Reproduction of the double-slit experiment through particle simulation and theoretical analysis

Shokin Shigemitsu
Department of Gastroenterology. Ushiku Aiwa General Hospital, Ushiku City, Ibaraki prefecture, Japan
Corresponding auther : Shokin Shigemitsu

Abstract
Double-slit experiments form the cornerstone of quantum theory.
This study simulated the movement of 100,000 particles individually based on a straightforward assumption.
We assumed that the movement of a particle is influenced by the particle preceding it.
Consequently, the simulation successfully reproduced the single-slit and double-slit experiment results.

Main text

Introduction: Double-slit experiments involving particles are fundamental to quantum theory.(1)
This study aims to reproduce the results of double-slit experiments through simulations, employing simple assumptions and computational programming.

Method: Particles in the simulation traversed slits individually, adhering to specific conditions during their passage.
・A moving particle is influenced by the particle preceding it. The horizontal distance traversed by the preceding particle, measured from the starting point of the current particle, is designated as r. The displacement, representing the horizontal shift ⊿x of the following particle, is inversely proportional to r. This process was repeated 100,000 times, and the resulting distribution was plotted.

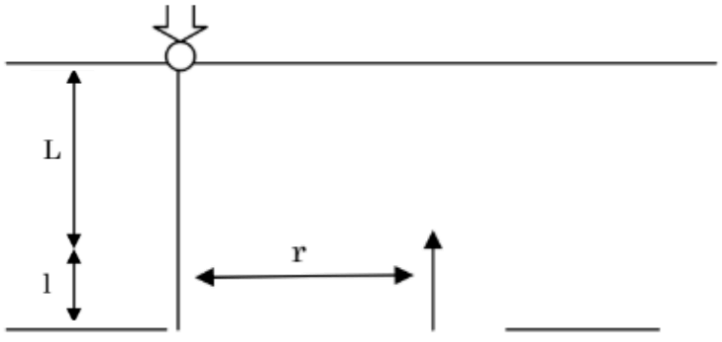


FIGURE1

$\Delta x = \frac{A}{r}$ (1)

In the simulation, *r* was treated as a continuous variable,
A was assumed to be constant for single-slit experiments and to have two conditional values for double-slit experiments.
Additionally, *A*, assuming the particle to be an electron, is defined as:

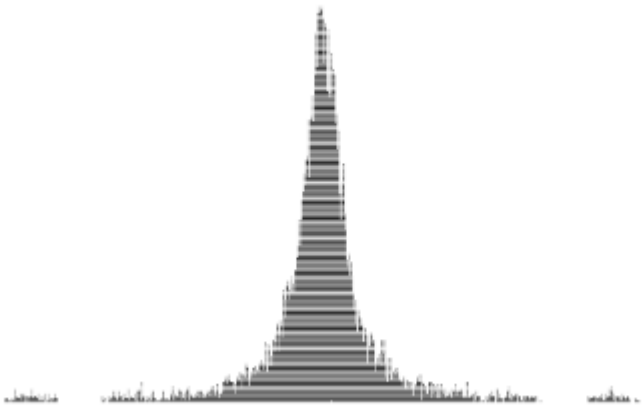
$$A = \frac{\mu e^2 nLl}{2\pi m} = const \quad (2)$$

where *n* is a constant. Its value is constant for single-slit experiments, while it takes two distinct values depending on whether the particles follow the same slit as its predecessor or a different one.

μ: magnetic permeability
e: electron charge
π: pi
m: mass of electron
L: slit-specific constant
l: slit-specific constant (smaller unit than L)
n: slit-specific constant (two values)
μ: $\mu=1.26\times10^{-6}$ [H/m]
e: $e=1.6\times10^{-19}$ [c]
π: 3.14
m: 9.1×10^{-31} [kg]
L: 1
l: 1
n: Single slit: 1000
Double slit: 1000 or 500000
It remains constant in single-slit experiments. In double-slit experiments, it varies depending on the relationship between successive particle paths.

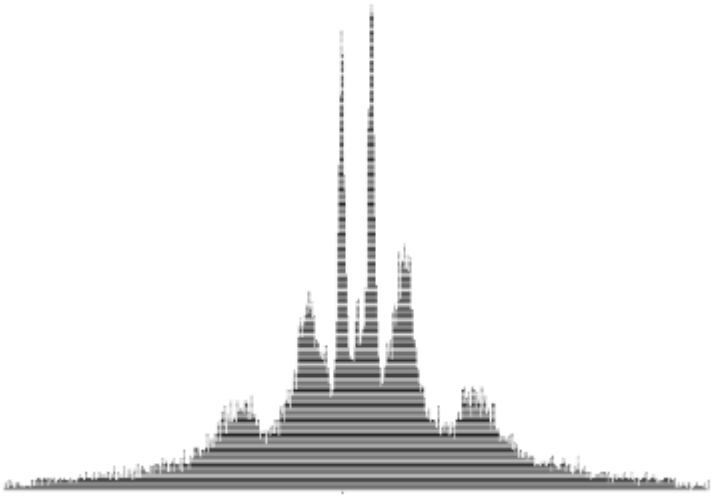
Result:

1. Single slit
Under the given assumptions, electrons reached a distance of 15 μm.



Graph 1

2. Double slit
Under the given assumptions, electrons reached a distance of 60 μm.



Graph 2

Consideration:
The simulation was conducted under the assumption that the trajectory of each particle was influenced by the path of the preceding particle.

We assumed the particles to be electrons, and the analysis was grounded in electromagnetic theory. (2)(3)
The concept of time was incorporated into the magnetic field generated using the Biot–Savart law.
Magnetic-field superposition was considered to occur only near the slit.

According to the Biot-Savart law, the magnetic field produced by a current flowing through a straight line can be expressed as:

$B = \frac{\mu I}{2\pi r}$ (3)

Substituting $I = env_0$

$B = \frac{\mu env_0}{2\pi r}$ (4)

The Lorentz force acting on an electron due to this magnetic field is given by:

$$F = ev_1B = \frac{\mu e^2 n v_0 v_1}{2\pi r} \quad (5)$$

We approximated the force acting on the particle to simplify the equations.
All particles were assumed to have identical velocities, i.e., $v_1 = v_0$.
Additionally, we assumed negligible l when L>>l.
Moreover, the L section was assumed to move horizontally at a constant velocity.

$$v = \frac{l}{v_0} \times \frac{F}{m} \quad (6)$$

$$t = \frac{l+L}{v_0} \quad (7)$$

Its horizontal velocity and transit time were v and t, respectively.
Substituting $\Delta x = v \times t$ yielded:

$$\Delta x = \frac{\mu e^2 nl(l+L)}{2\pi m} \times \frac{1}{r} \quad (8)$$

Finally, we obtained the formula used in the simulation by assuming L>>l.
l was assumed to differ based on the slit traversed by the particle for double-slit experiments, making n a binary variable.

$$I = \int\int \Delta x dx dt = \left| \int \psi(x) dx \right|^2 \quad (9)$$

The motions of electrons within the slit were superimposed over time, resulting in the final equation.

Conclusion

A double-slit experiment was simulated under the assumption that the trajectory of each particle was influenced by the preceding particle. The results were successfully reproduced using a straightforward definition and computational simulation.

Acknowledgement

We would like to thank Editage (www.editage.jp) for English language editing.

References

1. Feynman et al., The Feynman lectures on physics, Addition-Wesley reading, Volume 3 Ch.1 (1966)
2. Jönsson C, Electron diffraction at multiple slits, American Journal of Physics, Volume 42, Issue1 4-11 (1974, January)
3. Merli, P. G., Missiroli, G. F. & Pozzi, G, On the statistical aspect of electron interference phenomena, American Journal of Physics, Volume 44, Issue 3, 306–307 (1976, March)

【source program : single slit】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
ATOM InitApp(HINSTANCE);
BOOL InitInstance(HINSTANCE, int);

LPCTSTR lpszClassName = TEXT("win01");

int WINAPI WinMain(HINSTANCE hCurInst, HINSTANCE hPreInst,
   LPSTR lpsCmdLine, int nCmdShow)

{
  MSG msg;
  BOOL bRet;

  if (!InitApp(hCurInst))
    return FALSE;
  if (!InitInstance(hCurInst, nCmdShow))
    return FALSE;
  while ((bRet = GetMessage(&msg, NULL, 0, 0)) !=0){
    if(bRet == -1){
       break;
    }else {
      TranslateMessage(&msg);
      DispatchMessage(&msg);
      }
    }
    return (int)msg.wParam;
}

ATOM InitApp(HINSTANCE hInst)
{
  WNDCLASSEX wc;
  wc.cbSize = sizeof(WNDCLASSEX);
  wc.style = CS_HREDRAW | CS_VREDRAW;
  wc.lpfnWndProc = WndProc;
  wc.cbClsExtra = 0;
  wc.cbWndExtra = 0;
  wc.hInstance = hInst;
  wc.hIcon = (HICON)LoadImage(NULL, MAKEINTRESOURCE(IDI_APPLICATION),
    IMAGE_ICON, 0, 0, LR_DEFAULTSIZE | LR_SHARED);
  wc.hCursor = (HCURSOR)LoadImage(NULL, MAKEINTRESOURCE(IDC_ARROW),
```

```
    IMAGE_CURSOR, 0, 0, LR_DEFAULTSIZE | LR_SHARED);
  wc.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
  wc.lpszMenuName = NULL;
  wc.lpszClassName = lpszClassName;
  wc.hIconSm = (HICON)LoadImage(NULL,MAKEINTRESOURCE(IDI_APPLICATION),
    IMAGE_ICON, 0, 0, LR_DEFAULTSIZE | LR_SHARED);

  return (RegisterClassEx(&wc));
}

BOOL InitInstance(HINSTANCE hInst, int nCmdShow)
{
  HWND hWnd;

  hWnd = CreateWindow(lpszClassName,
    TEXT("single-slit"), WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
    NULL, NULL, hInst, NULL);
    if (!hWnd)
       return FALSE;
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);
  return TRUE;
}

LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp)
{
  int X[1990];
  int l=1990;
  int q=10000;
  double V1=1;
  double V2=1;
  double pi=3.14159265358979323846;
  double e=1.6*pow(10.0,-19.0);
  double m=9.1*pow(10.0,-31.0);
  double u=1.256637*pow(10.0,-6.0);
  double k;
  double g;
  double L=1;
  double r;
  double z;
  int f;
  int h;
  int i;
  int x;
  int y;
  srand((unsigned int) time(0));
  k=e*1000000/(2*pi);
```

```
g=u*e*k*L*L*V1/(2*m*V2)*pow(10.0,14.0);
int a;
int b;
for(i = 0; i < l; i++){
       X[i]=250;
    }
  do{
  x = rand() * l/ (1.0 + RAND_MAX);
  }while(x<950||1000<x);
  h=x;
  X[x]--;

HDC hdc;
PAINTSTRUCT ps;

switch (msg){
  case WM_PAINT:
   hdc = BeginPaint(hWnd, &ps);

for(i = 0; i < q; i++){
 do{
  do{
  x = rand() * l/ (1.0 + RAND_MAX);
  }while(x<950||1000<x);
  }while(x==h);

  r=x-h;
  z=-g/r;
  y=z/500;

  f=x+y;
  X[f]--;
  h=f;

         a=f-500;
         b=2*X[f];
  SetPixel(hdc,a,b,RGB(0,0,0));
}

     EndPaint(hWnd, &ps);
     break;

  case WM_DESTROY:
    PostQuitMessage(0);
    break;
  default:
    return (DefWindowProc(hWnd, msg, wp, lp));
  }
```

```
    return 0;
}
```

【source program : double slit】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
ATOM InitApp(HINSTANCE);
BOOL InitInstance(HINSTANCE, int);

LPCTSTR lpszClassName = TEXT("win01");

int WINAPI WinMain(HINSTANCE hCurInst, HINSTANCE hPreInst,
    LPSTR lpsCmdLine, int nCmdShow)

{
  MSG msg;
  BOOL bRet;

  if (!InitApp(hCurInst))
    return FALSE;
  if (!InitInstance(hCurInst, nCmdShow))
    return FALSE;
  while ((bRet = GetMessage(&msg, NULL, 0, 0)) !=0){
    if(bRet == -1){
        break;
    }else {
       TranslateMessage(&msg);
       DispatchMessage(&msg);
       }
    }
    return (int)msg.wParam;
}

ATOM InitApp(HINSTANCE hInst)
{
  WNDCLASSEX wc;
  wc.cbSize = sizeof(WNDCLASSEX);
  wc.style = CS_HREDRAW | CS_VREDRAW;
  wc.lpfnWndProc = WndProc;
  wc.cbClsExtra = 0;
  wc.cbWndExtra = 0;
```

```
  wc.hInstance = hInst;
  wc.hIcon = (HICON)LoadImage(NULL, MAKEINTRESOURCE(IDI_APPLICATION),
    IMAGE_ICON, 0, 0, LR_DEFAULTSIZE | LR_SHARED);
  wc.hCursor = (HCURSOR)LoadImage(NULL, MAKEINTRESOURCE(IDC_ARROW),
    IMAGE_CURSOR, 0, 0, LR_DEFAULTSIZE | LR_SHARED);
  wc.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
  wc.lpszMenuName = NULL;
  wc.lpszClassName = lpszClassName;
  wc.hIconSm = (HICON)LoadImage(NULL,MAKEINTRESOURCE(IDI_APPLICATION),
    IMAGE_ICON, 0, 0, LR_DEFAULTSIZE | LR_SHARED);

  return (RegisterClassEx(&wc));
}

BOOL InitInstance(HINSTANCE hInst, int nCmdShow)
{
  HWND hWnd;

  hWnd = CreateWindow(lpszClassName,
    TEXT("double-slit"), WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
    NULL, NULL, hInst, NULL);
    if (!hWnd)
       return FALSE;
    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);
  return TRUE;
}

LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp)
{
  int X[5990];
  int l=5990;
  int q=100000;
  double V1=1;
  double V2=1;
  double pi=3.14159265358979323846;
  double e=1.6*pow(10.0,-19.0);
  double m=9.1*pow(10.0,-31.0);
  double u=1.256637*pow(10.0,-6.0);
  double k;
  double g;
  double L=1;
  double r;
  double z;
  int f;
  int h;
  int i;
```

```
int x;
int y;
srand((unsigned int) time(0));
k=e*500000/(2*pi);
g=u*e*k*L*L*V1/(m*V2)*pow(10.0,14.0);
int a;
int b;
for(i = 0; i < l; i++){
       X[i]=250;
   }
  do{
  x = rand() * l/ (1.0 + RAND_MAX);
  }while(x<2850||2900<x&&x<3100||3150<x);
  h=x;
  X[x]++;

HDC hdc;
PAINTSTRUCT ps;

switch (msg){
  case WM_PAINT:
   hdc = BeginPaint(hWnd, &ps);

for(i = 0; i < q; i++){
 do{
  do{
  x = rand() * l/ (1.0 + RAND_MAX);
  }while(x<2850||2900<x&&x<3100||3150<x);
 }while(x==h);

  r=x-f;
  z=-g/r;

 if(r<-100||100<r){
  y=z;
  f=x+y;
  X[f]--;
 }
 else{
  y=z/500;
  f=x+y;
  X[f]--;
 }
  h=f;

         a=f/10+100;
         b=2*X[f];
  SetPixel(hdc,a,b,RGB(0,0,0));
```

```
  }

      EndPaint(hWnd, &ps);
      break;

    case WM_DESTROY:
      PostQuitMessage(0);
      break;
    default:
      return (DefWindowProc(hWnd, msg, wp, lp));
    }
    return 0;
}
```