i+1,j}w_{i-1,j}) + 4\lambda(v_{i+1,j}w_{i+1,j} - v_{i-1,j}w_{i-1,j})\} \\
&= (1 + \lambda^2)(v_{0,j}^2 + v_{1,j}^2 + v_{N-1,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{1,j}^2 + w_{N-1,j}^2 + w_{N,j}^2) \\
&+ 2(1 + \lambda^2) \sum_{i=2}^{N-2} (v_{i,j}^2 + w_{i,j}^2) + 2(1 - \lambda^2) \sum_{i=1}^{N-1} (v_{i+1,j}v_{i-1,j} + w_{i+1,j}w_{i-1,j}) \\
&+ 4\lambda(v_{N,j}w_{N,j} + v_{N-1,j}w_{N-1,j} - v_{1,j}w_{1,j} - v_{0,j}w_{0,j}).
\end{aligned}$$

$\lambda = 1$ を代入して、

$$\begin{aligned}
4 \sum_{i=1}^{N-1} (v_{i,j+1}^2 + w_{i,j+1}^2) &= 2(v_{0,j}^2 + v_{1,j}^2 + v_{N-1,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{1,j}^2 + w_{N-1,j}^2 + w_{N,j}^2) \\
&+ 4 \sum_{i=2}^{N-2} (v_{i,j}^2 + w_{i,j}^2) + 4(v_{N,j}w_{N,j} + v_{N-1,j}w_{N-1,j} - v_{1,j}w_{1,j} - v_{0,j}w_{0,j}) \\
&= 4 \left(\frac{1}{2}(v_{0,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{N,j}^2 - w_{1,j}^2 - w_{N-1,j}^2 - w_{1,j}^2 - w_{N-1,j}^2) \right) \\
&+ 4 \left(\sum_{i=1}^{N-1} (v_{i,j}^2 + w_{i,j}^2) + (v_{N,j}w_{N,j} + v_{N-1,j}w_{N-1,j} - v_{1,j}w_{1,j} - v_{0,j}w_{0,j}) \right)
\end{aligned}$$

より

$$\begin{aligned}
\sum_{i=1}^{N-1} (v_{i,j+1}^2 + w_{i,j+1}^2) &= \frac{1}{2} \{(v_{0,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{N,j}^2) - (w_{1,j}^2 + w_{N-1,j}^2 + w_{1,j}^2 + w_{N-1,j}^2)\} \\
&+ \sum_{i=1}^{N-1} (v_{i,j}^2 + w_{i,j}^2) + (v_{N,j}w_{N,j} + v_{N-1,j}w_{N-1,j} - v_{1,j}w_{1,j} - v_{0,j}w_{0,j})
\end{aligned}$$

E_{j+1} と E_j 差を取り、上の式を代入すると

$$\begin{aligned}
E_{j+1} - E_j &= \frac{h}{2} \left(\frac{1}{2}(v_{0,j+1}^2 + v_{N,j+1}^2 + w_{0,j+1}^2 + w_{N,j+1}^2) + \sum_{i=1}^{N-1} (v_{i,j+1}^2 + w_{i,j+1}^2) \right) \\
&\quad - \frac{h}{2} \left(\frac{1}{2}(v_{0,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{N,j}^2) + \sum_{i=1}^{N-1} (v_{i,j}^2 + w_{i,j}^2) \right) \\
&= \frac{h}{2} \left(\frac{1}{2} \{ (v_{0,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{N,j}^2) - (w_{1,j}^2 + w_{N-1,j}^2 + w_{1,j}^2 + w_{N-1,j}^2) \} \right) \\
&\quad + \frac{h}{2} (v_{N,j}w_{N,j} + v_{N-1,j}w_{N-1,j} - v_{1,j}w_{1,j} - v_{0,j}w_{0,j}) \\
&= \frac{h}{4} (v_{0,j+1}^2 + v_{N,j+1}^2 + w_{0,j+1}^2 + w_{N,j+1}^2 + 2(v_{N,j}w_{N,j} - v_{0,j}w_{0,j})) \\
&\quad - \frac{h}{4} ((v_{1,j} + w_{1,j})^2 + (v_{N-1,j} - w_{N-1,j})^2)
\end{aligned}$$

この式に、問題 (A) の $\lambda = 1$ の時の離散化した境界条件を代入すると、

$$\begin{aligned}
E_{j+1} - E_j &= \frac{h}{4} (0 + 0 + w_{0,j+1}^2 + w_{N,j+1}^2 + 2(0 - 0)) \\
&\quad - \frac{h}{4} ((w_{0,j+1})^2 + (-w_{N,j+1})^2) \\
&= 0.
\end{aligned}$$

また、問題 (B) の $\lambda = 1$ の時離散化した境界条件を代入すると、

$$\begin{aligned}
E_{j+1} - E_j &= \frac{h}{4} (v_{0,j+1}^2 + v_{N,j+1}^2 + 0 + 0 + 2(0 - 0)) \\
&\quad - \frac{h}{4} ((v_{0,j+1})^2 + (v_{N,j+1})^2) \\
&= 0.
\end{aligned}$$

また、 $\frac{\partial}{\partial x}u(0, t) = 0$ 、 $u(1, t) = 0$ 、の時の $\lambda = 1$ の離散化した境界条件を代入すると、

$$\begin{aligned}
E_{j+1} - E_j &= \frac{h}{4} (v_{0,j+1}^2 + 0 + 0 + w_{N,j+1}^2 + 2(0 - 0)) \\
&\quad - \frac{h}{4} ((v_{0,j+1})^2 + (-w_{N,j+1})^2) \\
&= 0.
\end{aligned}$$

ゆえに、いずれの場合も $\lambda = 1$ の時

$$E_{j+1} = E_j$$

となり、エネルギー保存則は成り立つ。

$\lambda = 1$ 以外の時の数値解析

(b) $0 < \lambda < 1$ の時

$$E_{j+1} < E_j$$

が予想される。そこで、縦軸にエネルギー E_j の底 10 の対数 $y = \log_{10} E_j$ を取り、横軸に $t = j\tau$ としたグラフは、図 5 である。

$t = 0$ に近いところは誤差として、このグラフは直線である。その傾きを計算する。

適当に選んだ 2 点 $(t, y) = (17.425, -1.999971) = (95.170000, -6.999750)$ から傾きを求めると、

$$\text{傾き} = \frac{(-6.999750) - (-1.999971)}{(95.170000) - (17.425)} = -0.064310$$

となる。よって、グラフの直線部を延長し $t = 0$ の y の値を C とすると、 E_j と $j\tau$ の関係は、

$$\begin{aligned} y &= -0.06431t + C \\ \log_{10} E_j &= -0.06431(j\tau) + C \\ E_j &= C' \cdot 10^{-0.06431(j\tau)} \quad (C' = 10^C) \\ E_j &= \frac{C'}{10^{0.06431(j\tau)}} \quad (j \text{ が大きいところ}) \end{aligned}$$

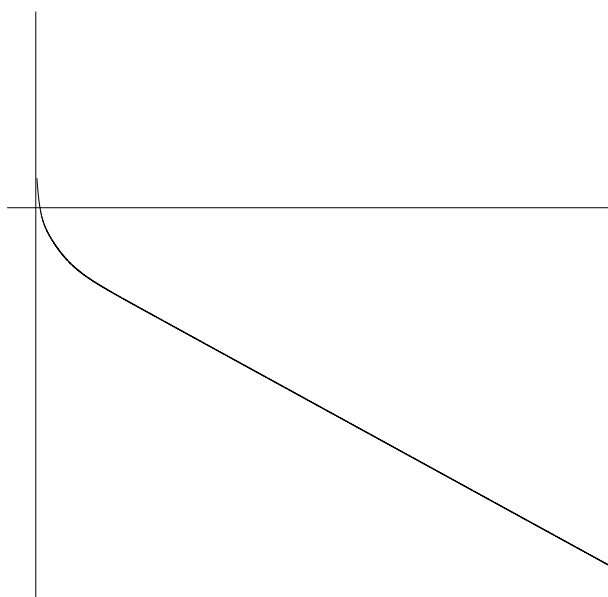


図 4.5: $N = 100, \lambda = 0.5, t = 100$ まで

j が大きいとき E_{j+1} と E_j の比を取ると、

$$\begin{aligned} \frac{E_{j+1}}{E_j} &= \frac{\frac{C'}{10^{0.06431(j+1)\tau}}}{\frac{C'}{10^{0.06431(j)\tau}}} \\ &= \frac{1}{10^{0.06431\tau}} \end{aligned}$$

よって、 E_j は、公比 $\frac{1}{10^{0.06431\tau}}$ の等比数列である。 $N = 100, \lambda = 0.5$ の時、 $\tau = 0.005$ になり、公比の値を計算すると、公比が $0.99926 < 1$ となり、

$$\lim_{j \rightarrow \infty} E_j = 0$$

となる。つまり、時間が経つにつれ E_j は 0 に近づき無限時間経つと 0 になる。

(c) $\lambda > 1$ の時

$$E_{j+1} > E_j$$

が予想される。そこで、縦軸にエネルギー E_j の底 10 の対数 $y = \log_{10} E_j$ を取り、横軸に $t = j\tau$ としたグラフは、図 6 のものである。

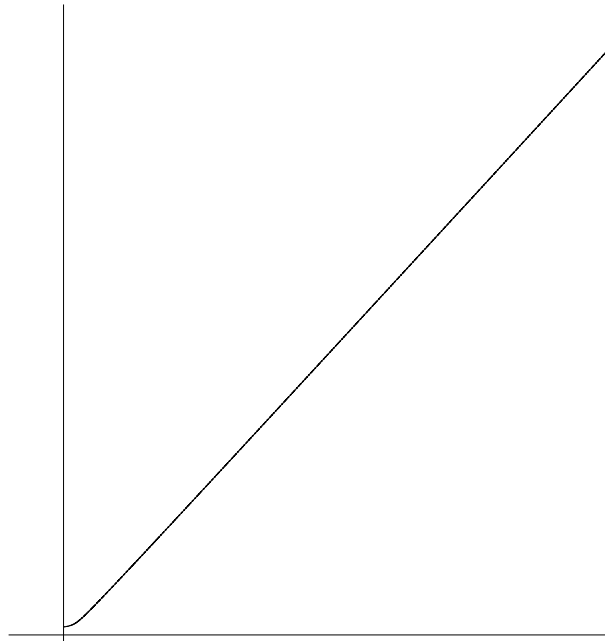


図 4.6: $N = 100$, $\lambda = 1.01$, $t = 100$ まで

$t = 0$ に近いところは誤差として、このグラフは直線である。その傾きを計算する。

適当に選んだ 2 点 $(t, y) = (36.289300, 29.995966) = (94.970300, 79.999598)$ から傾きを求めると、

$$\text{傾き} = \frac{79.999598 - 29.995966}{94.970300 - 36.289300} = 0.852126$$

よって、直線のグラフを延長し $t = 0$ の y の値を C とすると E_j と $j\tau$ の関係は、

$$\begin{aligned}y &= 0.852126t + C \\ \log_{10} E_j &= 0.852126(j\tau) + C \\ E_j &= C' \cdot 10^{0.852126(j\tau)} \quad (C' = 10^C, j \text{ が大きいところ})\end{aligned}$$

j が大きいとき E_{j+1} と E_j の比を取ると、

$$\begin{aligned}\frac{E_{j+1}}{E_j} &= \frac{C' \cdot 10^{0.852126(j+1)\tau}}{C' \cdot 10^{0.852126(j\tau)}} \\ &= 10^{0.852126\tau}\end{aligned}$$

よって、 E_j は、公比 $10^{0.852126\tau}$ の等比数列である。 $N = 100$ 、 $\lambda = 1.01$ の時、 $\tau = 0.010100$ になり公比の値を計算すると、公比が $1.02001 > 1$ となり、

$$\lim_{j \rightarrow \infty} E_j = \infty$$

となる。つまり、時間が経つにつれ E_j の値は大きくなり無限時間経つと ∞ に発散する。

4.2 のまとめ

λ の値を変えてみた時の変化を表にしてみる。

λ	τ	傾き	公比	エネルギー
0.1	0.001	-0.424413	0.999023	0 に収束
0.2	0.002	-0.205773	0.999053	0 に収束
0.3	0.003	-0.130034	0.999102	0 に収束
0.4	0.004	-0.090020	0.999171	0 に収束
0.5	0.005	-0.064298	0.999260	0 に収束
0.6	0.006	-0.045734	0.999368	0 に収束
0.7	0.007	-0.031495	0.999492	0 に収束
0.8	0.008	-0.021902	0.999597	0 に収束
0.9	0.009	-0.018834	0.999660	0 に収束
1.0	0.010	0.000000	1.000000	13.570706 で一定
1.1	0.011	7.510883	1.209540	∞ に発散
1.2	0.012	13.178307	1.439260	∞ に発散

傾きは、縦軸を $y = \log_{10} E_j$ 、横軸を $t = j\tau$ とした時のグラフの傾き、公比は、等比数列 E_j の公比である。Dirichlet 境界条件 (問題 (A)) で、 $N = 100$,

$$\varphi(x) = \begin{cases} \sin(5\pi x) & (0.4 < x < 0.6) \\ 0 & (0.4 < x < 0.6 \text{ 以外}) \end{cases}$$

$$\psi(x) = 0$$

の結果である。

$0 < \lambda < 1$ のとき、さらに詳しく λ と公比の関係を調べる (実験式を求める)。まず、下の表を作った。

λ	公比 a	$1.0 - \text{公比 } a$
0.1	0.376259	0.623741
0.2	0.622679	0.377321
0.3	0.741213	0.258787
0.4	0.812747	0.187253
0.5	0.862384	0.137616
0.6	0.899994	0.100006
0.7	0.929982	0.070018
0.8	0.950863	0.049137
0.9	0.962921	0.037079

公比 a は、ある時刻 t のエネルギー E_t と、ある時刻 t から時間を τ ずつ増やし、 $t + \tau, t + 2\tau, \dots, t + 1$ になった時のエネルギー E_{t+1} の比 $\frac{E_{t+1}}{E_t}$ の値である。

公比 a は、次の計算方法で求めた。

$$\text{公比 } a = \left(\frac{E_{j+1}}{E_j} \right)^{\frac{1.0}{\tau}}$$

この表をもとに、次のグラフを書いた。縦軸に $y = \log_{10}(1.0 - \text{公比 } a)$ 、横軸に $x = \lambda$ を取ったものである。

このグラフは、ほぼ直線になり傾きは $\frac{(-1.154791) - (-0.727570)}{0.7 - 0.4} \doteq -1.42$ となる。だから、 λ と公比 a の関係は、 C を任意定数として

$$\log_{10}(1.0 - \text{公比 } a) = -1.42\lambda + C$$

$$(1.0 - \text{公比 } a) = 10^{-1.42\lambda + C}$$

$$\text{公比 } a = C' \cdot 10^{-1.42\lambda} + 1.0 \quad (C' = -10^C)$$

$\lambda = 0.5$ の時、公比 $a=0.862384$ より

$$0.862 \doteq C' \cdot 10^{-0.712} + 1.0$$

$$C' \cdot 10^{-0.712} = -0.137$$

$$C' = -0.137 \cdot 10^{0.712}$$

$$C' = -0.709$$

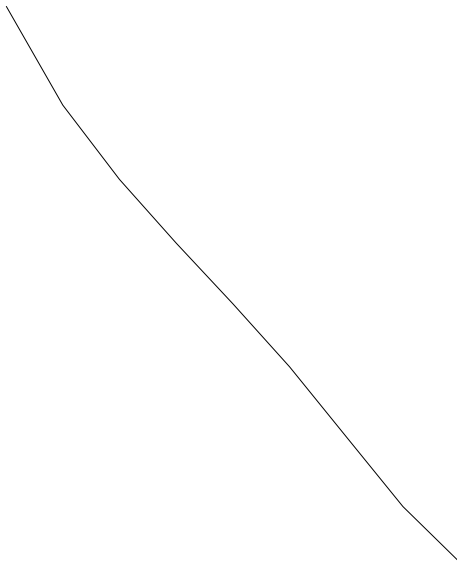


図 4.7: ほぼ直線

ゆえに、

$$\text{公比 } a = -0.709 \cdot 10^{-1.42\lambda} + 1.0$$

実際、値を代入してみると、だいたい表のとおりになる。
(小数点第2位で、少し値がずれる。)

4.4.3 数値解析 (N の値を変えて)

エネルギー E_j は、 N の値とどんな関係があるか、 N の値を変えてみたときのエネルギー E_j の変化をまとめた表である。境界条件、初期条件、傾き、公比は上の表で用いたものを採用する。

(a) $\lambda = 1.0$ として N の値を変えてみた表である。

N	エネルギー
100	13.570706
500	12.583746
1000	12.460376
5000	12.361680
10000	12.349343
50000	12.339473
100000	12.338239

N の値が大きいくほど、エネルギーは本来の値に近づく様に近似しているので、エネルギー E_j は、 N の値が小さいと本来の値より大きくなってしまふ。

(b) $0 < \lambda < 1$ のサンプルとして、 $\lambda = 0.5$ を採用し N の値を変えた表である。

N	τ	傾き	公比	エネルギー
100	0.005000	-0.064297	0.999260	0 に収束
300	0.001667	-0.025441	0.999902	0 に収束
500	0.001000	-0.023330	0.999946	0 に収束
800	0.000625	-0.020491	0.999971	0 に収束
1000	0.000500	-0.018531	0.999979	0 に収束

N が大きくなるほど、公比は 1 に近づく。その結果、 $N = \infty$ とすればエネルギーの保存則が成り立つだろう（結果からの予想）。しかし、数値解析を行なう時は $N = \infty$ と取ることは不可能。また、 $h = \frac{1}{N}$ なので、 $N = \infty$ と取ると、 $h = 0$ となり、分母 = 0 となる式があるので実用的ではない。

もっと詳しく N と公比の関係調べ。

N	公比 a	$1.0 - \text{公比 } a$
100	0.862384	0.137616
300	0.942904	0.057096
500	0.947431	0.052569
800	0.954659	0.045341
1000	0.958869	0.041131

公比 a は、ある時刻 t のエネルギー E_t と、ある時刻 t から時間を τ ずつ増やし、 $t + \tau, t + 2\tau, \dots, t + 1$ になった時のエネルギー E_{t+1} の比 $\frac{E_{t+1}}{E_t}$ の値である。公比 a は、次の計算で求めた。

$$\text{公比 } a = \left(\frac{E_{j+1}}{E_j} \right)^{\frac{1.0}{\tau}}$$

この表をもとに、次のグラフを書いた。縦軸に $y = \log_{10}(1.0 - \text{公比 } a)$ 、横軸に $x = N$ を取ったものである。(表以外の $N = 100 \sim N = 300$ までの適当な値も取ってグラフを書いた。)

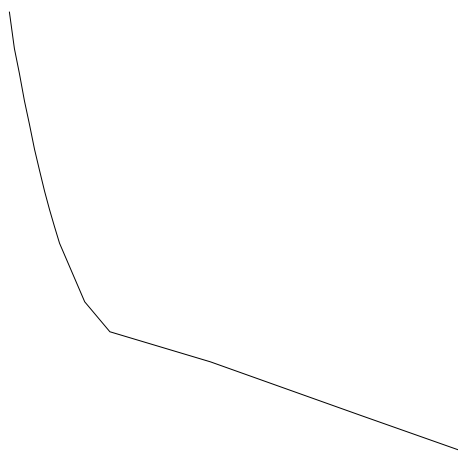


図 4.8: N が十分大きいと直線

$x = N = 100$ に近いところは誤差と考えると、 N が十分大きいとき、このグラフは、ほぼ直線となる。傾きは、 $\frac{(-1.385835) - (-1.279268)}{1000 - 500} = -0.000213$ より、 N と公比 a の関係は、 C を任意定数として

$$\begin{aligned}\log_{10}(1.0 - \text{公比 } a) &= -0.000213N + C \\ (1.0 - \text{公比 } a) &= 10^{-0.000213N+C} \\ \text{公比 } a &= C' \cdot 10^{-0.000213N} + 1.0 \quad (C' = -10^C, N \text{ が大きいところ})\end{aligned}$$

$N = 1000$ の時、公比 $a=0.958869$ より

$$\begin{aligned}0.958869 &= C' \cdot 10^{-0.213} + 1.0 \\ C' \cdot 10^{-0.213} &= -0.041131 \\ C' &= -0.041131 \cdot 10^{0.213} \\ C' &= -0.06716906\end{aligned}$$

ゆえに、

$$\text{公比 } a = -0.06716906 \cdot 10^{-0.000213N} + 1.0$$

実際、値を代入してみると、だいたい表のとおりになる。

(c) $\lambda > 1$ のサンプルとして、 $\lambda = 1.01$ を採用し N の値を変えた表である。

N	τ	傾き	公比	エネルギー
100	0.010100	0.852126	1.020015	∞ に発散
500	0.002020	4.271246	1.020065	∞ に発散
800	0.001262	6.838410	1.020070	∞ に発散
1000	0.001010	8.549850	1.020083	∞ に発散

N が小さいほど、公比は 1 に近づく。その結果、 $N = 0$ とすればエネルギーの保存則が成り立つだろう（結果からの予想）。 $N = 0$ では、分割をしないので差分方程式すら作れず全く意味がない。

4.5 安定性

4.5.1 λ の値と安定性

λ の値と安定性は、どうなるか数値解析をしてみた表である。境界条件は、問題 (A), (B) 及び、 $\frac{\partial}{\partial x}u(0,t) = 0$ 、 $u(1,t) = 0$ の、どの時でも、初期条件を境界条件を満たす任意の関数を取れば、次の表の通りになる。

($t = j\tau = 300$ まで。波動の状態は筆者の判断によるもの)

λ	安定性	波動の状態
0.1	安定	波動は止まっていく
0.3	安定	波動は止まっていく
0.5	安定	波動は止まっていく
0.7	安定	波動は止まっていく
0.9	安定	波動は止まっていく
1.0	安定	波動は止まらず、発散しない
1.1	不安定	発散
1.2	不安定	発散

この結果より、エネルギーと安定性には、共通するものがある。

(N の値を大きくするにつれエネルギーが減少する $0 < \lambda < 1$ とき、波動が止まっていくのは期待通り。)

4.6 まとめ

一次元の波動方程式を、フリードリクス (Friedrichs) の差分法を用いて解いた差分解の安定性は $0 < \lambda \leq 1$ (初期条件問題の場合の証明は参考文献,4、にある) である。 $0 < \lambda < 1$ の時、差分解は安定してるが時間が立つと 0 になり、波動は止まってしまい、エネルギーも 0 になる。 $\lambda = 1$ の時、差分解は安定し、エネルギーの保存則が成り立つ (これは、証明できた)。 $\lambda > 1$ の時、差分解は不安定で時間が立つと発散する。また、波動もエネルギーも発散する。よって、波動方程式の厳密解に最も近い差分解を求めるには、 $\lambda = 1$ を採用すれば良いだろう。

4.7 プログラム

問題 (A) を解くためのプログラムである。
(これから載せるプログラムを実行するには、 $N = 500$ 以上取ることがお勧めである。それ以下だと波動はギザギザになる。)

```
/*
 * move1d-e.c -- 1次元波動方程式の初期値境界値問題を解く。
 *   コンパイル:  ccx move1d-e.c
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    int i, j, Jmax, skip, N;
    double tau, h, lambda, dt, Tmax;
    double *u, *newu, *v, *newv, *w, *neww;
    double f(double), g(double), fx(double);

    /* N, λ を入力する */
    printf("区間の分割数 N = "); scanf("%d", &N);
    printf(" λ = "); scanf("%lf", &lambda);

    /* h を計算する */
    h = 1.0 / N;
    tau = lambda * h;
    printf(" τ = %f\n", tau);
    printf(" dt = "); scanf("%lf", &dt);
    skip = dt / tau;

    /* 最終時刻を入力する */
    printf("最終時刻 Tmax = "); scanf("%lf", &Tmax);

    /* ベクトル u, newu, v, newv, w, neww を用意する */
    u = malloc(sizeof(double) * (N+1));
    newu = malloc(sizeof(double) * (N+1));
    v = malloc(sizeof(double) * (N+1));
    newv = malloc(sizeof(double) * (N+1));
    w = malloc(sizeof(double) * (N+1));
    neww = malloc(sizeof(double) * (N+1));

    for (i = 0; i <= N; i++)
        u[i] = f(i * h);

    for (i = 0; i <= N; i++)
```



```

v[i] = g(i * h);

for (i = 0; i <=N; i++)
    w[i] = fx(i * h);

/* ウィンドウを出す */
openpl();
fspace2(-0.1, -1.1, 1.1, 1.1);

/* t=0 の状態を表示する */
fmove(0.0, u[0]);
for (i = 1; i <= N; i++)
    fcont(i * h, u[i]);

/* ループを何回まわるか計算する */
Jmax = Tmax / tau + 0.5;

/* Dirichlet 境界条件 */
u[0] = u[N] = 0.0;
v[0] = v[N] = 0.0;

for (j = 0; j <= Jmax; j++) {

    /* 差分方程式 (j -> j+1) */
    for (i = 1; i < N; i++){
        newv[i]= (v[i+1] + lambda* w[i+1])/ 2.0
            +(v[i-1] - lambda* w[i-1])/ 2.0;
        neww[i]=( lambda * v[i+1] + w[i+1])/ 2.0
            +(-lambda * v[i-1] + w[i-1])/ 2.0;
        newu[i]=( u[i+1] + u[i-1])/ 2.0 + tau * v[i];
    }

    neww[0] = lambda * v[1] + w[1];
    neww[N] = -lambda * v[N-1] + w[N-1];

    for (i = 1; i < N; i++){
        u[i] = newu[i];
        v[i] = newv[i];
        w[i] = neww[i];
    }

    w[0] = neww[0];
    w[N] = neww[N];

    if (j % skip == 0) {
        /* この時刻 (t=(j+1) τ) の状態を表示する */
        erase();
        fmove(0.0, u[0]);
        for (i = 1; i <= N; i++)

```

```

        fcont(i * h, u[i]);
    }
}
printf("終了しました。ウィンドウをクリックして下さい。 \n");
closepl();
}

```

```

double f(double x)
{
    if (x < 0.4)
        return 0.0;
    if (x > 0.6)
        return 0.0;
    else
        return sin(5.0 * M_PI * x);
}

```

```

double g(double x)
{
    return 0.0;
}

```

```

double fx(double x)
{
    if (x < 0.4)
        return 0.0;
    if (x > 0.6)
        return 0.0;
    else
        return 5.0 * M_PI * cos(5.0 * M_PI * x);
}

```

問題 (B) を解くためのプログラムである。

```

/* move1d-e.c -- 1次元波動方程式の初期値境界値問題を解く。
 * コンパイル: ccx move1d-e.c
 * du/dx = 0 ( x= 0.0 )
 * du/dx = 0 ( x= 1.0 )
 * の条件で解く。
 */

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

int main()
{
    int i, j, Jmax, skip, N;
    double tau, h, lambda, dt, Tmax, a, b, q, s;
    double *u, *newu, *v, *newv, *w, *neww;

```

```

double f(double), g(double), fx(double);

/* N, λ を入力する */
printf("区間の分割数 N = "); scanf("%d", &N);
printf(" λ = "); scanf("%lf", &lambda);

/* h を計算する */
h = 1.0 / N;
tau = lambda * h;
printf("τ=%f\n", tau);
printf(" dt = "); scanf("%lf", &dt);
skip = dt / tau;

/* 最終時刻を入力する */
printf("最終時刻 Tmax = "); scanf("%lf", &Tmax);

/* ベクトル u, newu, v, newv, w, neww を用意する */
u = (double *)malloc(sizeof(double) * (N+1));
newu = (double *)malloc(sizeof(double) * (N+1));
v = (double *)malloc(sizeof(double) * (N+1));
newv = (double *)malloc(sizeof(double) * (N+1));
w = (double *)malloc(sizeof(double) * (N+1));
neww = (double *)malloc(sizeof(double) * (N+1));

for (i = 0; i <= N; i++)
    u[i] = f(i * h);

for (i = 0; i <=N; i++)
    v[i] = g(i * h);

for (i = 0; i <=N; i++)
    w[i] = fx(i * h);

/* ウィンドウを出す */
openpl();
fspace2(-0.1, -1.1, 1.1, 1.1);

/* t=0 の状態を表示する */
fmove(0.0, u[0]);
for (i = 1; i <= N; i++)
    fcont(i * h, u[i]);

/* ループを何回まわるか計算する */
Jmax = (int)rint(Tmax / tau);

/* Dirichlet 境界条件 */
w[0] = 0.0;
w[N] = 0.0;

```

```

for (j = 0; j <= Jmax; j++) {

    /* 差分方程式 (j -> j+1) */
    for (i = 1; i < N; i++){
        newv[i]= (v[i+1] + lambda* w[i+1])/ 2.0
            +(v[i-1] - lambda* w[i-1])/ 2.0;
        neww[i]=( lambda * v[i+1] + w[i+1])/ 2.0
            +( -lambda * v[i-1] + w[i-1])/ 2.0;
        newu[i]=( u[i+1] + u[i-1])/ 2.0 + tau * v[i];
    }

    newv[0]= lambda * w[1] + v[1];
    newu[0]= u[1] + tau * v[0];
    newv[N]= -lambda * w[N-1] + v[N-1];
    newu[N]= u[N+1] + tau * v[N];
    neww[0]= w[0];
    neww[N]= w[N];

    for (i = 0; i <= N; i++){
        u[i] = newu[i];
        v[i] = newv[i];
        w[i] = neww[i];
    }

    if (j % skip == 0) {
        /* この時刻 (t=(j+1) τ) の状態を表示する */
        erase();
        fmove(0.0, u[0]);
        for (i = 1; i <= N; i++)
            fcont(i * h, u[i]);
    }
}
printf("終わりました。ウィンドウをクリックして下さい。 \n");
closepl();

}

double f(double x)
{
    if (x < 0.2)
        return 0.0;
    if (x > 0.6)
        return 0.0;
    else
        return sin(5.0 * M_PI * x);
}

double g(double x)
{

```

```
    return 0.0;
}

double fx(double x)
{
    if (x < 0.2)
        return 0.0;
    if (x > 0.6)
        return 0.0;
    else
        return 5.0 * M_PI * cos(5.0 * M_PI * x);
}
```

4.8 参考文献

1. 菊地文雄、山本昌宏 [17]
2. 高見穎郎、河村哲也 [19]
3. 登坂宣好、大西和栄 [20]
4. 山口昌哉、野木達夫 [21]

第5章 桂田メモ

5.1 対称双曲型方程式の問題への変換

5.1.1 対称双曲型微分方程式への変換

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

は

$$v = \frac{\partial u}{\partial t}, \quad w = \frac{\partial u}{\partial x}, \quad \vec{U} = \begin{pmatrix} v \\ w \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

とおくと、

$$\frac{\partial}{\partial t} \vec{U} = A \frac{\partial}{\partial x} \vec{U}$$

という対称双曲型方程式に変換される。

5.1.2 初期条件

区間 $(0, 1)$ における初期値境界値問題を考える。初期条件

$$u(x, 0) = \varphi(x), \quad \frac{\partial u}{\partial t}(x, 0) = \psi(x) \quad (x \in [0, 1])$$

から

$$v(x, 0) = \psi(x), \quad w(x, 0) = \varphi'(x) \quad (x \in [0, 1])$$

が導かれる。

5.1.3 境界条件

残るは境界条件であるが、まず v については比較的簡単である。

v について

Dirichlet 境界条件

$$u(0, t) = \alpha(t), \quad u(1, t) = \beta(t) \quad (t > 0)$$

の場合は

$$\frac{\partial u}{\partial t}(0, t) = \alpha'(t), \quad \frac{\partial u}{\partial t}(1, t) = \beta'(t) \quad (t > 0)$$

であるから v について

$$v(0, t) = \alpha'(t), \quad v(1, t) = \beta'(t) \quad (t > 0)$$

が導かれる。また Neumann 境界条件

$$u_x(0, t) = \gamma(t), \quad u_x(1, t) = \delta(t) \quad (t > 0)$$

の場合は

$$\frac{\partial^2 u}{\partial x \partial t}(0, t) = \frac{\partial^2 u}{\partial t \partial x}(0, t) = \gamma'(t), \quad \frac{\partial^2 u}{\partial x \partial t}(1, t) = \frac{\partial^2 u}{\partial t \partial x}(1, t) = \delta'(t) \quad (t > 0)$$

であるから v について

$$v_x(0, t) = \gamma'(t), \quad v_x(1, t) = \delta'(t) \quad (t > 0)$$

が導かれる。

w について

次は w の境界条件である。この場合は、 u について Neumann 境界条件

$$u_x(0, t) = \gamma(t), \quad u_x(1, t) = \delta(t) \quad (t > 0)$$

の場合は

$$w(0, t) = \gamma(t), \quad w(1, t) = \delta(t) \quad (t > 0)$$

と簡単になる。問題は u について Dirichlet 境界条件

$$u(0, t) = \alpha(t), \quad u(1, t) = \beta(t) \quad (t > 0)$$

を課してある場合である。境界においても波動方程式が成り立っていると考えると、

$$\frac{\partial w}{\partial x}(0, t) = \frac{\partial^2 u}{\partial x^2}(0, t) = \frac{\partial^2 u}{\partial t^2}(0, t) = \alpha''(t) \quad (t > 0),$$

同様にして

$$\frac{\partial w}{\partial x}(1, t) = \beta''(t) \quad (t > 0)$$

を課すことが考えられる (ほんまかいな)。

5.2 Friedrichs の差分法

$$\frac{\partial U}{\partial t} = A \frac{\partial U}{\partial x}$$

を

$$\frac{U_{i,j+1} - \frac{U_{i+1,j} + U_{i-1,j}}{2}}{\tau} = A \frac{U_{i+1,j} - U_{i-1,j}}{2h}$$

と離散化するのが **Friedrichs** の差分法である。

$U = \vec{U} = \begin{pmatrix} v \\ w \end{pmatrix}$, $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ の場合に考えてみる。両辺に τ をかけて $\lambda = \tau/h$ とおくと、

$$\vec{U}_{i,j+1} - \frac{1}{2}(\vec{U}_{i+1,j} + \vec{U}_{i-1,j}) = \frac{1}{2}\lambda A(\vec{U}_{i+1,j} - \vec{U}_{i-1,j}),$$

ゆえに

$$\vec{U}_{i,j+1} = \frac{1}{2}(I + \lambda A)\vec{U}_{i+1,j} + \frac{1}{2}(I - \lambda A)\vec{U}_{i-1,j}.$$

これは成分で書くと

$$\begin{pmatrix} v_{i,j+1} \\ w_{i,j+1} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & \lambda \\ \lambda & 1 \end{pmatrix} \begin{pmatrix} v_{i+1,j} \\ w_{i+1,j} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 1 & -\lambda \\ -\lambda & 1 \end{pmatrix} \begin{pmatrix} v_{i-1,j} \\ w_{i-1,j} \end{pmatrix},$$

すなわち

$$\begin{aligned} v_{i,j+1} &= \frac{1}{2} [(v_{i+1,j} + v_{i-1,j}) + \lambda(w_{i+1,j} - w_{i-1,j})], \\ w_{i,j+1} &= \frac{1}{2} [\lambda(v_{i+1,j} - v_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})] \end{aligned}$$

$i = 1, 2, \dots, N-1$ に対して $v_{i,j+1}$, $w_{i,j+1}$ はこの式を使ってすぐに求められる。
 u について Dirichlet 境界条件

$$u(0, t) = u(1, t) = 0 \quad (t > 0)$$

を課してある場合を考える。まず

$$\frac{\partial u}{\partial t}(0, t) = \frac{\partial u}{\partial t}(1, t) = 0 \quad (t > 0)$$

であるから

$$v(0, t) = v(1, t) = 0 \quad (t > 0)$$

ゆえに $v_{i,j+1}$ については、

$$v_{0,j+1} = v_{N,j+1} = 0 \quad (j = 0, 1, 2, \dots)$$

とするのが自然である。

$w_{0,j+1}$ については、まず

$$\frac{\partial w}{\partial x}(0, t) = \frac{\partial w}{\partial x}(1, t) = 0$$

となることから、いわゆる仮想格子点 x_{-1} を導入して、Neumann 境界条件を中心差分近似して

$$w_{-1,j+1} = w_{1,j+1}$$

を得る。未知数が増えた分は、 $i = 0$ についても差分方程式、すなわち

$$\begin{aligned} v_{0,j+1} &= \frac{1}{2} [(v_{1,j} + v_{-1,j}) + \lambda(w_{1,j} - w_{-1,j})], \\ w_{0,j+1} &= \frac{1}{2} [\lambda(v_{1,j} - v_{-1,j}) + (w_{1,j} + w_{-1,j})] \end{aligned}$$

が成り立つと仮定してつじつまを合わせる。まず第一の式に $v_{0,j+1} = 0$, $w_{1,j} = w_{-1,j}$ を代入して

$$v_{1,j} + v_{-1,j} = 0.$$

これから $v_{-1,j} = -v_{1,j}$ となるから、 $w_{-1,j} = w_{1,j}$ と合わせて第二式に代入して

$$w_{0,j+1} = \frac{1}{2} [\lambda(v_{1,j} + v_{1,j}) + (w_{1,j} + w_{1,j})] = \lambda v_{1,j} + w_{1,j}.$$

$w_{N,j+1}$ についても同様に、

$$v_{N,j+1} = 0,$$

$$w_{N+1,j} = w_{N-1,j},$$

$$\begin{aligned} v_{N,j+1} &= \frac{1}{2} [(v_{N+1,j} + v_{N-1,j}) + \lambda(w_{N+1,j} - w_{N-1,j})], \\ w_{N,j+1} &= \frac{1}{2} [\lambda(v_{N+1,j} - v_{N-1,j}) + (w_{N+1,j} + w_{N-1,j})] \end{aligned}$$

から

$$v_{N+1,j} = -v_{N-1,j},$$

さらに

$$w_{N,j+1} = \frac{1}{2} [\lambda(-v_{N-1,j} - v_{N-1,j}) + (w_{N-1,j} + w_{N-1,j})] = -\lambda v_{N-1,j} + w_{N-1,j}.$$

u を求めるには、 $\frac{\partial u}{\partial t} = v$ を

$$\frac{u_{i,j+1} - \frac{1}{2}(u_{i+1,j} + u_{i-1,j})}{\tau} = v_{i,j} \quad (i = 1, 2, \dots, N-1)$$

と近似して

$$u_{i,j+1} = \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \tau v_{i,j} \quad (i = 1, 2, \dots, N-1)$$

とすれば良い。

まとめると、

Friedrichs の差分法による差分方程式

$$\begin{aligned} u_{i,j+1} &= \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) + \tau v_{i,j} \quad (i = 1, 2, \dots, N-1), \\ v_{i,j+1} &= \frac{1}{2}[(v_{i+1,j} + v_{i-1,j}) + \lambda(w_{i+1,j} - w_{i-1,j})], \quad (i = 1, 2, \dots, N-1), \\ w_{i,j+1} &= \frac{1}{2}[\lambda(v_{i+1,j} - v_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})] \quad (i = 1, 2, \dots, N-1), \\ u_{0,j+1} &= u_{N,j+1} = 0, \\ v_{0,j+1} &= v_{N,j+1} = 0, \\ w_{0,j+1} &= \lambda v_{1,j} + w_{1,j}, \\ w_{N,j+1} &= -\lambda v_{N-1,j} + w_{N-1,j}. \end{aligned}$$

5.3 エネルギー保存則

$$\begin{aligned} v_{i,j+1}^2 + w_{i,j+1}^2 &= \frac{1}{4} \{ [(v_{i+1,j} + v_{i-1,j})^2 + 2\lambda(v_{i+1,j} + v_{i-1,j})(w_{i+1,j} - w_{i-1,j}) + \lambda^2(w_{i+1,j} - w_{i-1,j})^2] \\ &\quad + [\lambda^2(v_{i+1,j} - v_{i-1,j})^2 + 2\lambda(v_{i+1,j} - v_{i-1,j})(w_{i+1,j} + w_{i-1,j}) + (w_{i+1,j} + w_{i-1,j})^2] \} \\ &= \frac{1}{4} \{ (1 + \lambda^2)(v_{i+1,j}^2 + v_{i-1,j}^2) + 2(1 - \lambda^2)v_{i+1,j}v_{i-1,j} \\ &\quad + (1 + \lambda^2)(w_{i+1,j}^2 + w_{i-1,j}^2) + 2(1 - \lambda^2)w_{i+1,j}w_{i-1,j} \\ &\quad + 4\lambda(v_{i+1,j}w_{i+1,j} - v_{i-1,j}w_{i-1,j}) \} \\ &= \frac{1}{4} \{ (1 + \lambda^2)(v_{i+1,j}^2 + v_{i-1,j}^2 + w_{i+1,j}^2 + w_{i-1,j}^2) + 2(1 - \lambda^2)(v_{i+1,j}v_{i-1,j} + w_{i+1,j}w_{i-1,j}) \\ &\quad + 4\lambda(v_{i+1,j}w_{i+1,j} - v_{i-1,j}w_{i-1,j}) \}. \end{aligned}$$

$$\begin{aligned}
4 \sum_{i=1}^{N-1} (v_{i,j+1}^2 + w_{i,j+1}^2) &= (1 + \lambda^2) (v_{0,j}^2 + v_{1,j}^2 + v_{N-1,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{1,j}^2 + w_{N-1,j}^2 + w_{N,j}^2) \\
&\quad + 2(1 + \lambda^2) \sum_{i=2}^{N-2} (v_{i,j}^2 + w_{i,j}^2) \\
&\quad + 2(1 - \lambda^2) \sum_{i=1}^{N-1} (v_{i+1,j}v_{i-1,j} + w_{i+1,j}w_{i-1,j}) \\
&\quad + 4\lambda(v_{N,j}w_{N,j} + v_{N-1,j}w_{N-1,j} - v_{1,j}w_{1,j} - v_{0,j}w_{0,j})
\end{aligned}$$

特に $\lambda = 1$ の場合

$$\begin{aligned}
\sum_{i=1}^{N-1} (v_{i,j+1}^2 + w_{i,j+1}^2) &= \frac{1}{2} (v_{0,j}^2 + v_{1,j}^2 + v_{N-1,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{1,j}^2 + w_{N-1,j}^2 + w_{N,j}^2) \\
&\quad + \sum_{i=2}^{N-2} (v_{i,j}^2 + w_{i,j}^2) \\
&\quad + (v_{N,j}w_{N,j} + v_{N-1,j}w_{N-1,j} - v_{1,j}w_{1,j} - v_{0,j}w_{0,j}) \\
&= \sum_{i=0}^N (v_{i,j}^2 + w_{i,j}^2) \\
&\quad - \frac{1}{2} (v_{0,j}^2 + v_{1,j}^2 + v_{N-1,j}^2 + v_{N,j}^2 + w_{0,j}^2 + w_{1,j}^2 + w_{N-1,j}^2 + w_{N,j}^2) \\
&\quad - (v_{1,j}w_{1,j} + v_{0,j}w_{0,j} - v_{N,j}w_{N,j} - v_{N-1,j}w_{N-1,j}) \\
&= \sum_{i=0}^N (v_{i,j}^2 + w_{i,j}^2) \\
&\quad - \frac{1}{2} [(v_{0,j} + w_{0,j})^2 + (v_{1,j} + w_{1,j})^2 + (v_{N-1,j} - w_{N-1,j})^2 + (v_{N,j} - w_{N,j})^2]
\end{aligned}$$

それゆえ

$$v_{0,j+1}^2 + w_{0,j+1}^2 + v_{N,j+1}^2 + w_{N,j+1}^2 = \frac{1}{2} [(v_{0,j} + w_{0,j})^2 + (v_{1,j} + w_{1,j})^2 + (v_{N-1,j} - w_{N-1,j})^2 + (v_{N,j} - w_{N,j})^2]$$

であればエネルギー保存則が成り立つ。

簡単のため同次 Dirichlet 境界条件の場合

$$\begin{aligned}
w_{0,j+1}^2 + w_{N,j+1}^2 &= \frac{1}{2} [w_{0,j}^2 + (v_{1,j} + w_{1,j})^2 + (v_{N-1,j} - w_{N-1,j})^2 + w_{N,j}^2] \\
w_{0,j+1}^2 &= \frac{1}{2} [w_{0,j}^2 + (v_{1,j} + w_{1,j})^2], \quad w_{N,j+1}^2 = \frac{1}{2} [(v_{N-1,j} - w_{N-1,j})^2 + w_{N,j}^2]
\end{aligned}$$

第6章 2次元長方形領域における波動方程式の初期値境界値問題

6.1 Dirichlet 境界条件

$\Omega := (0, L) \times (0, H)$ における波動方程式の初期値境界値問題

$$(6.1) \quad \frac{\partial^2 u}{\partial t^2}(x, y, t) = \frac{\partial^2 u}{\partial x^2}(x, y, t) + \frac{\partial^2 u}{\partial y^2}(x, y, t) \quad ((x, y, t) \in \Omega \times (0, \infty)),$$

$$(6.2) \quad u(x, y, t) = 0 \quad ((x, y, t) \in \partial\Omega \times (0, \infty)),$$

$$(6.3) \quad u(x, y, 0) = f(x, y) \quad ((x, y) \in \bar{\Omega}),$$

$$(6.4) \quad \frac{\partial u}{\partial t}(x, y, 0) = g(x, y) \quad ((x, y) \in \bar{\Omega})$$

を考える。

Fourier の方法で解は求まり、

$$u(x, y, t) = \sum_{m,n=1}^{\infty} \sin \frac{m\pi x}{L} \sin \frac{n\pi y}{H} \left(a_{mn} \cos(\lambda_{mn} t) + b_{mn} \frac{\sin(\lambda_{mn} t)}{\lambda_{mn}} \right),$$

$$\lambda_{mn} = \sqrt{\left(\frac{m}{L}\right)^2 + \left(\frac{n}{H}\right)^2} \pi,$$

$$a_{mn} = \frac{4}{LH} \int_0^L \int_0^H f(x, y) \sin \frac{m\pi x}{L} \sin \frac{n\pi y}{H} dy dx,$$

$$b_{mn} = \frac{4}{LH} \int_0^L \int_0^H g(x, y) \sin \frac{m\pi x}{L} \sin \frac{n\pi y}{H} dy dx.$$

6.1.1 差分法のプログラム

差分法で解く以下のプログラムは、関口 [7] にあったプログラムを少し手直ししたものである。

$\left(\frac{1}{c^2} u_{tt} = u_{xx} + u_{yy}\right)$ という方程式に一般化することを考えると、プログラム中で c という変数を使っているのは、変更したい…)

```

/*
 * wave2d.c --- 2次元波動方程式 (同次 Dirichlet 境界条件)
 * コンパイル
 *      ccmg wave2d.c
 */

#include <stdio.h>
#include <math.h>
#include <matrix.h>
#define G_DOUBLE
#include <glsc.h>

double sqr(double x) { return x * x; }

double pi;

int main(void)
{

    int    Nx, Ny, i, j, n, skip, nMax, nfunc;
    double hx,hy,lambdax,lambday,lambdax2,lambday2,tau,Tmax,t,c,dt;
    matrix u1,u2,u3;
    double phi( double, double,int), psi( double, double,int);
    double exact(double x, double y, double t);
    char    message[100];

    pi = 4.0 * atan(1.0);

    printf("Nx = , Ny = : "); scanf("%d %d", &Nx, &Ny);

    hx = 1.0 / Nx; hy = 1.0/ Ny;
    printf("τ (限界は%g) : ", 1.0 / sqrt(1/(hx*hx) + 1/(hy*hy)));
    scanf("%lf", &tau);

    lambdax = tau / hx; lambday = tau / hy;
    printf("λ x = %g になりました。 \n", lambdax);

```

```

printf("λ y = %g になりました。 \n", lambday);
lambdax2 = lambdax * lambdax; lambday2 = lambday * lambday;
printf("λ x * λ x + λ y * λ y = %g になりました。 \n", lambdax2 + lambday2);

printf("Tmax = : "); scanf("%lf", &Tmax);
printf("Δ t = : "); scanf("%lf", &dt);
printf("nfunc = "); scanf("%d", &nfunc);

/* メモリの確保 */
if ((u1 = new_matrix(Nx + 1, Ny + 1)) == NULL) {
    fprintf(stderr, "配列 u1 を確保できませんでした, \n");
    exit(1);
}
if ((u2 = new_matrix(Nx + 1, Ny + 1)) == NULL) {
    fprintf(stderr, "配列 u2 を確保できませんでした, \n");
    exit(1);
}
if ((u3 = new_matrix(Nx + 1, Ny + 1)) == NULL) {
    fprintf(stderr, "配列 u3 を確保できませんでした, \n");
    exit(1);
}

c = 1.0 - lambdax2 - lambday2;

skip = dt / tau;

/* GLSC の開始 */
g_init("wave2d", 250.0, 180.0);
g_device(G_BOTH);
g_def_scale(0,
            0.0, 1.0, -4.0, 4.0,
            150.0, 35.0, 80.0, 140.0);
g_def_line(0, G_BLACK, 0, G_LINE_SOLID);
g_def_line(1, G_BLACK, 0, G_LINE_DOTS);
g_def_text(0, G_BLACK, 2);
g_sel_scale(0);

```

```

/* t=0 の時グラフを描く*/
for (i=0; i <= Nx; i++)
    for (j=0; j <= Ny; j++)
        u1[i][j] = phi(i * hx, j * hy, nfunc);
g_cls();
g_text(50.0, 10.0,
        "The numerical simulation of a 2-dimensional wave equation");
sprintf(message, "Nx=%d,Ny=%d,tau=%g,lambdax=%g,lambday=%g,Tmax=%g, dt=%g",
        Nx, Ny, tau, lambdax, lambday, Tmax, dt);
g_text(50.0, 20.0, message);
sprintf(message, "t = %g", 0 * tau);
g_text(50.0, 30.0, message);

/* 座標軸を描く */
g_sel_line(1);
g_move(-0.1, 0.0); g_plot(1.1, 0.0);
g_move(0.0, -0.1); g_plot(0.0, 2.0);
/* 断面図を描く */
g_sel_line(0);
g_move(0.0, u1[0][Ny/2]);
for (i=1; i<= Nx; i++)
    g_plot(i * hx, u1[i][Ny/2]);
/* 鳥瞰図を描く */
g_hidden2(5.0, 5.0, 1.0, -1.0, 1.0,
        15.0, 25.0, 30.0,
        20.0, 60.0, 100.0, 80.0,
        u1, Nx + 1, Ny + 1, 1,
        G_SIDE_NONE, 2, 1);
printf("u1 まで描けた\n");

/* t=τ の時のグラフを描く*/
for (i = 1; i < Nx; i++)
    for (j = 1; j < Ny; j++)
        u2[i][j] = c * u1[i][j]
            + 0.5 * lambdax2 * (u1[i+1][j] + u1[i-1][j])

```

```

        + 0.5 * lambday2 * (u1[i][j+1] + u1[i][j-1])
        + tau * psi(i * hx, j * hy, nfunc);

/* Dirichlet 境界条件 */
for (i = 0; i <= Nx; i++)
    u2[i][0] = u2[i][Ny] = 0.0;
for (j = 0; j <= Ny; j++)
    u2[0][j] = u2[Nx][j] = 0.0;

g_cls();
sprintf(message, "t = %g", tau);
g_text(50.0, 30.0, message);
/* 座標軸を描く */
g_sel_line(1);
g_move(-0.1, 0.0); g_plot(1.1, 0.0);
g_move(0.0, -0.1); g_plot(0.0, 2.0);
/* 断面図を描く */
g_sel_line(0);
g_move(0.0, u2[0][Ny/2]);
for (i = 1; i <= Nx; i++)
    g_plot(i * hx, u2[i][Ny/2]);
/* 鳥瞰図を描く */
g_hidden2(5.0, 5.0, 1.0, -1.0, 1.0,
          15.0, 25.0, 30.0,
          20.0, 60.0, 100.0, 80.0,
          u2, Nx + 1, Ny + 1, 1,
          G_SIDE_NONE, 2, 1);
printf("u2 まで描けた\n");

nMax = rint(Tmax /tau);

/* t=nτ (n ≧ 2) の時のグラフを描く*/
for (n = 2; n <= nMax; n++) {
    for (i = 1; i < Nx; i++)
        for (j = 1; j < Ny; j++)
            u3[i][j] = 2 * c * u2[i][j]

```



```

        + lambdax2 * (u2[i+1][j] + u2[i-1][j])
        + lambday2 * (u2[i][j+1] + u2[i][j-1])
        - u1[i][j];
/* Dirichlet 境界条件 */
for (i = 0; i <= Nx; i++)
    u3[i][0] = u3[i][Ny] = 0.0;
for (j = 0; j <= Ny; j++)
    u3[0][j] = u3[Nx][j] = 0.0;

/* 配列の交換 */
for (i = 0; i <= Nx; i++)
    for (j = 0; j <= Ny; j++) {
        u1[i][j] = u2[i][j];
        u2[i][j] = u3[i][j];
    }
if (n % skip == 0) {
    g_cls();
    /* 座標軸を描く */
    g_sel_line(1);
    g_move(-0.1, 0.0); g_plot(1.1, 0.0);
    g_move(0.0, -0.1); g_plot(0.0, 2.0);
    /* 断面図を描く */
    g_sel_line(0);
    g_move(0.0, u2[0][Ny/2]);
    for (i = 1; i <= Nx; i++)
        g_plot(i * hx, u2[i][Ny/2]);
    t = n * tau;
    sprintf(message, "t = %g", t);
    g_text(50.0, 30.0, message);
    /* 鳥瞰図を描く */
    g_hidden2(5.0, 5.0, 1.0, -1.0, 1.0,
              15.0, 25.0, 30.0,
              20.0, 60.0, 100.0, 80.0,
              u3, Nx + 1, Ny + 1, 1,
              G_SIDE_NONE, 2, 1);
    if (nfunc == 1)

```

```

        printf("t=%g, u[Nx/2] [Ny/2]=%g, 厳密解=%g\n",
               t, u2[Nx/2] [Ny/2], exact(0.5,0.5,t));
    }
}
printf("終了したらウィンドウをクリックして終了してください。 \n");
g_sleep(-1.0);
g_term();

return 0;
}

double h(double r)
{
    double a = 0.1, b = 0.2;
    if (r >= a && r <= b)
        return 4 * sqrt(r - a) * sqrt(r - b) / sqrt(sqrt(b - a));
    else
        return 0.0;
}

double dh(double r)
{
    double a = 0.1, b = 0.2;
    if (r >= a && r <= b)
        return 12 * (r - a) * (r - b) * (2 * r - a - b) / sqrt(sqrt(b - a));
    else
        return 0.0;
}

/* 初期条件： $\phi$ 、 $\psi$  */
double phi(double x, double y, int nfunc)
{
    if (nfunc == 1)
        return sin(pi * x) * sin(pi * y);
    else if (nfunc == 2 || nfunc == 3) {
        double r;

```

```

    r = sqrt(sqr(x - 0.5) + sqr(y - 0.5));
    if (r > 1e-4)
        return h(r) / r;
    else
        return 0.0;
}
else
    return 0.0;
}

double psi( double x, double y, int nfunc)
{
    if (nfunc == 1) {
        return 0.0;
    }
    else if (nfunc == 2) {
        double r;
        r = sqrt(sqr(x - 0.5) + sqr(y - 0.5));
        if (r > 1e-4)
            return - 1.0 * dh(r) / r;
        else
            return 0.0;
    }
    else if (nfunc == 3) {
        return 0.0;
    }
    else
        return 0.0;
}

double exact(double x, double y, double t)
{
    return cos(sqrt(2.0) * pi * t) * sin(pi * x) * sin(pi * y);
}

```

6.2 Neumann 境界条件

$\Omega := (0, L) \times (0, H)$ における波動方程式の初期値境界値問題

$$(6.5) \quad \frac{\partial^2 u}{\partial t^2}(x, y, t) = \frac{\partial^2 u}{\partial x^2}(x, y, t) + \frac{\partial^2 u}{\partial y^2}(x, y, t) \quad ((x, y, t) \in \Omega \times (0, \infty)),$$

$$(6.6) \quad \frac{\partial u}{\partial n}(x, y, t) = 0 \quad ((x, y, t) \in \partial\Omega \times (0, \infty)),$$

$$(6.7) \quad u(x, y, 0) = f(x, y) \quad ((x, y) \in \bar{\Omega}),$$

$$(6.8) \quad \frac{\partial u}{\partial t}(x, y, 0) = g(x, y) \quad ((x, y) \in \bar{\Omega})$$

を考える。ここで n は Ω の境界上の点における、 Ω の外向き単位法線ベクトルである。

Fourier の方法で解は求まり、

$$\begin{aligned} u(x, y, t) &= \frac{a_{00} + b_{00}t}{4} \\ &+ \frac{1}{2} \sum_{m=1}^{\infty} \cos \frac{m\pi x}{L} \left(a_{m0} \cos(\lambda_{m0}t) + b_{m0} \frac{\sin(\lambda_{m0}t)}{\lambda_{m0}} \right) \\ &+ \frac{1}{2} \sum_{n=1}^{\infty} \cos \frac{n\pi y}{H} \left(a_{0n} \cos(\lambda_{0n}t) + b_{0n} \frac{\sin(\lambda_{0n}t)}{\lambda_{0n}} \right) \\ &+ \sum_{m,n=1}^{\infty} \cos \frac{m\pi x}{L} \cos \frac{n\pi y}{H} \left(a_{mn} \cos(\lambda_{mn}t) + b_{mn} \frac{\sin(\lambda_{mn}t)}{\lambda_{mn}} \right), \\ \lambda_{mn} &= \sqrt{\left(\frac{m}{L}\right)^2 + \left(\frac{n}{H}\right)^2} \pi, \\ a_{mn} &= \frac{4}{LH} \int_0^L \int_0^H f(x, y) \cos \frac{m\pi x}{L} \cos \frac{n\pi y}{H} dy dx, \\ b_{mn} &= \frac{4}{LH} \int_0^L \int_0^H g(x, y) \cos \frac{m\pi x}{L} \cos \frac{n\pi y}{H} dy dx. \end{aligned}$$

メモ: 余弦関数の直交性

$m, n = 0, 1, 2, \dots$ に対して、

$$\int_0^1 \cos m\pi x \cos n\pi x dx = \begin{cases} 0 & (m \neq n) \\ \frac{1}{2} & (m = n \neq 0) \\ 1 & (m = n = 0). \end{cases}$$

$m, n, k, \ell = 0, 1, 2, \dots$ に対して、

$$\iint_{[0,1] \times [0,2]} \cos mx \cos ny \cos kx \cos \ell y \, dx \, dy = \begin{cases} 0 & (m \neq k \text{ or } n \neq \ell) \\ \frac{1}{4} & (m = k \neq 0 \text{ and } n = \ell \neq 0) \\ \frac{1}{2} & ((m = k = 0 \text{ and } n = \ell \neq 0) \\ & \text{or } (m = k \neq 0 \text{ and } n = \ell = 0)) \\ 1 & (m = k = n = \ell = 0). \end{cases}$$

6.2.1 差分法のプログラム

以下のプログラムは、関口 [7] にあったプログラムを少し手直したものである。仮想格子点を導入し、Neumann 境界条件の近似に、中心差分近似を用いている。

```

/*
 * wave2n.c --- 2次元波動方程式 (Neumann 境界条件)
 * コンパイル
 * ccmg wave2n.c
 */

#include <stdio.h>
#include <math.h>
#include <matrix.h>
#define G_DOUBLE
#include <glsc.h>

double sqr(double x) { return x * x; }

double pi;

int main(void)
{
    int    Nx, Ny, i, j, k, skip, Kmax, nfunc;
    double hx, hy, lambdax, lambday, lambdax2, lambday2, tau, Tmax, t, c, dt;
    matrix u1, u2, u3;
    double phi( double, double, int), psi( double, double, int);
    double exact(double x, double y, double t);
    char    message[100];

    pi = 4.0*atan(1.0);

    printf("Nx = , Ny = : "); scanf("%d %d", &Nx, &Ny);

    hx = 1.0 / Nx; hy = 1.0/ Ny;

```

```

printf("τ (限界は%g) : ", 1.0 / sqrt(1/(hx*hx) + 1/(hy*hy)));
scanf("%lf", &tau);

lambdax = tau / hx; lambday = tau / hy;
printf("λ x = %g になりました。\\n", lambdax);
printf("λ y = %g になりました。\\n", lambday);
lambdax2 = lambdax * lambdax; lambday2 = lambday * lambday;
printf("λ x * λ x + λ y * λ y = %g になりました。\\n", lambdax2 + lambday2);

printf("Tmax = : "); scanf("%lf", &Tmax);
printf("Δ t = : "); scanf("%lf", &dt);
printf("nfunc = "); scanf("%d", &nfunc);

/* メモリの確保 */
if ((u1 = new_matrix(Nx +1,Ny +1)) == NULL) {
    fprintf(stderr, "配列 u1 を確保できませんでした,\\n");
    exit(1);
}
if ((u2 = new_matrix(Nx +1,Ny +1)) == NULL) {
    fprintf(stderr, "配列 u2 を確保できませんでした,\\n");
    exit(1);
}
if ((u3 = new_matrix(Nx +1,Ny +1)) == NULL) {
    fprintf(stderr, "配列 u3 を確保できませんでした,\\n");
    exit(1);
}

c = 1.0 - lambdax2 - lambday2;

skip = dt / tau;

/* GLSC の開始 */
g_init("wave2d", 250.0, 180.0);
g_device(G_BOTH);
g_def_scale(0,
            0.0, 1.0, -4.0, 4.0,
            150.0, 35.0, 80.0, 140.0);
g_def_line(0, G_BLACK, 0, G_LINE_SOLID);
g_def_line(1, G_BLACK, 0, G_LINE_DOTS);
g_def_text(0, G_BLACK, 2);
g_sel_scale(0);

/* t=0 の時グラフを描く */
for (i=0; i<= Nx; i++)
    for (j=0; j<= Ny; j++)
        u1[i][j] = phi(i * hx , j * hy, nfunc);
g_cls();
g_text(50.0,10.0,
       "The numerical simulation of a 2-dimensional wave equation");

```

```

sprintf(message, "Nx=%d, Ny=%d, tau=%g, lambdax=%g, lambday=%g, Tmax=%g, dt=%g",
           Nx, Ny, tau, lambdax, lambday, Tmax, dt);
g_text(50.0, 20.0, message);
sprintf(message, "t = %g", 0 * tau);
g_text(50.0, 30.0, message);
g_sel_scale(0);

/* 座標軸を描く */
g_sel_line(1);
g_move(-0.1, 0.0); g_plot(1.1, 0.0);
g_move(0.0, -0.1); g_plot(0.0, 2.0);
/* 断面図を描く */
g_sel_line(0);
g_move(0.0, u1[0][0]);
for (i=1; i<= Nx; i++)
    g_plot(i * hx, u1[i][Ny/2]);
/* 鳥瞰図を描く */
g_hidden2(5.0, 5.0, 1.0, -1.0, 1.0,
          15.0, 25.0, 30.0,
          20.0, 60.0, 100.0, 80.0,
          u1, Nx + 1, Ny + 1, 1,
          G_SIDE_NONE, 4, 4);

printf("u1 まで描けた\n");

/* t=τ の時のグラフを描く */
for (i = 1; i < Nx; i++) {
    for (j = 1; j < Ny; j++)
        u2[i][j] = c * u1[i][j]
            + 0.5 * lambdax2 * (u1[i+1][j] + u1[i-1][j])
            + 0.5 * lambday2 * (u1[i][j+1] + u1[i][j-1])
+ tau * psi(i * hx, j * hy, nfunc);
}
for (i = 1; i < Nx; i++) {
    u2[i][0] = c * u1[i][0] + 0.5 * lambdax2 * (u1[i+1][0] + u1[i-1][0])
        + lambday2 * u1[i][1] + tau * psi(i * hx, 0.0, nfunc);
    u2[i][Ny] = c * u1[i][Ny] + 0.5 * lambdax2 * (u1[i+1][Ny] + u1[i-1][Ny])
        + lambday2 * u1[i][Ny-1] + tau * psi(i * hx, 1.0, nfunc);
}
for (j = 1; j < Ny; j++) {
    u2[0][j] = c * u1[0][j] + lambdax2 * u1[1][j]
        + 0.5 * lambday2 * (u1[0][j+1] + u1[0][j-1])
        + tau * psi(0.0, j * hy, nfunc);
    u2[Nx][j] = c * u1[Nx][j] + lambdax2 * u1[Nx-1][j]
        + 0.5 * lambday2 * (u1[Nx][j+1] + u1[Nx][j-1])
        + tau * psi(1.0, j * hy, nfunc);
}
u2[0][0] = c * u1[0][0] + lambdax2 * u1[1][0] + lambday2 * u1[0][1]

```

```

    + tau * psi(0.0, 0.0, nfunc);
u2[0][Ny] = c * u1[0][Ny] + lambdax2 * u1[1][Ny] + lambday2 * u1[0][Ny-1]
    + tau * psi(0.0, 1.0, nfunc);
u2[Nx][0] = c * u1[Nx][0] + lambdax2 * u1[Nx-1][0] + lambday2 * u1[Nx][1]
    + tau * psi(1.0, 0.0, nfunc);
u2[Nx][Ny] = c * u1[Nx][Ny] + lambdax2 * u1[Nx-1][Ny] + lambday2 *
    u1[Nx][Ny-1] + tau * psi(1.0, 1.0, nfunc);

g_cls();
sprintf(message,"t = %g", tau);
g_text(50.0, 30.0, message);
/* 座標軸を描く */
g_sel_line(1);
g_move(-0.1, 0.0); g_plot(1.1, 0.0);
g_move(0.0, -0.1); g_plot(0.0, 2.0);
/* 断面図を描く */
g_sel_line(0);
g_move(0.0, u2[0][0]);
for (i = 1; i < Nx; i++)
    g_plot(i * hx, u2[i][Ny/2]);
/* 鳥瞰図を描く */
g_hidden2(5.0, 5.0, 1.0, -1.0, 1.0,
          15.0, 25.0, 30.0,
          20.0, 60.0, 100.0, 80.0,
          u2, Nx + 1, Ny + 1, 1,
          G_SIDE_NONE, 4, 4);

printf("u2 まで描けた\n");

Kmax = rint(Tmax /tau);

/* t=n τ (n>1) の時のグラフを描く*/
for (k = 2; k <= Kmax; k++) {
    for (i = 1; i < Nx; i++) {
        for (j = 1; j < Ny; j++)
            u3[i][j] = 2 * c * u2[i][j]
                + lambdax2 * (u2[i+1][j] + u2[i-1][j])
                + lambday2 * (u2[i][j+1] + u2[i][j-1])
                - u1[i][j];
    }
    for (i = 1; i < Nx; i++) {
        u3[i][0] = 2 * c * u2[i][0] + lambdax2 * (u2[i+1][0] + u2[i-1][0])
            + 2 * lambday2 * u2[i][1] - u1[i][0];
        u3[i][Ny] = 2 * c * u2[i][Ny] + lambdax2 * (u2[i+1][Ny] + u2[i-1][Ny])
            + 2 * lambday2 * u2[i][Ny-1] - u1[i][Ny];
    }
    for (j = 1; j < Ny; j++) {
        u3[0][j] = 2 * c * u2[0][j] + 2 * lambdax2 * u2[1][j]
            + lambday2 * (u2[0][j+1] + u2[0][j-1]) - u1[0][j];
    }
}

```



```

    u3[Nx][j] = 2 * c * u2[Nx][j] + 2 * lambdax2 * u2[Nx-1][j]
      + lambday2 * (u2[Nx][j+1] + u2[Nx][j-1]) - u1[Nx][j];
  }
  u3[0][0] = 2 * c * u2[0][0] + 2 * lambdax2 * u2[1][0] + 2 * lambday2
    * u2[0][1] - u1[0][0];
  u3[0][Ny] = 2 * c * u2[0][Ny] + 2 * lambdax2 * u2[1][Ny] + 2 * lambday2
    * u2[0][Ny-1] - u1[0][Ny];
  u3[Nx][0] = 2 * c * u2[Nx][0] + 2 * lambdax2 * u2[Nx-1][0]
    + 2 * lambday2 * u2[Nx][1] - u1[Nx][0];
  u3[Nx][Ny] = 2 * c * u2[Nx][Ny] + 2 * lambdax2 * u2[Nx-1][Ny]
    + 2 * lambday2 * u2[Nx][Ny-1] - u1[Nx][Ny];
  /* 配列の交換 */
  for (i = 0; i <= Nx; i++)
    for (j = 0; j <= Ny; j++) {
      u1[i][j] = u2[i][j];
      u2[i][j] = u3[i][j];
    }
  if (k % skip == 0) {
    g_cls();
    /* 座標軸を描く */
    g_sel_line(1);
    g_move(-0.1, 0.0); g_plot(1.1, 0.0);
    g_move(0.0, -0.1); g_plot(0.0, 2.0);
    /* 断面図(x->u(x,1/2)のぐらふ)を描く */
    g_sel_line(0);
    g_move(0.0, u3[0][Ny/2]);
    for (i = 1; i <= Nx; i++)
      g_plot(i * hx, u3[i][Ny/2]);
    sprintf(message, "t = %g", k * tau);
    g_text(50.0, 30.0, message);
    /* 鳥瞰図を描く */
    g_hidden2(5.0, 5.0, 1.0, -1.0, 1.0,
              15.0, 25.0, 30.0,
              20.0, 60.0, 100.0, 80.0,
              u3, Nx + 1, Ny + 1, 1,
              G_SIDE_NONE, 4, 4);
  }
  t = tau * k;
  if (nfunc == 1)
    printf("t=%g, u[Nx/2][Ny/2]=%g, 厳密解=%g\n",
          t, u2[Nx/2][Ny/2], exact(0.5,0.5,t));
}
printf("終了したらウィンドウをクリックして終了してください。 \n");
g_sleep(-1.0);
g_term();

return 0;
}

```

```

double h(double r)
{
    double a = 0.1, b = 0.2;
    if (r >= a && r <= b)
        return 4 * sqr(r - a) * sqr(r - b) / sqr(sqr(b - a));
    else
        return 0.0;
}

double dh(double r)
{
    double a = 0.1, b = 0.2;
    if (r >= a && r <= b)
        return 12 * (r - a) * (r - b) * (2 * r - a - b) / sqr(sqr(b - a));
    else
        return 0.0;
}

/* 初期条件:  $\phi$ 、 $\psi$  */
double phi(double x, double y, int nfunc)
{
    if (nfunc == 1)
        return cos(pi * x) * cos(pi * y);
    else if (nfunc == 2 || nfunc == 3) {
        double r;
        r = sqrt(sqr(x - 0.5) + sqr(y - 0.5));
        if (r > 1e-4)
            return h(r) / r;
        else
            return 0.0;
    }
    else
        return 0.0;
}

double psi( double x, double y, int nfunc)
{
    if (nfunc == 1) {
        return 0.0;
    }
    else if (nfunc == 2) {
        double r;
        r = sqrt(sqr(x - 0.5) + sqr(y - 0.5));
        if (r > 1e-4)
            return - 1.0 * dh(r) / r;
        else
            return 0.0;
    }
    else if (nfunc == 3) {

```

```

    return 0.0;
}
else
    return 0.0;
}

double exact(double x, double y, double t)
{
    return cos(sqrt(2.0) * pi * t) * cos(pi * x) * cos(pi * y);
}

```

6.3 misc.

6.3.1 CFL 条件

Courant-Friedrichs-Lewy の条件について、きちんと調べて書いておきたい。

- 陽解法のスキームが安定であるための必要条件である (うっかり十分条件であると書いた学生もいたが、いつもそうであるわけではないはず)。
- 直観的な解釈が可能 「情報の伝達する速さが、波の速さ以上である」
- 1次元の波動方程式 $\frac{1}{c^2}u_{tt} = u_{xx}$ について、2章のような差分スキームを考えると、

$$\frac{\Delta x}{\Delta t} \geq c.$$

これは

$$\lambda := \frac{c\tau}{h} \leq 1$$

と書ける ($h = \Delta x$, $\tau = \Delta t$)。この λ は Courant 数と呼ばれる。

- Wikipedia の “Courant-Friedrichs-Lewy condition” には、

$$\frac{u\Delta t}{\Delta x} \leq C, \quad \frac{u_x\Delta t}{\Delta x} + \frac{u_y\Delta t}{\Delta y} \leq C, \quad \Delta t \sum_{i=1}^n \frac{u_{x_i}}{\Delta x_i} \leq C$$

のような不等式が (微分方程式抜きで) 書いてある。まあ、気持ちというか、書かれていない設定は想像できなくもない。

- この章のような2次元波動方程式に対する差分スキームの CFL 条件は、きっと

$$\lambda_x^2 + \lambda_y^2 \leq 1,$$

$$\lambda_x := \frac{c\tau}{h_x}, \quad \lambda_y := \frac{c\tau}{h_y}$$

なのであろう。

第7章 3次元直方体領域における波動方程式の初期値境界値問題

7.1 コメント

2001年度卒研に

関口洋正「3次元波動方程式の差分法による数値シミュレーション」

というのがある。

当時書いた解説から。

何と言っても3次元であるところが新しく、ちょっと自慢できるところである。例えば立方体領域の各辺を100分割する場合、格子点の個数は約100万なので、関数の値の記憶には8MB必要である。2階の方程式であるから特別な工夫をしない限り、その3倍の24MBのメモリーが必要となる。今だから簡単にパソコンで実行できる数値実験ということになる。

次元を3にすることに何の意味があるか？という問には、ホイヘンスの原理は空間の次元が3以上の奇数でないと成立しない、というところがミソである。

ホイヘンスの原理については、やはり2001年度卒研に

坪井泰洋「波動方程式の解の公式とホイヘンスの原理」

というのがある。

これも当時書いた解説から。

ホイヘンスの原理は 3 以上の奇数次元でのみ成立するという話は有名だが、その証明は読んだことがなかった。 n 次元波動方程式の基本解は載っている本が見つけにくかった (後になって溝畑 [8] にあることに気がついた)。坪井君は検索エンジンで <http://village.infoweb.ne.jp/~fwiz0276/pde03.htm> というページを見つけて (残念ながらもうなくなっている)、その解説を進めていった。

残念ながら少し消化不良のところがあって、後輩で、超関数を勉強する人が現れたら後を引き継いで完備化してもらいたい。

第8章 円盤領域における波動方程式の初期値境界値問題

金子 [22] が参考になる。

8.1 Dirichlet 境界値問題の厳密解

$\Omega := \{(x, y) \in \mathbf{R}^2; x^2 + y^2 < 1\}$ における初期値境界値問題

$$(8.1) \quad \frac{1}{c^2} u_{tt}(x, y, t) = \Delta u(x, y, t) \quad ((x, y) \in \Omega, t > 0),$$

$$(8.2) \quad u(x, y, t) = 0 \quad ((x, y) \in \partial\Omega, t > 0),$$

$$(8.3) \quad u(x, y, 0) = \phi(x, y), \quad u_t(x, y, 0) = \psi(x, y) \quad ((x, y) \in \bar{\Omega}).$$

Fourier の方法を用いる。 $u(x, y, t) = v(x, y)\eta(t)$ の形をしているものをすべて求める。

$$\frac{\eta''(t)}{c^2\eta(t)} = \frac{\Delta v(x, y)}{v(x, y)}.$$

これは定数であるから、 $-\lambda$ とおくと、

$$(8.4) \quad \eta''(t) = -c^2\lambda\eta(t),$$

$$(8.5) \quad -\Delta v(x, y) = \lambda v(x, y) \quad ((x, y) \in \Omega), \quad v(x, y) = 0 \quad ((x, y) \in \partial\Omega).$$

8.1.1 $-\Delta$ の固有値問題

極座標変換

$$x = r \cos \theta, \quad y = r \sin \theta \quad (0 \leq r \leq 1, \theta \in [0, 2\pi])$$

を用いて変数変換する。すなわち

$$V(r, \theta) := v(x, y) = v(r \cos \theta, r \sin \theta)$$

とおくと、

$$V_{rr}(r, \theta) + \frac{1}{r}V_r(r, \theta) + \frac{1}{r^2}V_{\theta\theta}(r, \theta) = -\lambda V(r, \theta) \quad (r \in (0, 1), \theta \in [0, 2\pi)).$$

$V(r, \theta) = R(r)\Theta(\theta)$ の形をしているものをすべて求める。

$$R''(r)\Theta(\theta) + \frac{1}{r}R'(r)\Theta(\theta) + \frac{1}{r^2}R(r)\Theta''(\theta) = -\lambda R(r)\Theta(\theta)$$

より

$$\frac{R''(r)}{R(r)} + \frac{1}{r} \frac{R'(r)}{R(r)} + \frac{1}{r^2} \frac{\Theta''(\theta)}{\Theta(\theta)} = -\lambda.$$

これから

$$\frac{R''(r)}{R(r)}r^2 + \frac{R'(r)}{R(r)}r + \lambda r^2 = -\frac{\Theta''(\theta)}{\Theta(\theta)}.$$

この等式の値も定数である。 μ とおくことによって、

$$(8.6) \quad r^2 R''(r) + r R'(r) + (\lambda r^2 - \mu) R(r) = 0 \quad (0 < r < 1),$$

$$(8.7) \quad \Theta''(\theta) = -\mu \Theta(\theta) \quad (\theta \in [0, 2\pi]).$$

Θ は周期 2π であるから、

$$(8.8) \quad \Theta(0) = \Theta(2\pi), \quad \Theta'(0) = \Theta'(2\pi).$$

(8.7), (8.8) の解は

$$\mu = n^2, \quad \Theta(\theta) = A \cos n\theta + B \sin n\theta \quad (A, B \text{ は任意定数; } n \in \mathbf{N} \cup \{0\}).$$

R については

$$R''(r) + \frac{1}{r}R'(r) + \left(\lambda - \frac{n^2}{r^2}\right)R(r) = 0, \quad R(1) = 0, \quad R(0) \text{ は有限.}$$

$$\rho := \sqrt{\lambda}r, \quad W(\rho) := R(r)$$

とおくと、

$$(8.9) \quad W''(\rho) + \frac{1}{\rho}W'(\rho) + \left(1 - \frac{n^2}{\rho^2}\right)W(\rho) = 0,$$

$$(8.10) \quad W(\sqrt{\lambda}) = 0, \quad W(0) \text{ は有限.}$$

Bessel の微分方程式である。一般解 $W(\rho) = AJ_n(\rho) + BY_n(\rho)$ (A, B は任意定数) である。ここで J_n は n 次の Bessel 関数、 Y_n は n 次の Neumann 関数である。

$$J_n(z) := \sum_{k=0}^{\infty} \frac{(-1)^k}{k!(n+k)!} \left(\frac{z}{2}\right)^{n+2k}, \quad Y_n(z) = \text{結局は使わないので省略.}$$

原点の近傍で有界であるためには $B = 0$ が必要である。ゆえに

$$W(\rho) = AJ_n(\rho).$$

ゆえに

$$R(r) = AJ_n(\sqrt{\lambda}r).$$

$R(1) = 0$ より、 $\sqrt{\lambda}$ は J_n の零点である。 J_n の正の零点は小さい方から順に並べることが出来る (どこかで集積したりはしない)。それを

$$\mu_{1,n} < \mu_{2,n} < \cdots < \mu_{m,n} < \cdots$$

とおくと、

$$\exists m \in \mathbf{N} \quad \text{s.t.} \quad \sqrt{\lambda} = \mu_{m,n}.$$

ゆえに $\lambda = \mu_{m,n}^2$, $R(r) = AJ_n(\mu_{m,n}r)$.

$$V(r, \theta) = J_n(\mu_{m,n}r) (A_{mn} \cos n\theta + B_{mn} \sin n\theta).$$

8.1.2 変数分離解

η に関する微分方程式は、 $\lambda = n^2$ のとき

$$\eta''(t) = -c^2 n^2 \eta(t) \quad (t > 0)$$

であるから、一般解は

$$\eta(t) = \begin{cases} At + B & (n = 0) \\ A \cos cnt + B \sin cnt & (n \neq 0) \end{cases}$$

である (A, B は任意定数)。

$$U(r, \theta, t) = J_n(\mu_{m,n}r) \cos n\theta (a_{mn} \cos cnt + b_{mn} \sin cnt) + J_n(\mu_{m,n}r) \sin n\theta (c_{mn} \cos cnt + d_{mn} \sin cnt)$$

$$\begin{aligned}
U(r, \theta, t) = & \frac{1}{2} \sum_{m=1}^{\infty} J_n(\mu_{m,0}r)(a_m t + b_m) \\
& + \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} J_n(\mu_{m,n}r) [\cos n\theta (a_{mn} \cos cnt + b_{mn} \sin cnt) + \sin n\theta (c_{mn} \cos cnt + d_{mn} \sin cnt)]
\end{aligned}$$

付録A 物理メモ

A.1 1次元波動方程式

A.1.1 管の中の気体の振動

有山 [23] による。

変位を ξ , 体積弾性率を K , 密度を ρ とすると、

$$\frac{\partial^2 \xi}{\partial t^2} = \frac{K}{\rho} \frac{\partial^2 \xi}{\partial x^2}.$$

振動に伴う膨張圧縮は速やかなので、断熱変化とみなすのが適当で、

$$K = \gamma p, \quad \gamma := \frac{c_p}{c_v}.$$

ここで c_p, c_v はそれぞれ (金原編 [24])。

$$\frac{\partial^2 \xi}{\partial t^2} = \gamma \frac{p}{\rho} \frac{\partial^2 \xi}{\partial x^2}.$$

管の端が閉じている場合、その点で $\xi = 0$.

管の端が開いている場合、その点で $\frac{\partial \xi}{\partial x} = 0$. 例えば管の断面が円形で、波長に比してその半径が小さい場合は、半径の 0.65 倍だけ管の外側の点で $\frac{\partial \xi}{\partial x} = 0$ が成り立つとすべき (開口端の補正, 小橋 [25])。

A.1.2 弾性体の棒の縦振動

$$\frac{\partial^2 \xi}{\partial t^2} = \frac{E}{\rho} \frac{\partial^2 \xi}{\partial x^2}.$$

A.2 2次元波動方程式

A.2.1 膜の振動

有山 [23] による。「膜というのは弦のようにずれの弾性を示さないものという意味」だそうである。

一定の張力、平面内にある枠で張られた一様な膜を仮定する。

膜の張力とは、微小な線分の両側の部分が互いに引き合う力の単位長さ辺りの量のことをいう。これは膜上の点だけでなく、線分の方にも依存している。

ここでは、線分の方によらず、大きさが一定で、力の方向が線分に垂直な場合のみを考える。液体の表面張力はこの良い例になっているらしいが、「適当に張った」ゴム膜も良い近似となる。

同一の固有値に対して異なる固有関数が存在することを縮退 (degeneration) という。

有山 [23] pp.154-155 を写経

板の振動は、膜と同じく2次元的な弾性体の振動であるが、数学的にかなり複雑な問題になるので、これもここでは触れないことにする。ただ、実験的に板の振動のようすを調べるには Chladni (クラドニ) の方法が良く知られている。板の中心を固定しておき、ふちの1点を指で軽くおさえ、他の点を弓でこすって振動させる。板の上に細かい砂をまいておくと砂は節線の上に集まる。指でおさえる場所、弓でこする場所を変えると色々異なる形の節線が現われる。このようにして得られる図形をクラドニの図形 (Chladni's figure) という。5-20 図にその例を示す。

鐘の振動は古くから学者の興味をひいたものとみえて多くの研究がある。その振動のようすは板の振動と似ていることが知られている。

A.3 棒の振動

棒の振動 (横振動ということ?) については、クーラント・ヒルベルト [26] (第2巻の第5章) に記述がある。

$$\frac{\partial^2 u}{\partial t^2} + \frac{\partial^4 u}{\partial x^4} = 0$$

という方程式で支配されるそうである。

A.4 板の振動

板の振動について、物理の教科書に説明が書いてあるのは発見出来なかった。工学系の本を探して、近藤 [27], 入江・小林 [28] を発見した。

板の振動は

$$\frac{\partial^2 u}{\partial t^2} + \Delta^2 u = 0$$

という方程式に従うそうである。

Gazzola-Grunau-Sweers [29] によると、Chladni の結果が知られてから、

$$\frac{\partial^2 u}{\partial t^2} + \frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4} = 0$$

という方程式を提唱した人もいたが、うまく行かなかったとか。

A.5 水の波

$$\frac{\partial^2 \eta}{\partial t^2} = gh \frac{\partial^2 \eta}{\partial x^2},$$

$$v = \sqrt{gh}.$$

$$v = \sqrt{\left(\frac{g\lambda}{2\pi} + \frac{2\pi T}{\rho\lambda}\right) \tanh \frac{2\pi h}{\lambda}}.$$

$T = 0$, $h/\lambda \ll 1$ の場合、上の場合に一致する。

水面の波の早さは波長によって変化する。このような性質を分散 (dispersion) と呼ぶ。

A.6 Schrödinger の波動方程式

この名前は正しいのかとも思うが、そう書いてある本も多いので。

関連図書

- [1] 松本英久：2次元領域における熱伝導方程式 — 円・円環領域に対する追求… —, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/1996-matsumoto.pdf> (1997年3月).
- [2] 養田孝：2次元領域における波動方程式の研究, 1997年度卒業研究レポート, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/1997-youda.pdf> (1998).
- [3] 金子裕司：S-W近似によって様々な領域の熱方程式を解くためのアルゴリズム, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2006-kaneko.pdf> (2007).
- [4] 久保田祥史：S-W近似によって様々な領域の熱方程式を解く, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2007-kubota.pdf> (2008).
- [5] 濱勇樹：S-W近似による楕円領域での波動方程式のシミュレーション, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2011-hama.pdf> (2012).
- [6] 坪井泰洋：波動方程式の解の公式とホイヘンスの原理, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/2001-tsuboi.pdf> (2002).
- [7] 関口洋正：3次元波動方程式の差分法による数値シミュレーション, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/2001-sekiguchi.pdf> (2002).
- [8] ^{みぞはた}溝畑茂：偏微分方程式論, 岩波書店 (1965).
- [9] 桂田祐史：楕円領域でのLaplacianの差分近似のために, <http://nalab.mind.meiji.ac.jp/~mk/labo/2004/ellipse.pdf> (2005年).
- [10] 中西謙太：2次元円盤領域における波動方程式の研究, 2004年度卒業研究レポート, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2004-nakanishi.pdf> (2005).

- [11] 三井康之：Java による波動方程式の数値シミュレーション, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/2001-mitsui.pdf> (2002).
- [12] 伊藤秀範:Java による波動方程式のシミュレーションプログラム ～1次元,2次元の波の動き～, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2004-itou.pdf> (2005).
- [13] 中村圭佑：2次元発展問題の可視化, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2007-nakamura.pdf> (2007).
- [14] 藤沼祐一：波動方程式に対する差分法, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/1998-fujinuma.pdf> (1999).
- [15] 伊藤雄一:二次元波動方程式の Friedrichs の方法による差分解, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/2001-itou.pdf> (2002).
- [16] John, F.: *Lectures on Advanced Numerical Analysis*, Gordon and Breach, Science Publishers, Inc. (1967), 邦訳: F. ジョン著, 藤田宏・名取亮訳, 数値解析講義, 産業図書 (1975).
- [17] 菊地文雄, 山本昌宏：微分方程式と計算機演習, 山海堂 (1991).
- [18] 笠井加奈子：波動方程式の初期値境界値問題における差分解への収束証明, <http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/2001-kasai.pdf> (2002).
- [19] 高見^{ひでお} 穎郎・河村哲也：偏微分方程式の差分解法, 東京大学出版会 (1994).
- [20] 登坂宣好, 大西和榮：偏微分方程式の数値シミュレーション, 東京大学出版会 (1991).
- [21] 山口昌哉^{まさや}, 野木達夫：数値解析の基礎, 共立出版 (1969), 2009年「復刊 数値解析の基礎」というタイトルで出版されました。
- [22] 金子 晃^{あきら}：偏微分方程式入門, 東京大学出版会 (1998).
- [23] 有山正孝：振動・波動, 基礎物理学選書, 裳華房 (1970, 1986).
- [24] 金原^{かねはら} 寿郎^{としろう}編：基礎物理学 上, 裳華房 (1963).

- [25] ^{こはし}小橋豊：音と音波, 基礎物理学選書 4, 裳華房 (1969), 良い本だと思います。常識に欠ける私には必携の必需品です。
- [26] R. クーラント, D. ヒルベルト：数理物理学の方法 1–4, 東京図書 (1959–1968), 齋藤利弥監訳, 丸山滋弥訳。
- [27] 近藤恭平：工学基礎 振動論, 培風館 (1993).
- [28] 入江敏博, 小林幸徳：機械振動学通論, 朝倉書店 (2006).
- [29] Gazzola, F., Grunau, H.-C. and Sweers, G.: *Polyharmonic Boundary Value Problems: Positivity Preserving and Nonlinear Higher Order Elliptic Equations in Bounded Domains*, Springer (2010).