

複屈折現象解説ソフトの作成

長谷川 能三

大阪市立科学館 学芸課

概要

2004年秋に偏光をテーマとしたサイエンスショー「見えたり、見えなくなったり」を行なった。その中で、偏光板にセロハンテープや荷造り用透明テープを貼り、もう1枚の偏光板を通して見ると、鮮やかな色がついて見えるという実験を行なった。この現象はテープの複屈折性による現象であるために、テープを偏光板の軸に平行、もしくは垂直に貼ると色づいては見えない。これまで、友の会会報誌「月刊うちゅう」でこの現象の解説を行なったこともあるが、動きのない図では理解が難しかった。

そこで、動きをつけられるコンピュータソフトを作成し、これを用いて友の会の例会で複屈折についての解説も行なったので、複屈折現象、および、その解説ソフトの内容等についてここで報告する。

1. 複屈折

そもそも光が屈折する、つまり光の速さが物質内で遅くなるのは、光(電磁波)の振動する電場によって物質が分極を起こすからであり、分極のしやすさを示す誘電率によって物質の屈折率が決まる。このため、等方的な物質では屈折率は電場の向きによらず一定であるが、異方性のある物質では電場の向きによって屈折率が異なる。

異方性のある物質としては、単位格子が立方格子などではない結晶や、分子の向きがある程度そろった高分子など、さまざまなものがある。

セロハンテープの場合、テープの長さの方向と幅の方向で誘電率が異なるため、電場の振動の方向、つまり偏光の向きによって屈折率が異なる。しかし、今の場合、セロハンテープの厚み方向の誘電率がどうなっているかは考える必要がなく、また、セロハンテープの長さの方向と幅の方向が互いに垂直であるために、電場をこの2つの方向の成分に分けて考えるだけでよい。

2. ソフトの内容

このソフトでは、偏光板を固定して、いろいろな向きにセロハンテープを貼るのではなく、セロハンテープの方向を固定し、このセロハンテープの表と裏に、それぞれ任意の方向の偏光板を重ねるとい

う形をとった。

このソフトを起動すると、中段左に表側(入射側)に重ねた偏光板を表わす絵、中段右に裏側に重ねた偏光板を表わす絵が表示される。最初はどちらの偏光板も斜め45度にセットされているが、上下の矢印キーで表側の偏光板を、左右の矢印キーで裏側の偏光板を回転させることができる。

ここで、入射される光は偏光板の向きでできる偏光面を持つ光に限られるとし、その電場の振動の左右方向(セロハンテープの幅の方向)成分を上段に、上下方向(セロハンテープの長さの方向)成分を下段に表示している。中央のグレーの部分はセロハンテープの厚みの中であり、この部分では、屈折率に従って、波長が変えてある。

セロハンテープの厚みより左側では、電場の振動の左右方向の成分と上下方向の成分が同位相である(偏光板の向きによって、表示上逆位相の場合もある)。しかし、テープの厚み内では、左右方向の振動の波長と、上下方向の振動の波長が異なるため、位相にずれが生じる。このため、例えば起動した初期状態では、セロハンテープの表側では右上と左下の間を振動していた電場が、セロハンテープの裏側では左上と右下の間で振動するようになっており、表側と同じ方向に向けた偏光板を通過できる電場の成分はほとんどないことがわか

る。

ところが、表側の偏光板の向きを水平または垂直にすると、そもそも電場が左右方向の成分だけ、もしくは上下方向の成分だけをもつため、位相のずれが生じず、セロハンテープの裏側に達しても電場の振動方向に変化がないことがわかる。

さらにこのソフトでは、数字キーによってセロハンテープの重ね枚数を指定したり、[R][Y][G][B]キーによって、光の波長を変えたり波長の異なる複数の光を同時に表示することができる。このため、セロハンテープを貼り重ねる枚数と偏光板の向きによって、透過する色が変化することがわかる。

また、[A]キーを押すと、裏側の偏光板の絵が消え、代わりにセロハンテープの裏側での電場の動きがそのまま表わされる。これにより、セロハンテープを通過した後の光は、単に偏光面が変化しているのではなく、楕円偏光や円偏光になることもわかる。

3. 最後に

このソフトは複屈折の解説に使用することを目的としているため、プログラム内で使用しているさまざまな数値(光の波長、テープの厚み、セロハンテープの屈折率など)は、見やすくなるような適当な数値を選んでおり、実際の光の波長やセロハンテープの厚みや屈折率などとは異なっている。

なお、このソフトはHSP言語によって記述しており、プログラムのソースは以下に掲載の通りである。また、<http://www.sci-museum.kita.osaka.jp/~nozo/> からダウンロードすることもできる。

しかし、私が現在使っているコンピュータではうまく動作しているが、他の環境でのテストを行っていないため、動作の保証はできない。特に、ディスプレイの解像度は、**1024×768**でうまく表示されるようにいろいろな部分のサイズを選んでいるので、解像度の低いディスプレイでは、一部しか表示されないかもしれないが、ご了承ください。

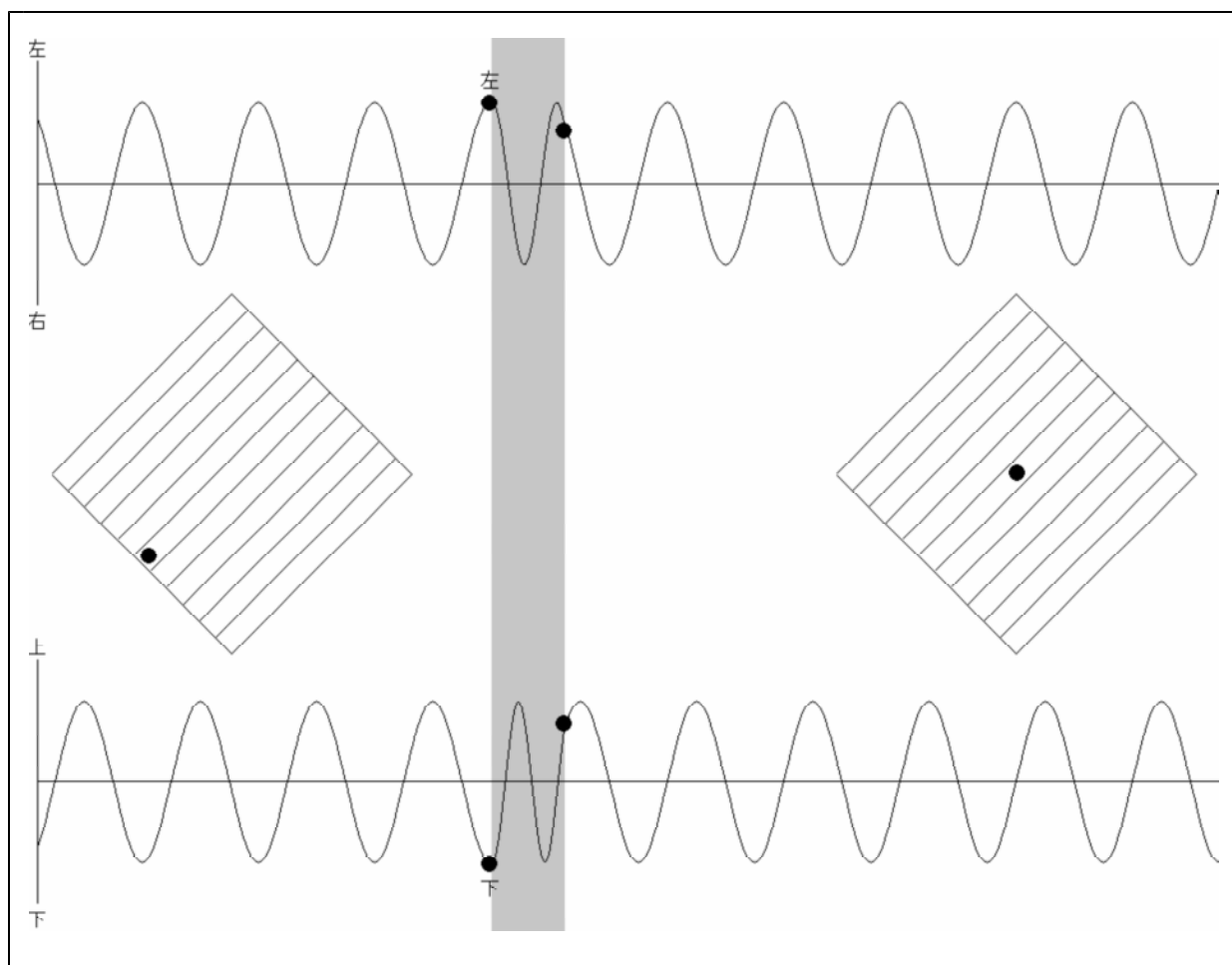


図1. 起動直後の画面

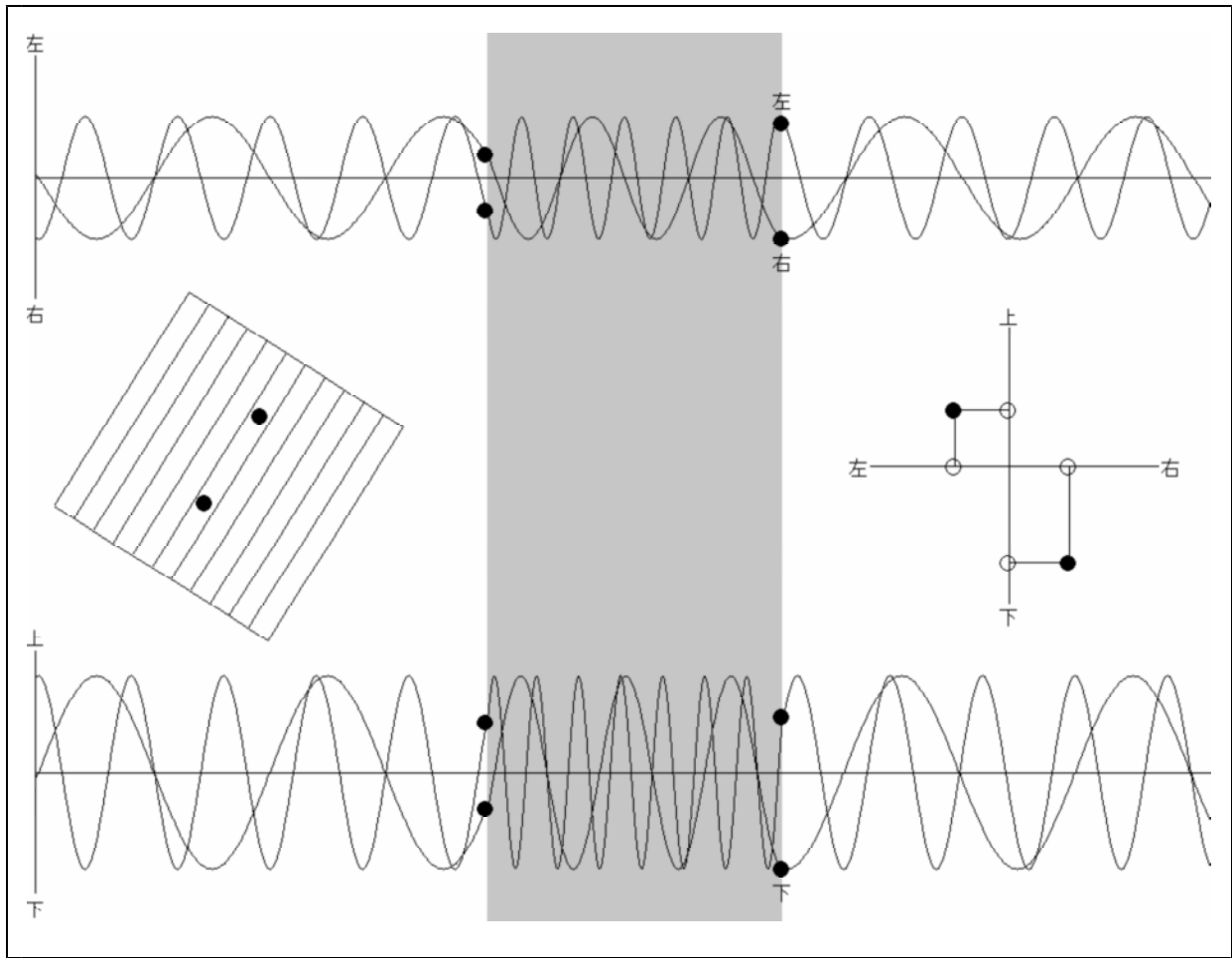


図2. 以下の操作を行なった後の画面

- ・ 表側の偏光板を回転
- ・ 長波長と短波長の2つの光を同時表示
- ・ セロハンテープの重ね枚数を2枚から4枚に変更
- ・ 裏側の偏光板の絵をなくし、電場の動きをそのまま表示

```

#include "hspexit.as"
#fbt 8
:-----
:初期設定
wx=dispx : wy=dispy           ; 画面サイズ
bgscr 2,wx,wy,1,0,0,wx,wy    ; ウィンドウ初期化
gsel 2,2
color 0,0,0 : boxf

*parameter
emath 8 : i=256              ; 数値計算精度8ビット
m = 180                      ; 屈折率(水平方向)
nv = 220                     ; 屈折率(垂直方向)
amp = 100                    ; 振幅
th = 128                     ; 膜厚
a1 = 32                      ; 入射側偏光板角度
a2 = 32                      ; 射出側偏光板角度

redraw 0
color 0,0,0 : boxf

rd=-1 : y|=1 : gr=1 : bl=-1   ; 初期状態は緑のみ表示
ax= 1 : pl= 1 : d= 1

repeat 100000
  j = cnt
  *pause
  getkey k,32 : if k=1 : goto *pause ; [space]でポーズ
  getkey k,82 : if k=1 : rd=-rd     ; [r]で赤の表示・非表示
  getkey k,89 : if k=1 : y|=y|     ; [y]で黄の表示・非表示
  getkey k,71 : if k=1 : gr=-gr    ; [g]で緑の表示・非表示
  getkey k,66 : if k=1 : bl=-bl    ; [b]で青の表示・非表示
  sp = (rd+yl+gr+bl+8)/6
  getkey k,48 : if k=1 : th=0

```

```

getkey k,49 : if k=1 : th=64
getkey k,50 : if k=1 : th=128
getkey k,51 : if k=1 : th=192
getkey k,52 : if k=1 : th=256
getkey k,53 : if k=1 : th=320
getkey k,54 : if k=1 : th=384
getkey k,55 : if k=1 : th=448
getkey k,56 : if k=1 : th=512
; [0]~[8]で膜厚調整

getkey k,38 : if k=1 : a1=a1-1
getkey k,40 : if k=1 : a1=a1+1
if a<0 : a1=a1+256
if a>256 : a1=a1-256
getkey k,72 : if k=1 : a1=64
getkey k,86 : if k=1 : a1=0
ems in sn,a1 : emocs cs,a1
ah=amp*sn/256 : av=amp*cs/256
; 水平成分・垂直成分の振幅

getkey k,37 : if k=1 : a2=a2-1
getkey k,39 : if k=1 : a2=a2+1
; [←][→]で射出側偏光板角度調整

getkey k,65 : if k=1 : ax=-ax : pl=-1 ; [a]で射出側座標表示/非表示
getkey k,70 : if (k=1)&(ax=1) : pl=-pl ; [f]で射出側偏光板表示/非表示

getkey k,68 : if k=1 : d=-d         ; [d]で射出側ドット表示/非表示

getkey k,69 : if k=1 : end         ; [e]で終了

color 64,64,64 : boxf 399,0,397+th,763 : boxf 399,525,397+th,768
; 膜の表示

if ax-pl>0 [
  color 255,255,255
  pos 730,375 : line 980,375
  pos 850,255 : line 850,495
  pos 730-8-10,375-8 : mes "左"
  pos 980-8+10,375-8 : mes "右"
  pos 850-8,255-8-10 : mes "上"
  pos 850-8,495-8+10 : mes "下"
]

if ax+pl<2 : x0=850 : y0=375 : a=a2 : gosub *filter
x0=175 : y0=375 : a=a1 : gosub *filter

```

```

if ax=p|>0 [
  pos yh-125+850-8,yv-640+375-8 : mes "●"
  pos 850-8,yv-640+375-8 : mes "○"
  pos yh-125+850-8,375-8 : mes "○"

  pos 850,yv-640+375 : line yh-125+850,yv-640+375
  pos yh-125+850,375 : line yh-125+850,yv-640+375
]

if ax+p|2 [
  emsin sn2,a2 : emcos cs2,a2
  dd=((yh-125)*sn2/256)+((yv-640)*cs2/256)
  pos (dd*sn2/256)+850-8,(dd*cs2/256)+375-8 : mes "●"
]

n=700-th : xs = 400+th : nn=100
p0h = ph
gosub *h_wave

p0v = pv
gosub *v_wave

return

;-----
*h_wave
pos xh,yh
repeat n
  i=cnt
  ph = i*nn*256/100/length+p0h
  emsin snh,ph
  xh = xs+i-1 : yh = -(snh*ah/256)+125
  line xh,yh
loop
return

;-----
*h_move

```

```

if rd=1 : length = 200 : color 255,0,0 : gosub *main
if y|=1 : length = 133 : color 255,255,0 : gosub *main
if gr|=1 : length = 100 : color 0,255,0 : gosub *main
if b|=1 : length = 80 : color 0,0,255 : gosub *main

color 255,255,255
pos 8,20 : line 8,230 : pos 8,535 : line 8,745
pos 8,125 : line 1024,125 : pos 8,640 : line 1024,640
pos 0,0 : mes "左" : pos 0,234 : mes "右"
pos 0,515 : mes "上" : pos 0,749 : mes "下"

redraw 1
redraw 0
color 0,0,0 : boxf
await 30

loop
await 1000
end

;-----
*main
n=381 : nn=100 : xs=9

xn = 8 : yh = 125 : p0h = -j*so*256/length
gosub *h_wave
gosub *h_move

xv = 8 : yv = 340 : p0v = -j*so*256/length
gosub *v_wave
gosub *v_move

pos yh-125+175-8,yv-640+375-8 : mes "●"

n=th : xs=400
p0h = ph : nn=nh
gosub *h_wave
if d|=1 : gosub *h_move

p0v = pv : nn=nv
gosub *v_wave
if d|=1 : gosub *v_move

```

```

pos xh-8,yh-8 : mes "●"
if cs>254 : return
if cs<-254 : return

if ah>0 [
  if snh>220 : pos xh-8,yh-28 : mes "左"
  if snh<-220 : pos xh-8,yh+12 : mes "右"
]

if ah<0 [
  if snh<-220 : pos xh-8,yh-28 : mes "左"
  if snh>220 : pos xh-8,yh+12 : mes "右"
]

return

;-----
*V_wave

pos xv,yv
repeat n
  i=cnt
  pv = i*mn*256/100/length +p0v
  emsin snv,pv
  xv = xs+i-1 : yv = -(snv*av/256)+640
  line xv,yv
loop

return

;-----
*V_move

pos xv-8,yv-8 : mes "●"

if sn>254 : return
if sn<-254 : return

if av>0 [
  if snv>220 : pos xv-8,yv-28 : mes "上"
  if snv<-220 : pos xv-8,yv+12 : mes "下"
]

```

```

if av<0 [
  if snv<-220 : pos xv-8,yv-28 : mes "上"
  if snv>220 : pos xv-8,yv+12 : mes "下"
]

return

;-----
*filter

color 255,255,255

repeat 12
  i=cnt*2-11
  x1=i*10 : x2=x1 : y1=110 : y2=-110
  gosub *rot
  x1=x0+x1 : x2=x0+x2 : y1=y0+y1 : y2=y0+y2
  pos x1,y1 : line x2,y2
loop

x1=-110 : y1=110 : x2=110 : y2=110
gosub *rot
x1=x0+x1 : x2=x0+x2 : y1=y0+y1 : y2=y0+y2
pos x1,y1 : line x2,y2

x1=-110 : y1=-110 : x2=110 : y2=-110
gosub *rot
x1=x0+x1 : x2=x0+x2 : y1=y0+y1 : y2=y0+y2
pos x1,y1 : line x2,y2

return

*rot

emsin sn,a : emcos cs,a

xx1 = ((x1*cs)+(y1*sn))/256
yy1 = (-(x1*sn)+(y1*cs))/256

xx2 = ((x2*cs)+(y2*sn))/256
yy2 = (-(x2*sn)+(y2*cs))/256

x1=xx1 : y1=yy1 : x2=xx2 : y2=yy2

return

```