

Duffing 方程的 Matlab 实现

张昊鹏

导师：路宽

2021 年 5 月 31 日

目录

1	问题的引入	2
1.1	一维弹簧质量块模型	2
1.2	伪二维 Jeffcott 转子模型	2
2	求解方法	3
2.1	近似求解方法——谐波平衡法	3
2.2	数值求解方法——龙格库塔方法	4
3	近似解析解	5
3.1	谐波平衡解的推导	5
3.2	编程实现	6
3.3	结果分析	8
4	数值解	10
4.1	编程实现	10
4.2	结果分析	13

1 问题的引入

无论是在理论研究还是在工程应用中，Duffing 方程都具有广泛的应用。数学、力学、电学领域的学者都对 Duffing 方程进行了深入的研究。本节简要介绍了具有 Duffing 型刚度的一维弹簧质量块模型、伪二维 Jeffcott 转子模型以引出 Duffing 方程。

1.1 一维弹簧质量块模型

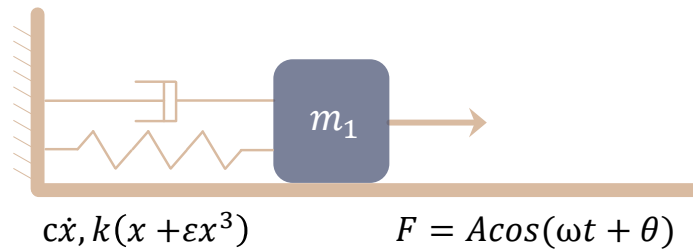


图 1: 具有 Duffing 型刚度和粘性阻尼的一维弹簧质量块模型

图.1 是具有 Duffing 型刚度和粘性阻尼的一维弹簧质量块模型，其无量纲形式的动力学微分方程为：

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2(x + \varepsilon x^3) = A\omega_0^2 \cos(\omega t + \theta) \quad (1)$$

其中 ζ 为阻尼比； ω_0 为系统固有频率； ε 为小参数，描述系统非线性支持力； A 为系统激励幅值。

1.2 伪二维 Jeffcott 转子模型

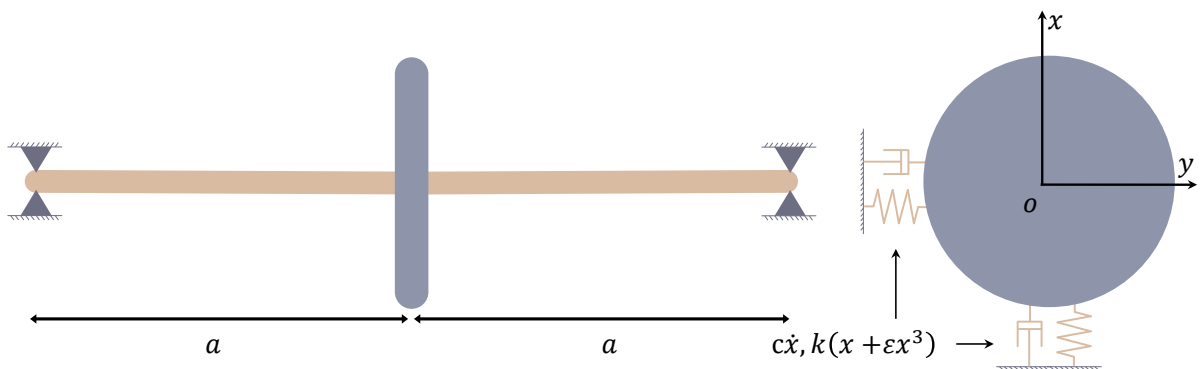


图 2: 具有 Duffing 型支撑刚度的 Jeffcott 模型

图.2 是具有 Duffing 型支撑刚度的 Jeffcott 模型, 转轴与转盘固连, 两端由铰支座提供支撑; 假设系统受到简谐激励; 将转轴提供的支撑刚度和阻尼简化为 xoy 面的面内刚度和阻尼, 同时将支撑部分由金属橡胶减震片导致的非线性支承刚度也简化在转盘的面内, 可以得到系统的动力学微分方程如下:

$$\begin{aligned} \ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2(x + \varepsilon x^3) &= B\omega_0^2 \cos(\omega t + \theta) \\ \ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2(y + \varepsilon y^3) &= B\omega_0^2 \sin(\omega t + \theta) \end{aligned} \quad (2)$$

其中 ζ 为阻尼比; ω_0 为系统固有频率; ε 为小参数, 描述系统非线性支持力; B 为系统激励幅值。此时的激振力来自于转盘自身: 因加工误差的存在转盘的质心和形心不重合, 在转动时就会产生一个绕转动轴的激振力。

2 求解方法

2.1 近似求解方法——谐波平衡法

各种近似方法中, 谐波平衡法是使用最为简便的, 并且应用范围不仅限于非线性系统。该方法的基本思想是: 将激励项与方程的解展开为傅里叶级数。为了保证系统惯性力、作用力的各阶谐波分量相平衡, 必须令动力学方程两端同阶谐波的系数相等, 进而得到包含未知系数的代数方程组, 解该方程即可确定待定的傅里级数的系数。

讨论如下普遍形式的非线性系统受迫振动:

$$\ddot{x} + f(x, \dot{x}) = F(t) \quad (3)$$

设 $F(t)$ 为偶函数, 且不含常值分量。当实验测得系统的周期为 $T = \frac{2\pi}{\omega}$, 可将 $F(t)$ 展开为周期为 T 的傅里叶级数:

$$F(t) = \sum_{n=1}^{\infty} f_n \cos(n\omega t) \quad (4)$$

$$f_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} F(t) \cos(n\omega t) dt \quad (n = 1, 2, \dots) \quad (5)$$

同时, 估计方程的解也以频率 ω 周期变化, 也可展开为傅里叶级数:

$$x(t) = A_0 + \sum [A_n \cos(n\omega t) + B_n \sin(n\omega t)] \quad (6)$$

将已展开为傅里叶级数的激励项 $F(t)$ 和响应 $x(t)$ 代入动力学微分方程式。函数 $f(x, \dot{x})$ 中包含了非线性项、阻尼项, 通常是 x, \dot{x} 的多项式, 对于非线性项, 将解的形式带入该项后,

通常能利用三角公式化为各阶一次谐波的形式。接着令等式两边各阶谐波前的系数相等,得到包含未知系数的无穷维代数方程组。

通过合理预先假定谐波的个数,即可从有限的代数方程中解出待定系数,进而确定各阶谐波的振幅与频率之间的关系。当傅里叶级数收敛时,谐波频率越高,振幅越小,所以可以用有限项代替无穷级数。

2.2 数值求解方法——龙格库塔方法

龙格库塔 (Runge-Kutta) 方法,简称 R-K 方法,由求解常微分方程的欧拉方法发展而来,具体的推导方法参考任意一本数值分析教科书。

这里给出最经典的一个四阶龙格库塔公式,以求解动力学微分方程 $\dot{y} = f(x, y)$:

$$y_{n+1} = y_n + \frac{h}{6} (K_1 + 2K_2 + 2K_3 + K_4) \quad (7)$$

$$\begin{aligned} K_1 &= f(x_n, y_n) \\ K_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1\right) \\ K_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2\right) \\ K_4 &= f(x_n + h, y_n + hK_3) \end{aligned} \quad (8)$$

上述公式是求解一阶常微分方程组的龙格库塔公式,但动力学微分方程一般为二阶,并且常以方程组的形式出现。所以在求解时,一般对高阶常微分方程进行降阶,得到对应的一阶常微分方程组。降阶方法参考各种网站、教科书,非常简单。

如果不想降阶,就想有个公式直接求解二阶动力学微分方程组怎么办?想偷懒的同学可以使用以下龙格库塔公式,以求解二阶微分方程组。

设待求解的二阶微分方程组为:

$$\begin{cases} y''_1 = f_1(x, y_1, y_2, \dots, y_N, y'_1, y'_2, \dots, y'_N) \\ y''_2 = f_2(x, y_1, y_2, \dots, y_N, y'_1, y'_2, \dots, y'_N) \\ \vdots \\ y''_N = f_N(x, y_1, y_2, \dots, y_N, y'_1, y'_2, \dots, y'_N) \end{cases} \quad (9)$$

相应的龙格库塔公式为:

$$\begin{aligned} y_{n+1,1}(i) &= y_{n,1}(i) + \frac{h}{6} (K_{11}(i) + 2K_{21}(i) + 2K_{31}(i) + K_{41}(i)) \\ y_{n+1,2}(i) &= y_{n,2}(i) + \frac{h}{6} (K_{21}(i) + 2K_{22}(i) + 2K_{23}(i) + K_{24}(i)) \end{aligned} \quad (10)$$

其中 $y_{n+1,1}(i)$ 和 $y_{n+1,2}(i)$ 分别代表求解到第 $n+1$ 步的 y_i 和 y'_i , i 为因变量 y 的序号,共

有 N 个因变量。式中：

$$\begin{aligned}
 K_{11}(i) &= y_{n,2}(i) \\
 K_{12}(i) &= f_i(x_n, y_{n,1}(1), y_{n,1}(2), \dots, y_{n,1}(N), y_{n,2}(1), y_{n,2}(2), \dots, y_{n,2}(N)) \\
 K_{21}(i) &= y_{n,2}(i) + \frac{h}{2}K_{12}(i) \\
 K_{22}(i) &= f_i(x_n + \frac{h}{2}, y_{n,1}(1) + \frac{h}{2}K_{11}(1), y_{n,1}(2) + \frac{h}{2}K_{11}(2), \dots, y_{n,1}(N) + \frac{h}{2}K_{11}(N), \\
 &\quad y_{n,2}(1) + \frac{h}{2}K_{12}(1), y_{n,2}(2) + \frac{h}{2}K_{12}(2), \dots, y_{n,2}(N) + \frac{h}{2}K_{12}(N)) \\
 K_{31}(i) &= y_{n,2}(i) + \frac{h}{2}K_{22}(i) \\
 K_{32}(i) &= f_i(x_n + \frac{h}{2}, y_{n,1}(1) + \frac{h}{2}K_{21}(1), y_{n,1}(2) + \frac{h}{2}K_{21}(2), \dots, y_{n,1}(N) + \frac{h}{2}K_{21}(N), \\
 &\quad y_{n,2}(1) + \frac{h}{2}K_{22}(1), y_{n,2}(2) + \frac{h}{2}K_{22}(2), \dots, y_{n,2}(N) + \frac{h}{2}K_{22}(N)) \\
 K_{41}(i) &= y_{n,2}(i) + \frac{h}{2}K_{32}(i) \\
 K_{42}(i) &= f_i(x_n + \frac{h}{2}, y_{n,1}(1) + hK_{31}(1), y_{n,1}(2) + hK_{31}(2), \dots, y_{n,1}(N) + hK_{31}(N), \\
 &\quad y_{n,2}(1) + \frac{h}{2}K_{32}(1), y_{n,2}(2) + \frac{h}{2}K_{32}(2), \dots, y_{n,2}(N) + \frac{h}{2}K_{32}(N))
 \end{aligned} \tag{11}$$

3 近似解析解

以 Jeffcott 模型为例，本节将给出 Duffing 方程的谐波解推导过程和相应的 Matlab 关键代码。

3.1 谐波平衡解的推导

因为两个方向的振动相互独立，因此以 x 方向为例，推导系统的一阶近似解析解。

1. 根据激励的形式，假设解的形式为：

$$x = A \cos \omega t \tag{12}$$

将 式.12 代入系统动力学微分方程式.2 的第一式，并利用三角变换

$$\cos^3 \alpha = \frac{3 \cos \alpha + \cos 3\alpha}{4} \tag{13}$$

得到：

$$\left[A(1 - s^2) + \frac{3}{4}\varepsilon A^3 \right] \cos \omega t - (2\zeta s A) \sin \omega t = B(\cos \theta \cos \omega t - \sin \theta \sin \omega t) \tag{14}$$

上式省略了高次谐波， $s = \frac{\omega}{\omega_0}$ 表示频率比。令上式两边一次谐波的系数相等，得到：

$$\begin{aligned} A(1 - s^2) + \frac{3}{4}\varepsilon A^3 &= B \cos \theta \\ 2\zeta s A &= B \sin \theta \end{aligned} \quad (15)$$

从上式消去参数 θ ，导出幅频特性关系式：

$$\frac{A}{B} = \frac{1}{\sqrt{(1 - s^2 + \frac{3}{4}\varepsilon A^2)^2 + (2\zeta S)^2}} \quad (16)$$

2. 假设解的形式为 $x = A \cos \omega t + A \sin \omega t$ ，按照相同的方法推导得出系统幅频关系式：

$$\frac{A}{B} = \frac{1}{\sqrt{(1 - s^2 + \frac{3}{2}\varepsilon A^2)^2 + (2\zeta S)^2}} \quad (17)$$

3.2 编程实现

根据 式.16，取小参数 $\varepsilon = +0.04$ （硬弹簧）及 $\varepsilon = -0.04$ （软弹簧），可以得到一阶近似的幅频响应曲线。注意，本部分代码并不完整，只给出了关键代码，每一行关键代码都有相关解释。

首先，定义并输入初始参数。

```

1   %Input initial value
2   format long; %display 15-bit precision of data with class-double
3   global B epsilon xi;
4   syms s A ;
5   epsilon=[0 0.01 0.02 0.03 0.04];
6   B=1;
7   xi=0.05;

```

第二步，进行求解。

```

1   %Solve Amplitude-Frequency of Duffing Equation
2   s_start=0; s_end=2.5; s_int=0.025; s_number=(s_end-s_start)/s_int;
3   s=linspace(s_start,s_end,s_number);
4   s_Duffing_AF=zeros(3,s_number*size(xi,2));
5   for j=1:size(xi,2)
6       for i=1:(s_end-s_start)/s_int

```

```

7      Duffing_AF = (B / ...
          sqrt((1-s(i)^2+0.75*epsilon*A^2)^2+(2*xi(j)*s(i))^2))-A;
8      %vpa() can get the numerical roots of equation solved by solve();
9      %double(): solve() get the data with class()=sym,so using ...
          double() to transfer the class of data
10     Duffing_AF=double(vpa(solve(Duffing_AF,A)));% solve
11     Duffing_AF=Duffing_AF(abs(imag(Duffing_AF))<eps(Duffing_AF));
12     %abs(imag(?)<eps(?)) delete imaginary number
13     Duffing_AF=real(Duffing_AF(Duffing_AF>0));% find the data which ...
          are greater than 0,and insure the class of data is double
14     if size(Duffing_AF,1)==1
15         s_Duffing_AF(1,i+s_number*(j-1))=Duffing_AF;    ...
          %i+s_number*(j-1): storing result with different 'xi' ...
          in one matrix
16     elseif size(Duffing_AF,1)==2
17         s_Duffing_AF(1:2,i+s_number*(j-1))=Duffing_AF;
18     else
19         s_Duffing_AF(1:3,i+s_number*(j-1))=Duffing_AF;
20     end
21 end
22 end

```

第三步，绘图。

```

1      %a.Find zero point,then delete it. b.Polt
2      a=find(s_Duffing_AF==0);
3      s_Duffing_AF(a)=NaN;
4      i=0;j=0;
5      c_color=colormap(copper(size(xi,2)));sz = 10;%color and size of dots
6      for i=1:size(xi,2)
7          for j=1:size(s_Duffing_AF,1)
8              scatter(s,s_Duffing_AF(j,1+s_number*(i-1):s_number*i), sz, ...
                  c_color(i,:), 'filled');
9          hold on
10         end
11     end
12     hold off

```

3.3 结果分析

图.3和图.4给出了以 ζ 为参数的幅频响应曲线族。可以看出非线性系统的受迫振动有与线性系统类似的幅频曲线，不过支撑曲线族的骨架不是直线，而是朝频率增大方向 ($\varepsilon > 0$) 或减小方向 ($\varepsilon < 0$) 弯曲，从而使整个曲线族朝一侧倾斜。

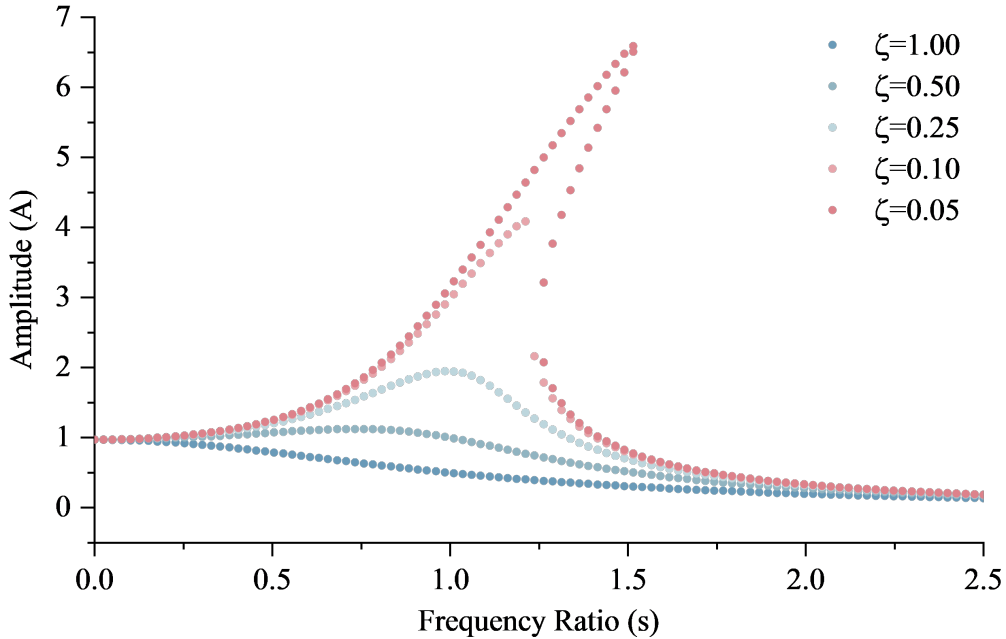


图 3: 以 ζ 为参数的幅频特性曲线族 ($\varepsilon = +0.04$)

从图.3中可以看出，当阻尼比趋近于 1 时，系统在频域上是稳定的，一个频率唯一对应一个幅值。改变系统固有参数后（增大阻尼比），系统在某一区间出现分岔现象，阻尼比 $\zeta = 0.05$ 且频率比在区间 $s \in (1.25, 1.5)$ 时每个频率对应 3 个幅值，即系统出现不稳定运动。同样的现象也出现在图.4中。具有软弹簧特性的转子系统以较低的频率运行时，系统也极有可能出现不稳定的运动状态。当阻尼比为 $\zeta = 0.05$ ，同时改变非线性参数 ε ，可以得到以 ε 为参数的系统幅频特性曲线，如图.5所示。

当非线性参数 $\varepsilon = 0$ 时，系统简化为线性系统，逐渐增大 ε 系统出现分岔现象的区间逐渐增大，意味着不稳定区域逐渐增大。同时幅频曲线的出现峰值时的频率逐渐推后，幅值也相应的有所下降。

不改变其他参数，根据式.17计算系统的幅频特性曲线，可以看出曲线没有较大的改变。在阻尼比 $\zeta = 0.05$ 的情况下，对比不同解的形式下求解得到的系统的幅频曲线的变化，如图 16(b) 所示。可以看出当 $x = A \cos \omega t + A \sin \omega t$ 时求解得到系统的幅频曲线向左弯曲，在不稳定段以及靠近一阶共振频率时两者差别明显，在稳定段幅频特性几乎一致。

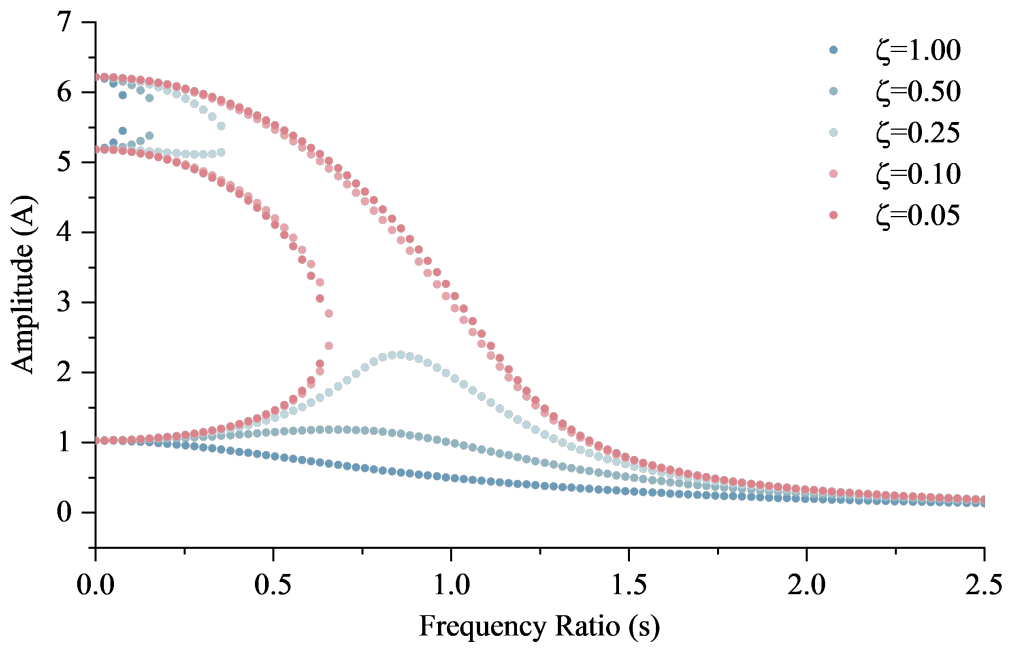


图 4: 以 ζ 为参数的幅频特性曲线族 ($\varepsilon = +0.04$)

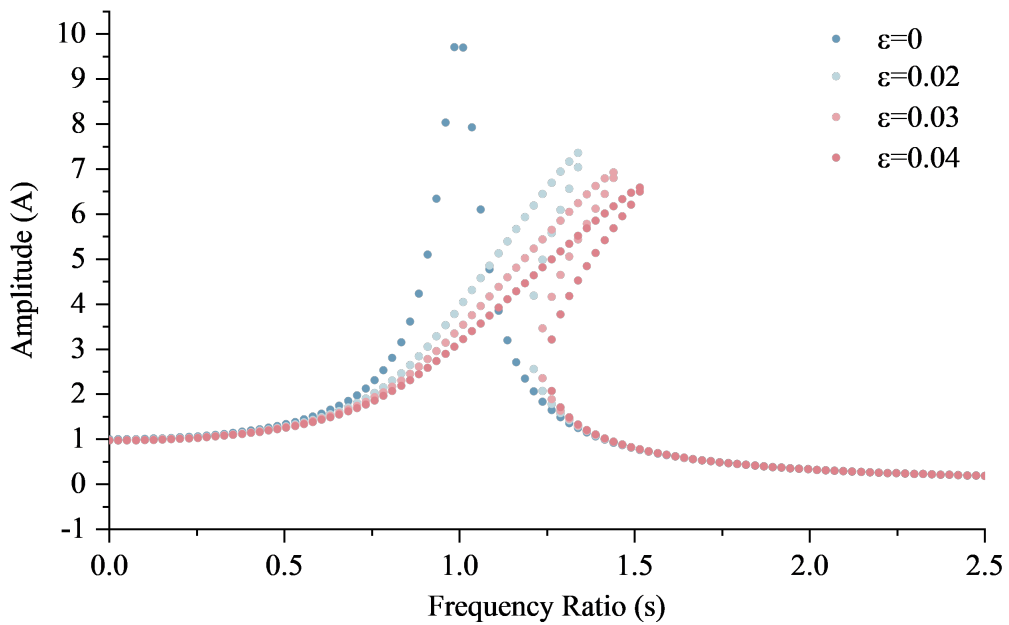


图 5: 以 ε 为参数的幅频特性曲线族 ($\zeta = 0.05$)

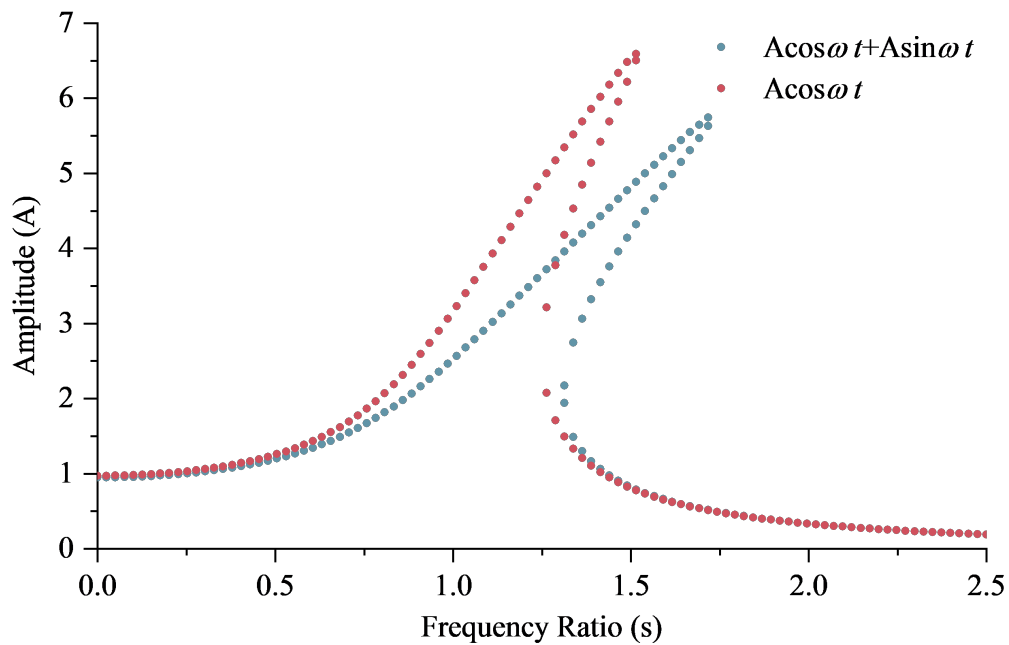


图 6: 两种解的形式求解结果对比

4 数值解

4.1 编程实现

前期工作 1: 定义 Duffing 方程函数 *.m* 文件。

```

1   %Save the Duffing equation
2   function dy=Duffing(t,yn,y_n)
3
4       global B epsilon xi omicron0 omicron;
5       dy=B*omicron0^2*cos(omicron*t) - 2*xi*y_n - ...
6           omicron0^2*(yn+epsilon*yn^3);
7   end

```

前期工作 2: 定义龙格库塔求解函数 *.m* 文件。

```

1   function [yn1,y_n1] = RungeKutta(fun,xn,yn,y_n,h) %#ok<*INUSD>
2   %{
3   This function aims to solve differential equation with two orders step
4   by step.

```

```

5   Input :
6   fun:the differential equation ,fun is column vector;  y''=fun(xn,yn,y_n)
7   xn:initial value
8   yn:initial value;could be sigal or column vector; y(n)
9   y_n:initial value with one order;could be sigal or column vector; ...
    y'(n)
10  h:step length
11  Output:
12  yn1:y(n+1),step length=h
13  y_n1:y'(n+1),step length=h
14  %}
15  if nargin == 3
16      h = 1;                %the default step size is 1
17      %input K11-K24
18      k11 = y_n;
19      k21 = fun(xn,yn,y_n);
20      k12 = y_n+h/2*k21;
21      k22 = fun(xn+h/2, yn+h/2*k11, y_n+h/2*k21);
22      k13 = y_n+h/2*k22;
23      k23 = fun(xn+h/2, yn+h/2*k12, y_n+h/2*k22);
24      k14 = y_n+h*k23;
25      k24 = fun(xn+h, yn+h*k13, y_n+h*k23);
26      yn1 = yn + h/6*(k11+ 2*k12+ 2*k13 +k14);
27      y_n1 = y_n + h/6*(k21 +2*k22 +2*k23 +k24);
28
29  else
30      %input K11-K24
31      k11 = y_n;
32      k21 = fun(xn,yn,y_n);
33      k12 = y_n+h/2*k21;
34      k22 = fun(xn+h/2, yn+h/2*k11, y_n+h/2*k21);
35      k13 = y_n+h/2*k22;
36      k23 = fun(xn+h/2, yn+h/2*k12, y_n+h/2*k22);
37      k14 = y_n+h*k23;
38      k24 = fun(xn+h, yn+h*k13, y_n+h*k23);
39      yn1 = yn + h/6*(k11+ 2*k12+ 2*k13 +k14);
40      y_n1 = y_n + h/6*(k21 +2*k22 +2*k23 +k24);
41  end
42
43  end

```

首先，定义并初始化参数。

```

1  global omicron omicron0 xi epsilon
2  omicron0=1;
3  xi=0.05;
4  epsilon=0.04;
5  %When np=400 and t_num=4000, the runtime completing the whole program ...
   is 83s;
6  np=400;%calculating times about excited frequency
7  Omega1 = linspace(0,2.5,np);%Forward sweep
8  Omega2 = linspace(2.5,0,np);%Backward sweep
9  s_A1=zeros(length(Omega1),1);%displacement
10 s_A2=zeros(length(Omega2),1);%displacement
11 s_A1_1=[];s_A2_1=[];
12 t_start=0; t_end=400; t_num=(t_end-t_start)*10;
13 step=(t_end-t_start)/t_num;%control the accuracy
14 t=linspace(t_start,t_end,t_num);

```

第二步，正向扫频求解。

```

1  %Forward sweep
2  yn=0;y_n=0;
3  for i=1:1:length(Omega1)
4      omicron=Omega1(i);%Change the excited frequency
5      s_yn=zeros(t_num,1);s_y_n=zeros(t_num,1);%Pre-allocated memory
6
7      for j=1:1:length(t)
8          [s_yn(j,:),s_y_n(j,:)]=RungeKutta(@Duffing,t(j),yn,y_n,step);
9          yn=s_yn(j,:);y_n=s_y_n(j,:);
10     end
11     nn=length(s_yn);
12     ynmax=max(s_yn(nn-round(nn/2):nn));
13     y_nmax=max(s_y_n(nn-round(nn/2):nn));
14     s_A1(i)=ynmax;%displacement
15     yn=ynmax;y_n=y_nmax;
16 end

```

第三步，反向扫频求解。

```

1  i=0;j=0;
2  yn=0;y_n=0;
3  for i=1:1:length(Omega2)

```

```

4      omicron=Omega2(i);%Change the excited frequency
5      s_yn=zeros(t_num,1);s_y_n=zeros(t_num,1);%Pre-allocated memory
6
7      for j=1:1:length(t)
8          [s_yn(j,:),s_y_n(j,:)]=RungeKutta(@Duffing,t(j),yn,y_n,step);
9          yn=s_yn(j,:);y_n=s_y_n(j,:);
10     end
11     nn=length(s_yn);
12     ynmax=max(s_yn(nn-round(nn/2):nn));
13     y_nmax=max(s_y_n(nn-round(nn/2):nn));
14     s_A2(i)=ynmax;%displacement
15     %yn=ynmax;y_n=y_nmax;
16     yn=0;y_n=0;
17 end

```

第四步，画图。

```

1      figure
2      plot(Omega1,s_A1)
3      hold on
4      plot(Omega2,s_A2)

```

4.2 结果分析

使用四阶龙格库塔公式求得不同频率下系统在时域下的响应，并在稳态段提取系统的幅值，得到系统的幅频特性曲线。分别构造递增频率数列（正向扫频）和递减频率数列（反向扫频），以第 i 次系统稳态响应位移和速度幅值作为第 $i+1$ 次计算时的系统初值，其中 $i=0$ 时取系统位移和速度为 0。

计算 $B=1$, $\zeta=0.05$, $\varepsilon=0.04$ 时系统的幅频曲线，如图.7所示。其中红色线条代表正向扫频结果，蓝色线条代表反向扫频结果。可以看出只有正向扫频可以得到幅频曲线极大值点前一段的幅频特性，而不稳定段的下半部分只能通过反向扫频得到。但是极大值点左端的一部分幅频特性并不能通过已有的数值方法计算得到。

对比数值解和一阶近似解析解，可以看出当近似解析解的形式为 $x=A\cos\omega t$ 时，除去幅频曲线极大值的左端部分区间，数值解和谐波平衡法得到的一阶近似解析解相同。但是，当近似解析解的形式为 $x=A\cos\omega t+A\sin\omega t$ 时，近似解析解的结果在不稳定段和一阶共振频率附近并不准确。因此，使用谐波平衡法求解系统幅频特性曲线时，需要严格根据激励的形式假设解的形式，并且可以得到足够精确的系统幅频特性表达式。

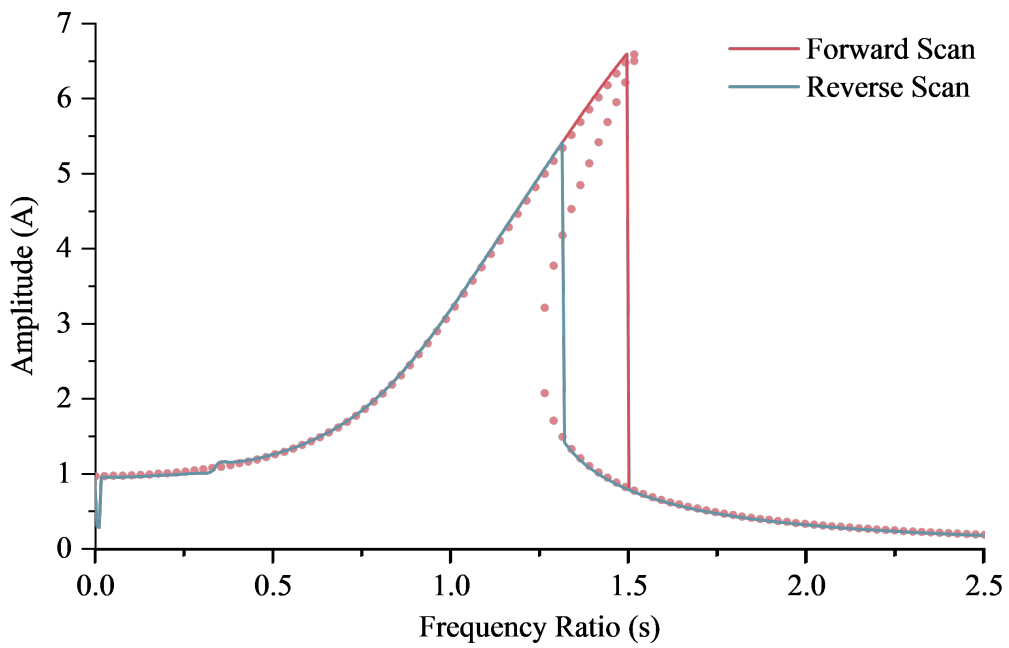


图 7: 幅频特性曲线 (数值解, 设解为 $x = A \cos \omega t$)

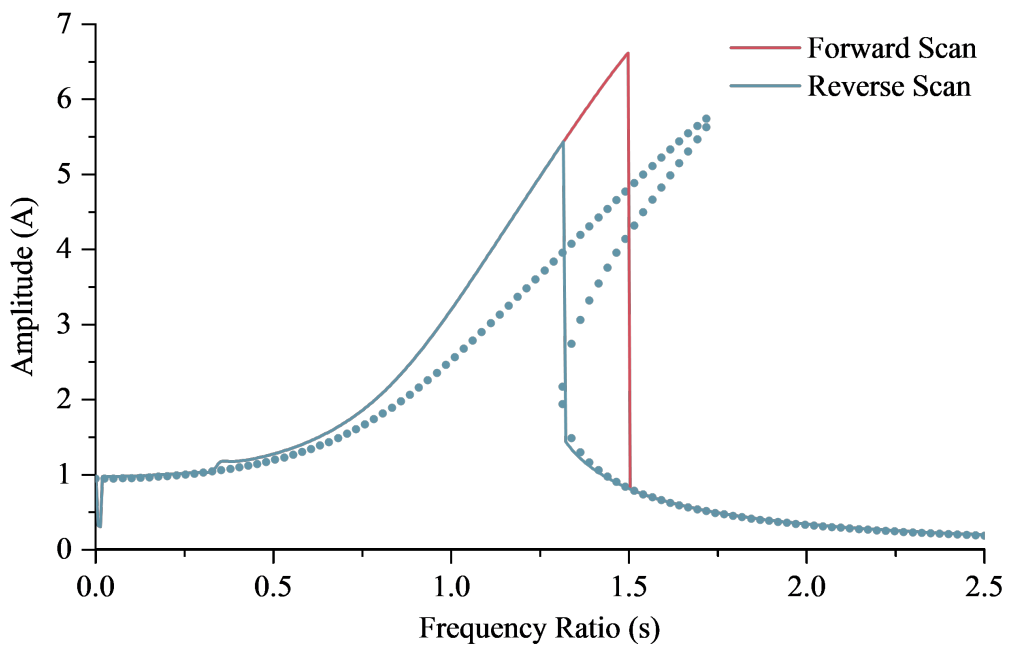


图 8: 幅频特性曲线 (数值解, 设解为 $x = A \cos \omega t + A \sin \omega t$)