

### 計算物理学実習 2021 年度課題 No. 3

1. 理工学 ITC のワークステーションルームのワークステーションにログインし、作業用ディレクトリディレクトリ `~/comp-phys/j3` を作成する。作業はこのディレクトリで行う。

2. テキスト 71 頁の課題 4-A

(a) ディレクトリ `/home/kyozai/jy/comp-phys/sec4` の下のファイル `wave.f90` を作業用ディレクトリ `~/comp-phys/j3` にコピーする。

(b) `wave.f90` の概略の PAD を図 J3-1 に、変数名の対応表を図 J3-2 に、プログラムのリストを図 J3-3(a), (b), (c) に示す。この PAD と変数名の対応表を参考にプログラムのリストを読み、各ブロックでの処理を理解する。以下の点に注意すること。

- `real(kind=8)::`, `complex(kind=8)::` は、それぞれ倍精度実数型、倍精度複素数型を表す。
- `integer,parameter:: idim=4097` は、変数 `idim` が整数型で値が 4097 のパラメタであることを示す。パラメタとは、プログラム中で値が変わらない定数という意味である。
- `cdexp(c)` は倍精度複素数型の引数の指数関数を与える倍精度複素数型の組み込み関数。
- `dconjg(c)` は倍精度複素数型の引数の複素共役を与える倍精度複素数型の組み込み関数。
- 複素数  $2 + 3i$  を与える倍精度複素数型の定数は `(2.0d0, 3.0d0)` と書く。

- |                                     |                 |
|-------------------------------------|-----------------|
| <pre>do while(条件)     ⋮ enddo</pre> | は while 型反復を表す。 |
|-------------------------------------|-----------------|

- |  |   |  |  |
|--|---|--|--|
| <pre>do i=n,m     ⋮ enddo</pre>                | は | <table border="1" data-bbox="683 1547 1094 1704"><tr><td><pre>do 行番号 i=n,m     ⋮     行番号 continue</pre></td></tr></table> と同等。 | <pre>do 行番号 i=n,m     ⋮     行番号 continue</pre> |
| <pre>do 行番号 i=n,m     ⋮     行番号 continue</pre> |   |  |  |

- グラフのプロットに関係無い出力行には、先頭に `#` を付けてある。  
gnuplot では `#` で始まる行はコメント行として無視されるので、こうすることにより、出力を直接 gnuplot で利用できるようになる。グラフのプロットに gnuplot を用いない場合でも、`grep` コマンドを用いれば `#` のついた行だけを集めることも、取り除くことも簡単にできる。

(c)  $L = 4, \sigma = 0.1, k_0 = 20, x_0 = -0.5$  として、 $\Delta x = 1/64, 1/128, 1/256$  の場合に、時刻  $t = 0$  から  $t = 0.05$  迄の  $\langle x \rangle, \langle (x - \langle x \rangle)^2 \rangle$  の時間変化を計算する。テキスト 66 頁の注意を読んで Jmax, Nmax, N1max を選ぶこと。

- パラメーターの入力は、入力すべきパラメーターを書き込んだ入力ファイルを作り、リダイレクションを使うと便利。
- 結果もリダイレクションを使いファイルに出力すると便利。

(d) ガウス波束に対する厳密解  $\langle x \rangle = x_0 + k_0 t, \langle (x - \langle x \rangle)^2 \rangle = \sigma^2 \left( 1 + \frac{t^2}{4\sigma^4} \right)$  を数値的に計算するプログラムを PAD を用いて設計し、FORTRAN プログラムを書く。時刻  $t = 0$  から  $t = 0.05$  迄  $N1max \times \Delta t$  おきに厳密解を計算し出力する。

(e)  $\Delta x = 1/64, 1/128, 1/256$  の場合の結果と厳密解の結果を  $\langle x \rangle$  対  $t$  および  $\langle (x - \langle x \rangle)^2 \rangle$  対  $t$  のグラフとして描き印刷する。

(f)  $\Delta x = 1/64, 1/128, 1/256$  の場合に、 $t = 0.05$  における  $\langle x \rangle, \langle (x - \langle x \rangle)^2 \rangle$  を計算し、横軸を  $\Delta x^2$ 、縦軸を  $\langle x \rangle$  または  $\langle (x - \langle x \rangle)^2 \rangle$  としてプロットする。空間の差分化による誤差が  $\Delta x^2$  に比例しているとして  $t = 0.05$  における  $\langle x \rangle, \langle (x - \langle x \rangle)^2 \rangle$  の値を推定する。その推定値をガウス波束に対する厳密解の値と比較する。ガウス波束に対する厳密解の値を  $\Delta x^2 = 0$  の軸上にプロットして比較すると良い。グラフを印刷する。

### 3. テキスト 71 頁の課題 4-A 追加

$|\Psi(x, t)|^2, \text{Re}\Psi(x, t), \text{Im}\Psi(x, t)$  対  $x$  を同時にプロットしたグラフを時刻  $t = 0$  と  $t = 0.05$  の 2 つの場合に描き印刷する。 $\Delta x = 1/256$  とする。wave.f90 を元にして、必要な出力が得られるようにプログラムを書き換える。

### 4. テキスト 72 頁の課題 4-B

(a) wave.f90 を元にして、ポテンシャルのある場合を扱うためにプログラムを書き換える。サンプルプログラムは、'implicit none' 宣言をしているので、新しく導入する全ての変数は宣言をしなければならないが、以下では特に断らない。

- $x_j$  におけるポテンシャルの値  $V(x_j)$  を配列要素  $V(j)$  に格納するために、宣言 に倍精度実数配列  $V$  を宣言する。
- ポテンシャルを決めるパラメーター  $x_1, x_2, V_0$  の変数名を  $x1, x2, V0$  とし、その値を読み込む部分を入力に付け加える。

- 配列要素  $v(j)$  にポテンシャルの値  $V(x_j)$  を設定するプログラムブロック **ポテンシャル** を PAD を用いて設計する。全体の PAD の中にこのブロックを配置し、FORTRAN プログラムを書く。

- **ポテンシャル** の PAD の例を図 J3-4 に示す。

▷ この PAD では、 $-L < x_1 < x_2 < L$  であることは **入力** で検査されていると仮定している。

▷  $x(j)$  が  $x_1$  あるいは  $x_2$  と等しいかどうかの比較は、倍精度実数の比較なので、deps の誤差を許すように比較する。

- $x_j$  におけるポテンシャルの値  $V(x_j)$  を配列要素  $v(j)$  に格納してあるとして、**1 ステップ積分** の中の

```
do j=2,jmax
  cdp(j)=(cps(j+1)-2.0d0*cps(j)+cps(j-1))*w*ci*dt
enddo
```

の部分

```
do j=2,jmax
  cdp(j)=((cps(j+1)-2.0d0*cps(j)+cps(j-1))*w-V(j)*cps(j))*ci*dt
enddo
```

のように書き換える必要がある。一行が長くなるのが気になる場合には、例えば、

```
do j=2,jmax
  cdp(j)= &
    ((cps(j+1)-2.0d0*cps(j)+cps(j-1))*w-V(j)*cps(j))*ci*dt
enddo
```

のように、継続行を使って 2 行に分けても良い。

- 波束の反射率、透過率を測定するために、ポテンシャルの左側の存在確率  $P_{\text{left}} = \int_{-L}^{x_1} |\Psi(x)|^2 dx$ 、ポテンシャルの中の存在確率  $P_{\text{center}} = \int_{x_1}^{x_2} |\Psi(x)|^2 dx$ 、ポテンシャルの右側の存在確率  $P_{\text{right}} = \int_{x_2}^L |\Psi(x)|^2 dx$  を計算し、 $P_{\text{left}}$ ,  $P_{\text{center}}$ ,  $P_{\text{right}}$  およびそれらの和  $P_{\text{left}} + P_{\text{center}} + P_{\text{right}}$  を出力するように、**集計** を書き換える。十分時間が経過し、 $P_{\text{center}} \simeq 0$  となった時の  $P_{\text{left}}$  が反射率、 $P_{\text{right}}$  が透過率を与える。

▷  $V(x_j) \neq 0$  である最小の  $j$  を  $j_1$  とする。

$V(x_j) \neq 0$  である最大の  $j$  を  $j_2$  とする。このとき、

$$P_{\text{left}} = \sum_{j=2}^{j_1-1} |\Psi_j|^2 \Delta x$$

$$P_{\text{center}} = \sum_{j=j_1}^{j_2} |\Psi_j|^2 \Delta x$$

$$P_{\text{right}} = \sum_{j=j_2+1}^{J_{\text{max}}} |\Psi_j|^2 \Delta x$$

のように計算できる。

▷  $j_1, j_2$  は ポテンシャル 中で計算しておく。

$j$  を 2 から 1 ずつ増やしていった初めて  $V(x_j) \neq 0$  となった時の  $j$  の値が  $j_1$  になる。

$j$  を  $J_{\text{max}}$  から 1 ずつ減らしていった初めて  $V(x_j) \neq 0$  となった時の  $j$  の値が  $j_2$  になる。

do while 文を使うと便利。

(b)  $k_0 = 30, 35, 40, 45, 50, 55, 60$  の中から 1 つの  $k_0$  の値を選ぶ。 $\sigma = 0.4, 0.6, 0.8$  の中から 1 つの  $\sigma$  の値を選ぶ。この  $k_0$  と  $\sigma$  に対して、反射率、透過率を計算する。 $\Delta x$  は  $1/64, 1/128, 1/256$  などと変化させる。

(c) 横軸を  $\Delta x^2$ 、縦軸を反射率または透過率として結果をプロットする。空間の差分化による誤差が  $\Delta x^2$  に比例しているとして  $\Delta x \rightarrow 0$  の極限における反射率、透過率を推定する。

## 5. レポート

- テキスト 71 頁の課題 4-A と実習課題 No. 3 の 2 頁の課題 4-A 追加とテキスト 72 頁の課題 4-B のレポートとして以下のものを提出してください。

- 自分で書いたプログラムのソース・ファイルと説明。
- 計算に用いたパラメーターとその説明。
- 出力のグラフを印刷したもの
- 考察

締切りは 12/8(水) です。

pdf 形式のファイルとして、keio.jp/授業支援/レポートから提出してください。

- 余裕のある人は課題 4-C, 4-D のレポートも提出してください。

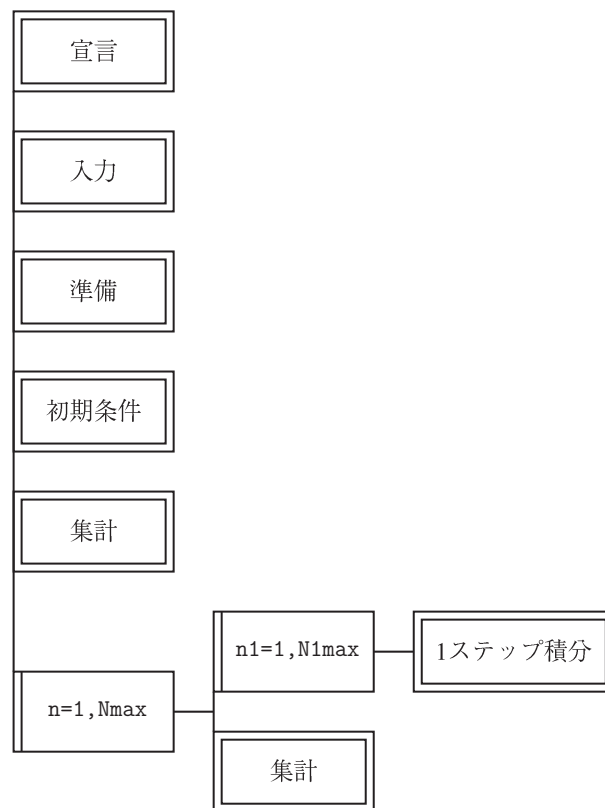


図 J3-1

変数名	対応する量		
a(k)	$a_k$	=	65 頁上の PAD 中の係数
b(k)	$b_k$	=	65 頁上の PAD 中の係数
r(j)	$\rho_j$	=	$ \Psi_j ^2$
x(j)	$x_j$	=	$\Delta x \times (j - 1) - L$
cp(j)	$\Psi_j$	=	$\Psi(x_j)$
cs(j)	$s_j$	=	65 頁上の PAD 中の $s$ の第 $j$ 成分
cps(j)	$\Psi_j^*$	=	65 頁上の PAD 中の $\Psi^*$ の第 $j$ 成分
cdp(j)	$\Delta\Psi_j$	=	65 頁上の PAD 中の $\Delta\Psi$ の第 $j$ 成分
jmax	Jmax	=	空間の分点数
nmax	Nmax	=	出力回数
n1max	N1max	=	1 回の出力あたりのステップ数
rl	$L$		
sig	$\sigma$		
rk	$k_0$		
x0	$x_0$		
dt	$\Delta t$		
dx	$\Delta x$	=	$2L/J_{\max}$
w	$\frac{1}{2} \frac{1}{\Delta x^2}$	=	$\frac{1}{2} \frac{\partial^2}{\partial x^2} \Psi$ の差分計算に出てくる係数
ci	$i$	=	虚数単位
xav	$\langle x \rangle$		
xs	$\langle (x - \langle x \rangle)^2 \rangle$		

図 J3-2

```

=====
! Schroedinger equation
!   A free particle in one dimension
!   <x> and <(x-<x>)**2> for a Gaussian wave packet
!   modified for the f90 form by J.Yamauchi 2011-10-06
=====
program wave
!=====
!   sengen
!=====
implicit none
integer,parameter:: idim=4097
real(kind=8):: a(4), b(4), r(idim), x(idim)
complex(kind=8):: cp(idim), cs(idim), cps(idim), cdp(idim)
!
integer:: iok,jmax,nmax,nlmax,j,n,n1,m
real(kind=8):: rl,sig,rk,x0,dt,dx,w,xj,sum,rnf,xav,xs
complex(kind=8):: ci
!=====
!   nyuuryoku
!=====
iok=0
do while (iok.eq.0)
  write(7,*) ' input Jmax,' &
    , ' where Jmax must be even and 2<= Jmax <=',idim-1,'.'
  read(*,*) jmax
  if (jmax.le.0) then
    write(7,*)'error: Jmax must be positive.'
  else if (jmax.ge.idim) then
    write(7,*)'error: Jmax must be less than',idim,'.'
  else if (mod(jmax,2).ne.0) then
    write(7,*) 'error: Jmax must be even.'
  else
    iok=1
  endif
enddo
!
iok=0
do while (iok.eq.0)
  write(7,*) ' input Nmax(> 0).'
  read(*,*) nmax
  if(nmax.le.0) then
    write(7,*)'error: Nmax must be positive.'
  else
    iok=1
  endif
enddo
!
iok=0
do while (iok.eq.0)
  write(7,*) ' input Nlmax (> 0).'
  read(*,*) nlmax
  if(nlmax.le.0) then
    write(7,*)'error: Nlmax must be positive.'
  else
    iok=1
  endif
enddo
!
write(*,')( '# Jmax = ',i12)') jmax
write(*,')( '# Nmax = ',i12)') nmax
write(*,')( '# Nlmax = ',i12)') nlmax
!

```

図 J3-3(a)

```

iok=0
do while (iok.eq.0)
  write(7,*)'  input L(>0).'
  read(*,*) r1
  if(r1.le.0.0d0) then
    write(7,*)'error: L must be positive.'
  else
    iok=1
  endif
enddo
!
iok=0
do while (iok.eq.0)
  write(7,*)'  input sig(>0).'
  read(*,*) sig
  if(sig.le.0.0d0) then
    write(7,*)'error: sig must be positive.'
  else
    iok=1
  endif
enddo
!
write(7,*)'  input k0.'
read(*,*) rk
write(7,*)'  input x0.'
read(*,*) x0
!
iok=0
do while (iok.eq.0)
  write(7,*)'  input dt.'
  read(*,*) dt
  if(dt.le.0.0d0) then
    write(7,*)'error: dt must be positive.'
  else
    iok=1
  endif
enddo
!
write(*,')( '# L      = ',e18.8e3)' r1
write(*,')( '# sig    = ',e18.8e3)' sig
write(*,')( '# k0     = ',e18.8e3)' rk
write(*,')( '# x0     = ',e18.8e3)' x0
write(*,')( '# dt     = ',e18.8e3)' dt
!=====
! junbi
!=====
a(1)=0.0d0
a(2)=0.5d0
a(3)=0.5d0
a(4)=1.0d0
b(1)=1.0d0/6.0d0
b(2)=1.0d0/3.0d0
b(3)=1.0d0/3.0d0
b(4)=1.0d0/6.0d0
dx=2.0d0*r1/dbl(jmax)
w =0.5d0/dx**2
ci=(0.0d0,1.0d0)
do j=1,jmax+1
  x(j)=dx*dbl(j-1)-r1
enddo

```

図 J3-3(b)



```

!=====
! shoki-jookken
!=====
sum=0.0d0
do j=2,jmax
  xj=x(j)
  cp(j)=cdexp(ci*rk*xj-((xj-x0)/(2.0d0*sig))**2)
  sum=sum+cp(j)*dconjg(cp(j))
enddo
cp(1)=(0.0d0,0.0d0)
cp(jmax+1)=(0.0d0,0.0d0)
rnf=1.0d0/dsqrt(sum*dx)
do j=2,jmax
  cp(j)=rnf*cp(j)
enddo
!
n=0
write(*,'( ''#''/ ''# time'',12x,'<x> ',12x,'<(x-<x>)**2>'')')
!=====
! shuukei
!=====
do j=2,jmax
  r(j)=cp(j)*dconjg(cp(j))
enddo
xav=0.0d0
do j=2,jmax
  xav=xav+x(j)*r(j)*dx
enddo
xs=0.0d0
do j=2,jmax
  xs=xs+((x(j)-xav)**2)*r(j)*dx
enddo
write(*,'(3e18.8e3)') dt*n1max*n,xav,xs
!
cdp(:)=0.0d0
!=====
do n=1,Nmax
  !=====
  do n1=1,N1max
    !=====
    ! 1 step sekibun
    !=====
    do j=2,jmax
      cs(j)=(0.0d0,0.0d0)
    enddo
    do m=1,4
      do j=2,jmax
        cps(j)=cp(j)+a(m)*cdp(j)
      enddo
      do j=2,jmax
        cdp(j)=(cps(j+1)-2.0d0*cps(j)+cps(j-1))*w*ci*dt
      enddo
      do j=2,jmax
        cs(j)=cs(j)+b(m)*cdp(j)
      enddo
    enddo
    do j=2,jmax
      cp(j)=cp(j)+cs(j)
    enddo
  enddo
  !=====
  ! shuukei
  !=====
  do j=2,jmax
    r(j)=cp(j)*dconjg(cp(j))
  enddo
  xav=0.0d0
  do j=2,jmax
    xav=xav+x(j)*r(j)*dx
  enddo
  xs=0.0d0
  do j=2,jmax
    xs=xs+((x(j)-xav)**2)*r(j)*dx
  enddo
  write(*,'(3e18.8e3)') dt*n1max*n,xav,xs
enddo
!=====
end program wave

```

図 J3-3(c)

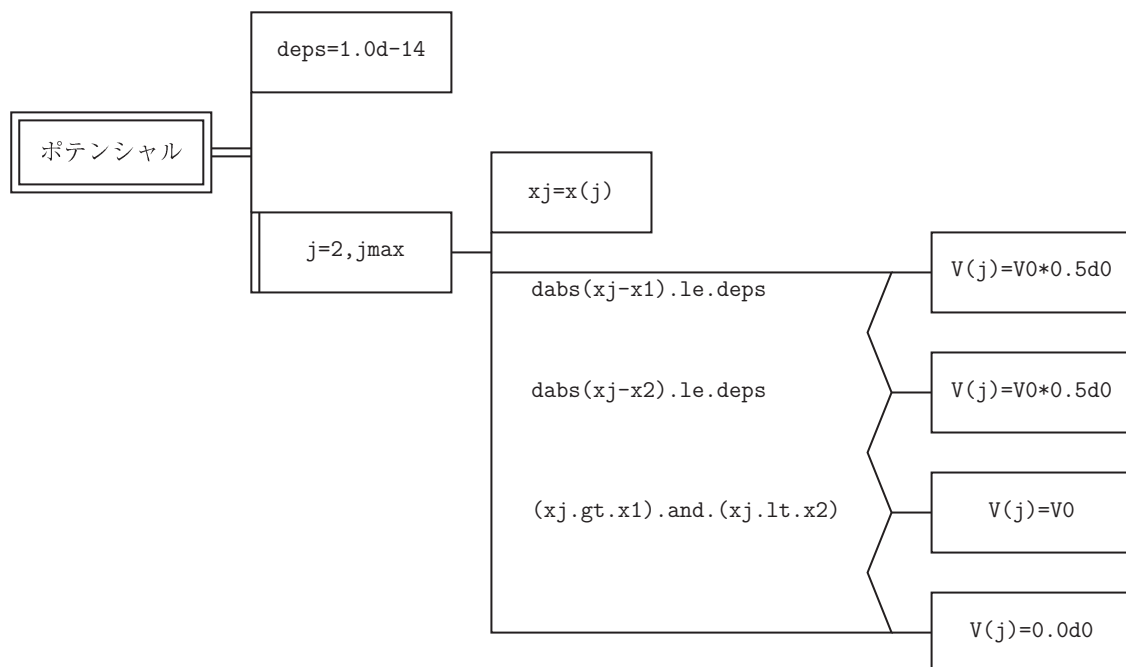


図 J3-4