

3次元楕円体のS-W近似による波動方程式のシミュレーション

4年2組 30番 松澤 京平

2019年2月21日

目次

1	概要	3
2	3次元楕円体を2次元平面へ変数変換する	3
3	楕円盤領域での Shortley-Weller 近似	5
3.1	Shortley-Weller 近似とは	5
3.2	求める領域を差分格子で覆う	6
3.3	格子点の領域内外の判定	6
3.4	境界上の不等間隔格子点を取る	7
3.5	境界の座標の計算	7
3.6	S-W 近似の差分方程式	8
3.6.1	$r > 0$ の場合	8
3.6.2	$r = 0$ の場合	9
3.7	差分スキームの安定性条件,CFL 条件	9
3.8	丸め誤差への対処	11
3.9	差分方程式を解くために	12
4	数値計算	13
4.1	楕円領域の焦点	13
4.2	初期値の設定	13
4.3	実行結果と考察	14
A	付録	30
A.1	hadouDaentai.c	30

1 概要

2011 年度桂田研卒業の浜勇樹先輩の「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」[1]では、2次元の楕円領域で片方の焦点から発せられた波がもう片方の焦点に向かって反射し収束する様子のシミュレーションに挑戦をした。結果として3次元での計算の必要性を解いて終わっている。そこで2013年度桂田研卒業の嵯峨野美希先輩の「楕円体の酒場」[2]では、3次元の回転楕円領域で検証をした。しかし、プログラムの計算時間が長いことにより十分な試行回数が得られなかった。

そこで今回は、3次元の波動方程式を、解の対称性を用いて2次元に変数変換してシミュレーションを行うことができるかということを試した。計算は2次元の波動方程式として扱うため、浜勇樹先輩のプログラムを改良し、波動方程式を解くことによって再現できるかを試した。

2 3次元楕円体を2次元平面へ変数変換する

楕円領域 $\frac{x^2}{a^2} + \frac{z^2}{b^2} < 1$ を z 軸周りの回転してできる回転楕円体領域を Ω_1 とする:

$$\Omega_1 = \{(x, y, z); \frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{b^2} < 1\}$$

このとき Ω_1 において以下の波動方程式の初期値境界値問題を与える。

$$\begin{cases} \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} & ((x, y, z) \in \Omega_1, t > 0) \\ u(x, y, z, 0) = \phi_1(x, y, z) & ((x, y, z) \in \Omega_1) \\ \frac{\partial u}{\partial t}(x, y, z, 0) = \psi_1(x, y, z) & ((x, y, z) \in \Omega_1) \\ u(x, y, z, t) = 0 & ((x, y, z) \in \partial\Omega_1, t > 0) \end{cases}$$

c は正定数であり、初期値 $\phi_1(x, y, z), \psi_1(x, y, z)$ は既知関数である。

ここで Ω は回転楕円体なので、初期値が回転軸について回転対称ならば、波動方程式の解は対称になる。したがって、ある3変数関数 $U(r, z, t)$ が存在して

$$u(x, y, z, t) \rightarrow U(\sqrt{x^2 + y^2}, z, t)$$

が成り立つ。

$r = \sqrt{x^2 + y^2}$ から

$$\begin{aligned}\frac{\partial r}{\partial x} &= \frac{x}{\sqrt{x^2 + y^2}} \\ &= \frac{x}{r} \\ \frac{\partial^2 r}{\partial x^2} &= \frac{1}{\sqrt{x^2 + y^2}} - \frac{x^2}{(\sqrt{x^2 + y^2})^3} \\ &= \frac{1}{r} - \frac{x^2}{r^3}\end{aligned}$$

したがって

$$\begin{aligned}\frac{\partial u}{\partial x} &= \frac{\partial r}{\partial x} \cdot \frac{\partial U}{\partial r} \\ \frac{\partial^2 u}{\partial x^2} &= \frac{\partial}{\partial x} \left(\frac{\partial r}{\partial x} \frac{\partial U}{\partial r} \right) \\ &= \left(\frac{\partial}{\partial x} \frac{\partial r}{\partial x} \right) \frac{\partial U}{\partial r} + \left(\frac{\partial}{\partial x} \frac{\partial U}{\partial r} \right) \frac{\partial r}{\partial x} \\ &= \frac{\partial^2 r}{\partial x^2} \frac{\partial U}{\partial r} + \left(\frac{\partial x}{\partial r} \right)^2 \frac{\partial^2 U}{\partial r^2} \\ &= \frac{1}{r} \frac{\partial U}{\partial r} - \frac{x^2}{r^3} \frac{\partial U}{\partial r} + \frac{x^2}{r^2} \frac{\partial^2 U}{\partial r^2}\end{aligned}$$

y についても同様に計算を行うと

$$\frac{\partial^2 u}{\partial y^2} = \frac{1}{r} \frac{\partial U}{\partial r} - \frac{y^2}{r^3} \frac{\partial U}{\partial r} + \frac{y^2}{r^2} \frac{\partial^2 U}{\partial r^2}$$

したがってこれらの和は、

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= \frac{2}{r} \frac{\partial U}{\partial r} - \frac{x^2 + y^2}{r^3} \frac{\partial U}{\partial r} + \frac{x^2 + y^2}{r^2} \frac{\partial^2 U}{\partial r^2} \\ &= \frac{1}{r} \frac{\partial U}{\partial r} + \frac{\partial^2 U}{\partial r^2}\end{aligned}$$

したがって

$$\frac{1}{c^2} \frac{\partial^2 U}{\partial t^2} = \frac{1}{r} \frac{\partial U}{\partial r} + \frac{\partial^2 U}{\partial r^2} + \frac{\partial^2 U}{\partial z^2}$$

以上より、領域 Ω_2 を $\Omega_2 := \{(r, z); \frac{r^2}{a^2} + \frac{z^2}{b^2} < 1 \wedge r < 0\}$ とし、その境界を

$$\Gamma_1 := \{(r, z); \frac{r^2}{a^2} + \frac{z^2}{b^2} = 1 \wedge r > 0\}$$

$$\Gamma_2 := \{(r, z); r = 0 \wedge -b < z < b\}$$

とすると

$$\begin{cases} \frac{1}{c^2} \frac{\partial^2 U}{\partial t^2} = \frac{1}{r} \frac{\partial U}{\partial r} + \frac{\partial^2 U}{\partial r^2} + \frac{\partial^2 U}{\partial z^2} & ((x, y, z) \in \Omega_2, t > 0) \\ U(r, z, 0) = \phi_2(r, z) & ((r, z) \in \Omega_2) \\ \frac{\partial U}{\partial t}(r, z, 0) = \psi_2(r, z) & ((r, z) \in \Omega_2) \\ U(r, z, t) = 0 & ((r, z) \in \Gamma_1, t > 0) \\ \frac{\partial U}{\partial r}(r, z, t) = 0 & ((r, z) \in \Gamma_2, t > 0) \end{cases}$$

c は正定数であり、初期値 $\phi_2(x, y, z), \psi_2(x, y, z)$ は既知関数である。

浜勇樹先輩の「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」[1] を参考にし、C 言語を用いてプログラミングを行い、計算結果を GLSC を用いて可視化した。

3 楕円盤領域での Shortley-Weller 近似

3.1 Shortley-Weller 近似とは

通常、差分法は等間隔格子点を用いて行われる。これは領域が直方体の場合には問題ないが、円盤領域などには合わない。これらの領域で問題を解くには、変数変換によって領域を長方形に変換することで差分法を適用するのが普通である。しかし、以下説明する Shortley-Weller 近似ではそのままの領域に対して差分法を行うことが出来る。簡単のため 1 変数関数 f の場合で説明する。

$$(1) \quad f(x+h) = f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \frac{1}{3!}f^{(3)}(x)h^3 + \frac{1}{4!}f^{(4)}(x+\theta h)h^4$$

$$(2) \quad f(x-h') = f(x) - f'(x)h' + \frac{1}{2}f''(x)h'^2 - \frac{1}{3!}f^{(3)}(x)h'^3 + \frac{1}{4!}f^{(4)}(x+\theta'h')h'^4$$

(1) - (2) をすると

$$\begin{aligned} f(x+h) - f(x-h') &= f'(x)(h+h') + \frac{1}{2}f''(x)(h^2 - h'^2) + \frac{1}{3!}f^{(3)}(x)(h^3 + h'^3) \\ &\quad + \frac{1}{4!}f^{(4)}(x+\theta h - \theta'h')(h^4 - h'^4) \end{aligned}$$

ゆえに

$$f'(x) = \frac{f(x+h) - f(x-h')}{h+h'} + O(h-h') + O\left(\frac{h^3 + h'^3}{h+h'}\right) + O\left(\frac{h^4 + h'^4}{h+h'}\right)$$

また、(1)/ h + (2)/ h を整理すると

$$\begin{aligned} \frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h')}{h'} &= \frac{1}{2}f''(x)(h+h') + \frac{1}{3!}f^{(3)}(x)(h^2 - h'^2) \\ &\quad + \frac{1}{4!}(f^{(4)}(x+\theta h)h^3 + f^{(4)}(x-\theta h')h'^3) \end{aligned}$$

ゆえに

$$f''(x) = \frac{2}{h+h'} \left(\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h')}{h'} \right) + O(h-h') + O\left(\frac{h^3 + h'^3}{h+h'}\right)$$

となり、この近似を $f'(x)$, $f''(x)$ の Shortley-Wellwe 近似と呼ぶ。以下では、Shortley-Weller 近似のことを S-W 近似と略記する。

3.2 求める領域を差分格子で覆う

領域 Ω_2 を覆う正方領域 (もしくは長方領域) の左辺の r 座標を r_{\min} , 右辺の r 座標を r_{\max} , 下辺の z 座標を z_{\min} , 上辺の z 座標を z_{\max} , r 方向の分割数を N_r , z 方向の分割数を N_z , r 方向の格子点の間隔を h_r , z 方向の格子点の間隔を h_z , 時間の刻み幅 $\tau > 0$ とおく。

$$\begin{aligned} h_r &= \frac{r_{\max} - r_{\min}}{N_r} \\ h_z &= \frac{z_{\max} - z_{\min}}{N_z} \end{aligned}$$

が成り立つ。さらに、 r_i, z_j を

$$\begin{aligned} r_i &= r_{\min} + ih_r \quad (0 \leq i \leq N_r) \\ z_j &= z_{\min} + jh_z \quad (0 \leq j \leq N_z) \end{aligned}$$

とする。

3.3 格子点の領域内外の判定

楕円領域 Ω_2 に対して、関数 F を $F(r, z) = a^2b^2 - b^2r^2 - a^2z^2$ で定めると、任意の $P = (r, z)$ に対して、以下が成り立ち、任意の点 P が領域の内部、外部、境界

のどこにあるか判断できる。

(1) $F(r, z) > 0$ のとき

$$\begin{cases} r > 0 \text{ のとき領域内部} \\ r = 0 \text{ のとき境界上} \\ r < 0 \text{ のとき領域外部} \end{cases}$$

(2) $F(r, z) = 0$ のとき

$$\begin{cases} r \geq 0 \text{ のとき境界上} \\ r < 0 \text{ のとき領域外部} \end{cases}$$

(3) $F(r, z) < 0$ のとき領域外部

3.4 境界上の不等間隔格子点を取る

等間隔の格子点によって分けられた領域内のある格子点 $P = (r, z)$ に対して、その東西南北の等間隔格子点が領域外に出てしまう場合、境界上に新しく不等間隔格子点を取る。西側、東側、北側、南側の点をそれぞれ

$$P_w = (r_w, z), P_e = (r_e, z), P_n = (r, z_n), P_s = (r, z_s)$$

と置く。新たに出来た左右上下の格子点の間隔をそれぞれ

$$\begin{aligned} h_w &= r_w - r = \varepsilon_w h_r, & h_e &= r - r_e = \varepsilon_e h_r \\ h_n &= z_n - z = \varepsilon_n h_z, & h_s &= z - z_s = \varepsilon_s h_z \end{aligned}$$

$$(0 < \varepsilon_w, \varepsilon_e, \varepsilon_n, \varepsilon_s \leq 1)$$

と置く。

3.5 境界の座標の計算

格子点の間に存在する境界の座標を計算する。楕円領域 Ω_2 の境界 Γ_1 上では

$$\frac{r^2}{a^2} + \frac{z^2}{b^2} = 1 \quad (r > 0)$$

が満たされ、 r, z についてそれぞれについて解くと

$$r = \pm a \sqrt{1 - \frac{z^2}{b^2}}, \quad z = \pm b \sqrt{1 - \frac{r^2}{a^2}}$$

となる。これによって境界上の r 座標、 z 座標が分かるので、格子点の東南北のどの等間隔格子点が境界外に出るかに合わせて、 P_e, P_n, P_s を決める。ただし、境界 Γ_2 上の r は $r = 0$ を満たす。よって西側が境界外に出るときは $P_w = (0, z)$ となる。

3.6 S-W 近似の差分方程式

3.6.1 $r > 0$ の場合

通常差分方程式では、

$$\frac{1}{r} U_r + U_{rr} + U_{zz} \cong \frac{1}{r_i} \frac{U_e - U_w}{2h_r} + \frac{U_w - 2U + U_e}{h_r^2} + \frac{U_n - 2U + U_s}{h_z^2}$$

S-W 近似では、

$$\frac{1}{r} U_r + U_{rr} + U_{zz} \cong \frac{1}{r_i} \frac{U_e - U_w}{h_w + h_e} + \frac{2}{h_w + h_e} \left(\frac{U_e - U}{h_e} - \frac{U - U_w}{h_w} \right) + \frac{2}{h_n + h_s} \left(\frac{U_n - U}{h_n} - \frac{U - U_s}{h_s} \right)$$

と近似する。波動方程式

$$\frac{1}{c^2} u_{tt} = \frac{1}{r} U_r + U_{rr} + U_{zz}$$

の左辺を 2 階の中心差分で近似すると、差分方程式は

$$\begin{aligned} \frac{1}{c^2} \frac{U_{i,j}^{n+1} - 2U_{i,j}^n + U_{i,j}^{n-1}}{\tau^2} &= \frac{1}{r_i} \frac{U_e - U_w}{h_w + h_e} + \frac{2}{h_w + h_e} \left(\frac{U_e - U_{i,j}^n}{h_e} - \frac{U_{i,j}^n - U_w}{h_w} \right) \\ &\quad + \frac{2}{h_n + h_s} \left(\frac{U_n - U_{i,j}^n}{h_n} - \frac{U_{i,j}^n - U_s}{h_s} \right) \end{aligned}$$

という形になる。 $U_{i,j}^{n+1}$ について整理すると、

$$\begin{aligned} U_{i,j}^{n+1} &= 2U_{i,j}^n - U_{i,j}^{n-1} + \frac{c^2 \tau^2}{r_i} \frac{U_e - U_w}{h_w + h_e} + \frac{2c^2 \tau^2}{h_w + h_e} \left(\frac{U_e - U_{i,j}^n}{h_e} - \frac{U_{i,j}^n - U_w}{h_w} \right) \\ &\quad + \frac{2c^2 \tau^2}{h_n + h_s} \left(\frac{U_n - U_{i,j}^n}{h_n} - \frac{U_{i,j}^n - U_s}{h_s} \right) \end{aligned}$$

となる。これは $h_w = h_e = h_n = h_s = h$ のとき、通常差分方程式になる。

3.6.2 $r = 0$ の場合

$r = 0$ では、波動方程式の左辺の分母に r があるため解くことができない。元の方程式

$$\frac{1}{c^2} w_{tt} = w_{xx} + w_{yy} + w_{zz}$$

について考える。

$W_{i,j,k} = w(x_i, y_j, z_k)$ として $i = 0, j = 0$ を代入し、2階中心差分近似すると

$$\frac{\partial^2 w}{\partial x^2}(0, 0, z, t) = \frac{W_{1,0,k} - 2W_{0,0,k} + W_{-1,0,k}}{h_x^2} + O(h_x^2)$$

$$\frac{\partial^2 w}{\partial y^2}(0, 0, z, t) = \frac{W_{0,1,k} - 2W_{0,0,k} + W_{0,-1,k}}{h_y^2} + O(h_y^2)$$

$h_x = h_y = h$ とする。また、解の対称性より

$$\begin{aligned} \frac{\partial^2 w}{\partial x^2}(0, 0, z, t) + \frac{\partial^2 w}{\partial y^2}(0, 0, z, t) &\cong \frac{W_{1,0,k} + W_{-1,0,k} + W_{0,1,k} + W_{0,-1,k} - 4W_{0,0,k}}{h^2} \\ &= \frac{4}{h^2}(W_{1,0,k} - W_{0,0,k}) \end{aligned}$$

次に $h = h_r$ とし、それぞれの座標を $w(x_i, y_j, z_k) = U(r_l, z_k)$ を用いて対応させると、 $r = 0$ での差分方程式は、

$$\frac{1}{c^2} \frac{U_{l,k}^{n+1} - 2U_{i,j}^n + U_{l,k}^{n-1}}{\tau^2} = \frac{4}{h_r^2} (U_{1,k}^n - U_{0,k}^n) + \frac{2}{h_n + h_s} \left(\frac{U_n - U_{l,k}^n}{h_n} - \frac{U_{l,k}^n - U_s}{h_s} \right)$$

$U_{l,k}^{n+1}$ について整理すると、

$$U_{l,k}^{n+1} = 2U_{0,k}^n - U_{0,k}^{n-1} + \frac{4c^2\tau^2}{h_r^2} (U_{1,k}^n - U_{0,k}^n) + \frac{2c^2\tau^2}{h_n + h_s} \left(\frac{U_n - U_{l,k}^n}{h_n} - \frac{U_{l,k}^n - U_s}{h_s} \right)$$

となる。

3.7 差分スキームの安定性条件, CFL 条件

Wikipedia[4] には

CFL 条件 (Courant-Friedrichs-Lewy Condition じょうけん) とは、コンピュータシミュレーションの計算 (数値解析) において、「情報が伝播する速さ」を「実際の現象で波や物理量が伝播する速さ」よりも早くしなければならないという必要条件のことである。

とある。

今回の波動方程式の場合に当てはめて考えると、等間隔格子を使った差分法の場合

$$\left(\frac{c\tau}{h_r}\right)^2 + \left(\frac{c\tau}{h_z}\right)^2 \leq 1$$

となる。これを变形すると、

$$\tau \leq \sqrt{\frac{h_r^2 h_z^2}{c^2(h_r^2 + h_z^2)}}$$

という τ の条件が得られる。

次に、S-W 近似における安定性条件について考える。まず、

$$\frac{1}{r}U_r + U_{rr} + U_{zz} \cong \frac{1}{r} \frac{U_e - U_w}{h_w + h_e} + \frac{2}{h_w + h_e} \left(\frac{U_e - U}{h_e} - \frac{U - U_w}{h_w} \right) + \frac{2}{h_n + h_s} \left(\frac{U_n - U}{h_n} - \frac{U - U_s}{h_s} \right)$$

の右辺に $h_w = \varepsilon_w h_r, h_e = \varepsilon_e h_r, h_n = \varepsilon_n h_z, h_s = \varepsilon_s h_z$ を代入すると

$$\frac{1}{r}U_r + U_{rr} + U_{zz} \cong \frac{1}{r} \frac{U_e - U_w}{\varepsilon_w h_r + \varepsilon_e h_r} + \frac{2}{\varepsilon_w h_r + \varepsilon_e h_r} \left(\frac{U_e - U}{\varepsilon_e h_r} - \frac{U - U_w}{\varepsilon_w h_r} \right) + \frac{2}{\varepsilon_n h_z + \varepsilon_s h_z} \left(\frac{U_n - U}{\varepsilon_n h_z} - \frac{U - U_s}{\varepsilon_s h_z} \right)$$

となる。これを变形すると、

$$\begin{aligned} \frac{1}{r}U_r + U_{rr} + U_{zz} \cong & \frac{1}{r} \frac{U_e - U_w}{h_r(\varepsilon_w + \varepsilon_e)} + \frac{2(\varepsilon_w U_e + \varepsilon_e U_w)}{h_r^2 \varepsilon_w \varepsilon_e (\varepsilon_w + \varepsilon_e)} + \frac{2(\varepsilon_s U_n + \varepsilon_n U_s)}{h_z^2 \varepsilon_s \varepsilon_n (\varepsilon_s + \varepsilon_n)} \\ & - \frac{2(\varepsilon_w \varepsilon_e h_r^2 + \varepsilon_s \varepsilon_n h_z^2)}{\varepsilon_w \varepsilon_e \varepsilon_s \varepsilon_n h_r^2 h_z^2} U \end{aligned}$$

となる。ここで

$$U_{tt} \cong \frac{1}{c^2} \frac{U_{i,j}^{n+1} - 2U_{i,j}^n + U_{i,j}^{n-1}}{\tau^2}$$

より、差分方程式は、

$$\begin{aligned} \frac{1}{c^2} \frac{U_{i,j}^{n+1} - 2U_{i,j}^n + U_{i,j}^{n-1}}{\tau^2} = & \frac{1}{r_i} \frac{U_e - U_w}{h_r(\varepsilon_w + \varepsilon_e)} + \frac{2(\varepsilon_w U_e + \varepsilon_e U_w)}{h_r^2 \varepsilon_w \varepsilon_e (\varepsilon_w + \varepsilon_e)} + \frac{2(\varepsilon_s U_n + \varepsilon_n U_s)}{h_z^2 \varepsilon_s \varepsilon_n (\varepsilon_s + \varepsilon_n)} \\ & - \frac{2(\varepsilon_w \varepsilon_e h_r^2 + \varepsilon_s \varepsilon_n h_z^2)}{\varepsilon_w \varepsilon_e \varepsilon_s \varepsilon_n h_r^2 h_z^2} U \end{aligned}$$

となる。 U^{n+1} について解くと

$$U^{n+1} = -U^{n-1} + \left\{ \frac{1}{r_i} \frac{U_e - U_w}{h_r(\varepsilon_w + \varepsilon_e)} + \frac{2(\varepsilon_w U_e + \varepsilon_e U_w)}{h_r^2 \varepsilon_w \varepsilon_e (\varepsilon_w + \varepsilon_e)} \right\} c^2 \tau^2 + \left\{ 2 - \frac{2c^2 \tau^2 (\varepsilon_w \varepsilon_e h_r^2 + \varepsilon_s \varepsilon_n h_z^2)}{\varepsilon_w \varepsilon_e \varepsilon_s \varepsilon_n h_r^2 h_z^2} \right\} U_{i,j}^n$$

となる。第三項の係数が正になるようにすると、

$$2 \geq \frac{2c^2\tau^2(\varepsilon_w\varepsilon_e h_r^2 + \varepsilon_s\varepsilon_n h_z^2)}{\varepsilon_w\varepsilon_e\varepsilon_s\varepsilon_n h_r^2 h_z^2}$$

となる。これを τ について整理すると、

$$\tau \leq \sqrt{\frac{\varepsilon_w\varepsilon_e\varepsilon_s\varepsilon_n h_r^2 h_z^2}{c^2(\varepsilon_w\varepsilon_e h_r^2 + \varepsilon_s\varepsilon_n h_z^2)}}$$

となる。

$\varepsilon_w = \varepsilon_e = \varepsilon_n = \varepsilon_s = 1$ のとき、この式は上に述べた等間隔格子の CFL 条件に一致しているので、2次元波動方程式を S-W 近似によって解く場合の CFL 条件であると考えられる。

また、

$$\varepsilon = \min\{\varepsilon_w, \varepsilon_e, \varepsilon_n, \varepsilon_s\}$$

とおく。

3.8 丸め誤差への対処

格子点が境界上にある場合に、プログラムが丸め誤差のせいで領域の内部だと判断することがある。よってその対策をプログラムに組み込む必要がある。何の対策もしなければ、その等間隔格子点上を領域の内部だと勘違いをして ε を極端に小さい値として返す。double 型では、有効数字はだいたい $10^{-14} \sim 10^{-15}$ ぐらいである。この対策として $\varepsilon_r = 10^{-13} \sim 10^{-14}$ というものを置き、これを用いて点 (x, y) が領域の内部、外部、境界のどこに位置するのか判断する。

(1) $F(r, z) > \varepsilon_r$ のとき

$$\begin{cases} r > \varepsilon_r \text{ のとき領域内部} \\ r \geq \varepsilon_r \text{ のとき境界上} \\ r < \varepsilon_r \text{ のとき領域外部} \end{cases}$$

(2) $F(r, z) \leq \varepsilon_r$ のとき

$$\begin{cases} r \geq \varepsilon_r \text{ のとき境界上} \\ r < \varepsilon_r \text{ のとき領域外部} \end{cases}$$

(3) $F(r, z) < \varepsilon_r$ のとき領域外部

上記のように判断することにする。また、この誤差は、領域の形や N_r, N_z の大小によって多少変化する。なので、もし ε が ε_r を下回るとき、そのことを出力するようにプログラムを組み込んでおく。

3.9 差分方程式を解くために

差分方程式を見ると $n+1$ ステップ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 $n-1$ での値が必要になる。これは、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る。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、 $n=1$ での値も必要になる。[3] を参考に、そのための計算を行う。 r, z を固定した 1 変数関数 $t \mapsto U(r, z, t)$ の $t=0$ におけるテーラー展開を行う。

$$U(r, z, \tau) \cong U(r, z, 0) + \tau \frac{\partial U}{\partial t}(r, z, 0) + \frac{\tau^2}{2} \frac{\partial^2 U}{\partial t^2}(r, z, 0)$$

において、関係式

$$\frac{1}{c^2} \frac{\partial^2 U}{\partial t^2}(r, z, 0) = \frac{1}{r} \frac{\partial U}{\partial r}(r, z, 0) + \frac{\partial^2 U}{\partial r^2}(r, z, 0) + \frac{\partial^2 U}{\partial z^2}(r, z, 0) = \Delta \phi_2$$

が成立する。

$$\begin{aligned} U(r, z, \tau) &\cong U(r, z, 0) + \tau \frac{\partial U}{\partial t}(r, z, 0) + \frac{\tau^2 c^2}{2} \left\{ \frac{1}{r} \frac{\partial U}{\partial r}(r, z, 0) + \frac{\partial^2 U}{\partial r^2}(r, z, 0) + \frac{\partial^2 U}{\partial z^2}(r, z, 0) \right\} \\ &\cong \phi_2(r_i, z_j) + \tau \psi_2(r_i, z_j) \\ &\quad + \frac{c^2 \tau^2}{2} \left\{ \frac{1}{r_i} \frac{U_e - U_w}{h_w + h_e} + \frac{2}{h_w + h_e} \left(\frac{U_e - U_{i,j}^0}{h_e} - \frac{U_{i,j}^0 - U_w}{h_w} \right) + \frac{2}{h_n + h_s} \left(\frac{U_n - U_{i,j}^0}{h_n} - \frac{U_{i,j}^0 - U_s}{h_s} \right) \right\} \end{aligned}$$

これを整理すると

$$\begin{aligned} U_{i,j}^1 &= \phi_2(r_i, z_j) + \tau \psi_2(r_i, z_j) \\ &\quad + c^2 \tau^2 \left\{ \frac{1}{2r_i} \frac{U_e - U_w}{h_w + h_e} + \frac{1}{h_w + h_e} \left(\frac{U_e - U_{i,j}^0}{h_e} - \frac{U_{i,j}^0 - U_w}{h_w} \right) + \frac{1}{h_n + h_s} \left(\frac{U_n - U_{i,j}^0}{h_n} - \frac{U_{i,j}^0 - U_s}{h_s} \right) \right\} \end{aligned}$$

という式が得られる。 $r=0$ のときも同様に計算すると

$$U_{0,j}^1 = \phi_2(0, z_j) + \tau \psi_2(0, z_j) + \frac{2c^2 \tau^2}{h_r^2} (U_{1,j}^0 - U_{0,j}^0) + \frac{c^2 \tau^2}{h_n + h_s} \left(\frac{U_n - U_{i,j}^0}{h_n} - \frac{U_{i,j}^0 - U_s}{h_s} \right)$$

という式が得られる。これで数値計算の準備ができた。

4 数値計算

4.1 楕円領域の焦点

領域 Ω_2 で波動方程式を解くにあたって、 $(0, z_0)$ を $a < b$ の場合の z 軸上にある正の値の焦点の座標とする。以下の実験では $a = 3, b = 5$ とする。すなわち

$$\Omega_2 := \{(r, z); \frac{r^2}{9} + \frac{z^2}{25} < 1 \wedge r > 0\}$$

とする。 $z_0 = \sqrt{b^2 - a^2} = 4$ であり、 $(0, 4)$ は楕円の焦点となる。

4.2 初期値の設定

検証したいのは、片方の焦点から発せられた波がもう片方の焦点に向かって反射し収束するのかということである。よって片方の焦点上にピークをもち領域内のいたるところで 0 になるような初期値をあたえる。この実験ではこのような初期値を 2 種類用意した。

(nfunc=1)

$a < b$ の場合 $H(r, z) = K \exp(-M\{r^2 + (z - z_0)^2\})$ を用意し、

$$\phi_2(r, z) = \begin{cases} H(r, z) - 0.1 & (H(r, z) \geq 0.1) \\ 0 & (H(r, z) < 0.1) \end{cases}$$

とし、 ψ_2 については

$$\psi_2(r, z) = 0$$

とした。

(nfunc=2)

$$f(p) = \begin{cases} (p^2 - 1)^4 & (p < 1) \\ 0 & (p \geq 1) \end{cases}$$

$$g(x, y) = K f\left(\frac{\|x\|}{0.5}\right)$$

を用意する。 g は $x = 0$ で最大値 K 、 $\|x\| > 0.5$ で $f(x) = 0$ となる。

$$\phi_2(r, z) = g(r, z - z_0).$$

とし、 ψ_2 については

$$\psi_2(r, z) = 0$$

とした。 K, M で初期値の波のサイズ（幅、高さ）を変更できる。今回は $K = 5$, $M = 10$ とした。

4.3 実行結果と考察

楕円は平面上の二つの焦点からの距離の和が一定であるという性質を持っている。短軸 $a = 3$ 、長軸 $b = 5$ 、波の速さ $c = 1$ という条件から片方の焦点から出た波が境界で反射し、もう片方の焦点へ行くのにかかる時間は 10 と予想される。

また、格子の分割数と時間の刻み幅を

(1) $N_r = N_z = 500, \tau = 0.001$

(2) $N_r = N_z = 1000, \tau = 0.0005$

(3) $N_r = N_z = 2000, \tau = 0.00025$

(4) $a = b = 2, N_r = N_z = 500, \tau = 0.002$

(5) $N_r = N_z = 600, \tau = 0.0005$ で $t = 10$ 前後の焦点を拡大したプロットと実行した。

(6) $N_r = N_z = 500, \tau = 0.001$ と $N_r = N_z = 2000, \tau = 0.00025$ の場合の各時刻での最大値をプロットした図も作成をした。

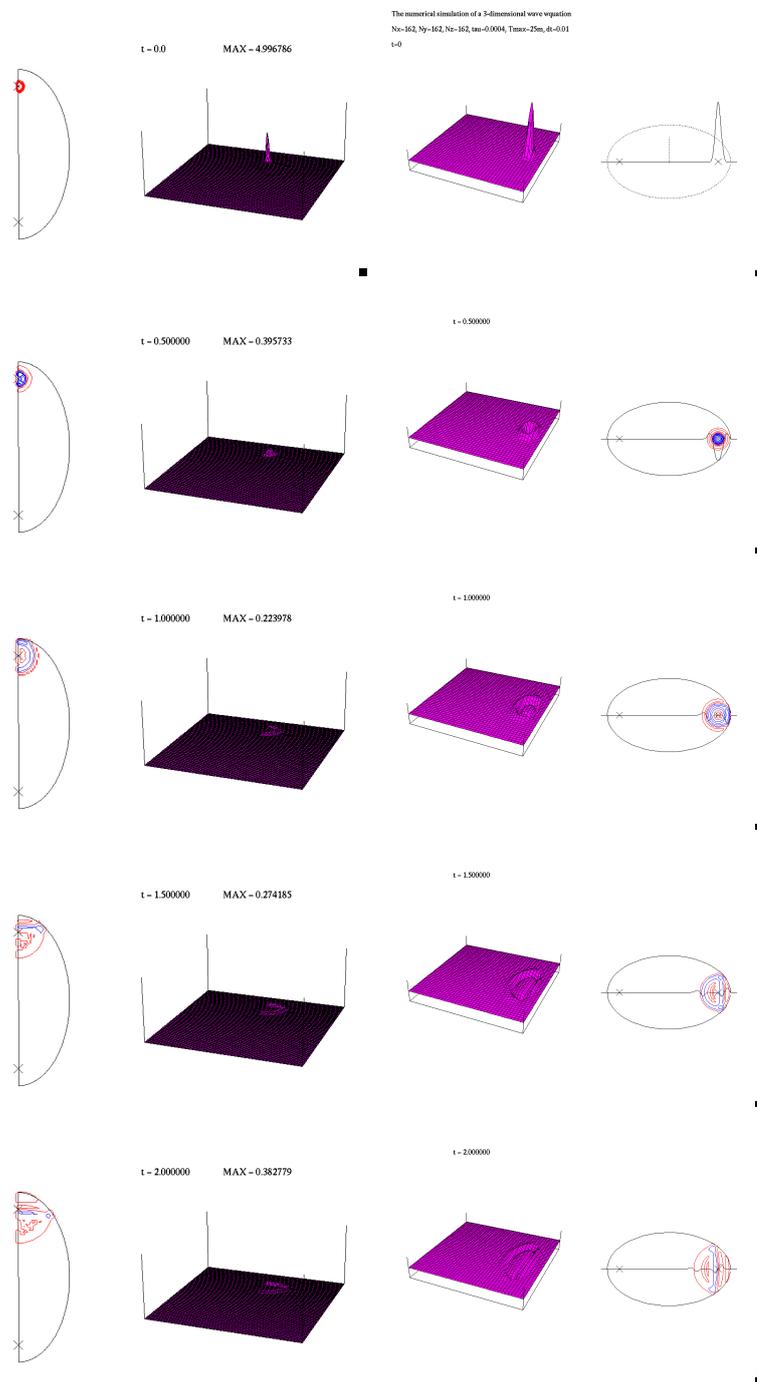
以下に考察を述べる。

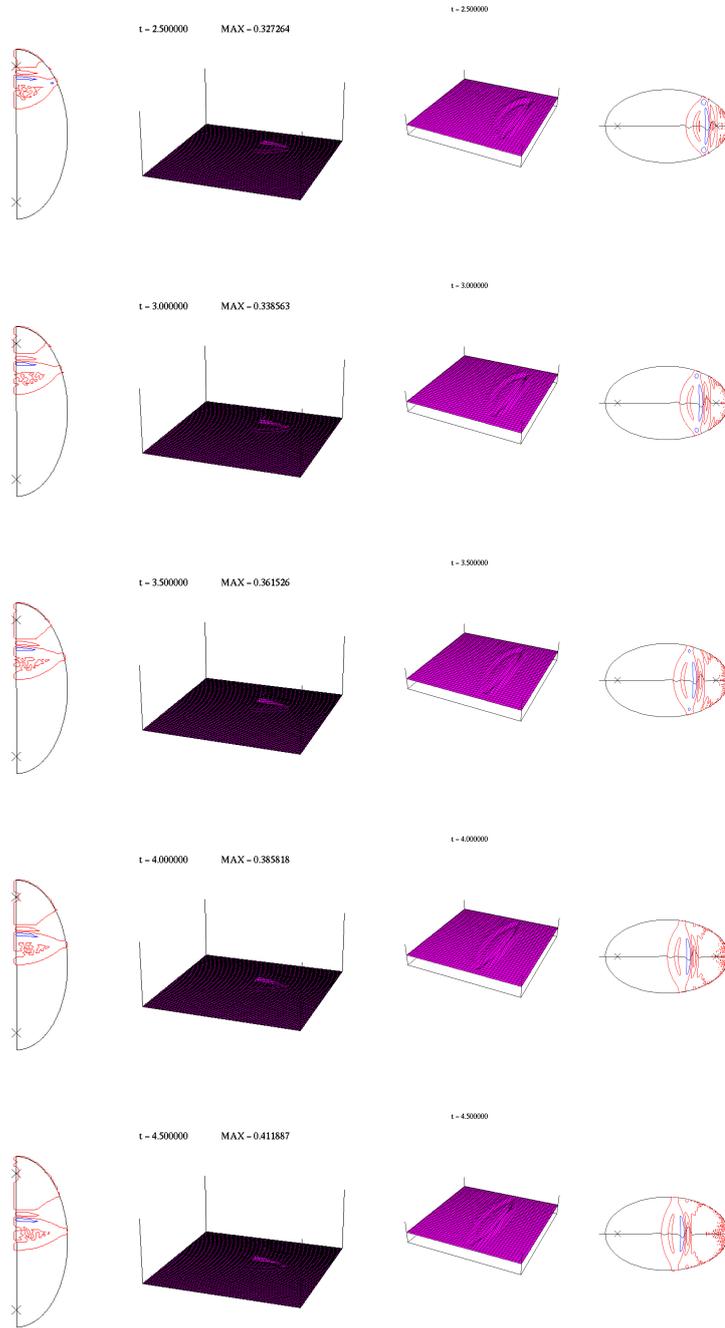
- (1),(2),(3) から、 $t = 10$ で反対の焦点に反射していく様子が見て取れた。また $t = 22.5$ 付近で元の焦点に戻る様子もわかった。これらの挙動にほぼ違いがないことから、ある程度精度の良い計算が行えていると言える。
- (5) は左の等高線図の緑の長方形の範囲を拡大して右の鳥瞰図にプロットしている。 $t = 10$ 付近で焦点に小さい山ができていく様子が検証できた。
- $t = 20$ で元に戻らないことから、1次元の計算ではないため解は t に関して周期関数ではないことがわかった。
- 3次元の計算結果との比較からも、これらの結果は過去の卒業研究と同じような結果になっていることがわかる。

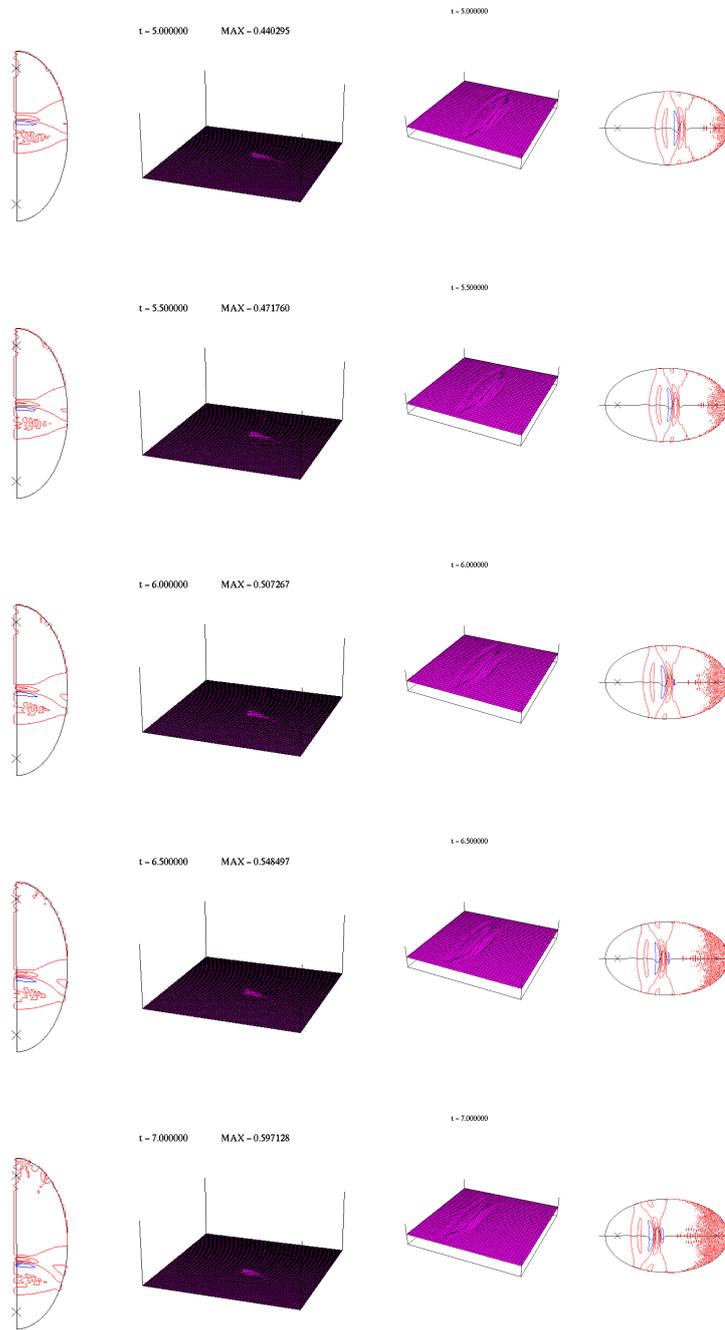
- (4) は $a = b$ のため球領域での計算結果となっている。 $a = b = 2$ では中心から反射して中心に戻るのに $t = 4$ と考えられるが、 $t = 4, 8, 12$ で中心に山ができてる様子が見て取れた。

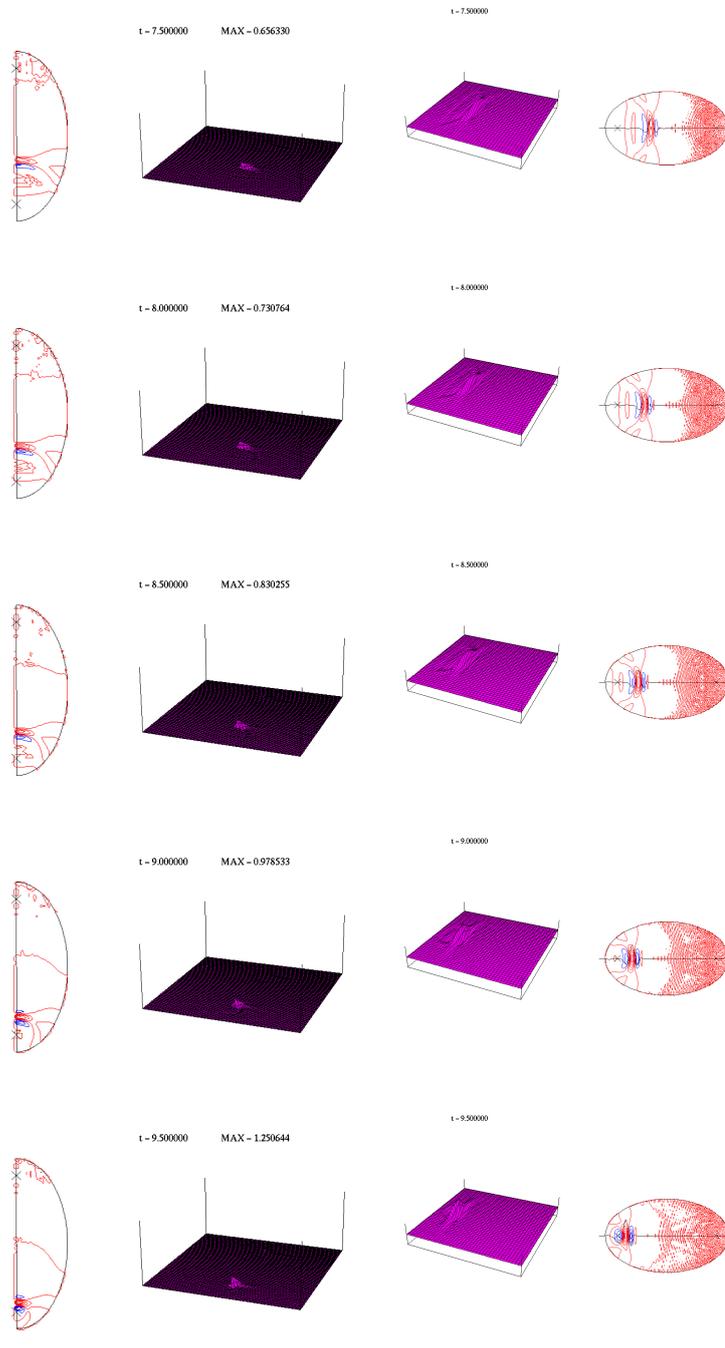
以下に (3),(4),(5),(6) の結果を載せる。(3) に関しては、比較のために 3次元の計算結果も載せてある。

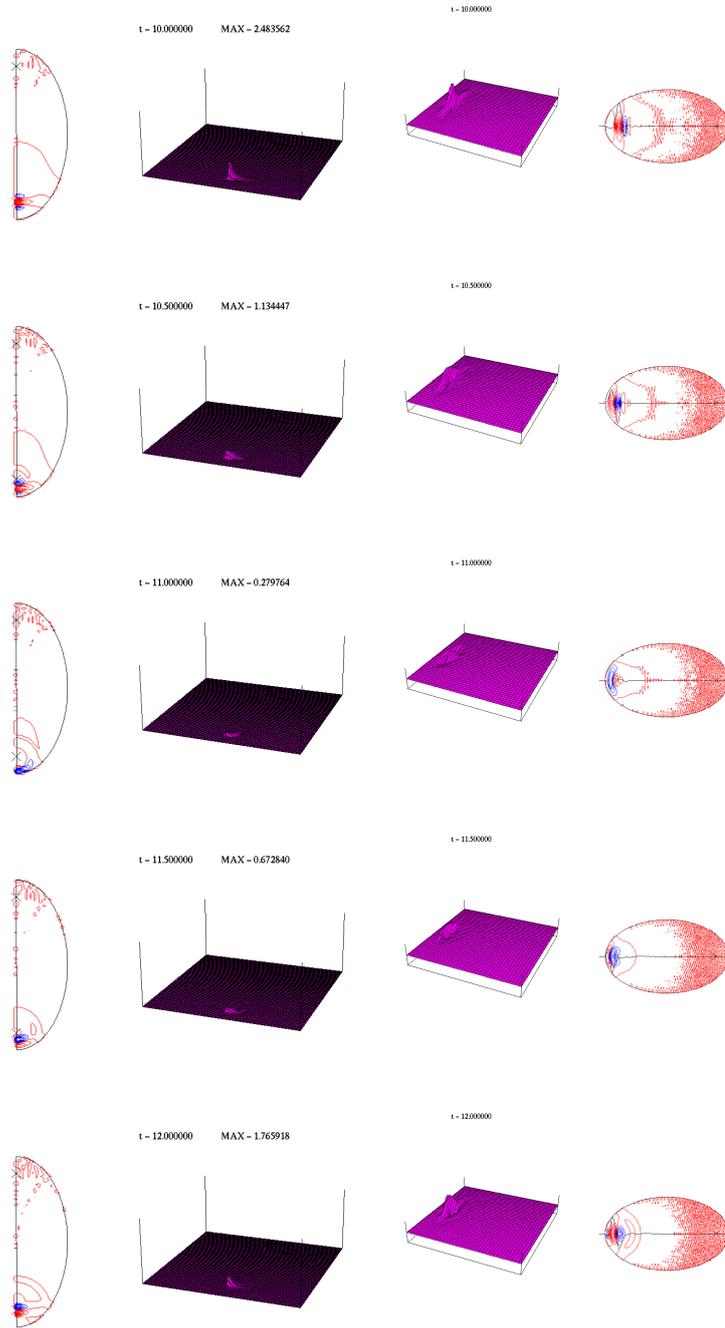
(3) $N_r = N_z = 2000, \tau = 0.00025$ と 3次元プロット

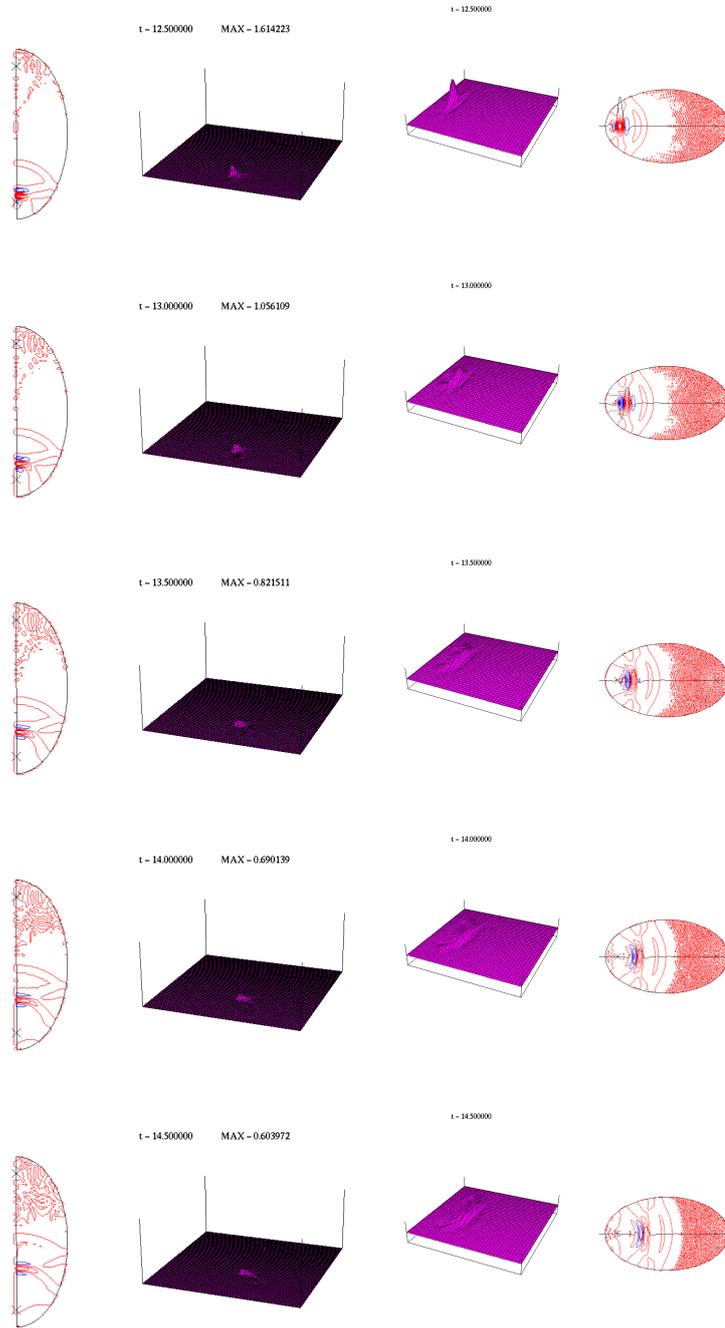


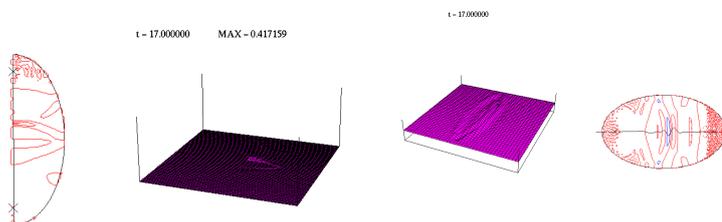
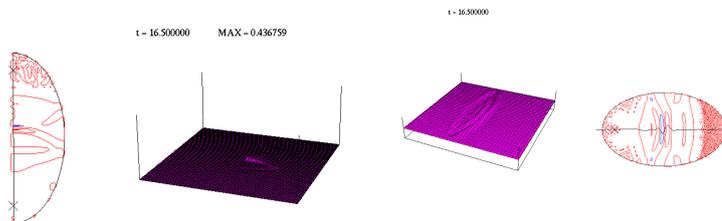
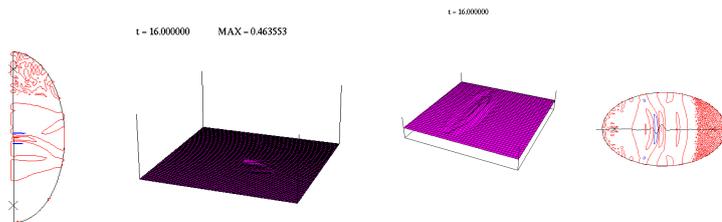
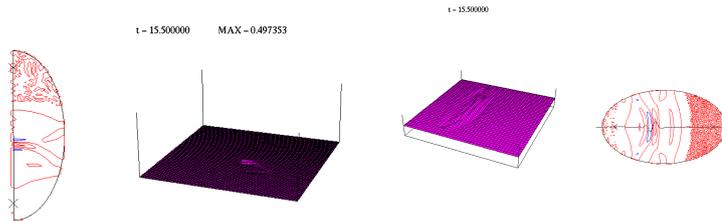
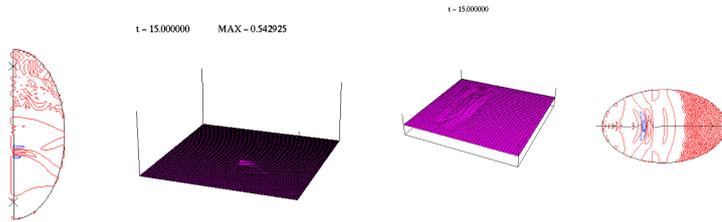


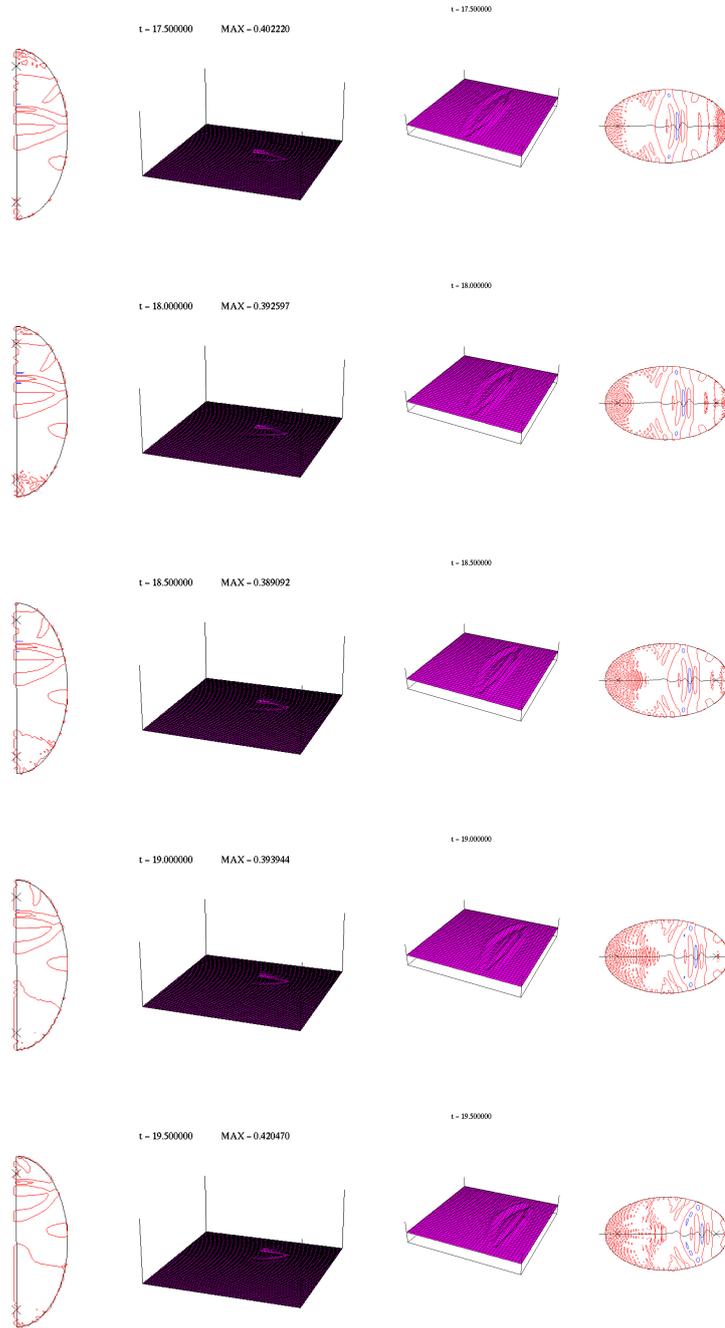


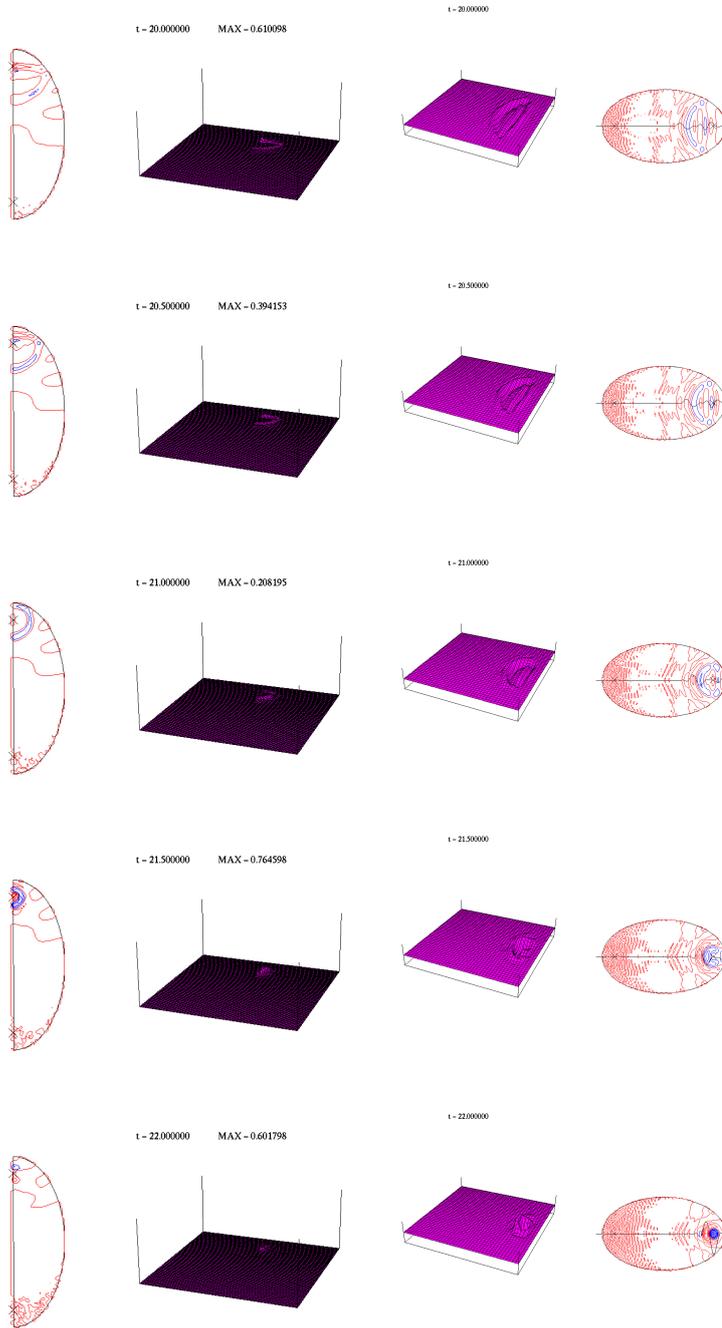


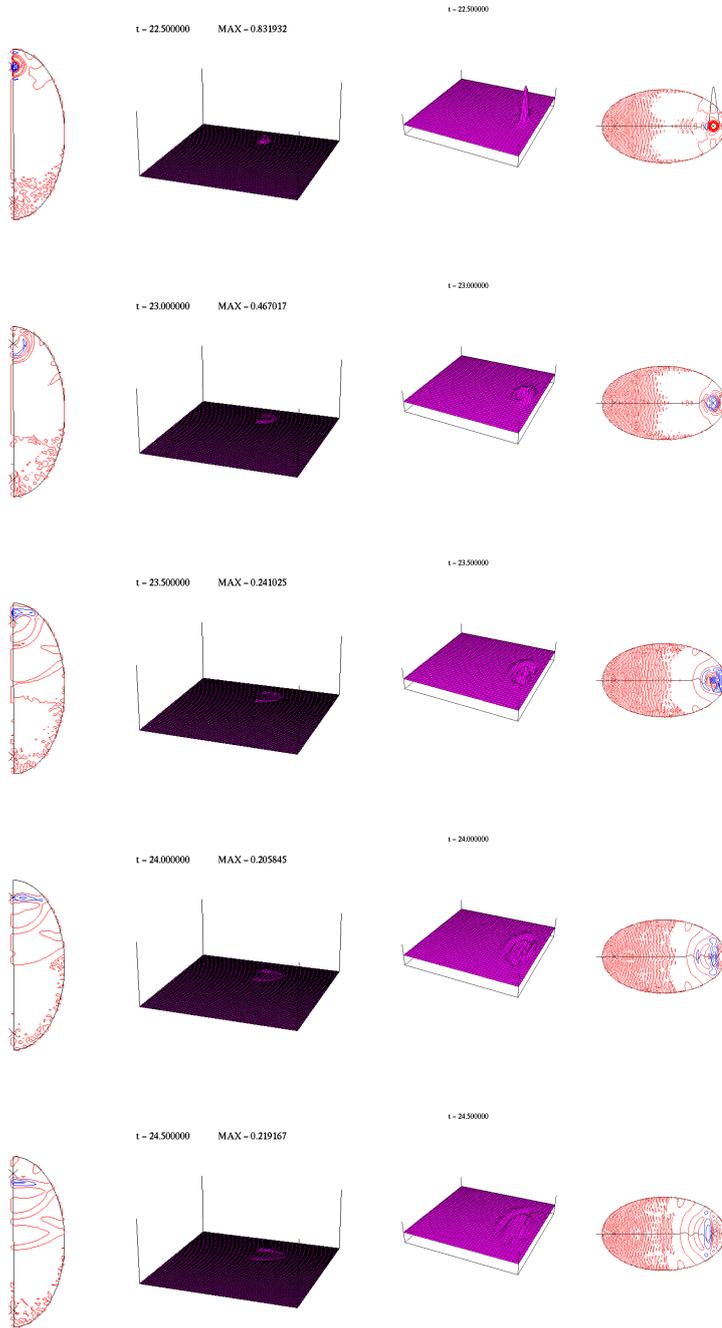




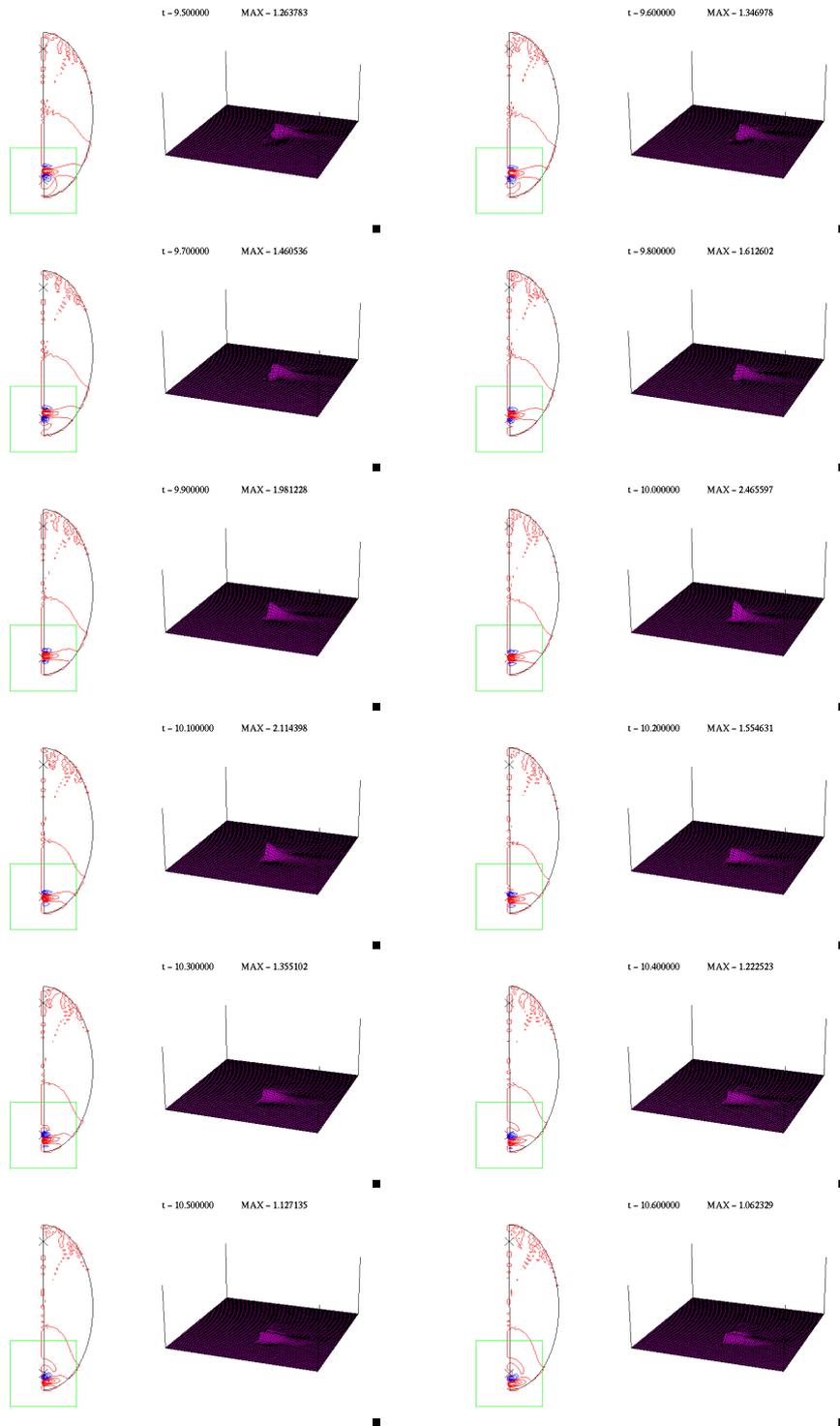




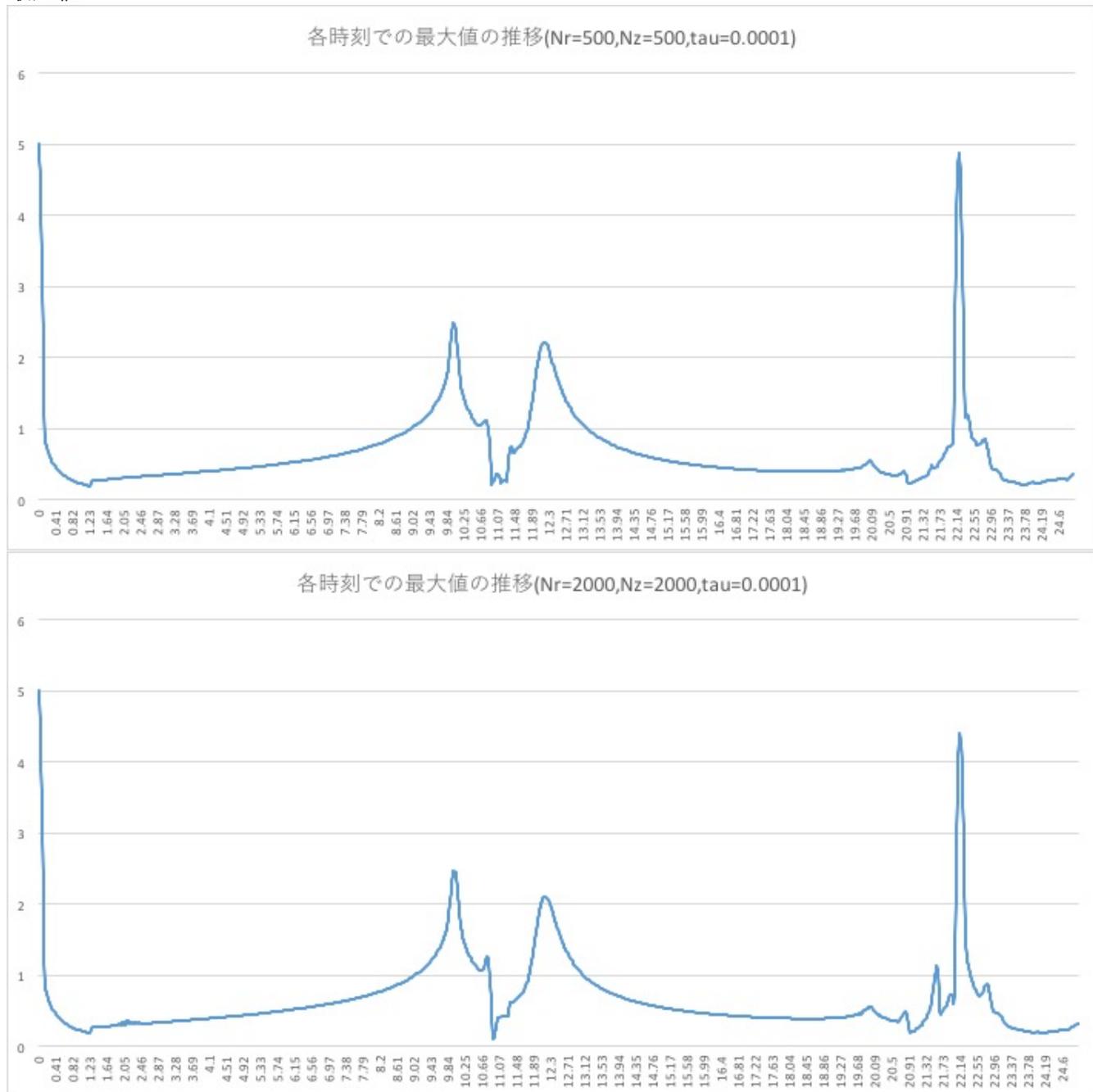




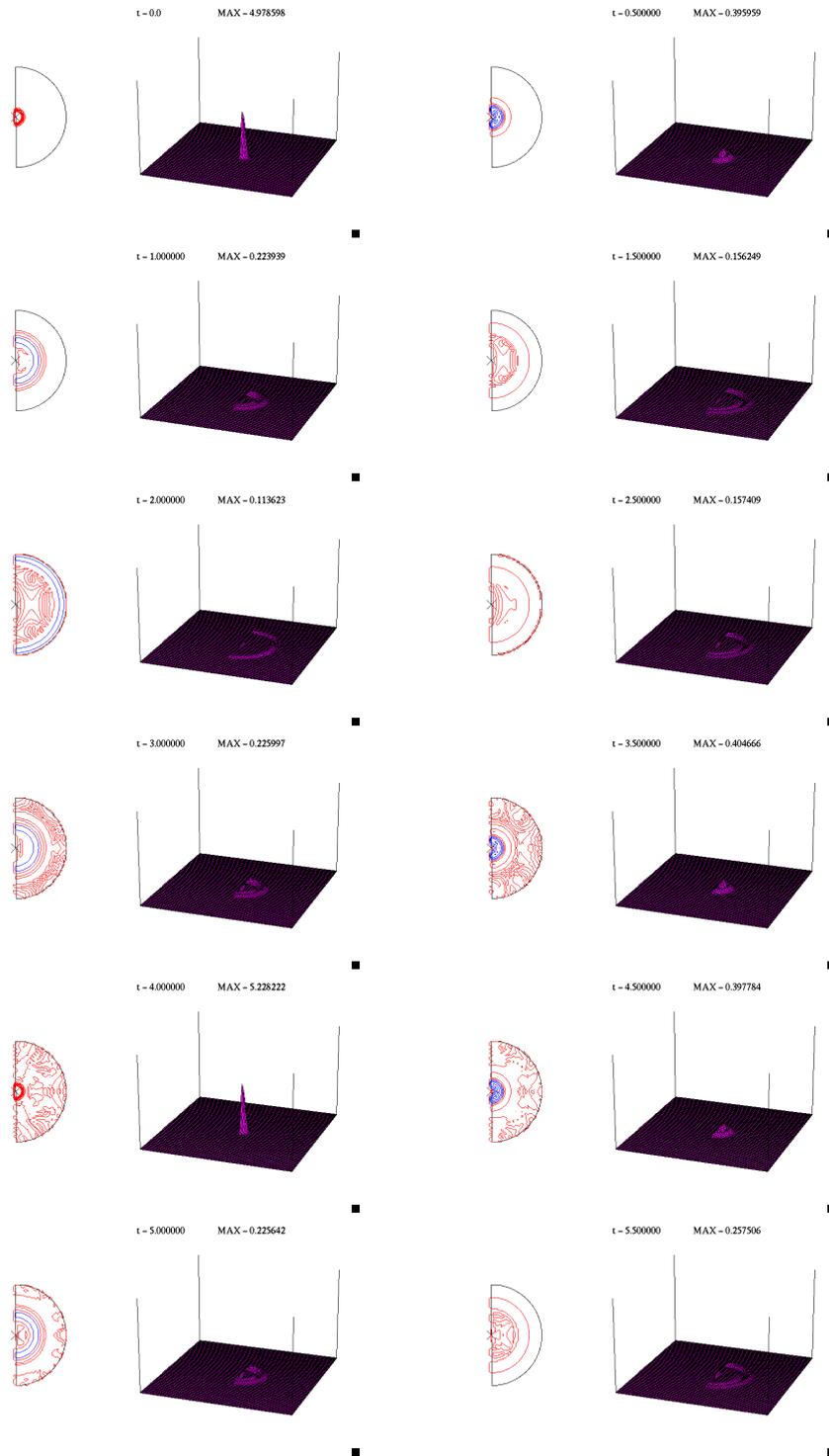
(5) $N_r = N_z = 500, \tau = 0.001$ で $t = 10$ 前後の焦点付近

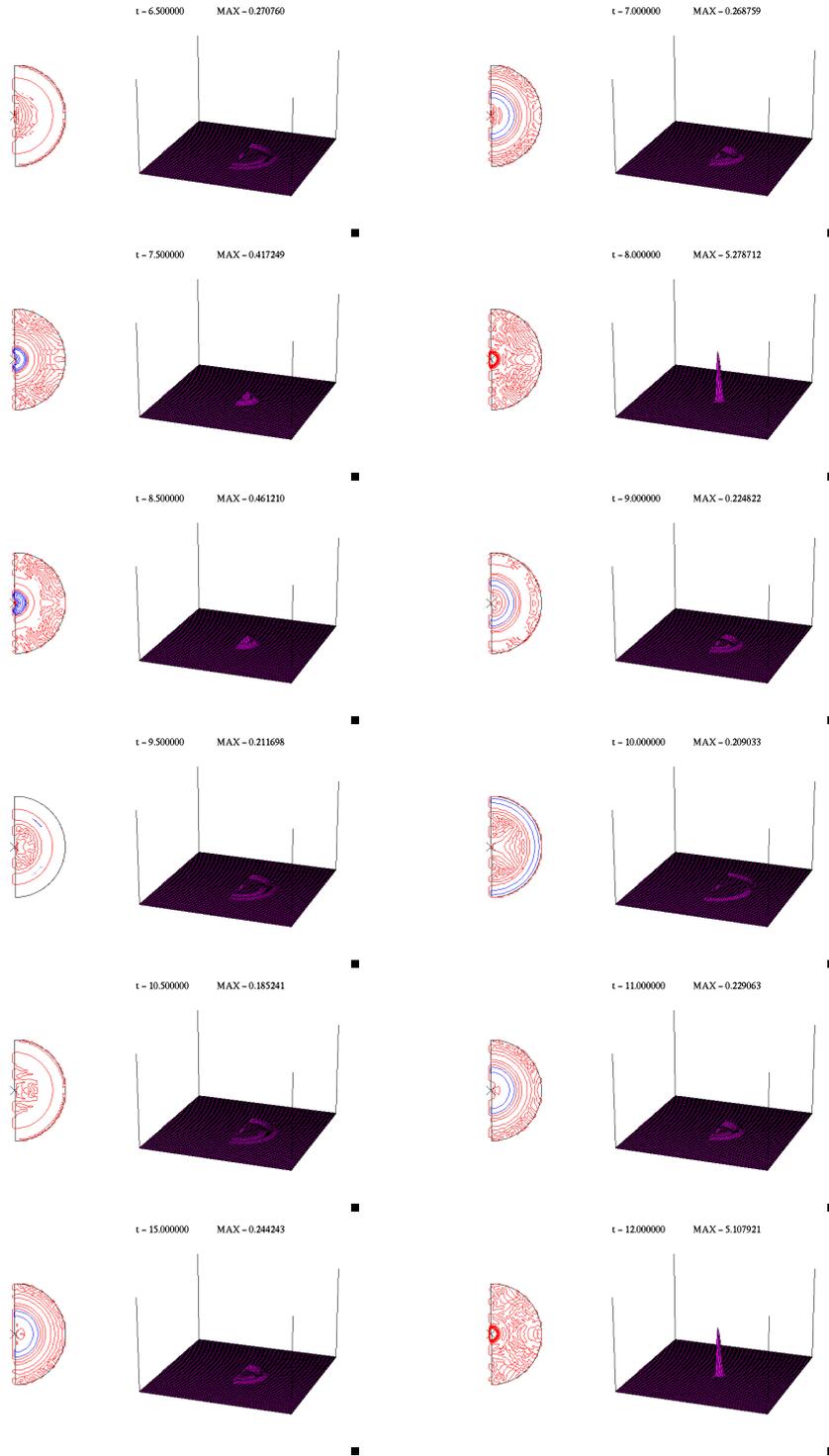


(6) $N_r = N_z = 500, \tau = 0.001$ と $N_r = N_z = 2000, \tau = 0.00025$ の場合の各時刻での
 最大値



参考 (4) $a = b = 2, N_r = N_z = 500, \tau = 0.002$





A 付録

A.1 hadouDaentai.c

```
/*
hadouDaentai.c --- 3次元楕円体領域の波動方程式を変数変換して2次元の計
算でSW近似を用いて解く
*/

#include <stdio.h>
#include <math.h>
#define M (10)
#define K (5)
double a, b, c, rmin, rmax, zmin, zmax, distance, r0, z00;

/* to use matrix, new_matrix() */
#include <matrix.h>

/* to use GLSC */
#define G_DOUBLE
#include <glsc.h>
/* #include <glsc2.h> */

/* 2乗の関数 */
double d(double r)
{
    return r * r;
}

/* 領域の内部・境界・外部の判定用 */
double F(double r, double z)
{
    return d(a) * d(b) - (d(b) * d(r) + d(a) * d(z));
}

/* 東側の境界 */
```

```

double E(double r, double z)
{
    return a * sqrt(1 - d(z) / d(b));
}

/* 西側の境界 */
double W(double r, double z)
{
    return z;
}

/* 北側の境界 */
double N(double r, double z)
{
    return b * sqrt(1 - d(r) / d(a));
}

/* 南側の境界 */
double S(double r, double z)
{
    return -b * sqrt(1 - d(r) / d(a));
}

/* 初期条件 */
double f(double p){
    if(p<1){
        return pow(d(p)-1, 4.0);
    }
    else
        return 0;
}

double g(double x, double y){
    return f(sqrt(d(x)+d(y))/0.5);
}

double phi(double r, double z, int nfunc){
    double H;

```

```

if(nfunc==1){
    if (a > b) {
        H = K * exp(-M * (d(r - r0) + d(z)));
    } else if (b > a) {
        H = K * exp(-M * (d(r) + d(z - z00)));
    } else {
        H = K * exp(-M * (d(r) + d(z)));
    }
    if (H - 0.1 >= 0)
        return H - 0.1;
    else
        return 0;
}
if(nfunc==2){
    return K*g(r ,z-z00);
}
else
return 0;
}

double psi(double r, double z){
    return 0;
}

/* 大小比較の関数 */
double min(double r, double z)
{
    if (r < z) {
        return r;
    } else
    return z;
}
double max(double r, double z)
{
    if (r < z) {
        return z;
    }
}

```

```

    } else
    return r;
}

int main()
{
    int Nr, Nz, Nrd, Nzd, spacez, spacer, i, j, n, skip, nMax;
    int nfunc;
    double hr, hz, tau, Tmax, t, dt, ew, es, ee, en, er, mine, maxtau;
    double maxu2, maxu3;
    double r, z, hw, he, hn, hs;
    double ri, riw, rie, zj, zjs, zjn;
    double uw, ue, un, us;
    matrix u1, u2, u3, ud;
    double W_MARGIN, H_MARGIN, WIDTH, HEIGHT;
    int k;
    char str[100];
    double sheta;
    double pi;
    pi=4*atan(1.0);
    /* 波の速さ */
    c = 1.0;
    /* 楕円の長軸、短軸 */
    a = 3.0;
    b = 5.0;
    /* 領域の r 軸、z 軸の長さ */
    rmin = -6;
    rmax =6;
    zmin = -6;
    zmax =6;
    /* 分割数 */
    printf("Nr= ");
    scanf("%d", &Nr);
    printf("Nz= ");
    scanf("%d", &Nz);
    /* 初期条件のピークの位置 */

```

```

r0 = sqrt(d(a) - d(b));
z00 = sqrt(d(b) - d(a));
/* 距離 */
distance = 60;
/* 格子間の長さ */
hr = (rmax - rmin) / Nr;
hz = (zmax - zmin) / Nz;
printf("hr=%g, hz=%g\n", hr, hz);
/* 誤差 */
er = 1.0e-13;
mine = 1.0;
maxtau = 1.0;
/*最大値計算の準備*/
maxu2=0.0;
maxu3=0.0;
/*****最小εと最大τを求める *****/
for (i = 0; i <= Nr; i++) {
    ri = rmin + i * hr;
    riw = rmin + (i - 1) * hr;
    rie = rmin + (i + 1) * hr;
    for (j = 0; j <= Nz; j++) {
        zj = zmin + j * hz;
        zjs = zmin + (j - 1) * hz;
        zjn = zmin + (j + 1) * hz;
        if (F(ri, zj) >= er && ri >= 0) {
            /* WEST */
            if (riw < 0) {
                hw = ri - W(ri, zj);
                ew = hw / hr;
            }
            hw = hr;
            ew = 1.0;
            /* SOUTH */
            if (F(ri, zjs) >= er) {
                hs = hz;
                es = 1.0;
            }
        }
    }
}

```

```

} else if (fabs(F(ri, zjs)) < er) {
    hs = hz;
    es = 1.0;
} else {
    hs = zj - S(ri, zj);
    es = hs / hz;
}
/* EAST */
if (F(rie, zj) >= er) {
    he = hr;
    ee = 1.0;
} else if (fabs(F(rie, zj)) < er) {
    ee = 1.0;
    he = hr;
} else {
    he = E(ri, zj) - ri;
    ee = he / hr;
}
/* NORTH */
if (F(ri, zjn) >= er) {
    en = 1.0;
    hn = hz;
} else if (fabs(F(ri, zjn)) < er) {
    en = 1.0;
    hn = hz;
} else {
    hn = N(ri, zj) - zj;
    en = hn / hz;
}
} else {
    ew = ee = en = es = 1.0;
    hw = he = hr;
    hn = hs = hz;
}
mine = min(mine, min(min(ew, ee), min(en, es)));
maxtau = min(maxtau,

```

```

        sqrt((ew * es * ee * en * d(hr) * d(hz)) /
            (d(hr) * ew * ee + d(hz) * es * en)));
    }
}
printf("εの最小値= %g\n", mine);
printf("τの最大値=%g\n", maxtau);
if (mine < er) {
    printf("mine=%g: ***** εが小さすぎます*****\n", mine);
}
if ((u1 = new_matrix(Nr + 1, Nz + 1)) == NULL) {
    fprintf(stderr, "配列 u1 を確保できませんでした。");
    exit(1);
}
if ((u2 = new_matrix(Nr + 1, Nz + 1)) == NULL) {
    fprintf(stderr, "配列 u2 を確保できませんでした。");
    exit(1);
}
if ((u3 = new_matrix(Nr + 1, Nz + 1)) == NULL) {
    fprintf(stderr, "配列 u3 を確保できませんでした。");
    exit(1);
}
//間引く鳥瞰図の配列
Nrd=100;
Nzd=100;
if ((ud = new_matrix(Nrd + 1, Nzd + 1)) == NULL) {
    fprintf(stderr, "配列 u3 を確保できませんでした。");
    exit(1);
}
printf("Tmax= ");
scanf("%lf", &Tmax);
printf("τ (≦ %g )== ", maxtau);
scanf("%lf", &tau);
printf("Δ t= ");
scanf("%lf", &dt);
printf("nfunc: ");
scanf("%d", &nfunc);

```

```

skip = rint(dt / tau);
if (skip == 0) {
    printf("Δ t が小さすぎるので、Δ t= τ とします。 \n");
    skip = 1;
    dt = skip * tau;
}

/* 初期値の入力 */
t = 0.0;
for (i = 0; i <= Nr; i++) {
    ri = rmin + i * hr;
    for (j = 0; j <= Nz; j++) {
        zj = zmin + j * hz;
        u1[i][j] = phi(ri, zj, nfunc);
    }
}

/* u2 の計算 */
for (i = 0; i <= Nr; i++) {
    ri = rmin + i * hr;
    riw = rmin + (i - 1) * hr;
    rie = rmin + (i + 1) * hr;
    for (j = 0; j <= Nz; j++) {
        zj = zmin + j * hz;
        zjn = zmin + (j + 1) * hz;
        zjs = zmin + (j - 1) * hz;
        if (F(ri, zj) >= er && ri >= 0) {
            /* WEST */
            if (riw < 0) {
                hw = ri - W(ri, zj);
                uw = 0;
            } else {
                hw = hr;
                uw = u1[i - 1][j];
            }
            /* EAST */
            if (F(rie, zj) >= er) {

```

```

        he = hr;
        ue = u1[i + 1][j];
    } else if (fabs(F(rie, zj)) < er) {
        he = hr;
        ue = u1[i + 1][j];
    } else {
        he = E(ri, zj) - ri;
        ue = 0;
    }
    /* NORTH */
    if (F(ri, zjn) >= er) {
        hn = hz;
        un = u1[i][j + 1];
    } else if (fabs(F(ri, zjn)) < er) {
        hn = hz;
        un = u1[i][j + 1];
    } else {
        hn = N(ri, zj) - zj;
        un = 0;
    }
    /* SOUTH */
    if (F(ri, zjs) >= er) {
        hs = hz;
        us = u1[i][j - 1];
    } else if (fabs(F(ri, zjs)) < er) {
        hs = hz;
        us = u1[i][j - 1];
    } else {
        hs = zj - S(ri, zj);
        us = 0;
    }
    if (ri == 0) {
        u2[i][j] = u1[i][j] + tau * psi(ri, zj)
            + 2 * d(tau) * d(c) * (uw - u1[i][j]) / d(hr)
            + d(tau) * d(c) * ((ue - u1[i][j]) / hn -
                (u1[i][j] - us) / hs) / (hn +

```

```

hs);
    } else {
        u2[i][j] = u1[i][j] + tau * psi(ri, zj)
            + d(tau) * d(c) * ((ue - uw) / ((hw + he) * 2 * ri)
                + ((ue - u1[i][j]) / he -
                    (u1[i][j] - uw) / hw) / (he +
                    hw)
                + ((un - u1[i][j]) / hn -
                    (u1[i][j] - us) / hs) / (hn +
                    hs));
    }
} else {
    u1[i][j] = 0;
    u2[i][j] = 0;
    hw = he = hr;
    hn = hs = hz;
    uw = ue = un = us = 0;
}
//最大値の計算
maxu2 = max(maxu2, u2[i][j]);
}
}
printf("Max= %g \n", maxu2);
//表示を間引く
spacer=Nr/Nrd;
spacez=Nz/Nzd;
for(i = 0; i <= Nrd; i++){
    for(j = 0; j <= Nzd; j++){
        ud[i][j] = u1[i * spacer][j * spacez];
    }
}
W_MARGIN = 10.0;
H_MARGIN = 10.0;
WIDTH = 100.0;
HEIGHT = 100.0;
g_init("Meta", 2 * WIDTH + 3 * W_MARGIN, HEIGHT + 2 * H_MARGIN);

```

```

g_device(G_BOTH);
g_capture_set("");
g_def_scale(0, rmin, rmax, zmin, zmax, W_MARGIN, H_MARGIN, WIDTH,
            HEIGHT);
g_def_line(0, G_BLACK, 0, G_LINE_SOLID);
g_sel_scale(0);
sprintf(str, "MAX = %lf", maxu2);
g_text(WIDTH + 6 * W_MARGIN, H_MARGIN, str);
g_text(WIDTH + 2 * W_MARGIN, H_MARGIN, "t = 0.0");
g_move(a, 0);
for (k = 0; k < 158; k++) {
    sheta = k * 0.01;
    g_plot(a * cos(sheta), b * sin(sheta));
}
for (k = 472; k < 629; k++) {
    sheta = k * 0.01;
    g_plot(a * cos(sheta), b * sin(sheta));
}
g_marker_type ( -4 );
g_marker( 0, z00 );
g_marker( 0, -z00 );
g_line_color(G_BLUE);
for (k = -10; k <= 0; k++) {
    g_contln(rmin, rmax, zmin, zmax, ud[0], Nrd + 1, Nzd + 1, k * 0.2);
}
g_line_color(G_RED);
for (k = 0; k <= 10; k++) {
    g_contln(rmin, rmax, zmin, zmax, ud[0], Nrd + 1, Nzd + 1, k * 0.2);
}
g_hidden(rmax - rmin, zmax - zmin, 5.0, 0.0, 10.0,
         distance, 20.0, 20.0,
         WIDTH + 2 * W_MARGIN, H_MARGIN, WIDTH, HEIGHT,
         ud[0], Nrd + 1, Nzd + 1, 1, G_SIDE_NONE, 1, 1);
g_capture();
g_sleep(1.0);
nMax = rint(Tmax / tau);

```

```

for (n = 2; n <= nMax; n++) {
    maxu3=0;
    /*****領域*****/
    for (i = 0; i <= Nr; i++) {
        ri = rmin + i * hr;
        riw = rmin + (i - 1) * hr;
        rie = rmin + (i + 1) * hr;
        for (j = 0; j <= Nz; j++) {
            zj = zmin + j * hz;
            zjn = zmin + (j + 1) * hz;
            zjs = zmin + (j - 1) * hz;
            if (F(ri, zj) >= er && ri >= 0) {
                /* WEST */
                if (riw < 0) {
                    hw = ri - W(ri, zj);
                    uw = 0;
                } else {
                    hw = hr;
                    uw = u2[i - 1][j];
                }
                /* EAST */
                if (F(rie, zj) >= er) {
                    he = hr;
                    ue = u2[i + 1][j];
                } else if (fabs(F(rie, zj)) < er) {
                    he = hr;
                    ue = u2[i + 1][j];
                } else {
                    he = E(ri, zj) - ri;
                    ue = 0;
                }
                /* NORTH */
                if (F(ri, zjn) >= er) {
                    hn = hz;
                    un = u2[i][j + 1];
                } else if (fabs(F(ri, zjn)) < er) {

```

```

        hn = hz;
        un = u2[i][j + 1];
    } else {
        hn = N(ri, zj) - zj;
        un = 0;
    }
    /* SOUTH */
    if (F(ri, zjs) >= er) {
        hs = hz;
        us = u2[i][j - 1];
    } else if (fabs(F(ri, zjs)) < er) {
        hs = hz;
        us = u2[i][j - 1];
    } else {
        hs = zj - S(ri, zj);
        us = 0;
    }
    if (ri == 0) {
        u3[i][j] = 2.0 * u2[i][j] - u1[i][j]
            + 4.0 * d(tau) * d(c) * (ue - u2[i][j]) / d(hr)
            + 2.0 * d(tau) * d(c) * ((un - u2[i][j]) / hn -
                (u2[i][j] -
                    us) / hs) / (hn +
                    hs);
    } else {
        u3[i][j] = 2.0 * u2[i][j] - u1[i][j]
            + d(tau) * d(c) * (ue - uw) / (ri * (hw + he))
            + 2.0 * d(tau) * d(c) * ((ue - u2[i][j]) / he -
                (u2[i][j] -
                    uw) / hw) / (he + hw)
            + 2.0 * d(tau) * d(c) * ((un - u2[i][j]) / hn -
                (u2[i][j] -
                    us) / hs) / (hn +
                    hs);
    }
} else {

```

```

        u1[i][j] = 0;
        u2[i][j] = 0;
        u3[i][j] = 0;
        hw = he = hr;
        hn = hs = hz;
        uw = ue = un = us = 0;
    }
    //最大値の計算
    maxu3 = max(maxu3,u3[i][j]);
}
}
for (i = 0; i <= Nr; i++) {
    for (j = 0; j <= Nz; j++) {
        u1[i][j] = u2[i][j];
        u2[i][j] = u3[i][j];
    }
}
//表示を間引く
for(i = 0; i <= Nrd; i++){
    for(j = 0; j <= Nzd; j++){
        ud[i][j] = u2[i * spacer][j * spacez];
    }
}
t = tau * n;
if (n % skip == 0) {
    printf("Max= %g \n", maxu3);
    g_cls();
    g_line_color(G_BLACK);
    g_move(a, 0);
    for (k = 0; k < 158; k++) {
        sheta = k * 0.01;
        g_plot(a * cos(sheta), b * sin(sheta));
    }
    for (k = 472; k < 629; k++) {
        sheta = k * 0.01;
        g_plot(a * cos(sheta), b * sin(sheta));
    }
}

```

```

    }
    g_marker_type ( -4 );
    g_marker( 0, z00 );
    g_marker( 0, -z00 );
    sprintf(str, "MAX = %lf", maxu3);
    g_text(WIDTH + 6 * W_MARGIN, H_MARGIN, str);
    sprintf(str, "t = %lf", t);
    g_text(WIDTH + 2 * W_MARGIN, H_MARGIN, str);
    g_line_color(G_BLUE);
    for (k = -10; k <= 0; k++) {
        g_contln(rmin, rmax, zmin, zmax, ud[0], Nrd + 1, Nzd + 1,
                k * 0.2);
    }
    g_line_color(G_RED);
    for (k = 0; k <= 10; k++) {
        g_contln(rmin, rmax, zmin, zmax, ud[0], Nrd + 1, Nzd + 1,
                k * 0.2);
    }
    g_hidden(rmax - rmin, zmax - zmin, 5.0, 0.0, 10.0,
            distance, 20.0, 20.0,
            WIDTH + 2 * W_MARGIN, H_MARGIN, WIDTH, HEIGHT,
            ud[0], Nrd + 1, Nzd + 1, 1, G_SIDE_NONE, 1, 1);
    g_capture();
}
}

/* マウスでクリックされるのを待つ */
g_sleep(-1.0);

/* ウィンドウを消す */
g_term();
return 0;
}

```

参考文献

- [1] 浜 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」2011年度桂田研卒業研究レポート
<http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/2011-hama.pdf>
- [2] 嵯峨野 美希, 「楕円体の酒場」2013年度桂田研卒業研究レポート
<http://nalab.mind.meiji.ac.jp/~mk/labo/report/pdf/2013-sagano.pdf>
- [3] 桂田 祐史, 「波動方程式の差分法」
<http://nalab.mind.meiji.ac.jp/~mk/labo/text/wave.pdf>
- [4] *wikipedia*, *Courant-Friedrichs-Lewy condition* (2019年2月21日)
http://en.wikipedia.org/wiki/Courant%E2%80%93Friedrichs%E2%80%93Lewy_condition