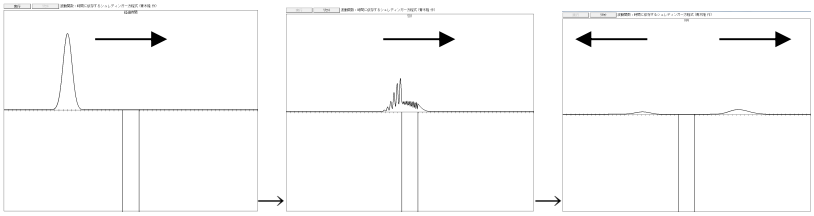
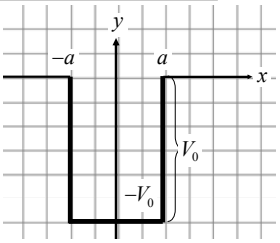
シュレディンガー方程式の解法（中級偏）



１個の量子とポテンシャルの障壁との一次元の衝突をコンピュータでシミュレーションします。角井戸型ポテンシャルを用い、出発時の波束は典型的な量子を記述するガウス型波束を用います。波動関数を各位置すべてにおいてシュレディンガー方程式に従って計算し、入射波束のガウス型波束が角井戸型のポテンシャル障壁に衝突する様子を導きます。（上図のようなグラフィックス化は省きます）

【1】量子力学



量子力学は複素数で記述される力学です。基本になる運動方程式を

- **シュレディンガー方程式**

と言い、波動関数を $\overset{\frown}{\psi}(x,t)$（プサイ(psi)）と表すと、

$$i\hbar\frac{\partial\psi(x,t)}{\partial t}=\left[\frac{1}{2m}\left(-i\hbar\frac{\partial}{\partial x}\right)^2+V(x)\right]\psi(x,t) \Leftarrow V(x)=\begin{cases}0 \cdots |x|>a\\ -V_0 \cdots |x|\le a\end{cases}$$

を満たします。量子の位置 $x(t)$ と運動量 $p(t)$ は、

- 観測毎に値が異なり

平均値しか観測できずに、その値は

$$x(t)=\int_{-\infty}^{\infty}\psi^*(x,t)\,x\psi(x,t)\,dx$$

$$p(t)=\int_{-\infty}^{\infty}\psi^*(x,t)\,i\hbar\frac{\partial\psi(x,t)}{\partial x}\,dx$$

で計算されます。従って、量子の運動は、

- $\psi(x,t)$ がわかればよい

ことになります。とくに、

- $|\psi(x,t)|^2$ は、位置 x に粒子がいる確率

になります。全領域（x：$-\infty\rightarrow+\infty$）に粒子がいる確率は 100%なので

$$\int_{-\infty}^{\infty}|\psi(x,t)|^2\,dx=1$$

になっています。この $\psi(x,t)$ を知るには、微分方程式の**シュレディンガー方程式**を解く必要があります。

【2】差分方程式

解くべき微分方程式は

$$i\hbar\frac{\partial\psi(x,t)}{\partial t}=-\frac{\hbar^2}{2m}\psi''(x,t)+V(x)\psi(x,t)\quad\left(\psi''(x,t)=\frac{\partial^2\psi(x,t)}{\partial x^2}\right)$$

です。これは、

- 位置xの値を固定してtの変化による$\psi(x,t)$を計算する

ことになります。差分に直す微分は、2種類で、

- $\frac{\partial\psi(x,t)}{\partial t}$　　- $\frac{\partial^2\psi(x,t)}{\partial x^2}\left(=\psi''(x,t)\right)$

になります。微分を差分に直すには、**テイラー展開**をつかいます。テイラー展開は

$$f(x+a)=\sum_{n=0}^{\infty}\frac{a^n}{n!}f^{(n)}(x)=f(x)+a\boxed{f'(x)}+\frac{a^2}{2!}\boxed{f''(x)}+\frac{a^3}{3!}f'''(x)+\cdots$$

で与えられ、第１項と第２項に該当する微分があります。差分の数値計算では、

$$\left.\begin{aligned}dt\\\frac{dx}{dt}=f(x)\end{aligned}\right\}\Rightarrow\left.\begin{aligned}\Delta t\\\frac{\Delta x}{\Delta t}=f(x)\end{aligned}\right\}\Rightarrow\left.\begin{aligned}x_{\text{次の時刻}}-x_{\text{現時刻}}=\boxed{\Delta x}\\\boxed{\Delta x}=f(x)\Delta t\end{aligned}\right\}\Rightarrow\left.\begin{aligned}x_{\text{次の時刻}}=x_{\text{現時刻}}+\boxed{\Delta x}\\x_{\text{次の時刻}}=x_{\text{現時刻}}+f(x)\Delta t\end{aligned}\right\}$$

のように、

$$\begin{cases}t_{\text{次の時刻}}=t_{\text{現時刻}}+\Delta t\\x_{\text{次の時刻}}=x_{\text{現時刻}}+\Delta x\end{cases}$$

として次の値を決めていきます。一つ前の値は

$$\begin{cases}t_{\text{前の時刻}}=t_{\text{現時刻}}-\Delta t\\x_{\text{前の時刻}}=x_{\text{現時刻}}-\Delta x\end{cases}$$

です。**位置xについて**の差分方程式は、$x+\Delta x$と$x-\Delta x$の2つのテイラー展開：

$$\psi(x+\Delta x,t)=\sum_{n=0}^{\infty}\frac{\Delta x^n}{n!}\psi^{(n)}(x,t)=\psi(x,t)+\Delta x\psi'(x,t)+\frac{\Delta x^2}{2!}\psi''(x,t)+\frac{\Delta x^3}{3!}\psi'''(x,t)+\cdots$$

$$\psi(x-\Delta x,t)=\sum_{n=0}^{\infty}\frac{(-\Delta x)^n}{n!}\psi^{(n)}(x,t)=\psi(x,t)-\Delta x\psi'(x,t)+\frac{\Delta x^2}{2!}\psi''(x,t)-\frac{\Delta x^3}{3!}\psi'''(x,t)+\cdots$$

を利用して、

$$\psi(x+\Delta x,t)-\psi(x,t)=\Delta x\boxed{\psi'(x,t)}+\overbrace{\frac{\Delta x^2}{2!}\psi''(x,t)+\frac{\Delta x^3}{3!}\psi'''(x,t)+\cdots}^{\fallingdotseq 0\text{にするには、充分小さい}\Delta x\text{をつかう。}\Delta x\gg\Delta x^2\gg\Delta x^3\text{とできる。}}=\Delta x\boxed{\psi'(x,t)}$$

$$\psi(x+\Delta x,t)+\psi(x-\Delta x,t)=2\psi(x,t)+2\frac{\Delta x^2}{2!}\boxed{\psi''(x,t)}+\underbrace{\cdots}_{\fallingdotseq 0\text{にするには、充分小さい}\Delta x\text{をつかう。}\Delta x^2\gg\Delta x^3\text{とできる。}}$$

$$=2\psi(x,t)+2\frac{\Delta x^2}{2!}\boxed{\psi''(x,t)}$$

より、

- 充分小さいΔx

$\Delta x = 0.01$

$\Delta x = 0.001 \Rightarrow$ 計算精度がより良い

など

の時には

$$\begin{cases} \psi'(x,t) = \dfrac{\psi(x+\Delta x,t)-\psi(x,t)}{\Delta x} \\ \psi''(x,t) = \dfrac{\psi(x+\Delta x,t)-2\psi(x,t)+\psi(x-\Delta x,t)}{\Delta x^2} \end{cases}$$

と差分であらわすことができます。**時間の微分**は

$$\frac{\partial\psi(x,t)}{\partial t} = \frac{\psi(x,t+\Delta t)-\psi(x,t)}{\Delta t}$$

になります。

【3】シュレディンガー方程式

微分方程式:

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\psi''(x,t)+V(x)\psi(x,t)$$

は、差分方程式として

$$i\hbar\frac{\psi(x,t+\Delta t)-\psi(x,t)}{\Delta t} = -\frac{\hbar^2}{2m}\frac{\psi(x+\Delta x,t)-2\psi(x,t)+\psi(x-\Delta x,t)}{\Delta x^2}+V(x)\psi(x,t)$$

になります。この差分方程式は、$\Delta t \to 0$ & $\Delta x \to 0$ で微分方程式に帰着します。これより、

$$\overbrace{\psi(x,t+\Delta t)}^{\text{時刻}t+\Delta t} = \overbrace{\psi(x,t)-\frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi(x+\Delta x,t)-2\psi(x,t)+\psi(x-\Delta x,t)}{\Delta x^2}-V(x)\psi(x,t)\right]\Delta t}^{\text{時刻}t}$$

として次の時刻 $t+\Delta t$ の値を決めていきます。まず、最初の波動関数は $t=t_0$ で分かっているとして与えます。標準な関数として、図 1 のような形状（**ガウス型波束**と呼ばれる）の

$$\psi(x,t_0) = \frac{1}{\sqrt[4]{2\pi D^2}}\exp\left[-\frac{(x-\langle x\rangle)^2}{4D^2}+\frac{i\langle p\rangle x}{\hbar}\right]$$

図 1 : $\hbar = 1,\ D = 0.035,\ \langle x\rangle = -0.3,\ \langle p\rangle = 50\pi$

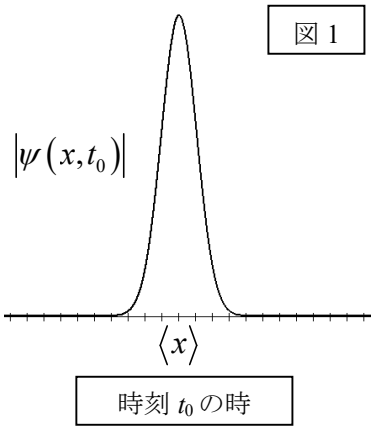


図 1

時刻 t_0 の時

にします。これを用いて、$x(t_0)$, $p(t_0)$ を計算

$$x(t_0) = \int_{-\infty}^{\infty}\psi^*(x,t_0)\,x\psi(x,t_0)\,dx$$

$$p(t_0) = \int_{-\infty}^{\infty}\psi^*(x,t_0)\,i\hbar\frac{\partial\psi(x,t_0)}{\partial x}\,dx$$

すると

$$\begin{cases} x(t_0) = \langle x\rangle \\ p(t_0) = \langle p\rangle \end{cases}$$

になります。$|\psi(x,t_0)|$ は

- 時刻 t_0 に量子を位置 x に見つける確率

- 大雑把に$x=\langle x\rangle$の付近の確率が大きい
  - ✧ 時刻t_0の量子の位置はだいたい$\langle x\rangle \Rightarrow x(t_0)=\langle x\rangle$

を与えています。また、

- 無限遠方には量子は存在しないと見なすので、$x=\pm\infty$では、
  - $\psi(x,t)=0$

とする

を仮定します。

波動関数のx方向の値は、

$$\boxed{x_0} \Rightarrow \boxed{x_1}=x_0+\Delta x \Rightarrow \boxed{x_2}=x_1+\Delta x \Rightarrow \boxed{x_3}=x_2+\Delta x \Rightarrow \cdots$$

$$\boxed{t_0}\ \psi(x_n,t_0)=\frac{1}{\sqrt[4]{2\pi D^2}}\exp\left[-\frac{(x_n-\langle x\rangle)^2}{4D^2}+\frac{i\langle p\rangle x_n}{\hbar}\right](n=0,1,2,3\cdots)$$

のように、$t=t_0$で、$\psi(x,t_0)$の値をx_0から順番にΔxづつ増やして計算しておきます。次に、

$$t_1=t_0+\Delta t$$

で、$\psi(x,t_0)$より、$\psi(x,t_0+\Delta t)$を計算します:

$$\psi\left(x,\overbrace{t_0+\Delta t}^{t_1}\right)=\psi(x,t_0)-\frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi(x+\Delta x,t_0)-2\psi(x,t_0)+\psi(x-\Delta x,t_0)}{\Delta x^2}-V(x)\psi(x,t_0)\right]\Delta t$$

それを用い、x_0から順番にΔxづつ増やして

$$\begin{cases}\psi(x_0,t_1)=\psi(x_0,t_0)+\dfrac{i}{\hbar}\left[\dfrac{\hbar^2}{2m}\dfrac{\psi(x_1,t_0)-2\psi(x_0,t_0)+\boxed{\psi(x_{-1},t_0)}}{\Delta x^2}-V(x_0,t_0)\psi(x_0,t_0)\right]\Delta t \\ \psi(x_n,t_1)=\psi(x_n,t_0)+\dfrac{i}{\hbar}\left[\dfrac{\hbar^2}{2m}\dfrac{\psi(x_{n+1},t_0)-2\psi(x_n,t_0)+\psi(x_{n-1},t_0)}{\Delta x^2}-V(x_n,t_0)\psi(x_n,t_0)\right]\Delta t\end{cases}$$

各x_n $(n=0,1,2,3,\cdots)$で計算します。次に、時間をすすめ

$$t_2=t_1+\Delta t$$

にして、$\psi(x,t_1)$より、$\psi(x,t_2)$を計算し、

$$\begin{cases}\psi(x_0,t_2)=\psi(x_0,t_1)+\dfrac{i}{\hbar}\left[\dfrac{\hbar^2}{2m}\dfrac{\psi(x_1,t_1)-2\psi(x_0,t_1)+\boxed{\psi(x_{-1},t_1)}}{\Delta x^2}-V(x_0,t_1)\psi(x_0,t_1)\right]\Delta t \\ \psi(x_n,t_2)=\psi(x_n,t_1)+\dfrac{i}{\hbar}\left[\dfrac{\hbar^2}{2m}\dfrac{\psi(x_{n+1},t_1)-2\psi(x_n,t_1)+\psi(x_{n-1},t_1)}{\Delta x^2}-V(x_n,t_1)\psi(x_n,t_1)\right]\Delta t\end{cases}$$

のように次々と時間をΔtづつ増やして、計算を繰り返します。ここで、

- $\psi(x_0,t_1)$の計算には、$\psi(x_{-1},t_0)$も必要

なので、

$$\psi\left(\boxed{x_{-1}},t_0\right)=\frac{1}{\sqrt[4]{2\pi D^2}}\exp\left[-\frac{\left(\boxed{x_{-1}}-\langle x\rangle\right)^2}{4D^2}+\frac{i\langle p\rangle\boxed{x_{-1}}}{\hbar}\right]$$

も計算しておきます。従って、

$$x=\boxed{x_{-1}},\ x_0,\ x_1,\ x_2,\ x_3,\ \cdots$$

の並びでΔxづつ増加させ値を決めておきます。

【4】ルンゲ･クッタ法

差分方程式は、

$$\psi\left(x,t+\Delta t\right)=\psi\left(x,t\right)-\frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi\left(x+\Delta x,t\right)-2\psi\left(x,t\right)+\psi\left(x-\Delta x,t\right)}{\Delta x^2}-V\left(x\right)\psi\left(x,t\right)\right]\Delta t$$

より、$t=t_n$のとき、$t+\Delta t$は

$$t_{n+1}=t_n+\Delta t$$

となるので、

$$\psi\left(x,t_{n+1}\right)=\psi\left(x,t_n\right)-\frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi\left(x+\Delta x,t_n\right)-2\psi\left(x,t_n\right)+\psi\left(x-\Delta x,t_n\right)}{\Delta x^2}-V\left(x\right)\psi\left(x,t_n\right)\right]\Delta t$$

と表されます。これはオイラー法の

$$\boxed{x_{n+1}=x_n+f\left(x_n,t_n\right)\Delta t}$$

に当たります。あるいは$x\left(t_0\right)=x_0,\ x\left(t_1\right)=x_1,\ x\left(t_2\right)=x_2,\ x\left(t_3\right)=x_3,\ \cdots$と対応させると

$$x\left(t_{n+1}\right)=x\left(t_n\right)+f\left(x\left(t_n\right),t_n\right)\Delta t$$

になります。そこで、

$$x\left(t_n\right)\rightarrow\psi\left(x,t_n\right)$$

と対応させることができるので、ψでは$f\left(\psi\left(x,t_n\right),t_n\right)$を用いて

$$\psi\left(x,t_{n+1}\right)=\psi\left(x,t_n\right)+f\left(\psi\left(x,t_n\right),t_n\right)\Delta t$$
$$f\left(\psi\left(x,t_n\right),t_n\right)=-\frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi\left(x+\Delta x,t_n\right)-2\psi\left(x,t_n\right)+\psi\left(x-\Delta x,t_n\right)}{\Delta x^2}-V\left(x\right)\psi\left(x,t_n\right)\right]$$

と分かります。また、$x\left(t_0\right)=x_0,\ x\left(t_1\right)=x_1,\ x\left(t_2\right)=x_2,\ x\left(t_3\right)=x_3,\ \cdots$に対して

$$\psi\left(x,t_0\right)=\psi_0\left(x\right),\ \psi\left(x,t_1\right)=\psi_1\left(x\right),\ \psi\left(x,t_2\right)=\psi_2\left(x\right),\ \psi\left(x,t_2\right)=\psi_3\left(x\right),\ \cdots$$

という関係になります。従って、

$$\psi_{n+1}(x)=\psi_n(x)+f\left(\psi_n(x),t_n\right)\Delta t$$

$$f\left(\psi_n(x),t_n\right)=-\frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi_n(x+\Delta x)-2\psi_n(x)+\psi_n(x-\Delta x)}{\Delta x^2}-V(x)\psi_n(x)\right]$$

（t_nは無い）

と表記できます。実際には、t_n は使われていないです。

ルンゲ･クッタ法では、**【第５回 数値シミュレーション：１階の微分方程式（シラバス８・９回目）】－【４】Ｃ言語ﾌﾟﾛｸﾞﾗﾑ：Runge-Kutta 法**にてSTEP $\rightarrow \Delta t$ と読み直して

$$x_{n+1}=x_n+\overbrace{f\left(x_n,t_n\right)}^{k_1}\Delta t \Rightarrow x_{n+1}=x_n+\frac{k_1+2k_2+2k_3+k_4}{6}\Delta t$$

$$\begin{cases}k_1=f\left(x_n,t_n\right)\\ k_2=f\left(x_n+\frac{k_1}{2}\Delta t,t_n+\frac{\Delta t}{2}\right)\\ k_3=f\left(x_n+\frac{k_2}{2}\Delta t,t_n+\frac{\Delta t}{2}\right)\\ k_2=f\left(x_n+k_3\Delta t,t_n+\Delta t\right)\end{cases}$$

を計算します。ここでは、ψ について解くので、$x\rightarrow\psi$ にすると

$$\psi_{n+1}=\psi_n+k_1\Delta t \Rightarrow \psi_{n+1}=\psi_n+\frac{k_1+2k_2+2k_3+k_4}{6}\Delta t$$

$$\begin{cases}k_1=f\left(\psi_n,t_n\right)\\ k_2=f\left(\psi_n+\frac{k_1}{2}\Delta t,t_n+\frac{\Delta t}{2}\right)\\ k_3=f\left(\psi_n+\frac{k_2}{2}\Delta t,t_n+\frac{\Delta t}{2}\right)\\ k_2=f\left(\psi_n+k_3\Delta t,t_n+\Delta t\right)\end{cases}$$

になります。ところで、ψ の場合には、各位置 x で計算するので、区別するため x を付け、

$$\psi_n\rightarrow\psi_n(x),\ \psi_{n+1}\rightarrow\psi_{n+1}(x),\ k_{1,2,3,4}\rightarrow k_{1,2,3,4}(x)$$

とします。従って、

$$\psi_{n+1}(x)=\psi_n(x)+k_1(x)\Rightarrow\psi_{n+1}(x)=\psi_n(x)+\frac{k_1(x)+2k_2(x)+2k_3(x)+k_4(x)}{6}\Delta t$$

になります。ルンゲ･クッタ法は

$$\psi_{n+1}(x)=\psi_n(x)+\frac{k_1(x)+2k_2(x)+2k_3(x)+k_4(x)}{6}\Delta t$$

になり、

$$k_1(x)=f\left(\psi_n(x),t_n\right)=\frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi_n(x+\Delta x)-2\psi_n(x)+\psi_n(x-\Delta x)}{\Delta x^2}-V(x)\psi_n(x)\right]$$

$$k_2(x)=f\left(\psi_n(x)+\frac{k_1(x)}{2}\Delta t,t_n+\frac{\Delta t}{2}\right)$$

$$=\frac{i}{\hbar}\left[\begin{array}{l}\dfrac{\hbar^2}{2m}\dfrac{\overbrace{\psi_n(x+\Delta x)+\dfrac{k_1(x+\Delta x)}{2}\Delta t}^{x+\Delta x}-2\overbrace{\left(\psi_n(x)+\dfrac{k_1(x)}{2}\Delta t\right)}^{x}+\overbrace{\psi_n(x-\Delta x)+\dfrac{k_1(x-\Delta x)}{2}\Delta t}^{x-\Delta x}}{\Delta x^2}\\ -V(x)\overbrace{\left(\psi_n(x)+\dfrac{k_1(x)}{2}\Delta t\right)}^{x}\end{array}\right]$$

$$k_3(x)=f\left(\psi_n(x)+\frac{k_2(x)}{2}\Delta t,t_n+\frac{\Delta t}{2}\right)$$

$$=\frac{i}{\hbar}\left[\begin{array}{l}\dfrac{\hbar^2}{2m}\dfrac{\overbrace{\psi_n(x+\Delta x)+\dfrac{k_2(x+\Delta x)}{2}\Delta t}^{x+\Delta x}-2\overbrace{\left(\psi_n(x)+\dfrac{k_2(x)}{2}\Delta t\right)}^{x}+\overbrace{\psi_n(x-\Delta x)+\dfrac{k_2(x-\Delta x)}{2}\Delta t}^{x-\Delta x}}{\Delta x^2}\\ -V(x)\overbrace{\left(\psi_n(x)+\dfrac{k_2(x)}{2}\Delta t\right)}^{x}\end{array}\right]$$

$$k_4(x)=f\left(\psi_n(x)+k_3(x),t_n\right)$$

$$=\frac{i}{\hbar}\left[\begin{array}{l}\dfrac{\hbar^2}{2m}\dfrac{\overbrace{\psi_n(x+\Delta x)+k_3(x+\Delta x)\Delta t}^{x+\Delta x}-2\overbrace{\left(\psi_n(x)+k_3(x)\Delta t\right)}^{x}+\overbrace{\psi_n(x-\Delta x)+k_3(x-\Delta x)\Delta t}^{x-\Delta x}}{\Delta x^2}\\ -V(x)\overbrace{\left(\psi_n(x)+k_3(x)\Delta t\right)}^{x}\end{array}\right]$$

で与えられます。

【5】プログラムの準備

得られた数式をプログラムに合うように整理します。まず、使用する数値は、**シッフ著「量子力学（上）（物理学叢書 (2)、吉岡書店）」**にある数値を参考にします。

使用する数値

位置の進み： $\boxed{\Delta x=0.004}$

$$-250\Delta x\le x\le 250\Delta x\Rightarrow\begin{cases}-\dfrac{m_{\max}}{2}\Delta x\le x\le\dfrac{m_{\max}}{2}\Delta x\ \left(\boxed{m_{\max}=500}\right)\\ \boxed{x=\left(m-\dfrac{m_{\max}}{2}\right)\Delta x}\Rightarrow\overbrace{\left(m-\dfrac{m_{\max}}{2}\right)}^{m=0\Rightarrow-\frac{m_{\max}}{2}}\Delta x\le x\le\overbrace{\left(m-\dfrac{m_{\max}}{2}\right)}^{m=i_{\max}\Rightarrow\frac{m_{\max}}{2}}\Rightarrow\boxed{m=0\rightarrow m=m_{\max}}\end{cases}$$

時間の進み： $\boxed{\Delta t=\dfrac{\Delta x^2}{4}}$

$$0\le t\le 800\Delta t\Rightarrow 0\le t\le n_{\max}\Delta t\ \left(\boxed{n_{\max}=800}\right)\Rightarrow t=n\Delta t\Rightarrow\boxed{n=0\rightarrow n=n_{\max}}$$

シュレディンガー方程式：

$$i\hbar\frac{\partial\psi(x,t)}{\partial t}=-\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2}+V(x,t)\psi(x,t)\Leftarrow\boxed{\hbar=1,\ 2m=1}$$

ポテンシャル：$V(x,t)=\begin{cases}0\ \cdots|x|>a\\ -V_0\cdots|x|\le a\end{cases}\Leftarrow\boxed{V_0=(70.7\pi)^2,\ a=0.032\ (\text{幅}:0.064)}$

量子の初期（*t*=0）での波動関数：ガウス型波束：

$$\psi(x,t_0)=\frac{1}{\sqrt[4]{2\pi D^2}}\exp\left[-\frac{(x-\langle x\rangle)^2}{4D^2}+\frac{i\langle p\rangle x}{\hbar}\right]\Leftarrow\boxed{D=0.035,\ \langle x\rangle=-0.3,\ \langle p\rangle=50\pi}$$

です。プログラム内では

$$\psi(x,t_0)=\frac{1}{\sqrt[4]{2\pi D^2}}\exp\left[-\frac{(x-\langle x\rangle)^2}{4D^2}\right]\left(\cos\frac{\langle p\rangle x}{\hbar}+i\sin\frac{\langle p\rangle x}{\hbar}\right)$$

として用います。実部と虚部に分けて用います。

$\psi_n(x)$の n について

時間発展は、

$$\psi_n(x):n=0\xrightarrow{\text{時間経過}\rightarrow}n=1\rightarrow n=2\rightarrow\cdots$$

とnで追跡します。また、位置の変化は

$$x:x=\boxed{x_{-1}},x_0,x_1,x_2,\cdots\Rightarrow x_m$$

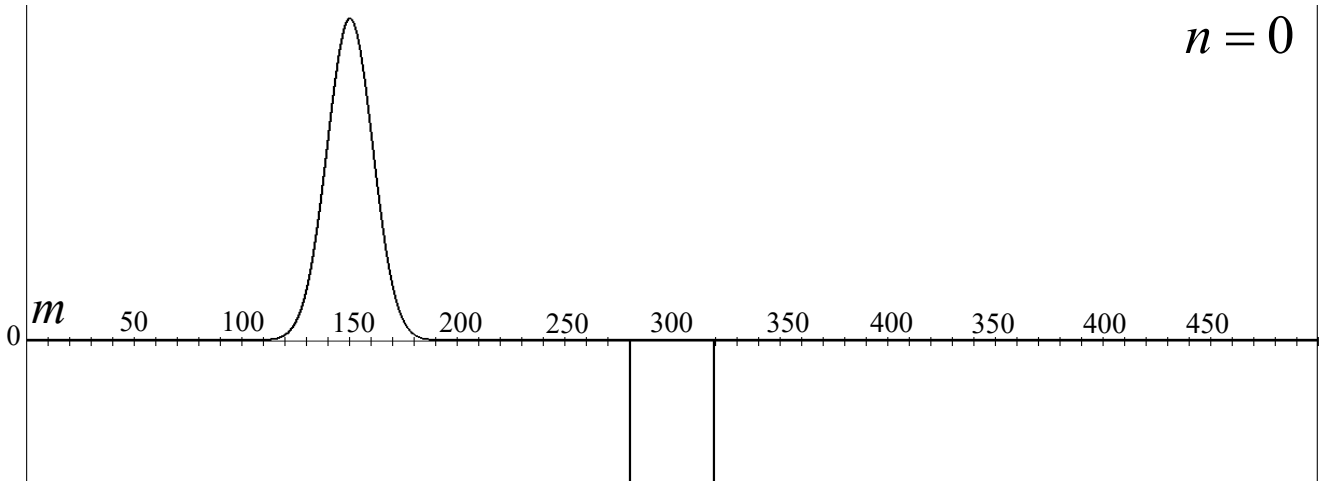
とmで追跡するとしましょう。プログラムでは

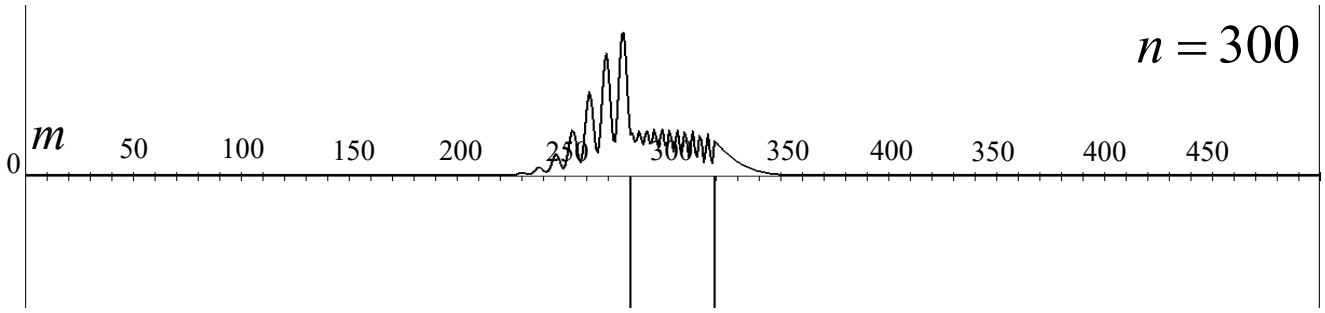
$$\begin{cases}m:\text{位置}x\\ n:\text{時間}t\end{cases}$$

の添え字を使います。mとnは

$$\begin{cases}x-\Delta x\Rightarrow m-1\\ x\Rightarrow m\\ x+\Delta x\Rightarrow m+1\end{cases}\quad\begin{cases}t\Rightarrow n\\ t+\Delta t\Rightarrow n+1\end{cases}$$

に対応します。





これを踏まえて、ある位置に於ける時間発展を配列(**【第２回　Ｃ言語の構成要素（シラバス３・４回目)】－【８】配列**)を使います。ここに、

- 列の添え字は**0から始まる**

に注意しましょう。従って、

$$x : x = \boxed{x_{-1}}, x_0, x_1, x_2, \cdots \Rightarrow x_m \Rightarrow x_{-1} : \boxed{m = -1}$$

の設定は、$m = \boxed{-1}$から始まりますが、

- $m = \boxed{-1}$はＣ言語の配列の添え字では、**使用不可**

です。そのため、

$$\psi\left(\overbrace{\boxed{x_0}}^{\boxed{m=1}}, t_1\right) = \psi\left(x_0, t_0\right) + \frac{i}{\hbar}\left[\frac{\hbar^2}{2m}\frac{\psi\left(x_1, t_0\right) - 2\psi\left(x_0, t_0\right) + \psi\left(\overbrace{\boxed{x_{-1}}}^{\boxed{m=0}}, t_1\right)}{\Delta x} - V\left(x_0, t_0\right)\psi\left(x_0, t_0\right)\right]$$

より

$$\begin{aligned}
&\boxed{m = 0} \Rightarrow x_{\boxed{-1}} \\
&m = 1 \Rightarrow x_0 \\
&m = 2 \Rightarrow x_1 \\
&\quad \vdots
\end{aligned}$$

の対応になります。

配列は、

$$\begin{cases}
\psi_n\left(x + \Delta x\right) \Rightarrow \psi\left[n+1\right]\left[m\right] \\
\psi_n\left(x\right) \Rightarrow \psi\left[n\right]\left[m\right] \\
\psi_n\left(x - \Delta x\right) \Rightarrow \psi\left[n-1\right]\left[m\right]
\end{cases}$$

$$k_{1,2,3,4}\left(x\right) \Rightarrow k_{1,2,3,4}\left[m\right]$$

として追跡します。同様に、ポテンシャルは

$$V\left(x\right) \Rightarrow V\left[m\right]$$

とします。これに従って、書き換えた表式が、

$$\psi[n+1][m]=\psi[n][m]+\frac{k_1[m]+2k_2[m]+2k_3[m]+k_4[m]}{6}\Delta t$$

です。ここに、

- $m=0$はx_{-1}に対応

し、計算は、$m=1$から始まり

$$\psi[n+1][1]=\psi[n][1]+\frac{k_1[1]+2k_2[1]+2k_3[1]+k_4[1]}{6}\Delta t$$

になります。また、

$$\begin{cases}
k_1[m]=\dfrac{i}{\hbar}\left[\dfrac{\hbar^2}{2m}\dfrac{\psi[n][m+1]-2\psi[n][m]+\psi[n][m-1]}{\Delta x^2}-V[m]\psi[n][m]\right] \\
k_2[m]=\dfrac{i}{\hbar}\left[\begin{array}{l}\dfrac{\hbar^2}{2m}\dfrac{\psi[n][m+1]+\dfrac{k_1[m+1]}{2}\Delta t-2\left(\psi[n][m]+\dfrac{k_1[m]}{2}\Delta t\right)+\psi[n][m-1]+\dfrac{k_1[m-1]}{2}\Delta t}{\Delta x^2} \\ -V[m]\left(\psi[n][m]+\dfrac{k_1[m]}{2}\Delta t\right)\end{array}\right] \\
k_3[m]=\dfrac{i}{\hbar}\left[\begin{array}{l}\dfrac{\hbar^2}{2m}\dfrac{\psi[n][m+1]+\dfrac{k_2[m+1]}{2}\Delta t-2\left(\psi[n][m]+\dfrac{k_2[m]}{2}\Delta t\right)+\psi[n][m-1]+\dfrac{k_2[m-1]}{2}\Delta t}{\Delta x^2} \\ -V[m]\left(\psi[n][m]+\dfrac{k_2[m]}{2}\Delta t\right)\end{array}\right] \\
k_4[m]=\dfrac{i}{\hbar}\left[\begin{array}{l}\dfrac{\hbar^2}{2m}\dfrac{\psi[n][m+1]+k_3[m+1]\Delta t-2\left(\psi[n][m]+k_3[m]\Delta t\right)+\psi[n][m-1]+k_3[m-1]\Delta t}{\Delta x^2} \\ -V[m]\left(\psi[n][m]+k_3[m]\Delta t\right)\end{array}\right]
\end{cases}$$

になり、これがシュレディンガー方程式の解法の主要部分です。プログラムでは、

- $k_1[m]: m=0\rightarrow m=m_{\max}$の**すべての値を計算した後**に$k_2[m]$を計算する

事になります。$k_{2,3,4}[m]$についても同様です。また、分子を一括して取り扱えるように

$$\begin{cases}
\psi[n][m+1]\cdots k_1[m] \\
\psi[n][m+1]+\dfrac{k_1[m+1]}{2}\Delta t\cdots k_2[m] \\
\psi[n][m+1]+\dfrac{k_2[m+1]}{2}\Delta t\cdots k_3[m] \\
\psi[n][m+1]+k_3[m+1]\Delta t\cdots k_4[m]
\end{cases}
\overset{\psi:\text{プサイ}\ (\text{psiと書く})}{\Rightarrow}
\boxed{\text{psi}+r\Delta t\times\text{k}}\Rightarrow\text{psi}=\psi[n][m+1]
\begin{cases}
r=0\cdots k=k_1[m] \\
r=\dfrac{1}{2}\cdots k=k_2[m] \\
r=\dfrac{1}{2}\cdots k=k_3[m] \\
r=1\cdots k=k_4[m]
\end{cases}$$

と見なして

$$\text{psi}+r\Delta t\times\text{k}$$

を用います。

プログラム用変数

プログラムで使用する変数を用意します。C 言語では、

- 使用する変数はすべて宣言

します。配列は大きさを指定します。ここでは、

- 配列（$\psi[n][m]$）の大きさは、添え字（m や n ）の総数で決定

されます。従って、

$$\overbrace{m=0\rightarrow m=m_{\max}}^{\text{総数}:\,m_{\max}+1}\Rightarrow \boxed{m\text{の総数}=m_{\max}+1}$$

$$\overbrace{n=0\rightarrow n=n_{\max}}^{\text{総数}:\,n_{\max}+1}\Rightarrow \boxed{n\text{の総数}=n_{\max}+1}$$

です。また、

$$\overbrace{\psi}^{\text{psi}}[n][m]\Rightarrow \text{psi}[n][m]\Rightarrow \boxed{\text{COMPLEX psi}[n\text{の総数}][m\text{の総数}];}$$

$$k_1[m]\Rightarrow \text{k1}[m]\Rightarrow \boxed{\text{COMPLEX k1}[m\text{の総数}];}$$

$$k_2[m]\Rightarrow \text{k2}[m]\Rightarrow \boxed{\text{COMPLEX k2}[m\text{の総数}];}$$

$$k_3[m]\Rightarrow \text{k3}[m]\Rightarrow \boxed{\text{COMPLEX k3}[m\text{の総数}];}$$

$$k_4[m]\Rightarrow \text{k4}[m]\Rightarrow \boxed{\text{COMPLEX k4}[m\text{の総数}];}$$

$$V[m]\Rightarrow \boxed{\text{dounle V}[m\text{の総数}];}$$

更に

- 無限遠方の $x=\pm\infty$ では、$\psi(x,t)=0$

なので、ここでは、

$$\underbrace{\overbrace{\psi[n][0]}^{x=-\infty},\ \psi[n][1],\ \psi[n][2],\cdots,\ \psi[n][m],\cdots,\ \psi[n][m_{\max-1}],\ \overbrace{\psi[n][m_{\max}]}^{x=\infty}}_{m\text{の総数}=m_{\max+1}}$$

と見なして、

$$\boxed{\begin{aligned}&\psi[n][0]=0\\&\psi[n][n_{\max}]=0\end{aligned}}$$

が条件になります。

使用する数値を、#define を使って、わかりやすい名前で定義します。それらは、

- 自分で定義した事を思い出すために**大文字で記述**

しておきます。

#define PI 3.1415926536 $\Leftarrow \pi=3.1415926536$

define DX 0.004 $\Leftarrow \overbrace{\Delta}^{\text{Delta}}\overbrace{x}^{\text{X}}=0.004$

define DT (DX*DX/4.0) $\Leftarrow \overbrace{\Delta}^{\text{Delta}}\overbrace{t}^{\text{T}}=\dfrac{\Delta x^2}{4}$

#define M_MAX 500 $\Leftarrow -\frac{m_{\max}}{2}\Delta x \le x \le \frac{m_{\max}}{2}\Delta x$

#define M_TOTAL M_MAX+1 $\Leftarrow \overbrace{m\text{の}\underbrace{\text{総数}}_{\text{total}}}^{\text{m\_total}} = m_{\max}+1$

#define N_MAX 800 $\Leftarrow 0 \le t \le n_{\max}\Delta t$

#define N_TOTAL N_MAX+1 $\Leftarrow \overbrace{n\text{の}\underbrace{\text{総数}}_{\text{total}}}^{\text{n\_total}} = n_{\max}+1$

$\begin{cases}\text{\#define H\_SLASH 1.0}\\ \text{\#define MASS 0.5}\end{cases} \Leftarrow i\overbrace{\hbar}^{\text{h\_slash}}\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2\underbrace{m}_{\text{mass}}}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x,t)\psi(x,t)$

$\begin{cases}\text{\#define D 0.035}\\ \text{\#define X\_KAKKO } -0.3\\ \text{\#define P\_KAKKO 50.0*PI}\end{cases} \Leftarrow \psi(x,t_0) = \frac{1}{\sqrt[4]{2\pi D^2}}\exp\left[-\frac{(x-\langle x\rangle)^2}{4D^2} + \frac{i\langle p\rangle x}{\hbar}\right]$

$\begin{cases}\text{\#define A 0.032}\\ \text{\#define V0 (70.7*PI)*(70.7*PI)}\end{cases} \Leftarrow V(x,t) = \begin{cases}0 \cdots |x| > a\\ -V_0 \cdots |x| \le a\end{cases}$

【5】プログラム

新しいプロジェクト

- **ryousi（量子）**

を作成し、以下の３つのプログラムをプロジェクトに追加し、ビルドすること。

１）以下のプログラムを、[**fukusosuu.h**]として保存：

```
typedef struct tagCOMPLEX{
    double r;
    double i;
} COMPLEX;

COMPLEX add(COMPLEX z1, COMPLEX z2);
COMPLEX subtract(COMPLEX z1, COMPLEX z2);
COMPLEX multiply(COMPLEX z1, COMPLEX z2);
COMPLEX divide(COMPLEX z1, COMPLEX z2);
COMPLEX conjugate(COMPLEX z);
double absolute(COMPLEX z);
COMPLEX multiply_number(double x, COMPLEX z);
COMPLEX cexp(double re, double im);
```

２）以下のプログラムを、[**fukusosuu.c**]として保存：

該当箇所を【第７回 複素数の使用法（シラバス１２回目）】のプログラムからコピー可能

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

#include "fukusosuu.h"

// z1+z2
```

```
COMPLEX add(COMPLEX z1, COMPLEX z2)
{
    COMPLEX z;

    z.r = z1.r+z2.r;
    z.i = z1.i+z2.i;

    return z;
}

// z1-z2
COMPLEX subtract(COMPLEX z1, COMPLEX z2)
{
    COMPLEX z;

    z.r = z1.r-z2.r;
    z.i = z1.i-z2.i;

    return z;
}

// z1*z2
COMPLEX multiply(COMPLEX z1, COMPLEX z2)
{
    COMPLEX z;

    z.r = z1.r*z2.r - z1.i*z2.i;
    z.i = z1.r*z2.i + z1.i*z2.r;

    return z;
}

// z1/z2
COMPLEX divide(COMPLEX z1, COMPLEX z2)
{
    COMPLEX z;
    COMPLEX bunsi;

    if((z2.r == 0) && (z2.i == 0)){
        COMPLEX zero = {0, 0};

        printf("エラー：０で割っています¥n");

        return zero;
    }

    bunsi.r = z1.r*z2.r + z1.i*z2.i;
    bunsi.i = z1.r*z2.i - z1.i*z2.r;

    z.r = bunsi.r/(z2.r*z2.r+z2.i*z2.i);
    z.i = bunsi.i/(z2.r*z2.r+z2.i*z2.i);

    return z;
}

// z^*
COMPLEX conjugate(COMPLEX z)
```

```
{
    z.i = -z.i;

    return z;
}

// |z|
double absolute(COMPLEX z)
{
    return sqrt(z.r*z.r + z.i*z.i);
}

// x*z
COMPLEX multiply_number(double x, COMPLEX z)
{
    z.r = x*z.r;
    z.i = x*z.i;

    return z;
}

// exp(re+i*im)
COMPLEX cexp(double re, double im)
{
    COMPLEX z;

    z.r = exp(re)*cos(im);
    z.i = exp(re)*sin(im);

    return z;
}
```

３）以下のプログラムを、[**ryousi.c**]として保存

- COMPLEX calculate_k1(int m, int n)をコピーしてプログラム内に適宜追加すれば **calculate_k2** を簡単に作成できる。
- COMPLEX **calculate_k2**(int m, int n)をコピーしてプログラム内の K1 を K2, K3 に置き換えれば、calculate_k3, calculate_k4 を簡単に作成できる。

```
#define _CRT_SECURE_NO_WARNINGS

#include <windows.h>
#include <stdio.h>
#include <math.h>
#include "fukusosuu.h"

#define PI 3.1415926536
#define DX 0.004
#define DT DX*DX/4.0
#define M_MAX 500
#define M_TOTAL M_MAX+1
#define N_MAX 800
#define N_TOTAL N_MAX+1
#define H_SLASH 1
#define MASS 0.5
#define D 0.035
#define X_KAKKO -0.3
```

```
#define P_KAKKO 50.0*PI
#define A 0.032
#define V0 (70.7*PI)*(70.7*PI)

COMPLEX Psi[N_TOTAL][M_TOTAL];
COMPLEX K1[M_TOTAL];
COMPLEX K2[M_TOTAL];
COMPLEX K3[M_TOTAL];
COMPLEX K4[M_TOTAL];
double V[M_TOTAL];
COMPLEX AI = {0, 1};

double getX(int m)
{
    double x = (m-M_MAX/2.0)*DX;

    return x;
}

double getT(int n)
{
    double t = n*DT;

    return t;
}

COMPLEX hasoku(int m)
{
    COMPLEX z;
    double x = getX(m);
    double re, im;

    re = -(x-X_KAKKO)*(x-X_KAKKO)/4.0/D/D;
    im = P_KAKKO*x/H_SLASH;

    z = cexp(re, im);

    z = multiply_number(1.0/pow(2.0*PI*D*D, 1.0/4.0), z);

    return z;
}

double getV(int m)
{
    double abs_x;
    double x = getX(m);

    abs_x = fabs(x);

    if(abs_x > A){
        return 0.0;
    }
    else{
        return -V0;
    }
}
//最初に
```

```
void saishoni(void)
{
    int m, n;
    COMPLEX c_zero = {0.0, 0.0};

    for(n=0; n<=N_MAX; n++){
        // n=0      n=M_TOTAL
        Psi[n][0] = Psi[n][M_MAX] = c_zero;
    }

    for(m=1; m<=M_MAX-1; m++){
        Psi[0][m] = hasoku(m);
    }

    // m=0からm=M_MAXまで
    for(m=0; m<=M_MAX; m++){
        V[m] = getV(m);
    }
}

COMPLEX calculate_k1(int m, int n)
{
    COMPLEX z, w;

    z = multiply_number(2.0, Psi[n][m]);
    z = subtract(Psi[n][m+1], z);
    z = add(Psi[n][m-1], z);
    z = multiply_number(H_SLASH*H_SLASH/2.0/MASS/DX/DX, z);

    w = multiply_number(V[m], Psi[n][m]);
    z = subtract(z, w);

    z = multiply(AI, z);
    z = multiply_number(1.0/H_SLASH, z);

    return z;
}

COMPLEX calculate_psi_plus_k(COMPLEX psi, double r, COMPLEX k)
{
    COMPLEX z;

    z = multiply_number(r*DT, k);
    z = add(psi, z);

    return z;
}

COMPLEX calculate_k2(int m, int n)
{
    COMPLEX z, w, v;

    w = calculate_psi_plus_k(Psi[n][m], 1.0/2.0, K1[m]);
    v = multiply_number(V[m], w);
    z = multiply_number(2.0, w);

    w = calculate_psi_plus_k(Psi[n][m+1], 1.0/2.0, K1[m+1]);
```

```
    z = subtract(w, z);

    w = calculate_psi_plus_k(Psi[n][m-1], 1.0/2.0, K1[m-1]);
    z = add(w, z);

    z = multiply_number(H_SLASH*H_SLASH/2.0/MASS/DX/DX, z);
    z = subtract(z, v);

    z = multiply(AI, z);
    z = multiply_number(1.0/H_SLASH, z);

    return z;
}

COMPLEX calculate_k3(int m, int n)
{
    COMPLEX z, w, v;

    w = calculate_psi_plus_k(Psi[n][m], 1.0/2.0, K2[m]);
    v = multiply_number(V[m], w);
    z = multiply_number(2.0, w);

    w = calculate_psi_plus_k(Psi[n][m+1], 1.0/2.0, K2[m+1]);
    z = subtract(w, z);

    w = calculate_psi_plus_k(Psi[n][m-1], 1.0/2.0, K2[m-1]);
    z = add(w, z);

    z = multiply_number(H_SLASH*H_SLASH/2.0/MASS/DX/DX, z);
    z = subtract(z, v);

    z = multiply(AI, z);
    z = multiply_number(1.0/H_SLASH, z);

    return z;
}

COMPLEX calculate_k4(int m, int n)
{
    COMPLEX z, w, v;

    w = calculate_psi_plus_k(Psi[n][m], 1.0, K3[m]);
    v = multiply_number(V[m], w);
    z = multiply_number(2.0, w);

    w = calculate_psi_plus_k(Psi[n][m+1], 1.0, K3[m+1]);
    z = subtract(w, z);

    w = calculate_psi_plus_k(Psi[n][m-1], 1.0, K3[m-1]);
    z = add(w, z);

    z = multiply_number(H_SLASH*H_SLASH/2.0/MASS/DX/DX, z);
    z = subtract(z, v);

    z =multiply(AI, z);
    z = multiply_number(1.0/H_SLASH, z);
```

```
    return z;
}

COMPLEX runge_kutta(int m, int n)
{
    COMPLEX z, w;

    w = multiply_number(2.0, K2[m]);
    z = add(K1[m], w);
    w = multiply_number(2.0, K3[m]);
    z = add(z, w);
    z = add(z, K4[m]);

    z = multiply_number(DT/6.0, z);

    z = add(Psi[n][m], z);

    return z;
}

void calculate_psi(int n)
{
    int m;

    for(m=1; m<=M_MAX-1; m++){
        K1[m] = calculate_k1(m,n);
    }

    for(m=1; m<=M_MAX-1; m++){
        K2[m] = calculate_k2(m,n);
    }

    for(m=1; m<=M_MAX-1; m++){
        K3[m] = calculate_k3(m,n);
    }

    for(m=1; m<=M_MAX-1; m++){
        K4[m] = calculate_k4(m,n);
    }

    for(m=1; m<=M_MAX-1; m++){
        Psi[n+1][m] = runge_kutta(m,n);
    }
}

void writetitle(FILE *file_open, int n)
{
    printf(" ,時刻N=%d¥n", n);

    if(file_open != NULL){
        fprintf(file_open, " ,時刻N=%d¥n", n);
    }
}

void writedata(FILE *file_open, int m, double x)
{
    printf("%f, %10.3e¥n", getX(m), x);
```

```
    if(file_open != NULL){
        fprintf(file_open, "%f, %10.3e¥n", getX(m), x);
    }
}

int main(void)
{
    // 作成ファイル名はryousi.csvなど
    char name[] = "ryousi.csv";
    // ryousi.csvに時刻のマーク追加用
    char filename[100];
    // 作成ファイル追尾用変数
    FILE * file_open = NULL;
    double x;
    int m, n, n_out = N_MAX/4;

    wsprintf(filename, "%d%s", n_out, name);

    // ファイルの作成に失敗すると数値(file_open=NULL)を返す
    file_open = fopen(filename, "wt");
    writetitle(file_open, n_out);

    saishoni();

    for(n=0; n<=N_MAX-1; n++){
        calculate_psi(n);
    }

    n = n_out;

    for(m=0; m<=M_MAX; m++){
        // |φ(x,t)|
        x = absolute(Psi[n][m]);
        // |φ(x,t)|^2(=x*x)
        writedata(file_open, m, x*x);
    }

    // NULLのときにファイル作成失敗
    if(file_open == NULL){
        printf("¥n====>ファイル(%s)の作成に失敗しています。¥n====>既にファイルが
開かれているかもしれません。", filename);

        getchar();
    }
    else{
        HINSTANCE hinst;

        fclose(file_open);

        hinst = ShellExecute((HWND)0, "open", filename, NULL, NULL, SW_SHOWNORMAL);

        if(hinst <= (HINSTANCE)32){
            printf("¥n====>表計算ソフトの自動起動に失敗しました。");

            getchar();
        }
```

```
        }

        return 0;
    }
```

第８回目レポート

レポートの回数(８回目)、学生証番号と氏名を明記すること

必ず表紙を付け、最後のページ(の添付用)を表紙の次に入れる

（Ａ４レポート用紙使用のこと）

1） ３つのプログラム(fukusosuu.h, fukusosuu.c, ryousi.c)において、

- ３つプログラム
- プログラムを実行し、計算結果($x-\left|\psi(x,t)\right|$)データを出力して、次のｎの場合の5つグラフ(n_out = **N_MAX/4** の箇所を変更します)
  - n=0
  - n=**N_MAX/4**
  - n=N_MAX/2
  - n=3*N_MAX/4
  - n=N_MAX

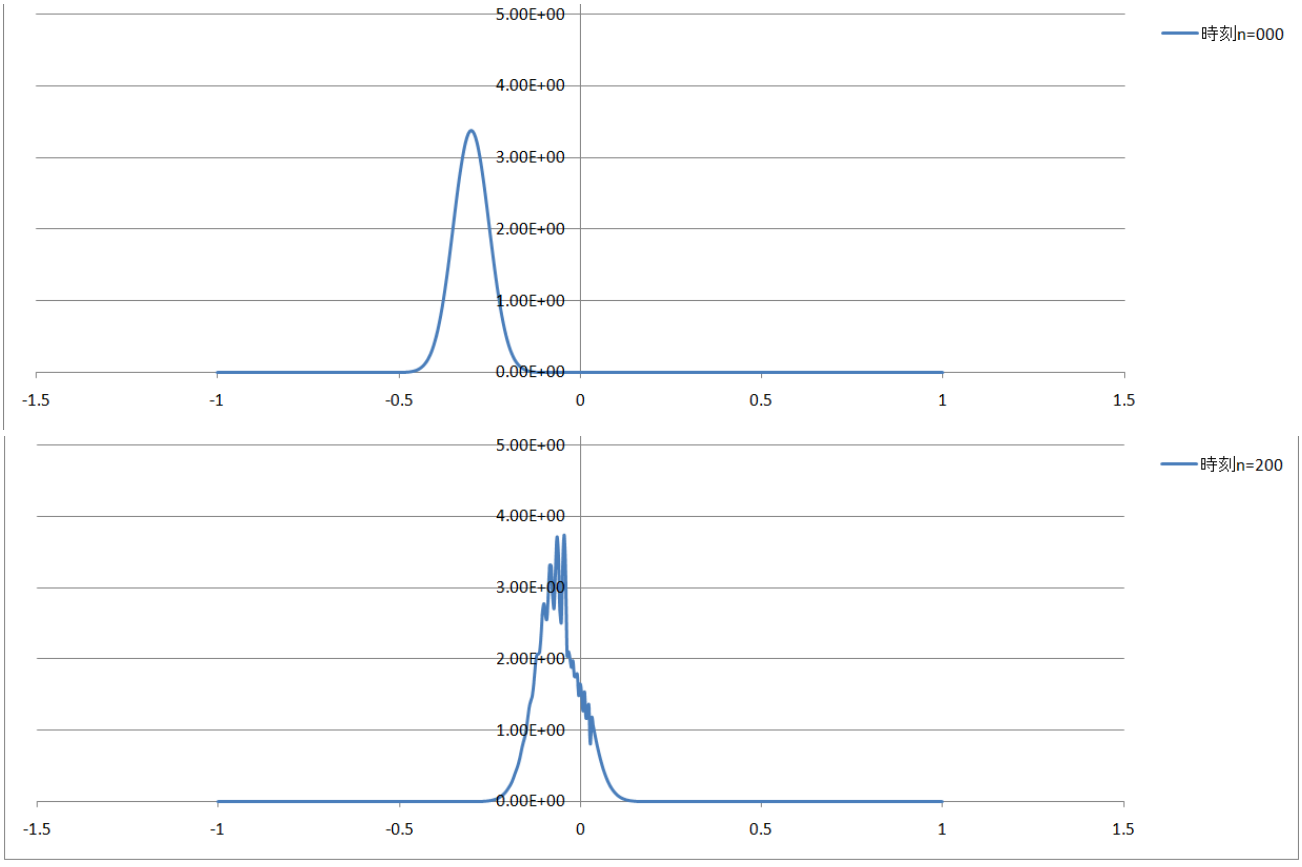
の7点です。グラフ作成時は、

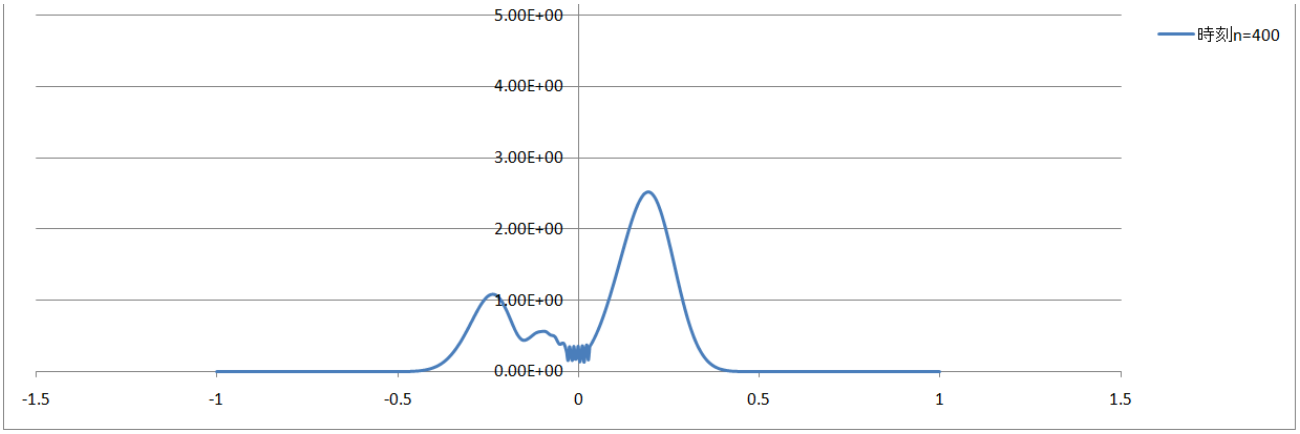
- **縦軸**の最大値を５にして表示する。
- [**グラフの移動**]を”新しいシート”にする。

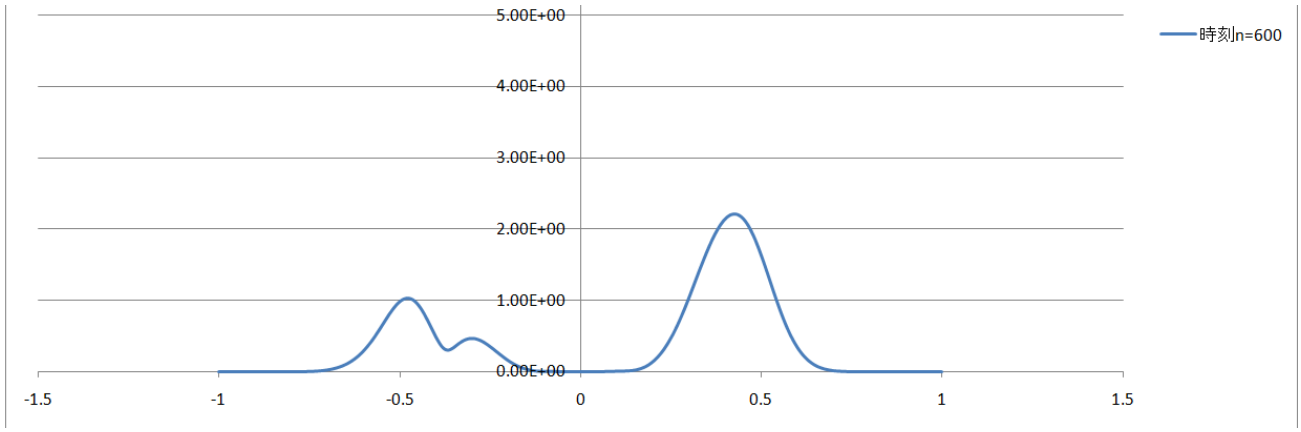
を実行する事。

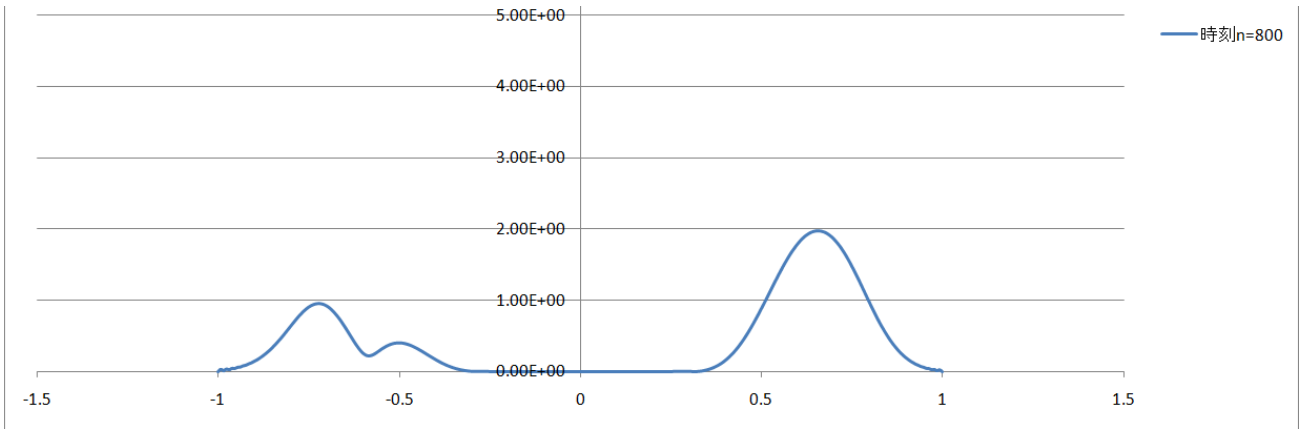
2） グラフからは、最初１つの波束しかないが、ポテンシャルを通過すると２つの波束に分かれてゆく様子が読み取れる。1つの量子をポテンシャルにぶつけると2つの量子に分裂するという事だろうか？この結果を説明せよ。

グラフは以下のようになります。

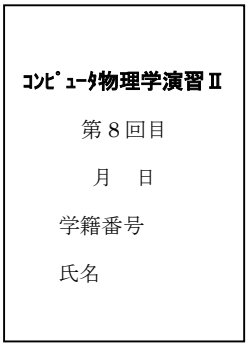



5.00E+00
4.00E+00
3.00E+00
2.00E+00
1.00E+00
0.00E+00
-1.5
-1
-0.5
0
0.5
1
1.5
時刻n=400


5.00E+00
4.00E+00
3.00E+00
2.00E+00
1.00E+00
0.00E+00
-1.5
-1
-0.5
0
0.5
1
1.5
時刻n=600


5.00E+00
4.00E+00
3.00E+00
2.00E+00
1.00E+00
0.00E+00
-1.5
-1
-0.5
0
0.5
1
1.5
時刻n=800

提出例(使える表紙は次のページ)



１ページ目

２ページ目

３ページ目以降

コンピュータ物理学演習 II

第８回目

月　日提出

学籍番号

氏名

第８回目レポート（添付用）

1）３つのプログラム(fukusosuu.h, fukusosuu.c, ryousi.c)において、

- ３つプログラムを作成
- プログラムを実行し、計算結果($x-\left|\psi\left(x,t\right)\right|$)データを出力して、次のｎの場合の5つグラフ(n_out = **N_MAX/4**の箇所を変更します)を作成
  - n=0
  - n=N_MAX/4
  - n=N_MAX/2
  - n=3*N_MAX/4
  - n=N_MAX

の合計７点です。グラフ作成時は、

- **縦軸**の最大値を 10 にして表示する。
- [**グラフの移動**]を”新しいシート”にする。

を実行する事。

2）グラフからは、最初１つの波束しかないが、ポテンシャルを通過すると２つの波束に分かれてゆく様子が読み取れる。1つの量子をポテンシャルにぶつけると2つの量子に分裂し、最後は ２ つの量子が残るという事だろうか？この結果を説明せよ。