２００４年度卒業研究

２次元円盤領域における波動方程式の研究

明治大学理工学部数学科

中　西　謙　太

指導教官　桂田　祐史

2005年2月28日

目 次

1　はじめに

今回私は、「2次元円盤領域における波動方程式」をテーマとして研究した。2次元円盤領域における波動方程式の初期値境界値問題を差分法を用いて解くことを目標とした。その方法として、陽解法と半陰スキームという2つの方法を用いて解いていった。また、解の安定性については1997年度桂田研、養田孝さんの卒業研究レポートを参考にした。

まず、円盤領域 $\Omega = \{(x, y); x^2 + y^2 < 1\}$ における2次元波動方程式

$$u_{tt} = u_{xx} + u_{yy}$$

を極座標変換し、円盤領域にして解いていくことにする。

2　空間2次元波動方程式（円盤領域）の初期値境界値問題

2.1　円盤領域の初期値境界値問題

解くべき領域の境界形状が円なので、極座標変換

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \quad (0 \le r \le 1, 0 \le \theta \le 2\pi) \end{aligned}$$

を行なう。ここで偏微分の関係式

$$\frac{\partial u}{\partial x} = \frac{\partial r}{\partial x}\frac{\partial u}{\partial r} + \frac{\partial \theta}{\partial x}\frac{\partial u}{\partial \theta} = r_x u_r + \theta_x u_\theta, \tag{1}$$

$$\frac{\partial u}{\partial y} = \frac{\partial r}{\partial y}\frac{\partial u}{\partial r} + \frac{\partial \theta}{\partial y}\frac{\partial u}{\partial \theta} = r_y u_r + \theta_y u_\theta \tag{2}$$

を用いる。式 (1)、(2) の u を x、y とおくと、

$$r_x x_r + \theta_x x_\theta = 1, \quad r_x y_r + \theta_x y_\theta = 0,$$
$$r_y x_r + \theta_y x_\theta = 0, \quad r_y y_r + \theta_y y_\theta = 1$$

となる。以上より r_x、θ_x、r_y、θ_y を解くと、

$$r_x = y_\theta/J, \quad \theta_x = -y_r/J, \tag{3}$$
$$r_y = -x_\theta/J, \quad \theta_y = x_r/J \tag{4}$$

となる。J はヤコビアンとよばれる量で、次式で定義される。

$$J = x_r y_\theta - x_\theta y_r.$$

(3)、(4) を (1)、(2) に代入すると $r-\theta$ 面の微係数のみで表された関係式

$$u_x = (y_\theta u_r - y_r u_\theta)/J, \tag{5}$$
$$u_y = (-x_\theta u_r + x_r u_\theta)/J \tag{6}$$

が得られる。2階微係数を求めるには、(3)、(4) を 2 度使う。すなわち、(1)、(2) の u の代わりに u_x、u_y とおき、整理すると、

$$\begin{aligned}
u_{xx} &= (y_\theta^2 u_{rr} - 2y_r y_\theta u_{r\theta} + y_r^2 u_{\theta\theta})/J^2 + [(y_\theta^2 y_{rr} - 2y_r y_\theta y_{r\theta} + y_r^2 y_{\theta\theta})(x_\theta u_r - x_r u_\theta) \\
&\quad +(y_\theta^2 x_{rr} - 2y_r y_\theta x_{r\theta} + y_r^2 x_{\theta\theta})(y_r u_\theta - y_\theta u_r)]/J^3, \\
u_{yy} &= (x_\theta^2 u_{rr} - 2x_r x_\theta u_{r\theta} + x_r^2 u_{\theta\theta})/J^2 + [(x_\theta^2 y_{rr} - 2x_r x_\theta y_{r\theta} + x_r^2 y_{\theta\theta})(x_\theta u_r - x_r u_\theta) \\
&\quad +(x_\theta^2 x_{rr} - 2x_r x_\theta x_{r\theta} + x_r^2 x_{\theta\theta})(y_r u_\theta - y_\theta u_r)]/J^3
\end{aligned}$$

が得られる。さらに $u_{xx}+u_{yy}$ を求めるために 2 式を加えると、

$$\begin{aligned}
u_{xx} + u_{yy} = (au_{rr} - 2bu_{r\theta} + cu_{\theta\theta})/J^3 + [(ax_{rr} - 2bx_{r\theta} + cx_{\theta\theta})(y_r u_\theta - y_\theta u_r) \\
+(ay_{rr} - 2by_{r\theta} + cy_{\theta\theta})(x_\theta u_r - x_r u_\theta)]/J^3, \\
a = x_\theta^2 + y_\theta^2, \quad b = x_r x_\theta + y_r y_\theta, \quad c = x_\theta^2 + y_\theta^2
\end{aligned}$$

となる。実際に計算すると、

$$\begin{aligned}
x_r = \cos\theta, \quad x_\theta = -r\sin\theta, \quad y_r = \sin\theta, \quad y_\theta = r\cos\theta, \\
a = x_\theta^2 + y_\theta^2 = r^2, \quad b = x_r x_\theta + y_r y_\theta = 0, \quad c = x_r^2 + y_r^2 = 1, \\
x_{rr} = 0, \quad x_{r\theta} = -\sin\theta, \quad x_{\theta\theta} = -r\cos\theta, \\
y_{rr} = 0, \quad y_{r\theta} = \cos\theta, \quad y_{\theta\theta} = -r\sin\theta
\end{aligned}$$

となるので、$u_{xx}+u_{yy}$ は、

$$\begin{aligned}
u_{xx} + u_{yy} &= \frac{r^2 u_{rr} + u_{\theta\theta}}{r^2} + \frac{1}{r^3}[-r\cos\theta(u_\theta \sin\theta - u_r r\cos\theta) - r\sin\theta(-u_r r\sin\theta - u_\theta\cos\theta)] \\
&= u_{rr} + \frac{1}{r^2}u_{\theta\theta} + \frac{1}{r}u_r
\end{aligned}$$

となり、問題は以下のようになる。

未知関数 $u(r,\theta,t)$（r, θ は空間座標。t は時刻。$0 \leq r \leq 1$, $0 \leq \theta \leq 2\pi$, $t \geq 0$）に関する 2 次元波動方程式を考える。

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} \quad (0 < r < 1,\ 0 \leq \theta < 2\pi,\ t > 0) \tag{7}$$

境界条件 (Dirichlet 境界条件) は、

$$u(1,\theta,t) = 0 \quad (t \geq 0, 0 \leq \theta < 2\pi) \tag{8}$$

となり、初期条件は、

$$u(r,\theta,0) = f(r,\theta) \quad (0 \leq r \leq 1,\ 0 \leq \theta \leq 2\pi) \tag{9}$$

$$\frac{\partial u}{\partial t}(r,\theta,0) = g(r,\theta) \quad (0 \leq r \leq 1,\ 0 \leq \theta \leq 2\pi) \tag{10}$$

となる。

2.2 Bessel 関数

この節の内容は NUMERICAL RECIPES in C を参考にした。任意の実数 ν に対して、ν 次第 1 種 Bessel 関数 $J_\nu(z)$ は

$$J_\nu(z) = (\frac{1}{2}z)^\nu \sum_{k=0}^{\infty} \frac{(-\frac{1}{4}z^2)^k}{k!\Gamma(\nu+k+1)}$$

で与えられる。ただし、これは一般には多価関数である。ν が 0 以上の整数の時には、任意の $z \in C$ に対して定義される。

第 2 種 Bessel 関数 $Y_\nu(z)$ は、ν が整数でないときには

$$Y_\nu(z) = \frac{J_\nu(z)\cos(\nu\pi) - J_\nu(z)}{\sin(\nu\pi)} \qquad (z \in R)$$

で、ν が整数 n に等しいときは

$$Y_\nu(z) = Y_n(z) = \lim_{\mu \to n} Y_\mu(z)$$

で定義される。

また変数 x が $x < \nu$ であるとき、特に $0 < x \ll \nu$ に対しては次の式で表される。

$$\begin{aligned} J_\nu(x) &\sim \frac{1}{\Gamma(\nu+1)}(\frac{1}{2}x)^\nu \quad (\nu \geq 0), \\ Y_0(x) &\sim \frac{2}{\pi}\ln(x), \qquad (\ln(x) = \log_e x) \\ Y_\nu(x) &\sim -\frac{\Gamma(\nu)}{\pi}(\frac{1}{2}x)^{-\nu} \quad (\nu > 0). \end{aligned}$$

次に $x > \nu$ であるとき、特に $x \gg \nu$ に対しては次の式で表される。

$$\begin{aligned} J_\nu(x) &\sim \sqrt{\frac{2}{\pi x}}\cos(x - \frac{1}{2}\nu\pi - \frac{1}{4}\pi), \\ Y_\nu(x) &\sim \sqrt{\frac{2}{\pi x}}\sin(x - \frac{1}{2}\nu\pi - \frac{1}{4}\pi) \end{aligned}$$

また $x \sim \nu$ での Bessel 関数の典型的な振幅は

$$\begin{aligned} J_\nu(\nu) &\sim \frac{2^{1/3}}{3^{2/3}\Gamma(\frac{2}{3})}\frac{1}{\nu^{1/3}} \sim \frac{0.4473}{\nu^{1/3}}, \\ Y_\nu(\nu) &\sim -\frac{2^{1/3}}{3^{1/6}\Gamma(\frac{2}{3})}\frac{1}{\nu^{1/3}} \sim -\frac{0.7748}{\nu^{1/3}} \end{aligned}$$

のように表される。

2.3 Bessel 関数を使った厳密解

まず (7),(8) の変数分離解を求める。$u(r,\theta,t)$ を

$$u(r,\theta,t) = R(r)\Theta(\theta)T(t) \tag{11}$$

とおく。式 (7) に式 (13) を代入し、$R(r)\Theta(\theta)T(t)$ で割ると、

$$\frac{T''}{T} = \frac{R''}{R} + \frac{1}{r}\frac{R'}{R} + \frac{1}{r^2}\frac{\Theta''}{\Theta} \tag{12}$$

となる。式 (14) が成り立つには、左辺、右辺が、定数関数の場合だけなので、その定数値を $-\lambda^2$ とおくと、

$$\frac{T''}{T} = \frac{R''}{R} + \frac{1}{r}\frac{R'}{R} + \frac{1}{r^2}\frac{\Theta''}{\Theta} = -\lambda^2.$$

よって

$$T'' + \lambda^2 T = 0 \tag{13}$$

となる。また、Θ についても同様に

$$\frac{\Theta''}{\Theta} = -r^2(\lambda^2 + \frac{R''}{R} + \frac{1}{r}\frac{R'}{R}) = -n^2 \qquad (n:\text{定数})$$

とおけるので、

$$\Theta'' + n^2\Theta = 0. \tag{14}$$

R は、λ^2、n^2 を使って整理すると、

$$r^2R'' + rR' + (\lambda^2 r^2 - n^2)R = 0 \tag{15}$$

となる。式 (13)、(14) より、

$$\begin{aligned} T(t) &= C_1\cos\lambda t + C_2\sin\lambda t, && (16)\\ \Theta(\theta) &= C_3\cos n\theta + C_4\sin n\theta && (17)\\ &\quad (C_1, C_2, C_3, C_4 : \text{定数},\quad n : \text{周期 } 2\pi \text{の解を持つために 0 以上の整数}) \end{aligned}$$

式 (17) は Bessel の方程式で、その解 $R(r)$ は、

$$R(r) = AJ_n(\lambda r) + BY_n(\lambda r) \qquad (A,\ B \text{ は定数})$$

である。J_n は $r=0$ で連続、Y_n は $r=0$ で不連続である。本題の条件では、$r=0$ で連続なので $B=0$ で

$$R(r) = AJ_n(\lambda r)$$

となる。一方、境界条件 $u(1,\theta,t)=0$ より

$$J_n(\lambda) = 0$$

が得られる。いいかえれば、$R(r)$ を円の境界上で 0 にするには、分離定数 λ を $J_n(\lambda)=0$ の根の一つであるように選ばなければならない。つまり、

$$\lambda = \nu_{n,m}$$

である。ここで、$\nu_{n,m}$ は $J_n(r)=0$ の小さい方から数えて m 番目の正の根である。以上より、

$$R(r) = AJ_n(\nu_{n,m} r)$$

となる。こうして変数分離解

$$u(r,\theta,t) = J_n(\nu_{n,m} r)(C_1 \cos \nu_{n,m} t + C_2 \sin \nu_{n,m} t)(C_3 \cos n\theta + C_4 \sin n\theta)$$

が求まった。

これらを重ね合わせすることにより波動方程式の混合問題の解が求まる。すなわち初期値を、

$$\begin{aligned} f(r,\theta) &= \sum_{n=0}^{\infty}\sum_{m=1}^{\infty} a^+_{n,m} J_n(\nu_{n,m} r)\cos n\theta \\ &\quad + \sum_{n=1}^{\infty}\sum_{m=1}^{\infty} a^-_{n,m} J_n(\nu_{n,m} r)\sin n\theta \end{aligned}$$

$$\begin{aligned} g(r,\theta) &= \sum_{n=0}^{\infty}\sum_{m=1}^{\infty} b^+_{n,m} J_n(\nu_{n,m} r)\cos n\theta \\ &\quad + \sum_{n=1}^{\infty}\sum_{m=1}^{\infty} b^-_{n,m} J_n(\nu_{n,m} r)\sin n\theta \end{aligned}$$

とおけば、解は

$$u(t,r,\theta) = \sum_{n=0}^{\infty}\sum_{m=1}^{\infty} a^+_{n,m} J_n(\nu_{n,m} r)\cos n\theta \cos(\nu_{n,m} t)$$

$$
\begin{aligned}
&+\sum_{n=1}^{\infty}\sum_{m=1}^{\infty} a_{n,m}^{-} J_n(\nu_{n,m} r)\sin n\theta \cos(\nu_{n,m} t) \\
&+\sum_{n=0}^{\infty}\sum_{m=1}^{\infty} b_{n,m}^{+} \frac{1}{\nu_{n,m}} J_n(\nu_{n,m} r)\cos n\theta \sin(\nu_{n,m} t) \\
&+\sum_{n=1}^{\infty}\sum_{m=1}^{\infty} b_{n,m}^{-} \frac{1}{\nu_{n,m}} J_n(\nu_{n,m} r)\sin n\theta \sin(\nu_{n,m} t)
\end{aligned}
$$

となる。

ここで $f(r,\theta), g(r,\theta)$ が与えられたとき $a_{n,m}^{+}, a_{n,m}^{-}, b_{n,m}^{+}, b_{n,m}^{-}$ を求める公式を紹介しておく。

$$
a_{n,m}^{+} = \frac{2}{\pi\{J_{n+1}(\nu_{n,m} r)\}^2}\int_0^1 J_n(\nu_{n,m} r)\{\int_{-\pi}^{\pi} f(r,\theta)\cos n\theta d\theta\} dr,
$$

$$
a_{n,m}^{-} = \frac{2}{\pi\{J_{n+1}(\nu_{n,m} r)\}^2}\int_0^1 J_n(\nu_{n,m} r)\{\int_{-\pi}^{\pi} f(r,\theta)\sin n\theta d\theta\} dr,
$$

$$
b_{n,m}^{+} = \frac{2}{\pi\{J_{n+1}(\nu_{n,m} r)\}^2}\int_0^1 J_n(\nu_{n,m} r)\{\int_{-\pi}^{\pi} g(r,\theta)\cos n\theta d\theta\} dr,
$$

$$
b_{n,m}^{-} = \frac{2}{\pi\{J_{n+1}(\nu_{n,m} r)\}^2}\int_0^1 J_n(\nu_{n,m} r)\{\int_{-\pi}^{\pi} g(r,\theta)\sin n\theta d\theta\} dr.
$$

ちなみに、実際のプログラムでは、厳密解を

$$
a_{0,1}^{+} = 1, b_{1,1}^{+} = 1,
$$

すなわち

$$
u(r,\theta,t) = J_0(\nu_{0,1} r)\cos(\nu_{0,1} t) + \frac{1}{\nu_{1,1}} J_1(\nu_{1,1} r)\cos\theta \sin(\nu_{1,1} t)
$$

とした。

2.3.1 2次元円盤のラプラシアンの固有値、固有関数

この部分は、金子「偏微分方程式入門」を参考にした。

円盤におけるラプラシアン $\triangle$ の固有値、固有関数について説明しておく。λ^2 が固有値で、U が λ^2 に属する固有関数であるとは、

$$\begin{aligned}&U_{rr}+\frac{1}{r}U_r+\frac{1}{r^2}U_{\theta\theta}=-\lambda^2 U,\\ &U=0 \quad (\text{円周上})\end{aligned}$$

をみたすことをいう。上の議論により、固有値 $\lambda^2=\nu_{nm}^2$ に属する固有関数として、

$$\begin{aligned}&J_n(\nu_{n,m}r)\cos n\theta, \qquad n=0,1,2,\cdots, \quad m=1,2,\cdots\\ &J_n(\nu_{n,m}r)\sin n\theta, \qquad n=1,2,\cdots, \qquad m=1,2,\cdots\end{aligned}$$

が得られた。

小さな固有値に属する固有関数について、節 ($u=0$ をみたす点の集合) の位置を見ていく。最初の固有値は $J_0(s)$ の第 1 零点 $\nu_{0,1}=2.40483\cdots$ に対応する単純固有値で、固有関数

$$J_0(\nu_{0,1}r)\cos 0\theta = J_0(\nu_{0,1}r)$$

は円盤の内部でいたるところ正であり、節を持たない。第 2 の固有値は $J_1(s)$ の第 1 零点 $\nu_{1,1}=3.83171\cdots$ に対応するもので重根であり、固有関数

$$J_1(\nu_{1,1}r)\cos\theta, \quad J_1(\nu_{1,1}r)\sin\theta$$

はそれぞれ y 軸、および x 軸を節に持つ（上の式が 0 になるのは θ が $0,\pi/2,\pi,3\pi/2$ のときだからである）。3 番目の固有値は $J_2(s)$ の第 1 零点 $\nu_{2,1}=5.13562\cdots$ に対応するもので重根であり、固有関数

$$J_2(\nu_{2,1}r)\cos 2\theta, \quad J_2(\nu_{2,1}r)\sin 2\theta$$

の節はそれぞれの両対角線の 2 本となる。4 番目の固有値は $J_0(s)$ の第 2 零点 $\nu_{0,2}=5.52008\cdots$ に戻り、固有関数

$$J_0(\nu_{0,2}r)\cos 0\theta = J_0(\nu_{0,2}r)$$

は J_0 の第 1 零点に対応した位置 $r=\nu_{0,1}/\nu_{0,2}=0.43565\cdots$ に円周状の節を一つ持つ。

混合問題の定常波は、固有関数に長方形と同様の時間関数因子をつければ得られる。すなわち、

$$J_n(\nu_{n,m}r)\begin{pmatrix}\cos\\ \text{または}\\ \sin\end{pmatrix} n\theta \begin{pmatrix}\cos\\ \text{または}\\ \sin\end{pmatrix}(\nu_{n,m}t)$$

の形となる。ここに sin と cos を縦に書いたところはどちらかを順不同で選択する意である。

3　円盤による差分法

3.1　陽解法

差分方程式の計算方法としてまず、陽解法から行う。領域の格子点を (r_i, θ_j) として各時間ステップごとに近似値を求める。

まず、差分近似式を導く。記号については h_r、h_θ、τ をそれぞれ r、θ、t 方向の格子間隔とする。空間変数 r の区間 $[0,1]$、θ の区間 $[0,2\pi]$ をそれぞれ N_r、N_θ 等分したものを間隔とし、時間変数 t については、τ を間隔とする。$r_i = ih_r$、$\theta_j = jh_\theta$、$t_n = n\tau$ での近似値を $u^n_{i,j}$ と書く。まず (7) の微分方程式の各項を中心差分近似、２階中心差分近似し、整理すると以下のようになる。

$$\frac{u^{n+1}_{i,j} - 2u^n_{i,j} + u^{n-1}_{i,j}}{\tau^2} = \frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{h_r^2} + \frac{1}{r_i}\frac{u^n_{i+1,j} - u^n_{i-1,j}}{2h_r} + \frac{1}{r_i^2}\frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{h_\theta^2}$$
$$(1 \leq i \leq N_r - 1,\ 0 \leq j \leq N_\theta - 1,\ n \geq 1)$$

$\frac{\tau}{h_r} = \lambda_r$、$\frac{\tau}{h_\theta} = \lambda_\theta$ として整理すると,

$$\begin{aligned}
u^{n+1}_{i,j} &= \lambda_r^2(u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}) + \frac{1}{r_i}\frac{\tau^2}{2h_r}(u^n_{i+1,j} - u^n_{i-1,j}) \\
&\quad + \frac{\lambda_\theta^2}{r_i^2}(u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}) + 2u^n_{i,j} - u^{n-1}_{i,j} \\
&= 2(1 - \lambda_r^2 - \frac{\lambda_\theta^2}{r_i^2})u^n_{i,j} + \lambda_r^2(u^n_{i+1,j} + u^n_{i-1,j}) + \frac{1}{r_i}\frac{\tau^2}{2h_r}(u^n_{i+1,j} - u^n_{i-1,j}) \\
&\quad + \frac{\lambda_\theta^2}{r_i^2}(u^n_{i,j+1} + u^n_{i,j-1}) - u^{n-1}_{i,j}
\end{aligned}$$

$$(1 \leq i \leq N_r - 1,\ 0 \leq j \leq N_\theta - 1,\ n \geq 1)$$

ただし、$u^n_{i,N_\theta} = u^n_{i,0}, u^n_{i,-1} = u^n_{i,N_\theta-1}$ とする。

初期条件より、

$$u^0_{i,j} = f_{i,j} \qquad (0 \leq i \leq N_r,\ 0 \leq j \leq N_\theta)$$

$\frac{\partial u}{\partial t}(r,\theta,0) = g(r,\theta)$ については Taylor の公式より、

$$\begin{aligned}
u(r,\theta,\tau) &= u(r,\theta,0) + \tau\frac{\partial u}{\partial t}(r,\theta,0) + \frac{\tau^2}{2}\frac{\partial^2 u}{\partial t^2}(r,\theta,0) + O(\tau^3) \\
&= f(r,\theta) + \tau g(r,\theta) + \frac{\tau^2}{2}(f_{rr}(r,\theta) + \frac{1}{r}f_r(r,\theta) + \frac{1}{r^2}f_{\theta\theta}(r,\theta)) + O(\tau^3)
\end{aligned}$$

導関数をいずれも中心差分商で近似すると、

$$\begin{aligned}
u^1_{i,j} &= f_{i,j} + \tau g_{i,j} + \frac{\tau^2}{2}(\frac{f_{i-1,j} - 2f_{i,j} + f_{i+1,j}}{h_r^2} + \frac{1}{r_i}\frac{f_{i+1,j} - f_{i-1,j}}{2h_r} + \frac{1}{r_i^2}\frac{f_{i,j-1} - 2f_{i,j} + f_{i,j+1}}{h_\theta^2}) \\
&= f_{i,j} + \tau g_{i,j} + \frac{1}{2}(\lambda_r^2(f_{i-1,j} - 2f_{i,j} + f_{i+1,j}) + \frac{1}{r_i}\frac{\tau^2}{2h_r}(f_{i+1,j} - f_{i-1,j}) \\
&\quad + \frac{\lambda_\theta^2}{r_i^2}(f_{i,j-1} - 2f_{i,j} + f_{i,j+1})) \\
&= (1 - \lambda_r^2 - \frac{\lambda_\theta^2}{r_i^2})f_{i,j} + \frac{1}{2}(\lambda_r^2(f_{i-1,j} + f_{i+1,j}) + \frac{1}{r_i}\frac{\tau^2}{2h_r}(f_{i+1,j} - f_{i-1,j}) \\
&\quad + \frac{\lambda_\theta^2}{r_i^2}(f_{i,j-1} + f_{i,j+1})) + \tau g_{i,j}
\end{aligned}$$

ただし、$f_{i,N_\theta} = f_{i,0}, f_{i,-1} = f_{i,N_\theta - 1}$ とする。

次に原点 $(r = 0)$ での微分方程式での近似を考える。(なぜなら $\frac{1}{r}, \frac{1}{r^2}$があるu$^{n+1}_{i,j}$ の近似式中では $r = 0$ では使えないので)

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$$

を元に戻して、

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

ここで半径 h_r, 中心が原点の円を描く、この円が x 軸 y 軸と交わる点における値は $u_{1,0}$,$u_{1,\frac{N_\theta}{4}}$, $u_{1,\frac{N_\theta}{2}}$,$u_{1,\frac{3N_\theta}{4}}$, 中心の値は $u_{0,0}$ となる。２階中心差分近似をして、

$$\frac{\partial^2 u}{\partial x^2}(0, 0, n\tau) = \frac{u_{1,0} - 2u_{0,0} + u_{1,N_\theta/2}}{h_r^2} + O(h_r)^2,$$

$$\frac{\partial^2 u}{\partial y^2}(0, 0, n\tau) = \frac{u_{1,N_\theta/4} - 2u_{0,0} + u_{1,3N_\theta/4}}{h_r^2} + O(h_r)^2$$

よってこれから、

$$\frac{\partial^2 u}{\partial x^2}(0, 0, n\tau) + \frac{\partial^2 u}{\partial y^2}(0, 0, n\tau) = \frac{u_{1,0} + u_{1,N_\theta/2} + u_{1,N_\theta/2} + u_{1,3N_\theta/4} - 4u_{0,0}}{h_r^2} + O(h_r)^2$$

以下、この回転を繰り返し行うことにより、

$$
\begin{aligned}
\Delta u(0,0,n\tau) &= \frac{u_{1,0}+u_{1,N_\theta/4}+u_{1,N_\theta/2}+u_{1,3N_\theta/4}-4u_{0,0}}{h_r^2}+O(h_r)^2 \\
&= \frac{u_{1,1}+u_{1,N_\theta/4+1}+u_{1,N_\theta/2+1}+u_{1,3N_\theta/4+1}-4u_{0,1}}{h_r^2}+O(h_r)^2 \\
&\ \ \vdots \\
&= \frac{4(\frac{1}{N_\theta}\sum_{j=0}^{N_\theta-1}u_{1,j}^n-u_{0,0}^n)}{h_r^2}+O(h_r)^2
\end{aligned}
$$

となる。そこで、

$$
\frac{u_{0,0}^{n+1}-2u_{0,0}^n+u_{0,0}^{n-1}}{\tau^2}=\frac{4(\frac{1}{N_\theta}\sum_{j=0}^{N_\theta-1}u_{1,j}^n-u_{0,0}^n)}{h_r^2}
$$

整理すると、

$$
u_{0,j}^{n+1}=2(1-2\lambda_r^2)u_{0,0}^n+\frac{4\lambda_r^2}{N_\theta}\sum_{j=0}^{N_\theta-1}u_{1,j}^n-u_{0,0}^{n-1}.
$$

という差分方程式が導かれる。
　また境界条件は、

$$
u_{N_r,j}^{n+1} = 0
$$

である。

3.2 半陰スキーム

後述する陽解法の安定性条件についての予想を元に考えると、安定性の条件を厳しくしているのは、

$$\Delta = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$$

の式の $\frac{1}{r^2}$ の存在である。

よって θ 方向の微分係数を陰的に扱うことにする。

$$\frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\tau^2} = \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{h_r^2} + \frac{1}{r_i}\frac{u_{i+1,j}^{n} - u_{i-1,j}^{n}}{2h_r} + \frac{1}{r_i^2}\frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{h_\theta^2}$$
$$(1 \le i \le N_r - 1,\ 0 \le j \le N_\theta - 1,\ n \ge 1)$$

$$(1 + 2\frac{\lambda_\theta^2}{r_i^2})u_{i,j}^{n+1} - \frac{\lambda_\theta^2}{r_i^2}(u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1}) = (2 - 2\lambda_r^2)u_{i,j}^{n} + \lambda_r^2[(1 + \frac{h_r}{2r_i})u_{i+1,j}^{n} + (1 - \frac{h_r}{2r_i})u_{i-1,j}^{n}] - u_{i,j}^{n-1} \quad (18)$$
$$(1 \le i \le N_r - 1,\ 1 \le j \le N_\theta - 1,\ n \ge 1)$$

ただし、$u_{i,N_\theta}^n = u_{i,0}^n$, $u_{i,-1}^n = u_{i,N_\theta - 1}^n$, 右辺に現れる $u_{N_r,j}^n$ は境界値から計算 (=$u(1, \theta_j, nt)$)、$u_{0,j}^n$ は（j が何であっても）原点での値とする。ここで

$$A_i := (1 + \tfrac{2\lambda_\theta^2}{r_i^2})I - \tfrac{\lambda_\theta^2}{r_i^2}J, \qquad I := N_\theta \text{ 次の単位行列},$$

$$J := \begin{pmatrix} 0 & 1 & & & 1 \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ 1 & & & 1 & 0 \end{pmatrix},$$

$$b_i^n = \begin{pmatrix} 2(1-\lambda_r^2)u_{i,0}^n + \lambda_r^2[(1+\frac{h_r}{2r_i})u_{i+1,0}^n + (1-\frac{h_r}{2r_i})u_{i-1,0}^n] \\ \vdots \\ 2(1-\lambda_r^2)u_{i,j}^n + \lambda_r^2[(1+\frac{h_r}{2r_i})u_{i+1,j}^n + (1-\frac{h_r}{2r_i})u_{i-1,j}^n] \\ \vdots \\ 2(1-\lambda_r^2)u_{i,N_{\theta-1}}^n + \lambda_r^2[(1+\frac{h_r}{2r_i})u_{i+1,N_{\theta-1}}^n + (1-\frac{h_r}{2r_i})u_{i-1,N_{\theta-1}}^n] \end{pmatrix} - \begin{pmatrix} u_{i,0}^{n-1} \\ \vdots \\ u_{i,j}^{n-1} \\ \vdots \\ u_{i,N_{\theta-1}}^{n-1} \end{pmatrix}$$

,

$$U_i^{n+1} := (u_{i,0}^{n+1}, \ldots, u_{i,N_\theta - 1}^{n+1})^T$$

とおくと (18) は

$$A_iU_i^{n+1} = b_i^n - U_i^{n-1} \quad (i = 1, 2, \cdots, N_r - 1) \tag{19}$$

と表される。

原点においては、陽解法のときと同様に

$$u_{0,0}^{n+1} = 2(1 - 2\lambda_r^2)u_{0,0}^n + \frac{4\lambda_r^2}{N_\theta}\sum_{j=0}^{N_\theta-1} u_{1,j}^n - u_{0,0}^{n-1}$$

を採用する。

また、$f(r,\theta)$ と $g(r,\theta)$ も同様に、

$$\begin{aligned} f(r,\theta) &= J_0(\nu_{0,1}r) & (0 \le r \le 1, 0 \le \theta \le 2\pi) \\ g(r,\theta) &= J_1(\nu_{1,1}r)\cos\theta & (0 \le r \le 1, 0 \le \theta \le 2\pi) \end{aligned}$$

とした。

(19) のタイプの係数行列を持った連立 1 次方程式の解き方とプログラムは桂田祐史 [熱方程式に対する差分法 II — 円盤領域、円柱領域、球における熱方程式 —] による。

4　差分解の安定性について

4.1　正方形領域の場合（参考）

まずは、参考として正方形領域においての安定性を挙げておく。この安定性の証明は 1997 年度桂田研, 養田孝さんの卒業論文を参考にした。

$\lambda_x^2 + \lambda_y^2 > 1$ となるように N_x、N_y、τ を選ぶと近似解が安定しないことがある。実は、

$$\lambda_x^2 + \lambda_y^2 \leq 1 \qquad (\frac{\tau^2}{h_x^2} + \frac{\tau^2}{h_y^2} \leq 1)$$

は解の安定性の十分条件であることが知られている。この条件を Courant-Friedrichs-Lewy の条件（CFL 条件）という。CFL 条件の証明は以下のとおりである。

$$u_{i,j}^k = g^k \exp[\sqrt{-1}(i\xi h_x + j\eta h_y)] \quad (g\text{: 増幅率}, \quad \xi, \eta\text{: 任意})$$

として、正方形領域の差分方程式に代入すると、

$$\begin{aligned} g^{k+1}\exp[\sqrt{-1}(i\xi h_x + j\eta h_y)] &= 2(1-\lambda_x^2+\lambda_y^2)g^k\exp[\sqrt{-1}(i\xi h_x + j\eta h_y)] \\ &\quad +\lambda_x^2(g^k\exp[\sqrt{-1}((i-1)\xi h_x + j\eta h_y)] \\ &\quad +g^k\exp[\sqrt{-1}((i+1)\xi h_x + j\eta h_y)]) \\ &\quad +\lambda_y^2(g^k\exp[\sqrt{-1}(i\xi h_x + (j-1)\eta h_y)] \\ &\quad +g^k\exp[\sqrt{-1}(i\xi h_x + (j+1)\eta h_y)]) \\ &\quad -g^{(k-1)}\exp[\sqrt{-1}(i\xi h_x + j\eta h_y)] \end{aligned}$$

整理すると、

$$\begin{aligned} g^2 &= [(\exp[\sqrt{-1}\xi h_x] - 2 + \exp[-\sqrt{-1}\xi h_x])\lambda_x^2 \\ &\quad +(\exp[\sqrt{-1}\eta h_y] - 2 + \exp[-\sqrt{-1}\eta h_y])\lambda_y^2 + 2]g - 1 \\ &= 2[\lambda_x^2(\cos\xi h_x - 1) + \lambda_y^2(\cos\eta h_y - 1) + 1]g - 1 \\ &= 2[\lambda_x^2(-2\sin^2\frac{\xi h_x}{2}) + \lambda_y^2(-2\sin^2\frac{\eta h_y}{2}) + 1]g - 1 \end{aligned}$$

よって、

$$g^2 - 2[1 - 2(\lambda_x^2\sin^2\frac{\xi h_x}{2} + \lambda_y^2\sin^2\frac{\eta h_y}{2})] + 1 = 0 \tag{20}$$

いま、式 (22) を　$g^2 - 2Ag + 1 = 0 \quad (A = 1 - 2\lambda_x^2 \sin^2 \frac{\xi h_x}{2} - 2\lambda_y^2 \sin^2 \frac{\eta h_y}{2})$ とおくと、

$$g = A \pm \sqrt{A^2 - 1}$$

$\mid g \mid \leq 1$ となるための条件を調べる。

$A^2 \ > \ 1$ のとき： $A > 1 \Rightarrow A + \sqrt{A^2 - 1} > 1,\ A < -1 \ \Rightarrow A - \sqrt{A^2 - 1} < 1$ で両方とも駄目。
$A^2 \ = \ 1$ のとき： $A = 1 \ \Rightarrow g = 1,\ \ A = 1 \ \Rightarrow g = -1$ 　で両方とも条件を満たす。
$A^2 \ < \ 1$ のとき： $g = A \pm \sqrt{1 - A^2}\sqrt{-1} \ \Rightarrow \mid g \mid = \sqrt{A^2 + (\sqrt{1 - A^2})^2} = 1$ 　で条件を満たす。

ゆえに、$\mid A \mid \leq 1$ のとき g が安定する。A の定義式を代入すると、

$$-1 \leq 1 - 2(\lambda_x^2 \sin^2 \frac{\xi h_x}{2} + \lambda_y^2 \sin^2 \frac{\eta h_y}{2}) \leq 1$$

となる。左の不等式から（右の不等式は明らか)、

$$\lambda_x^2 \sin^2 \frac{\xi h_x}{2} + \lambda_y^2 \sin^2 \frac{\eta h_y}{2} \leq 1$$

よって、

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

となる。

4.2　円領域 (陽解法) の場合

正方形領域の場合と同様に円領域の CFL 条件を導くことは難しい。しかし正方形の場合と同様にして安定性を考えると、安定性条件は

$$\lambda_r^2 + \frac{1}{r_i^2}\lambda_\theta^2 \leq 1 \qquad (i = 1, \cdots, N_r - 1)$$

であると予想される。いま $\frac{1}{r_i^2}$ が一番大きくなるのは r が一番小さくなるとき、つまり、$r_i = \frac{1}{N_r}$ のときであるから、安定条件は

$$\lambda_r^2 + N_r^2 \lambda_\theta^2 \leq 1$$

と書き換えることができる。

実際に $N_r = 20$、$N_\theta = 100$、$\tau = 0.0031$、$\lambda_r^2 + N_r^2\lambda_\theta^2 = 0.977618 < 1$ で安定である。$N_r = 20$、$N_\theta = 100$、$\tau = 0.0032$、$\lambda_r^2 + N_r^2\lambda_\theta^2 = 1.041617 > 1$ で不安定である。

4.2.1 厳密解との比較

厳密解と差分解（陽解法）との比較を行う。

厳密解を、

$$u(r,\theta,t) = J_0(\nu_{0,1}r)\cos(\nu_{0,1}t) + \frac{1}{\nu_{1,1}}J_1(\nu_{1,1}r)\cos\theta\sin(\nu_{1,1}t)$$

とすると初期条件 $f(r,\theta)$、$g(r,\theta)$ は、

$$\begin{aligned} f(r,\theta) &= J_0(\nu_{0,1}r) && (0 \le r \le 1, 0 \le \theta \le 2\pi) \\ g(r,\theta) &= J_1(\nu_{1,1}r)\cos\theta && (0 \le r \le 1, 0 \le \theta \le 2\pi) \end{aligned}$$

となる。

２つの最大誤差が、N_r, N_θ, τ を変えていくごとにどのように変化するかを調べる。$(x,y) = (r_i\cos\theta_j, r_i\sin\theta_j)$ とし、$(x,y) = (0.30, 0.00)$ を固定して誤差を計る。また、$\nu_{0,1} = 2.404825557695773, \nu_{1,1} = 3.831705970207512$ として計算した。

実験結果は以下のようになった。

N_r	N_θ	τ	最大誤差
5	25	0.48	1.40e-007
10	50	0.12×10^{-1}	1.64e-007
20	100	0.3×10^{-2}	1.94e-007
40	200	0.75×10^{-3}	2.23e-007
80	400	0.1875×10^{-4}	8.64e-008

N_r, N_θ をそれぞれ 2 倍にしていき、τ を 4 分の 1 していくとき、最大誤差のそれぞれの差は、ほぼ同じ値になった。

4.3 円領域 (半陰スキーム) の場合

陽解法の場合と同様に、このスキームが安定であるためには、

$$2 - 2\lambda_r^2 \geq 0$$

と予想できる。つまり安定条件は、

$$\lambda_r^2 \leq 1$$

実際に、$N_r = 20$、$N_\theta = 100$、$\tau = 0.048$、$\lambda_r^2 = 0.9216 < 1$ で安定である。
$N_r = 20$、$N_\theta = 100$、$\tau = 0.055$、$\lambda_r^2 = 1.0404 > 1$ で不安定である。

また、この結果からわかるように半陰スキームの場合、陽解法とは違い N_θ の値をどんなに小さくとろうが大きくとろうが、安定性の条件は N_θ によらないことがわかる。

4.3.1 厳密解との比較

厳密解と差分解（半陰スキーム）との比較を行う。
陽解法と同様に、厳密解を、

$$u(r,\theta,t) = J_0(\nu_{0,1}r)\cos(\nu_{0,1}t) + \frac{1}{\nu_{1,1}}J_1(\nu_{1,1}r)\cos\theta\sin(\nu_{1,1}t)$$

とすると初期条件 $f(r,\theta)$、$g(r,\theta)$ も、

$$\begin{aligned} f(r,\theta) &= J_0(\nu_{0,1}r) \qquad (0 \leq r \leq 1, 0 \leq \theta \leq 2\pi) \\ g(r,\theta) &= J_1(\nu_{1,1}r)\cos\theta \quad (0 \leq r \leq 1, 0 \leq \theta \leq 2\pi) \end{aligned}$$

となる。
陽解法と同様に考え、今度は $(x,y) = (0.15, 0.00)$ での点を固定し、$\nu_{0,1}$ と $\nu_{1,1}$ も同じ値として最大誤差を計る。
実験結果は以下のようになった。

N_r	N_θ	τ	最大誤差
5	25	0.16	1.42e-009
10	50	$0.8\ \times 10^{-1}$	1.42e-009
20	100	0.4×10^{-1}	1.39e-009
40	200	0.2×10^{-1}	1.40e-009
80	400	0.1×10^{-1}	1.42e-009

陽解法と同様に N_r, N_θ をそれぞれ 2 倍していき、今度は τ を 2 分の 1 倍していくとそれぞれの最大誤差はほぼ同じ値になった。

5　付録（プログラム）

以下、実験に用いたプログラムを紹介する。

２次元円領域 Dirichlet 境界条件（陽解法）

２次元円領域 Dirichlet 境界条件（半陰スキーム）

（なお、プログラムはすべて Java を用いて書いた。）

5.1 陽解法

```
/** ２次元波動方程式（円領域・陽解法）
 *
 *  <applet code = Wave2d width=500 height=500></applet>
 *
 */

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import corejava.*;
import Mitsui.*;
import cern.jet.math.*;

public class Wave2d extends Applet{
    MitsuiWorld_2 m = new MitsuiWorld_2();
    double xmin,xmax,ymin,ymax,zmin,zmax;
    private TextArea ta;

    public void init(){
        Graphics g=getGraphics();
        m.setGraphics(g);
        m.setScreenSize(getSize().width,getSize().height);
        //[a,b] × [c,d] × [e,f]
        xmin=-2.0;
        xmax=2.5;
        ymin=-2.0;
        ymax=2.5;
        zmin=-1.0;
        zmax=1.0;
        m.setArea2(xmin,xmax,ymin,ymax,zmin,zmax);
        m.setColor(3);
        add(ta = new TextArea(5,30));
        //ta.setBounds(0,0,10,7);
    }

    public void paint(Graphics g){

        int nr = 20;
        int nt = 20;
        double tau = 0.005;
        double Tmax = 1.0;
        double dt = 0.05;
        double skip = dt/tau+0.5;
        double hr = 1.0/nr;
        double ht = 2.0*Math.PI/nt;
        double lambdar = tau / hr;
        double lambdat = tau / ht;
```

```
double lambdar2 = lambdar * lambdar;
double lambdat2 = lambdat * lambdat;
double Kmax = (Tmax+0.1*tau)/tau;

double[][] u1 = new double[nr+1][nt+1];
double[][] u2 = new double[nr+1][nt+1];
double[][] u3 = new double[nr+1][nt+1];

//初期値代入
for(int i=0;i<nr;i++){
    for(int j=0;j<nt;j++){
        u1[i][j] = phi(i*hr,j*ht);
    }
}
{
    double mean = 0.0;
    for(int j=0;j<nt;j++)
        mean += u1[1][j];
    mean /= nt;
    for(int j=0;j<nt;j++)
        u1[0][j]=mean;
}
for(int i=0;i<=nr;i++){
    u1[i][nt]=u1[i][0];
}
for(int j=0;j<nt;j++){
    u1[nr][j]=0.0;
}

g.clearRect(0,0,getSize().width,getSize().height);
m.hideBirdView(u1,nr,nt,1);

for(int i=1;i<nr;i++){
    double ri=i*hr;
    for(int j=0;j<nt;j++){
        int jp1=j+1;
        if(jp1==nt){jp1=0;}
        int jm1=j-1;
        if(jm1==-1){jm1=nt-1;}
        u2[i][j]=(1.0-lambdar2-(lambdat2/(ri*ri)))*u1[i][j]
            +0.5*(lambdar2*(u1[i-1][j]+u1[i+1][j])
                  +((tau*tau)/(2.0*ri*hr))*(u1[i+1][j]-u1[i-1][j])
                  +(lambdat2/(ri*ri))*(u1[i][jm1]+u1[i][jp1]))
            +tau*psi(i*hr,j*ht);
    }
}
{
    double mean = 0.0;
    for(int j=0;j<nt;j++)
```

```
            mean += u2[1][j];
        mean /= nt;
        for(int j=0;j<nt;j++)
            u2[0][j]=mean;

    }
    for(int i=0;i<=nr;i++){
        u2[i][nt]=u2[i][0];
    }
    for(int j=0;j<nt;j++){
        u2[nr][j]=0.0;
    }

    g.clearRect(0,0,getSize().width,getSize().height);
    m.hideBirdView(u2,nr,nt,1);

    for(int k=1;k<=Kmax;k++){
        for(int i=1;i<nr;i++){
            double ri=i*hr;
            for(int j=0;j<nt;j++){
                int jp1=j+1;
                if(jp1==nt){jp1=0;}
                int jm1=j-1;
                if(jm1==-1){jm1=nt-1;}
                u3[i][j]=2.0*(1.0-lambdar2-(lambdat2/(ri*ri)))*u2[i][j]
                    +lambdar2*(u2[i+1][j]+u2[i-1][j])
                    +((tau*tau)/(2.0*ri*hr))*(u2[i+1][j]-u2[i-1][j])
                    +(lambdat2/(ri*ri))*(u2[i][jp1]+u2[i][jm1])-u1[i][j];
            }
        }

        //r=0
        {
            double mean=0.0;
            for(int j=0;j<nt;j++)
                mean += u3[1][j];
            mean /= nt;
            for(int j=0;j<nt;j++)
                u3[0][j]=mean;
        }

        //Dir(r=1)
        for(int j=0;j<nt;j++){
            u3[nr][j]=0.0;
        }
        //θ=2πのところはθ=0のところのコピー
        for(int i=0;i<=nr;i++){
            u3[i][nt]=u3[i][0];
        }
```

```
        g.clearRect(0,0,getSize().width,getSize().height);
        m.hideBirdView(u3,nr,nt,1);

        //u1 に u2 の値を u2 に u3 の値を代入
        for(int i=0;i<=nr;i++){
            for(int j=0;j<=nt;j++){
                u1[i][j] = u2[i][j];
                u2[i][j] = u3[i][j];
            }
        }

        double norm = maxnorm(u3,nr,nt);
        if(norm > 10.0)
            break;

        double t = tau*k;

        //誤差を測る
        double maxe = 0.0;
        double x = 0.0;
        double y = 0.0;
        double e = 0.0;
        for(int i=0;i<=nr;i++){
            double ri = i*hr;
            for(int j=0;j<=nt;j++){
                double theta_j = j * ht;
                e = Math.abs(exactu(ri,theta_j,t) - u3[i][j]);
                if(e > maxe){
                    maxe = e;
                    x = ri * Math.cos(theta_j);
                    y = ri * Math.sin(theta_j);
                }
            }
        }
        Format fmt = new Format("%7.2f");
        Format fmt_e = new Format("%7.2e");
        String str =ta.getText();
        if (str.length() > 1000)
            ta.setText("誤差"+fmt_e.form(e)+"\n(x,y)="
                       +fmt.form(x)+","+fmt.form(y)+")\n");
        else
            ta.setText(str+"誤差"+fmt_e.form(e)+"\n(x,y)="
                       +fmt.form(x)+","+fmt.form(y)+")\n");
        if(skip == 0){
            t = k*tau;
        }
    }
}
```

```
    //初期条件　ノルム
    public double maxnorm(double[][] u,int nr,int nt){
        int i0 = 0;
        int j0 = 0;
        double absu;
        double tmpmax = Math.abs(u[0][0]);
        for(int i=0;i<=nr;i++){
            for(int j=0;j<=nt;j++){
                if((absu = Math.abs(u[i][j]))>tmpmax){
                    tmpmax = absu;
                    i0 = i;
                    j0 = j;
                }
            }
        }
        return tmpmax;
    }

    double mu01 = 2.404825557695773;
    double mu11 = 3.831705970207512;
    double mu02 = 5.52008;

    //初期条件　φ
    public double phi(double r, double theta)
    {
        return Bessel.j0(mu01*r)+Bessel.j0(mu02*r);
    }

    //初期条件ψ
    public double psi(double r,double theta)
    {
        return 0.0;
    }

    //厳密解
    public double exactu(double r,double theta,double t){
        return 1.0*Bessel.j0(mu01*r)*Math.cos(mu01*t)
            +2.0*Bessel.j0(mu02*r)*Math.cos(mu02*t);
    }
}
```

5.2 半陰スキーム

```
/** 2次元波動方程式（円領域・半陰スキーム）
 *
 *  <applet code = Wave2d_i2 width=500 height=500></applet>
 *
 */

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import corejava.*;
import Mitsui.*;
import cern.jet.math.*;

public class Wave2d_i2 extends Applet{
    int nfunc=0;
    MitsuiWorld_2 m = new MitsuiWorld_2();
    double xmin,xmax,ymin,ymax,zmin,zmax;
    private TextArea ta;

    public void init(){
        Graphics g=getGraphics();
        m.setGraphics(g);
        m.setScreenSize(getSize().width,getSize().height);
        //[a,b] × [c,d] × [e,f]
        xmin=-2.0;
        xmax=2.5;
        ymin=-2.0;
        ymax=2.5;
        zmin=-1.0;
        zmax=1.0;
        m.setArea2(xmin,xmax,ymin,ymax,zmin,zmax);
        m.setColor(3);
        add(ta = new TextArea(5,30));
        //ta.setBounds(0,0,10,7);
    }

    public void paint(Graphics g){
        int nr = 20;
        int nt = 100;
        double tau = 0.045;
        double Tmax = 3.0;
        double dt = 0.05;
        double hr = 1.0/nr;
        double ht = 2.0*Math.PI/nt;
        double lambda_r = tau / hr;
        double lambda_t = tau / ht;
        double lambdar2 = lambda_r * lambda_r;
```

```
double lambdat2 = lambda_t * lambda_t;

double[][] u    = new double[nr+1][nt+1];
double[][] u1   = new double[nr+1][nt+1];
double[][] newu = new double[nr+1][nt+1];
double[][] al   = new double[nr][nt];
double[][] ad   = new double[nr][nt];
double[][] au   = new double[nr][nt];
double[][] ab   = new double[nr][nt];
double[][] ar   = new double[nr][nt];
double[]   b    = new double[nt+1];

if(dt < tau){
    dt = tau;
}

double skip = Math.rint(dt/tau);

g.clearRect(0,0,getSize().width,getSize().height);
m.hideBirdView(u,nr,nt,1);

//LU 分解
for(int i=1;i<nr;i++){
    double ri = i * hr;
    double ri2= ri * ri;
    double d  = 1+2*lambdat2/ri2;
    double od = -lambdat2 / ri2;
    for(int j=0;j<nt;j++){
        ad[i][j] = d;
        al[i][j] = au[i][j] = od;
        ab[i][j] = ar[i][j] = 0.0;
    }
    ab[i][0] = ab[i][nt-2] = ar[i][0] = ar[i][nt-2] = od;
    ptrilu0(nt,al[i],ad[i],au[i],ab[i],ar[i]);
}

for(int i=0;i<=nr;i++){
    double ri = i*hr;
    for(int j=0;j<=nt;j++){
        double theta_j = j*ht;
        u1[i][j] = phi(ri,theta_j);
    }
}
for(int i=0;i<=nr;i++){
    double ri = i*hr;
    for(int j=0;j<=nt;j++){
        double theta_j = j*ht;
        u[i][j] = phi(ri,theta_j) +tau*psi(ri,theta_j);
    }
```

```
}


//時間ループ
double nMax = Math.rint(Tmax / tau);
for(int n=1;n<=nMax;n++){
    for(int i=1;i<nr;i++){
        //右辺を作る
        double alpha = lambdar2 * (1.0+0.5/i);
        double beta  = lambdar2 * (1.0-0.5/i);
        for(int j=0;j<nt;j++){
            b[j] = 2*(1 - lambdar2)*u[i][j] + alpha * u[i+1][j]
                + beta * u[i-1][j] -u1[i][j];
        }
        //連立1次方程式を解く
        ptrisol0(nt,al[i],ad[i],au[i],ab[i],ar[i],b);
        //コピー
        for(int j=0;j<nt;j++)
            newu[i][j] = b[j];
        newu[i][nt] = b[0];
    }
    //Dir
    for(int j=0;j<=nt;j++)
        newu[nr][j] = 0.0;

    //原点
    double s = 0.0;
    for(int j=0;j<nt;j++)
        s += u[1][j];

    newu[0][0] = 4*lambdar2 * (s/nt-u[0][0]) + 2*u[0][0] - u1[0][0];

    for(int j=1;j<=nt;j++)
        newu[0][j] = newu[0][0];

    // newu, u, u1 更新
    for(int i=0;i<=nr;i++)
        for(int j=0;j<=nt;j++)
            u1[i][j] = u[i][j];
    for(int i=0;i<=nr;i++)
        for(int j=0;j<=nt;j++)
            u[i][j] = newu[i][j];

    g.clearRect(0,0,getSize().width,getSize().height);
    m.hideBirdView(u,nr,nt,1);

    double t = n * tau;

    //誤差を測る
```

```
        double maxe,e,x=0,y=0;
        maxe = 0;
        e = 0;
        for(int i=0;i<=nr;i++){
            double ri = i*hr;
            for(int j=0;j<=nt;j++){
                double theta_j = j * ht;
                e = Math.abs(exactu(ri,theta_j,t) - u[i][j]);
                if (e > maxe) {
                    maxe = e;
                    x = ri * Math.cos(theta_j);
                    y = ri * Math.sin(theta_j);
                }
            }
        }
        Format fmt = new Format("%7.2f");
        Format fmt_e = new Format("%7.2e");
        String str =ta.getText();
        if (str.length() > 1000)
            ta.setText("誤差"+fmt_e.form(e)+"\n(x,y)="
                    +fmt.form(x)+","+fmt.form(y)+")\n");
        else
            ta.setText(str+"誤差"+fmt_e.form(e)+"\n(x,y)="
                    +fmt.form(x)+","+fmt.form(y)+")\n");
        if(skip == 0){
            t = n*tau;
        }
    }
}

double mu01 = 2.404825557695773;
double mu11 = 3.831705970207512;

//初期値
public double phi(double r, double theta){
    return Bessel.j0(mu01*r);
}

public double psi(double r,double theta){
    return Bessel.j1(mu11*r)*Math.cos(theta);
}

//厳密解
public double exactu(double r,double theta,double t){
    return Bessel.j0(mu01*r)*Math.cos(mu01*t)+1/mu11*Bessel.j1(mu11*r)*
            Math.cos(theta)*Math.sin(mu11*t);
}

public double maxnorm(int m,int n,double[][] u){
```

```
    int i0 = 0;
    int j0 = 0;
    double absu;
    double tmpmax = Math.abs(u[0][0]);
    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            if((absu = Math.abs(u[i][j]))>tmpmax){
                tmpmax = absu;
                i0 = i;
                j0 = j;
            }
        }
    }
    return tmpmax;
}

public void ptrilu0(int n,
                    double[] al,double[] ad,double[] au,
                    double[] ab,double[] ar){
    int nm1 = n - 1;
    int nm2 = n - 2;
    for(int k=0;k<nm2;k++){
        int kp1 = k + 1;

        al[kp1] /= ad[k];
        ad[kp1] -= al[kp1]*au[k];
        if(kp1 < nm2)
            ar[kp1] -= al[kp1]*ar[k];
        else
            au[kp1] -= al[kp1]*ar[k];
        ab[k] = ab[k]/ad[k];
        if(kp1 < nm2)
            ab[kp1] -= ab[k]*au[k];
        else
            al[nm1] -= ab[k]*ar[k];
        ad[nm1] -= ab[k]*ar[k];
    }
    al[nm1] /= ad[nm2];
    ad[nm1] -= al[nm1]*au[nm2];
}

public void ptrisol0(int n,
                     double[] al,double[] ad,double[] au,
                     double[] ab,double[] ar,double[] b){
    int nm1 = n - 1;
    for(int k=0;k<nm1;k++){
        b[k+1] -= al[k+1]*b[k];
        if(k+1 < nm1 && k+1 > nm1)
            b[nm1] -= ab[k]*b[k];
```

```
    }
    b[nm1] = b[nm1]/ad[nm1];
    for(int k = n-2;k>=0;k--){
        b[k] -= au[k]*b[k+1];
        if(k+1 < nm1 && k+1 > nm1)
            b[k] -= ar[k]*b[nm1];
        b[k] /= ad[k];
    }
}
public void ptrilu1(int n,
                    double[] al,double[] ad,double[] au,
                    double[] ab,double[] ar){
    int nm1 = n - 1;
    for(int k=1;k<nm1;k++){
        int kp1 = k + 1;

        al[kp1] /= ad[k];
        ad[kp1] -= al[kp1] * au[k];
        if(kp1<nm1)
            ar[kp1] -= al[kp1]*ar[k];
        else
            au[kp1] -= al[kp1]*ar[k];

        ab[k] /= ad[k];
        if(kp1<nm1)
            ab[kp1] -= ab[k]*au[k];
        else
            al[n] -= ab[k]*au[k];
        ad[n] -= ab[k]*ar[k];
    }
    al[n] /= ad[nm1];
    ad[n] -= al[n]*au[nm1];
}

public void ptrisol1(int n,
                     double[] al,double[] ad,double[] au,
                     double[] ab,double[] ar,double[] b){
    for(int k=1;k<n;k++){
        b[k+1] -= al[k+1]*b[k];
        if(k+1<n && k+1>n)
            b[n] -= ab[k]*b[k];
    }
    b[n] = b[n]/ad[n];
    for(int k=n-1;k>=1;k--){
        b[k] -= au[k]*b[k+1];
        if(k+1 < n && k+1 > n)
            b[k] = b[k] - ar[k] * b[n];
        b[k] /= ad[k];
    }
```

```
    }
}
```

5.3 Bessel 関数

Bessel 関数を求めるコードついて、最初は NUMERICAL RECIPES in C を参考にしたが、後から

Colt Project (`http://dsd.lbl.gov/~hoschek/colt/`)

のライブラリィを利用することにした。プログラム先頭部分で

```
import cern.jet.math.*;
```

としているのが、colt ライブラリィを使うための宣言である。

6　あとがき

6.1　あとがき

私は今回２次元円盤領域における波動方程式について研究をしたのだが，もう少しいろいろな差分近似の方法を試してみたかったと思う。安定性の条件についてもいろいろな方法を研究してみたかったと思う。予想ではなく式によって証明をしたかった。さらにこれから円盤領域だけでなく楕円領域についても研究してくれる桂田研の学生がいてくれることを願っている。

Java についてももう少し勉強しておくべきだったと思う。

熱心に研究を手伝って下さり、申し訳ないと思うくらいに親身に見て下さった桂田先生に本当に深く感謝したい。

そして、４年間数学科でお世話になった先生方、そして友達に本当に「ありがとうございました。」と言いたい。この４年間で得たことを今後の自分の将来に役立てたい。

6.2　参考図書

偏微分方程式：スタンリー · ファーロウ著、伊理正夫、伊理由美訳：ワイリー · ジャパン

偏微分方程式入門：金子　晃著：東京大学出版会

数値解析入門 [増訂版]：山本哲郎著：サイエンス社

応用偏微分方程式：河村哲也著：共立出版株式会社

NUMERICAL RECIPES in C 〜 C 言語による数値計算のレシピ〜：William H.Press,Saul A.Teakolsky,William T.Vetterling,Brian P.Flannery 著、丹慶勝市、奥村晴彦、佐藤俊郎、小林誠訳 :技術評論社

養田孝、1997 年度卒業研究レポート：「２次元領域における波動方程式の研究」

桂田祐史, 熱方程式に対する差分法 II — 円盤領域、円柱領域、球における熱方程式 —, `http://www.math.meiji.ac.jp/~mk/labo/text/heat-fdm-2.pdf`