

# 第一次计算物理作业

杨梓鑫 学号: 2010301020023

March 27, 2013

\*1.5. Consider again a decay problem with two types of nuclei  $A$  and  $B$ , but now suppose that nuclei of type  $A$  decay into ones of type  $B$ , while nuclei of type  $B$  decay into ones of type  $A$ . Strictly speaking, this is not a “decay” process, since it is possible for the type  $B$  nuclei to turn back into type  $A$  nuclei. A better analogy would be a resonance in which a system can tunnel or move back and forth between two states  $A$  and  $B$  which have equal energies. The corresponding rate equations are

$$\frac{dN_A}{dt} = \frac{N_B}{\tau} - \frac{N_A}{\tau} \quad (1)$$

$$\frac{dN_B}{dt} = \frac{N_A}{\tau} - \frac{N_B}{\tau} \quad (2)$$

where for simplicity we have assumed that the two types of decay are characterized by the same time constant,  $\tau$ . Solve this system of equations for the numbers of nuclei,  $N_A$  and  $N_B$ , as functions of time. Consider different initial conditions, such as  $N_A = 100$ ,  $N_B = 0$ , etc., and take  $\tau = 1$ s. Show that your numerical results are consistent with the idea that the system reaches a steady state in which  $N_A$  and  $N_B$  are constant. In such a steady state, the time derivative  $dN_A/dt$  and  $dN_B/dt$  should vanish.

*Solution:*

The analytical solution goes like:

Since the total nuclei number  $N = N_A + N_B$  is a constant,

$$\begin{aligned} \frac{dN_A}{dt} &= \frac{N - N_A}{\tau} - \frac{N_A}{\tau} \\ \Rightarrow \frac{dN_A}{dt} &= \frac{N}{\tau} - 2\frac{N_A}{\tau} \\ \Rightarrow N_A(t) &= \frac{N}{2} + \frac{N}{2}e^{-\frac{2}{\tau}t} \end{aligned}$$

And

$$N_B(t) = \frac{N}{2} - \frac{N}{2}e^{-\frac{2}{\tau}t}$$

with the initial value  $N_A(0) = N$ ,  $N_B(0) = 0$

To run the resonance reaction numerically, the code of C++ programming will be written as:

```

1 #include <iostream>
2 #include "TGraph.h"
3 #include "TAxis.h"
4
5 double *reactionTime = NULL;
6 double *ParticleNum_A = NULL;
7 double *ParticleNum_B = NULL;
8
9
10 using namespace std;
11
12 void resonance(int numOfSteps, int N_A, int N_B, double tau)
13 {
14     double dt = 0.05;
15     int turnedNum = 0;
16     reactionTime = new double[numOfSteps];
17     ParticleNum_A = new double[numOfSteps];
18     ParticleNum_B = new double[numOfSteps];
19     for (int i = 0; i < numOfSteps ; i++)
20     {
21         reactionTime[i] = i * dt;
22         ParticleNum_A[i] = N_A;
23         ParticleNum_B[i] = N_B;
24         turnedNum = (N_B / tau - N_A / tau) * dt;
25         N_A += turnedNum;
26         N_B -= turnedNum;
27         turnedNum = (N_A / tau - N_B / tau) * dt;
28         N_B += turnedNum;
29         N_A -= turnedNum;
30         cout << reactionTime[i] << " " << N_A << " " <<
31             N_B << endl;
32     }
33 }
34
35
36
37 void plot()
38 {
39     int A, B, numOfSteps;
40     double tau;
41     cout<<"Initial number of type A = ";
42     cin>>A;
43     cout<<"Initial number of type B = ";
44     cin>>B;
45     cout<<"Time constant tau = ";
46     cin>>tau;
47     cout<<"The times of resonance reaction = ";
48     cin>>numOfSteps;
49
50     resonance(numOfSteps, A, B, tau);
51
52
53     TGraph* gr1 = new TGraph(numOfSteps, reactionTime,
54         ParticleNum_A);
55     TGraph* gr2 = new TGraph(numOfSteps, reactionTime,
56         ParticleNum_B);
57     // TCanvas* c1 = new TCanvas("c1", "Evolution of Resonance
58         between nuclei A and B", 200, 10, 600, 400);
59     gr1->GetXaxis()->SetTitle("Reaction Time");
60     gr1->GetYaxis()->SetTitle("Numbers of the remnant nuclei
61         ");

```

```
58     gr1->SetTitle("Nuclei Resonance between A and B");
59     gr1->SetMarkerStyle(22);
60     gr1->SetMarkerSize(1);
61     gr1->SetMarkerColor(kBlue);
62     gr1->Draw("AC*");
63
64     gr2->SetLineWidth(3);
65     gr2->SetMarkerStyle(21);
66     gr2->SetLineColor(2);
67     gr2->Draw("CP");
68 }
```

When running in *ROOT*, the above program allows us to input the initial values of  $N_A, N_B, \tau$ , and the times to resonant. By adjusting those values, we can observe the behavior of the system and check if it fits the analytical solution.

A rough glimpse of the figures tells us the symmetry along the  $Num = \frac{N}{2}$  line of  $N_A(t)$  and  $N_B(t)$ , and the resemblance to the exponential function. First, let's explore how the initial value of  $N_A$  will affect the system.

From Figure 1 to Figure 4, we set the initial  $N_A$  from 10000 to 50. As the figures shown, the larger of  $N_A$ , the better the numerical simulation fits the analytical result. If we want to enhance the fitness of smaller  $N_A$ , the infinitesimal time interval  $dt$  should be narrower than the present setting, too. In the figures, black curve with blue \* represents the value of  $N_B(t)$ , and the other one, red curve with black square represents the value of  $N_A(t)$ . The different parameter *Step* is approximately about the final steady time.

To explain the discrepancy of the final steady values of  $N_A(t)$  and  $N_B(t)$ , we have to notice that in Code line 19 - 31, we always run equation (1) first, and equation (2) next, which makes the value of  $N_B(t)$  always smaller than  $N_A(t)$  because  $dt$  is not infinitesimal and the equations happen at the same time in the reality. And this also shows why the smaller the initial  $N_A(t)$  i.e., the total number  $N$  is, the larger the difference between the  $N_A(t)$  and  $N_B(t)$  is. To fix it, we still have to decrease the value of  $dt$ .

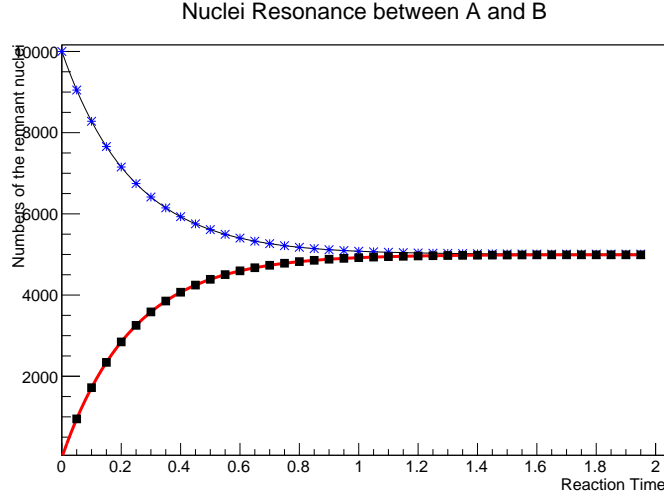


Figure 1:  $N_A = 10000, N_B = 0, \tau = 1, Step = 40$

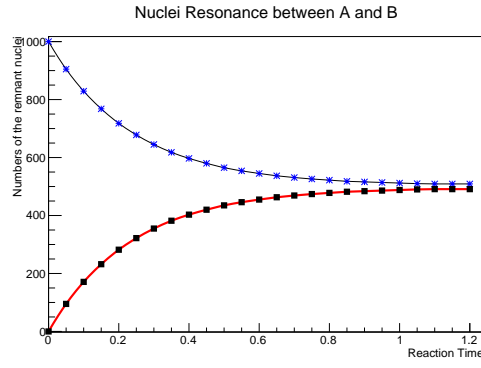


Figure 2:  $N_A = 1000, N_B = 0, \tau = 1, Step = 25$

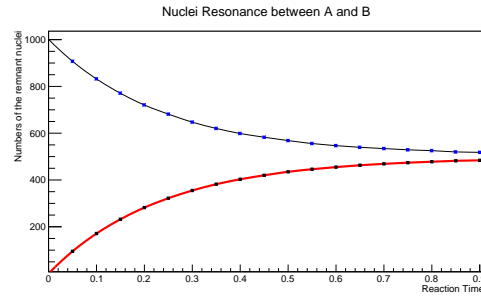


Figure 3:  $N_A = 100, N_B = 0, \tau = 1, Step = 30$

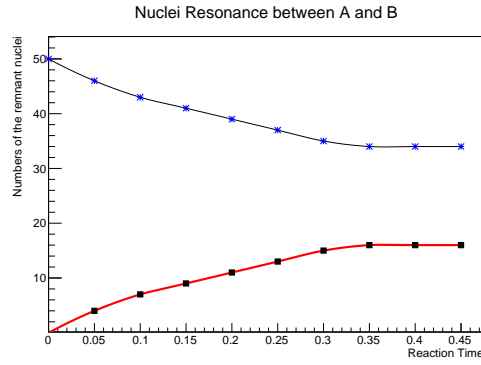


Figure 4:  $N_A = 50, N_B = 0, \tau = 1, Step = 10$

From Figure 5 to Figure 10, we explored the effect of the value of  $\tau$  on the system, by setting  $\tau$  from 0.2 to 10, and the total number  $N$  is fixed at 1000. As we can see from the equations, the time constant  $\tau$  mainly decide the time needed to be steady, which is represented by *Step* here, and the behavior of the exponential function, which is represented by the curvature of the curve here.

And the features of the following figures are consistent with the above discussion about  $\tau$ . With the increase of  $\tau$ , the *Step* is increasing and the curve is less and less curly. Also, the difference between the final steady values of  $N_A(t)$  and  $N_B(t)$  is increasing, which can still blame on the largeness of  $dt$  compare to  $\frac{1}{\tau}$  in the equations.

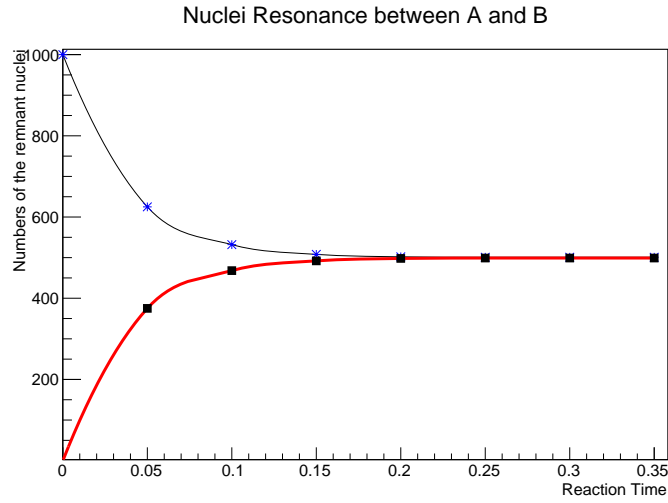


Figure 5:  $N_A = 1000, N_B = 0, \tau = 0.2, Step = 8$

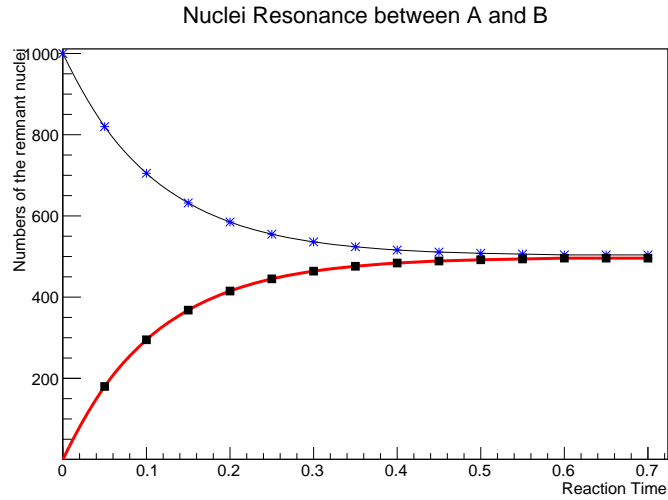


Figure 6:  $N_A = 1000, N_B = 0, \tau = 0.5, Step = 15$

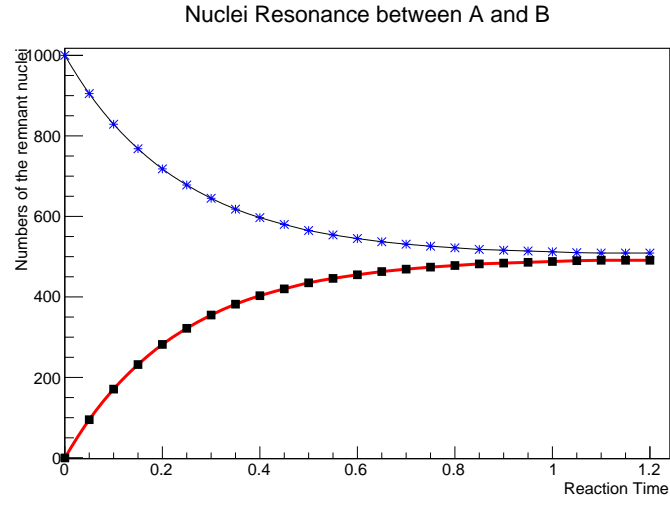


Figure 7:  $N_A = 1000, N_B = 0, \tau = 1, Step = 25$

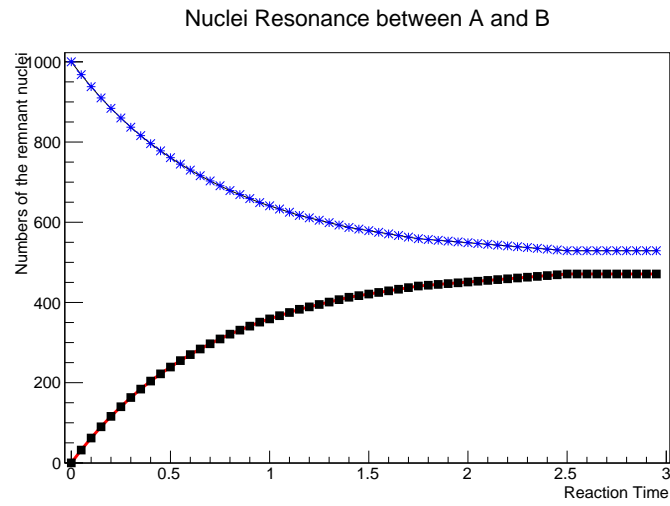


Figure 8:  $N_A = 1000, N_B = 0, \tau = 3, Step = 60$

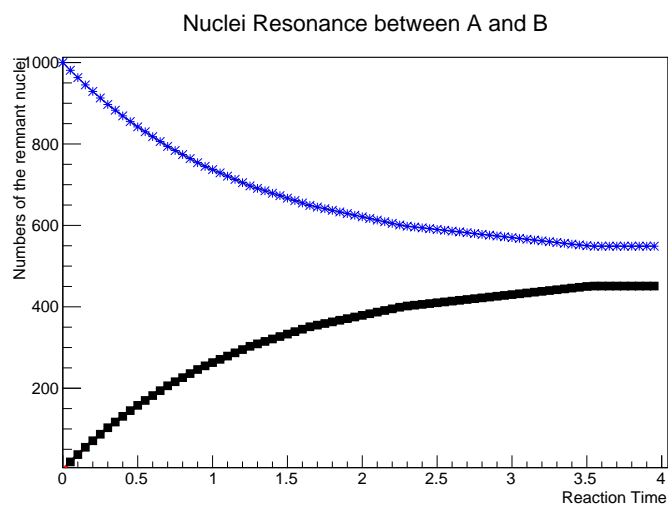


Figure 9:  $N_A = 1000, N_B = 0, \tau = 5, Step = 80$

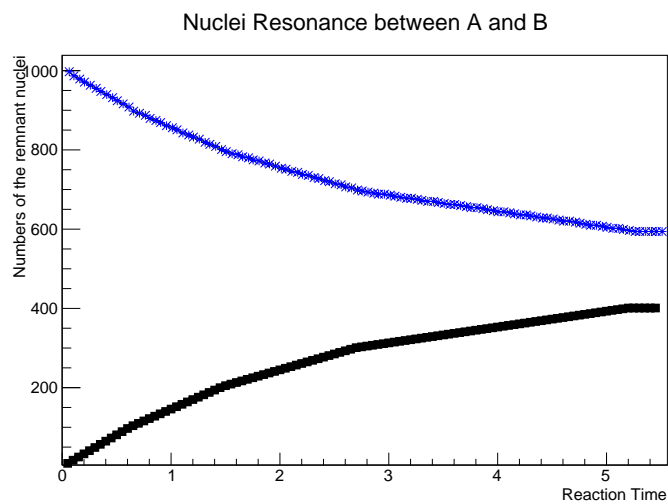


Figure 10:  $N_A = 1000, N_B = 0, \tau = 10, Step = 110$