

数值试验二 赵衍智 2020511041

试验一：体会数值方法的不稳定性对数值结果的影响

1. 显式欧拉法

(1). 理论分析

$$\because y_{n+1} = y_n + hf(x_n, y_n) = y_n - 50hy_n = (1 - 50h)y_n$$

假设 y_n 有误差 ϵ_n , 即实际计算值 $\bar{y}_n = y_n + \epsilon_n$

$$\therefore \bar{y}_{n+1} = (1 - 50h)\bar{y}_n = (1 - 50h)(y_n + \epsilon_n)$$

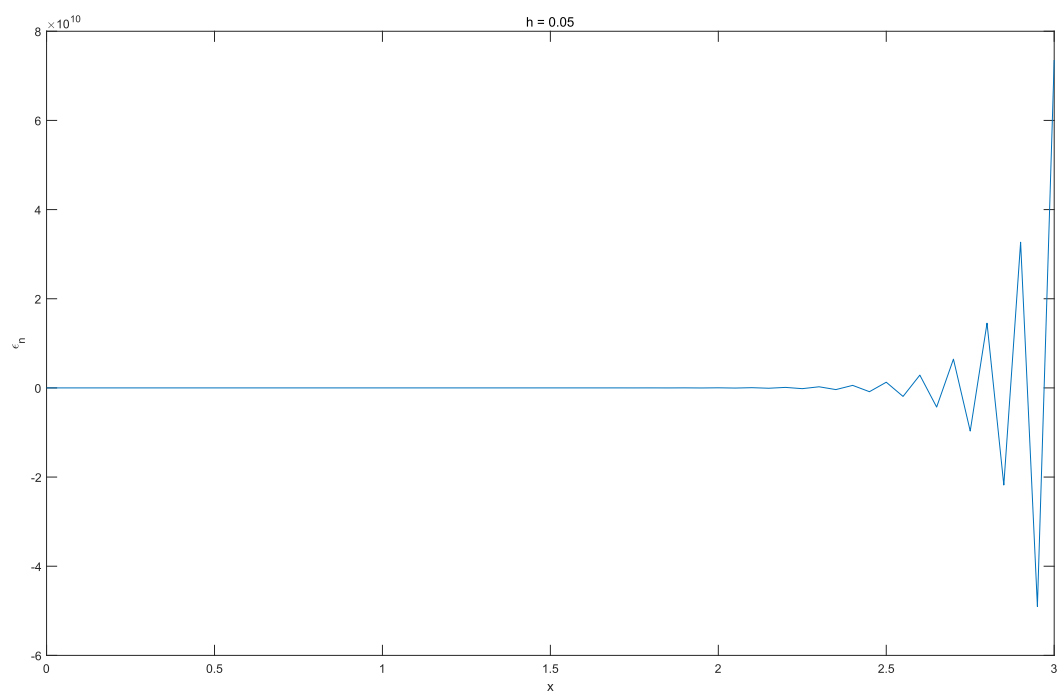
$$\therefore y_{n+1} \text{ 的误差 } \epsilon_{n+1} = \bar{y}_{n+1} - y_{n+1} = (1 - 50h)\epsilon_n$$

$$\therefore \text{迭代格式具有稳定性} \iff \frac{|\epsilon_{n+1}|}{|\epsilon_n|} \leq 1 \iff |1 - 50h| \leq 1 \iff 0 < h \leq 0.04$$

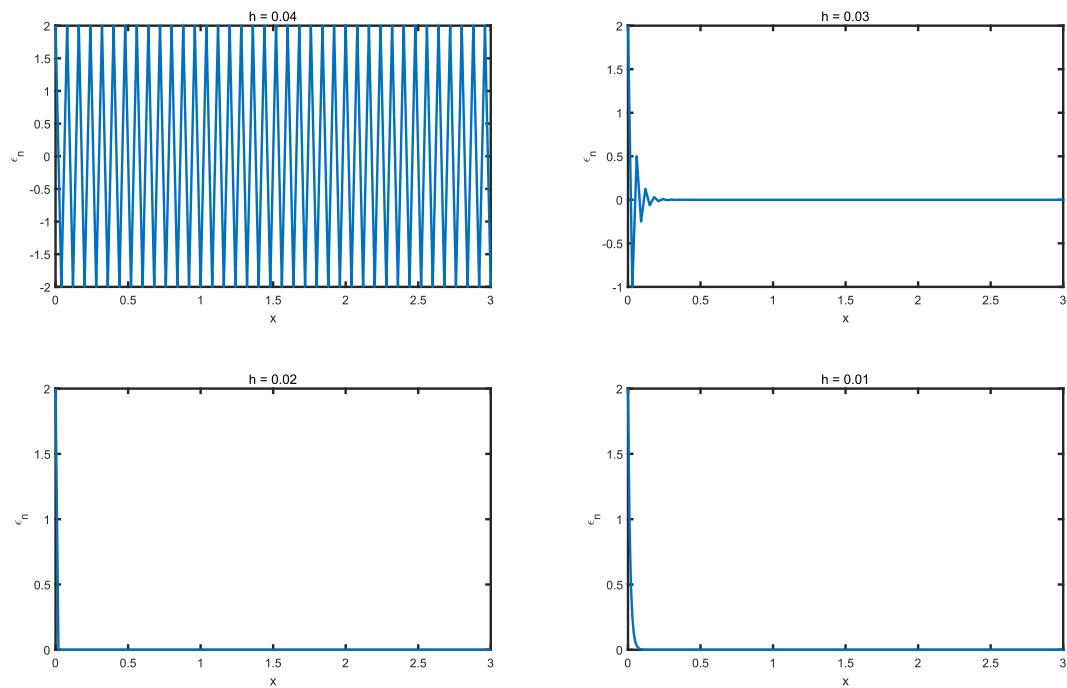
(2). 试验结果

在数值试验中一律取 $y_0 = y(0) = 100, \bar{y}_0 = 102$

- $h = 0.05$, 试验区间取为 $[0, 6]$



- $h = 0.04, 0.03, 0.02, 0.01$; 试验区间取为 $[0, 3]$



(3). 结果分析

- (i) $h = 0.05$ 时误差 ϵ_n 不断扩大
- (ii) $h = 0.04, 0.03, 0.02, 0.01$ 时迭代格式稳定
- (iii) $h = 0.04$ 时 ϵ_n 在2和 -2 处来回波动, 因为 $\frac{\epsilon_{n+1}}{\epsilon_n} = 1 - 50h = -1, n = 0, 1, 2, \dots$
- (iv) $h = 0.02$ 时 ϵ_n 衰减最快, 在 x_1 处衰减为零, 因为 $\frac{|\epsilon_{n+1}|}{|\epsilon_n|} = |1 - 50h| = 0, n = 0, 1, 2, \dots$

(4). 代码

- 以 $h = 0.04$ 为例：

```

1  clf
2  y0 = 100;
3  y0_ = 102;
4  h = 0.04; % 步长
5  l = 3; % 区间长度
6  m = l/h + 1;
7  x = 0:h:l;
8  Y = zeros(1,m);
9  Y_ = zeros(1,m);
10 Y(1) = y0;
11 Y_(1) = y0_;
12 for i = 2:1:m
13     Y(i) = (1 - 50*h)*Y(i-1);
14     Y_(i) = (1 - 50*h)*Y_(i-1);
15 end
16 epsilon = Y_ - Y;
17 subplot(2,2,1);
18 plot(x,epsilon);
19 xlabel('x');
20 ylabel('\epsilon_n');
21 title('h = 0.04');
```

2. 隐式欧拉法

(1). 理论分析

$$\because y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) = y_n - 50hy_{n+1}, \text{ 即 } y_{n+1} = \frac{y_n}{1 + 50h}$$

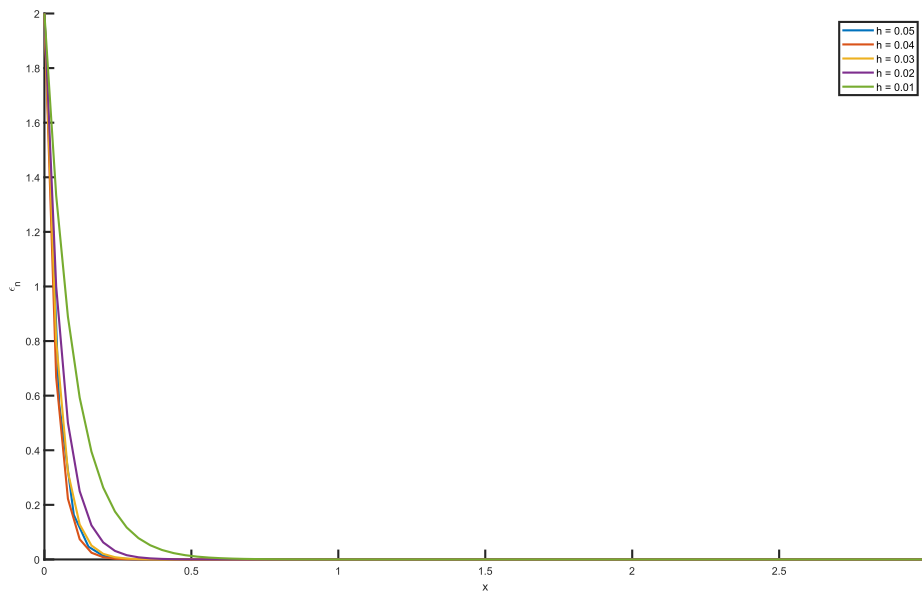
设 y_n 有误差 ϵ_n , 则实际计算值 $\tilde{y}_n = y_n + \epsilon_n$

$$\therefore \tilde{y}_{n+1} = \frac{\tilde{y}_n}{1 + 50h}, \quad y_{n+1} \text{ 有误差 } \epsilon_{n+1} = \tilde{y}_{n+1} - y_{n+1} = \frac{\epsilon_n}{1 + 50h}$$

$$\therefore \text{迭代格式具有稳定性} \iff \frac{|\epsilon_{n+1}|}{|\epsilon_n|} \leq 1 \iff \frac{1}{|1 + 50h|} \leq 1 \iff h > 0$$

(2). 试验结果

- h 取0.05, 0.04, 0.03, 0.02, 0.01; 试验区间取为[0, 3]
迭代格式均稳定



(3). 结果分析

随着 h 的增大, $|\epsilon_n|$ 衰减的速率增大, 因为 $\frac{|\epsilon_{n+1}|}{|\epsilon_n|} = \frac{1}{|1 + 50h|}$; h 增大, $\frac{|\epsilon_{n+1}|}{|\epsilon_n|} = \frac{1}{|1 + 50h|}$ 减小, ϵ_n 衰减加快

(4). 代码

- 以 $h = 0.05, 0.04$ 为例:

```
1  clf
2  y0 = 100;
3  y0_ = 102;
4
5  hold on
6  h1 = 0.05; % 步长
7  l = 3; % 区间长度
8  m = l/h1 + 1;
9  x = 0:h1:l;
10 Y = zeros(1,m);
11 Y_ = zeros(1,m);
12 Y(1) = y0;
13 Y_(1) = y0_;
14 for i = 2:1:m
15     Y(i) = Y(i-1)/(1 + 50*h1);
16     Y_(i) = Y_(i-1)/(1 + 50*h1);
17 end
18 epsilon = Y_ - Y;
19 plot(x,epsilon);
```

```
20 xlabel('x');
21 ylabel('\epsilon_n');
22
23 h2 = 0.04;
24 m = l/h2 + 1;
25 x = 0:h2:l;
26 Y = zeros(1,m);
27 Y_ = zeros(1,m);
28 Y(1) = y0;
29 Y_(1) = y0_;
30 for i = 2:1:m
31     Y(i) = Y(i-1)/(1 + 50*h2);
32     Y_(i) = Y_(i-1)/(1 + 50*h2);
33 end
34 epsilon = Y_ - Y;
35 plot(x,epsilon);
```

试验二：体会算法的保结构性

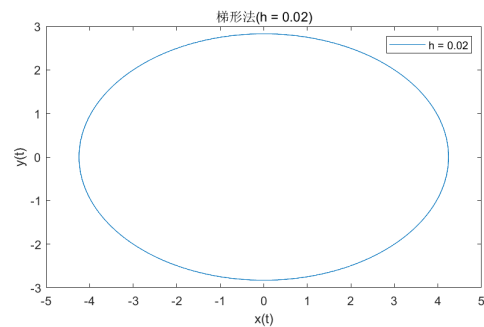
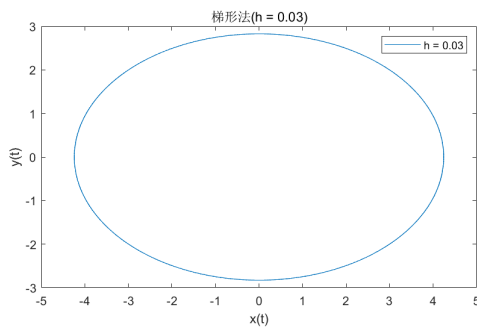
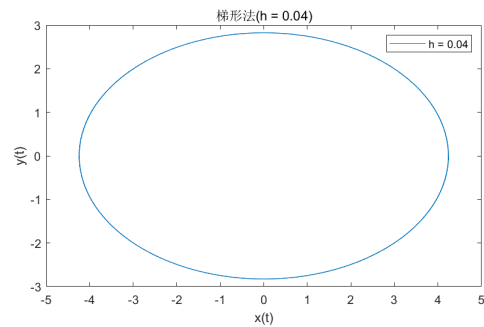
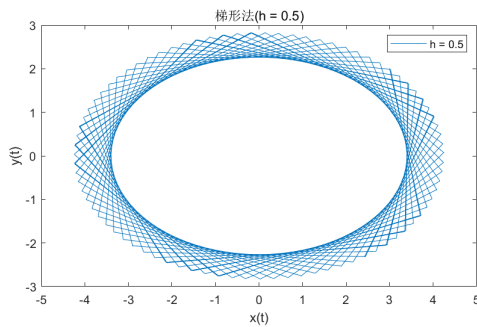
1. 梯形法

(1). 理论分析

$$\begin{aligned} \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} &= \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \frac{h}{2} \left[\begin{pmatrix} x'_n \\ y'_n \end{pmatrix} + \begin{pmatrix} x'_{n+1} \\ y'_{n+1} \end{pmatrix} \right] = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \frac{h}{2} \left[\begin{pmatrix} \frac{9}{2}y_n \\ -2x_n \end{pmatrix} + \begin{pmatrix} \frac{9}{2}y_{n+1} \\ -2x_{n+1} \end{pmatrix} \right] \\ \text{即} \quad \begin{cases} x_{n+1} - \frac{9h}{4}y_{n+1} &= x_n + \frac{9h}{4}y_n \\ y_{n+1} + hx_{n+1} &= y_n - hx_n \end{cases} \\ \text{即} \quad \begin{cases} y_{n+1} &= -\frac{8h}{4+9h^2}x_n + \frac{4-9h^2}{4+9h^2}y_n \\ x_{n+1} &= \frac{4-9h^2}{4+9h^2}x_n + \frac{18h}{4+9h^2}y_n \end{cases} \\ \therefore x_{n+1}^2 + \frac{9}{4}y_{n+1}^2 &= x_n^2 + \frac{9}{4}y_n^2 \\ \therefore x_n^2 + \frac{9}{4}y_n^2 &= x_0^2 + \frac{9}{4}y_0^2 = 18 \end{aligned}$$

(2). 试验结果

- $h = 0.5$ 时 $n = 1, 2, \dots, 100$; $h = 0.04, 0.03, 0.02$ 时 $n = 0, 1, 2, \dots, 2000$



(3). 结果分析

梯形法相图中的每个点都在椭圆 $x^2 + \frac{9}{4}y^2 = 18$ 上, 梯形法具有保结构性

(4). 代码

- 以 $h = 0.5$ 为例

```

1  clf
2  x_0 = 3;
3  y_0 = 2;
4
5  subplot(2,2,1);
6  h = 0.5;
7  n = 100;
8  X = zeros(1,n);
9  Y = X;
10 X(1) = x_0;
11 Y(1) = y_0;
12 for i = 2:1:n
13     X(i) = (4 - 9*h^2)*X(i-1)/(4+9*h^2) + 18*h*Y(i-1)/(4 + 9*h^2);
14     Y(i) = -8*h*X(i-1)/(4 + 9*h^2) + (4 - 9*h^2)*Y(i-1)/(4+9*h^2);
15 end
16 plot(X,Y);
17 xlabel('x(t)');
18 ylabel('y(t)');
19 legend('h = 0.5')
20 title('梯形法(h = 0.5)')

```

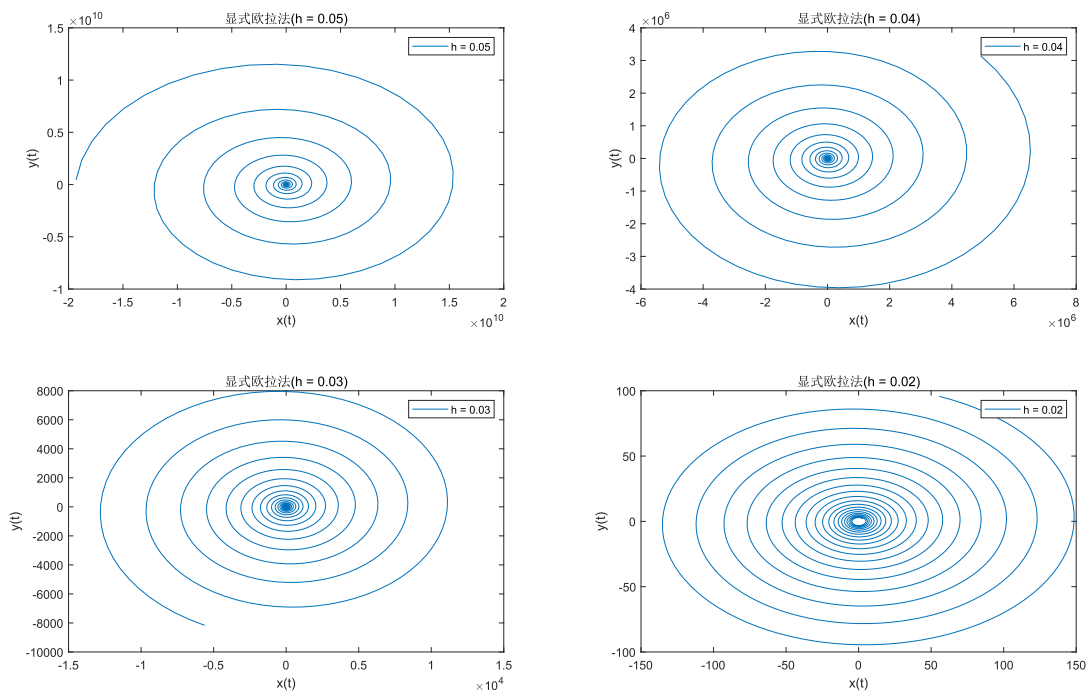
2. 显式欧拉法

(1). 理论分析

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + h \begin{pmatrix} x'_n \\ y'_n \end{pmatrix} = \begin{pmatrix} x_n + \frac{9h}{2}y_n \\ y_n - 2hx_n \end{pmatrix}$$

(2). 试验结果

- $n = 0, 1, 2, \dots, 2000$



(3). 结果分析

显式欧拉法相图中的点不在固定的轨迹上,改变 h 和 n ,点的轨迹随之改变。显式欧拉法不具有保结构性

(4). 代码

- 以 $h = 0.05$ 为例：

```
1      clf
2      x_0 = 3;
3      y_0 = 2;
4
5      %% 显式
6      subplot(2,2,1);
7      h = 0.05;
8      n = 2000;
9      X = zeros(1,n);
10     Y = X;
11     X(1) = x_0;
12     Y(1) = y_0;
13     for i = 2:1:n
14         X(i) = X(i-1) - 9*h*Y(i-1)/2;
15         Y(i) = Y(i-1) + 2*h*X(i-1);
16     end
17     plot(X,Y);
18     legend('h = 0.05');
19     xlabel('x(t)');
20     ylabel('y(t)');
21     title('显式欧拉法(h = 0.05)')
```

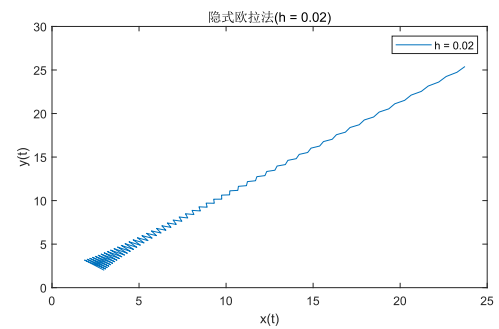
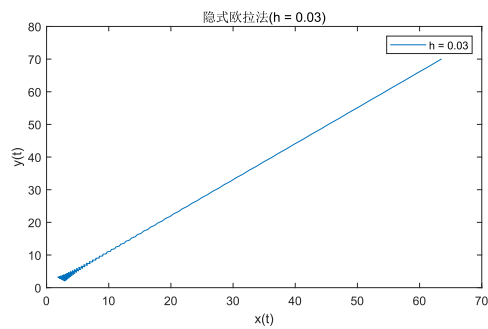
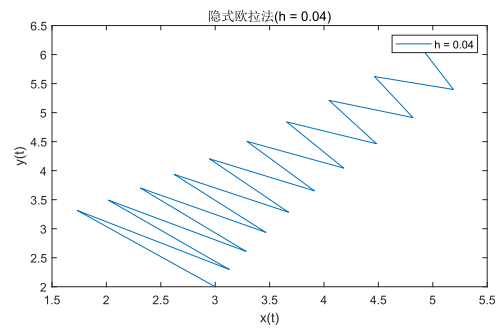
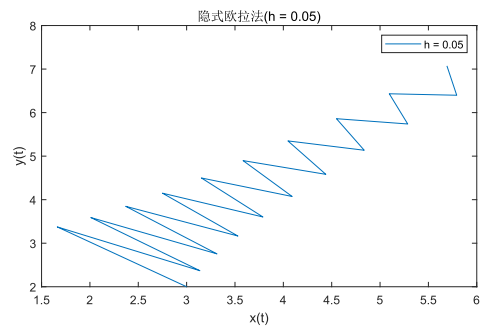
3. 隐式欧拉法

(1). 理论分析

$$\begin{aligned} \therefore \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} &= \begin{pmatrix} x_n \\ y_n \end{pmatrix} + h \begin{pmatrix} x'_{n+1} \\ y'_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} \frac{9h}{2} y_{n+1} \\ -2hx_{n+1} \end{pmatrix} \\ \therefore \begin{cases} y_{n+1} = -\frac{2h}{1+9h^2}x_n + \frac{1}{1+9h^2}y_n \\ x_{n+1} = \frac{1}{1+9h^2}x_n + \frac{9h}{2+18h^2}y_n \end{cases} \end{aligned}$$

(2). 试验结果

- $h = 0.05, 0.04$ 时 $n = 0, 1, 2, \dots, 20$; $h = 0.03, 0.02$ 时 $n = 0, 1, 2, \dots, 100$



(3). 结果分析

隐式欧拉法相图中的点不在固定的轨迹上,改变 h 和 n ,点的轨迹随之改变。隐式欧拉法不具有保结构性

(4). 代码

- 以 $h = 0.05$ 为例：

```

74  %% 隐式
75  subplot(2,2,1);
76  h = 0.05;
77  n = 20;
78  X = zeros(1,n);
79  Y = X;
80  X(1) = x_0;
81  Y(1) = y_0;
82  for i = 2:1:n
83      X(i) = -2*h*X(i-1)/(1+9*h^2) + Y(i-1)/(1 + 9*h^2);
84      Y(i) = X(i-1)/(1 + 9*h^2) + 9*h*Y(i-1)/(2+18*h^2);
85  end
86  plot(X,Y);
87  legend('h = 0.05');
88  xlabel('x(t)');
89  ylabel('y(t)');
90  title('隐式欧拉法(h = 0.05)')

```

4. 四阶 Runge-Kutta 法

(1). 理论分析

$$\text{记 } x'(t) = f(t, x, y) = \frac{9}{2}y(t), \quad y'(t) = g(t, x, y) = -2x(t)$$

$$\text{则四阶Runge-Kutta公式为: } \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \frac{h}{6} \left[\begin{pmatrix} k_1^{(1)} \\ k_1^{(2)} \end{pmatrix} + 2 \begin{pmatrix} k_2^{(1)} \\ k_2^{(2)} \end{pmatrix} + 2 \begin{pmatrix} k_3^{(1)} \\ k_3^{(2)} \end{pmatrix} + \begin{pmatrix} k_4^{(1)} \\ k_4^{(2)} \end{pmatrix} \right]$$

$$\therefore \begin{pmatrix} k_1^{(1)} \\ k_1^{(2)} \end{pmatrix} = \begin{pmatrix} f(t_n, x_n, y_n) \\ g(t_n, x_n, y_n) \end{pmatrix} = \begin{pmatrix} \frac{9}{2}y_n \\ -2x_n \end{pmatrix}$$

$$\begin{pmatrix} k_2^{(1)} \\ k_2^{(2)} \end{pmatrix} = \begin{pmatrix} f(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1^{(1)}, y_n + \frac{h}{2}k_1^{(2)}) \\ g(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1^{(1)}, y_n + \frac{h}{2}k_1^{(2)}) \end{pmatrix} = \begin{pmatrix} \frac{9}{2}(y_n + \frac{h}{2}k_1^{(2)}) \\ -2(x_n + \frac{h}{2}k_1^{(1)}) \end{pmatrix} = \begin{pmatrix} \frac{9}{2}y_n - \frac{9h}{2}x_n \\ -2x_n - \frac{9h}{2}y_n \end{pmatrix}$$

$$\begin{pmatrix} k_3^{(1)} \\ k_3^{(2)} \end{pmatrix} = \begin{pmatrix} f(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2^{(1)}, y_n + \frac{h}{2}k_2^{(2)}) \\ g(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2^{(1)}, y_n + \frac{h}{2}k_2^{(2)}) \end{pmatrix} = \begin{pmatrix} \frac{9}{2}(y_n + \frac{h}{2}k_2^{(2)}) \\ -2(x_n + \frac{h}{2}k_2^{(1)}) \end{pmatrix} = \begin{pmatrix} (\frac{9}{2} - \frac{81h^2}{8})y_n - \frac{9h}{2}x_n \\ (\frac{9h^2}{2} - 2)x_n - \frac{9h}{2}y_n \end{pmatrix}$$

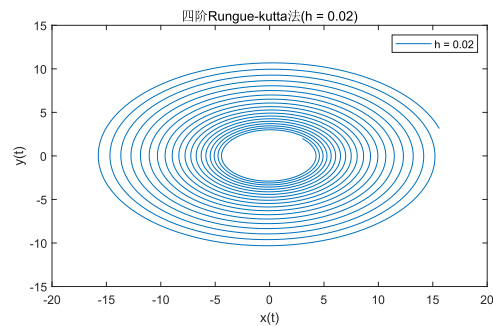
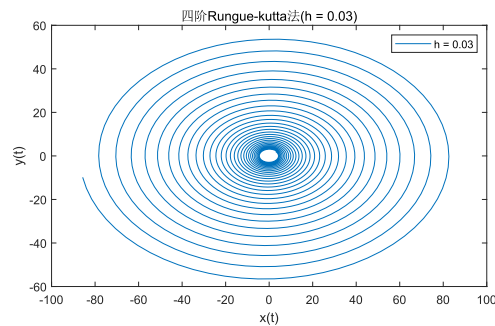
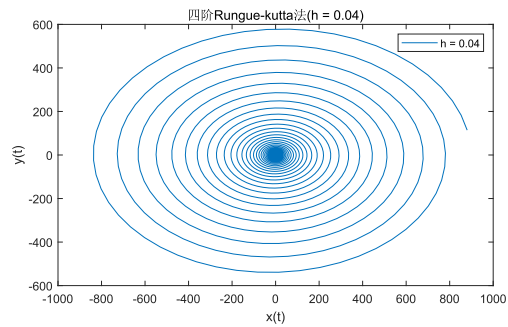
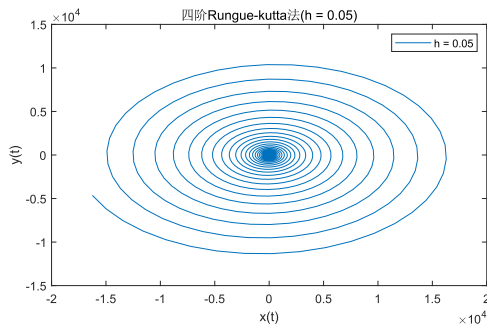
$$\begin{pmatrix} k_4^{(1)} \\ k_4^{(2)} \end{pmatrix} = \begin{pmatrix} f(t_n + h, x_n + hk_3^{(1)}, y_n + hk_3^{(2)}) \\ g(t_n + h, x_n + hk_3^{(1)}, y_n + hk_3^{(2)}) \end{pmatrix} = \begin{pmatrix} \frac{9}{2}(y_n + \frac{h}{2}k_3^{(2)}) \\ -2(x_n + \frac{h}{2}k_3^{(1)}) \end{pmatrix} = \begin{pmatrix} (\frac{9}{2} - \frac{81h^2}{8})y_n + (\frac{81h^3}{8} - \frac{9h}{2})x_n \\ (\frac{81h^3}{8} - \frac{9h}{2})y_n + (\frac{9h^2}{2} - 2)x_n \end{pmatrix}$$

$$\therefore \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \frac{h}{6} \begin{pmatrix} (-\frac{243h^2}{8} + 27)y_n + (\frac{81h^3}{8} - \frac{45h}{2})x_n \\ (\frac{81h^3}{8} - \frac{45h}{2})y_n + (\frac{27h^2}{2} - 12)x_n \end{pmatrix}$$

$$\therefore \begin{cases} x_{n+1} = (\frac{27h^4}{16} - \frac{15h^2}{4} + 1)x_n + (-\frac{81h^3}{16} + \frac{9h}{2})y_n \\ y_{n+1} = (\frac{9h^3}{4} - 2h)x_n + (\frac{27h^4}{16} - \frac{15h^2}{8} + 1)y_n \end{cases}$$

(2). 试验结果

- $n = 0, 1, 2, \dots, 2000$



(3). 结果分析

四阶Runge-Kutta法相图中的点不在固定的轨迹上,改变 h 和 n ,点的轨迹随之改变。四阶Runge-Kutta法不具有保结构性

(4). 代码

- 以 $h = 0.05, 0.04$ 为例：

```
29     clf
30     x_0 = 3;
31     y_0 = 2;
32     h = 0.05;
33     n = 2000;
34     X = zeros(1,n);
35     Y = X;
36     X(1) = x_0;
37     Y(1) = y_0;
38     for i = 2:1:n
39         X(i) = (27*h^4/16 - 15*h^2/4 + 1)*X(i-1) + (-81*h^3/16 + 9*h/2)*Y(i-1);
40         Y(i) = (9*h^3/4 - 2*h)*X(i-1) + (27*h^4/16 - 15*h^2/8 + 1)*Y(i-1);
41     end
42     subplot(2,2,1);
43     plot(X,Y);
44     xlabel('x(t)');
45     ylabel('y(t)');
46     legend('h = 0.05')
47     title('四阶Rungue-kutta法(h = 0.05)')
48
49     subplot(2,2,2);
50     h = 0.04;
51     n = 2000;
52     X = zeros(1,n);
53     Y = X;
54     X(1) = x_0;
55     Y(1) = y_0;
56     for i = 2:1:n
57         X(i) = (27*h^4/16 - 15*h^2/4 + 1)*X(i-1) + (-81*h^3/16 + 9*h/2)*Y(i-1);
58         Y(i) = (9*h^3/4 - 2*h)*X(i-1) + (27*h^4/16 - 15*h^2/8 + 1)*Y(i-1);
59     end
60     plot(X,Y);
61     xlabel('x(t)');
62     ylabel('y(t)');
63     legend('h = 0.04')
64     title('四阶Rungue-kutta法(h = 0.04)')
```

试验三：体会非线性方程的迭代求解

1. 将二阶方程组化为一阶方程组

$$\begin{aligned} \text{设 } \theta(t) = \theta_1(t), \quad \frac{d\theta_1(t)}{dt} = \theta_2(t) \\ \therefore \text{原二阶方程组可化为} \begin{cases} \frac{d\theta_2(t)}{dt} = -\sin \theta_1, \quad 0 < t \leq 10 \\ \frac{d\theta_1(t)}{dt} = \theta_2(t) \\ \theta_1(0) = \frac{\pi}{3} \\ \theta_2(0) = -\frac{1}{2} \end{cases} \end{aligned}$$

2. 隐式欧拉法

(1). 理论分析

$$\begin{pmatrix} \theta_{1(n+1)} \\ \theta_{2(n+1)} \end{pmatrix} = \begin{pmatrix} \theta_{1(n)} \\ \theta_{2(n)} \end{pmatrix} + h \begin{pmatrix} \theta'_{1(n+1)} \\ \theta'_{2(n+1)} \end{pmatrix} = \begin{pmatrix} \theta_{1(n)} \\ \theta_{2(n)} \end{pmatrix} + h \begin{pmatrix} \theta_{2(n+1)} \\ -\sin[\theta_{1(n+1)}] \end{pmatrix} \quad (*)$$

对一阶方程组(*)进行数值求解有多种求解方法, 本试验共尝试两种方法:

$$\text{[方法一]: } (*) \text{ 可化为 } \begin{cases} \theta_{1(n+1)} + h^2 \sin[\theta_{1(n+1)}] = \theta_{1(n)} + h\theta_{2(n)} \\ \theta_{2(n+1)} = \theta_{2(n)} - h \sin[\theta_{1(n+1)}] \end{cases}$$

通过Matlab内置函数 `vpasolve()` 直接计算 $\theta_{1(n+1)}$ 和 $\theta_{2(n+1)}$ 的数值解

$$\text{[方法二]: } (*) \text{ 可化为 } \begin{cases} \theta_{1(n+1)} = \theta_{1(n)} + h\theta_{2(n+1)} \\ \theta_{2(n+1)} = \theta_{2(n)} - h \sin[\theta_{1(n+1)}] \end{cases}, \text{ 迭代求解 } \theta_{1(n+1)} \text{ 与 } \theta_{2(n+1)}:$$

先用显式欧拉法求初始迭代值 $\theta_{1(n+1)}^{(0)}$ 和 $\theta_{2(n+1)}^{(0)}$, 即 $\theta_{1(n+1)}^{(0)} = \theta_{1(n)} + h\theta_{2(n)}$, $\theta_{2(n+1)}^{(0)} = \theta_{2(n)} - h \sin[\theta_{1(n)}]$

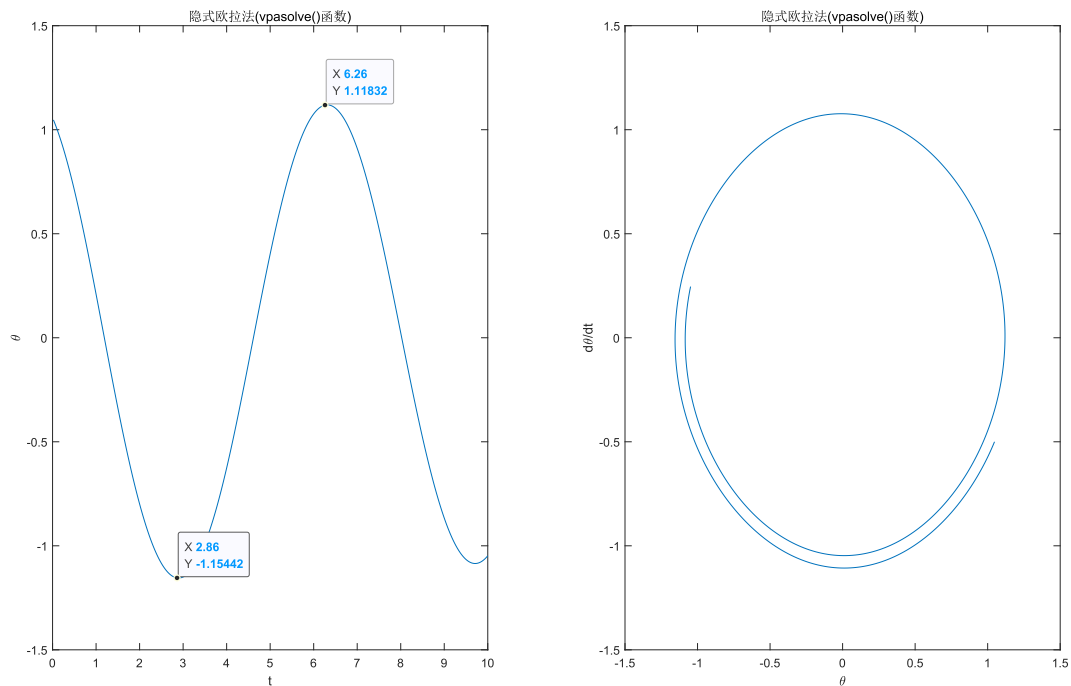
将初始迭代值代入隐式欧拉公式, 即: $\theta_{1(n+1)}^{(1)} = \theta_{1(n)} + h\theta_{2(n+1)}^{(0)}$, $\theta_{2(n+1)}^{(1)} = \theta_{2(n)} - h \sin[\theta_{1(n+1)}^{(0)}]$

$$\therefore \theta_{1(n+1)}^{(k+1)} = \theta_{1(n)} + h\theta_{2(n+1)}^{(k)}, \quad \theta_{2(n+1)}^{(k+1)} = \theta_{2(n)} - h \sin[\theta_{1(n+1)}^{(k)}]$$

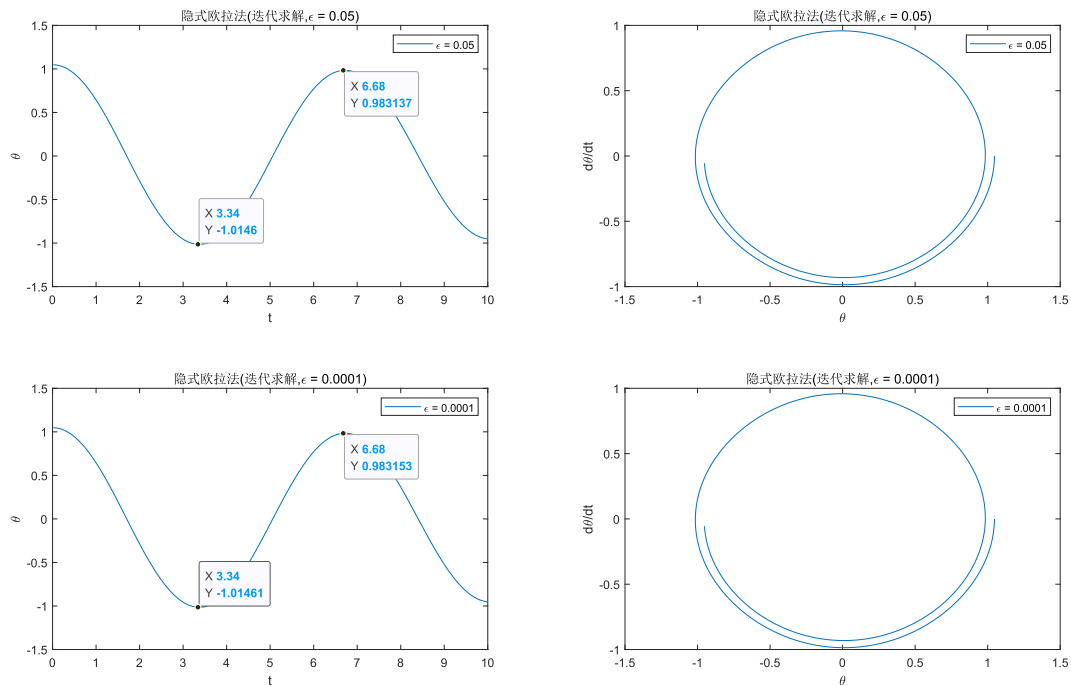
当 $|\theta_{1(n+1)}^{(k+1)} - \theta_{1(n+1)}^{(k)}| < \epsilon$, $|\theta_{2(n+1)}^{(k+1)} - \theta_{2(n+1)}^{(k)}| < \epsilon$ 时迭代终止, 然后计算 $\theta_{1(n+2)}$ 和 $\theta_{2(n+2)}$

(2). 试验结果

- [方法一]



- [方法二]



(3). 结果分析

- (i) [方法二 $\epsilon = 0.05$]和[方法二 $\epsilon = 0.0001$]的极值点和极值几乎相同
- (ii) [方法一]与[方法二 $\epsilon = 0.05$](或[方法二 $\epsilon = 0.0001$])的极值点与极值有较明显的差别,其原因是[方法一]的vpasolve()函数的原理是直接求数值解,与[方法二]存在精度上的差异
- (iii) 隐式欧拉法[方法一],[方法二]不保结构

(4). 代码

- [方法一]

```
1  clc;clf;
2  syms x;
3  h = 0.02;
4  equ = (x + h^2*sin(x)==1);%equ = sin(x)==1;
5  answ = vpasolve(equ)
6  h = 0.02;
7  n = 10/h;
8  syms x;
9  theta1 = zeros(1,n);
10 theta2 = theta1;
11 theta1(1) = pi/3;
12 theta2(1) = -1/2;
13 for i = 2:1:n
14     equ = (x + h^2*sin(x) == theta1(i-1) + h*theta2(i-1));
15     answ = vpasolve(equ);
16     theta1(i) = answ;
17     theta2(i) = theta2(i-1) - h*sin(answ);
18 end
19 subplot(1,2,2);
20 plot(theta1,theta2);
21 xlabel('\theta');
22 ylabel('d\theta/dt');
23 title('隐式欧拉法(vpasolve()函数)');
24 t = h:h:n*h;
25 subplot(1,2,1);
26 plot(t,theta1);
27 xlabel('t');
28 ylabel('\theta');
29 title('隐式欧拉法(vpasolve()函数)');
```

- [方法二] 以 $\epsilon = 0.05$ 为例：

```
28 h = 0.02;
29 n = 10/h;
30 t = h:h:n*h;
31 theta1 = zeros(1,n);
32 theta2 = theta1;
33 epsilon = 0.05;
34 theta1(1) = pi/3;
35 for i = 2:1:n
36     the1_0 = theta1(i-1) + h*theta2(i-1); % 初始迭代值
37     the2_0 = theta2(i-1) - h*sin(theta1(i-1)); % 初始迭代值
38     the1_ = theta1(i-1) + h*the2_0; % 第一次迭代值
39     the2_ = theta2(i-1) - h*sin(the1_0); % 第一次迭代值
40     while (abs(the1_1 - the1_) > epsilon) && (abs(the2_2 - the2_) > epsilon)
41         the1_1 = the1_; % 保留前一次迭代的值
42         the2_2 = the2_; % 保留前一次迭代的值
43         the1_ = theta1(i-1) + h*the2_2; % 第2,3,...次迭代值
44         the2_ = theta2(i-1) - h*sin(the1_1); % 第2,3,...次迭代值
45     end
46     theta1(i) = the1_;
47     theta2(i) = the2_;
48 end
49 subplot(2,2,1);
50 plot(t,theta1);
51 xlabel('t');
52 ylabel('\theta');
53 legend('\epsilon = 0.05')
54 title('隐式欧拉法(迭代求解,\epsilon = 0.05)')
55 subplot(2,2,2);
56 plot(theta1,theta2);
57 xlabel('\theta');
58 ylabel('d\theta/dt');
59 legend('\epsilon = 0.05')
60 title('隐式欧拉法(迭代求解,\epsilon = 0.05)')
61
```

3. 梯形法

(1). 理论分析

$$\begin{pmatrix} \theta_{1(n+1)} \\ \theta_{2(n+1)} \end{pmatrix} = \begin{pmatrix} \theta_{1(n)} \\ \theta_{2(n)} \end{pmatrix} + \frac{h}{2} \begin{pmatrix} \theta'_{1(n)} + \theta'_{1(n+1)} \\ \theta'_{2(n)} + \theta'_{2(n+1)} \end{pmatrix} = \begin{pmatrix} \theta_{1(n)} \\ \theta_{2(n)} \end{pmatrix} + \frac{h}{2} \begin{pmatrix} \theta_{2(n)} + \theta_{2(n+1)} \\ -\sin[\theta_{1(n)}] - \sin[\theta_{1(n+1)}] \end{pmatrix} \quad (**)$$

对一阶方程组(*)进行数值求解有多种求解方法, 本试验共尝试两种方法:

[方法一]: (**)可化为
$$\begin{cases} \theta_{1(n+1)} + \frac{h^2}{4} \sin[\theta_{1(n+1)}] = \theta_{1(n)} + h\theta_{2(n)} - \frac{h^2}{4} \sin[\theta_{1(n)}] \\ \theta_{2(n+1)} = \theta_{2(n)} - \frac{h}{2} (\sin[\theta_{1(n)}] + \sin[\theta_{1(n+1)}]) \end{cases}$$

通过Matlab内置函数 `vpasolve()` 直接计算 $\theta_{1(n+1)}$ 和 $\theta_{2(n+1)}$ 的数值解

[方法二]: (**)可化为
$$\begin{cases} \theta_{1(n+1)} = \theta_{1(n)} + \frac{h}{2}\theta_{2(n)} + \frac{h}{2}\theta_{2(n+1)} \\ \theta_{2(n+1)} = \theta_{2(n)} - \frac{h}{2} (\sin[\theta_{1(n)}] + \sin[\theta_{1(n+1)}]) \end{cases}$$
, 迭代求解 $\theta_{1(n+1)}$ 和 $\theta_{2(n+1)}$:

先用显式欧拉法求初始迭代值 $\theta_{1(n+1)}^{(0)}$ 和 $\theta_{2(n+1)}^{(0)}$: 即 $\theta_{1(n+1)}^{(0)} = \theta_{1(n)} + h\theta_{2(n)}$, $\theta_{2(n+1)}^{(0)} = \theta_{2(n)} - h \sin[\theta_{1(n)}]$

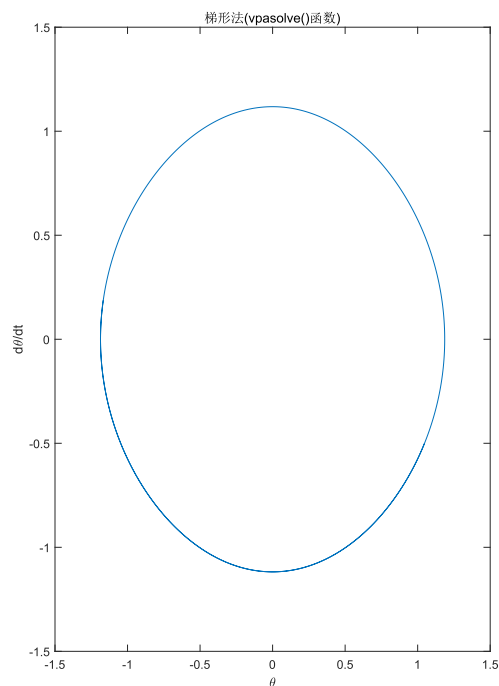
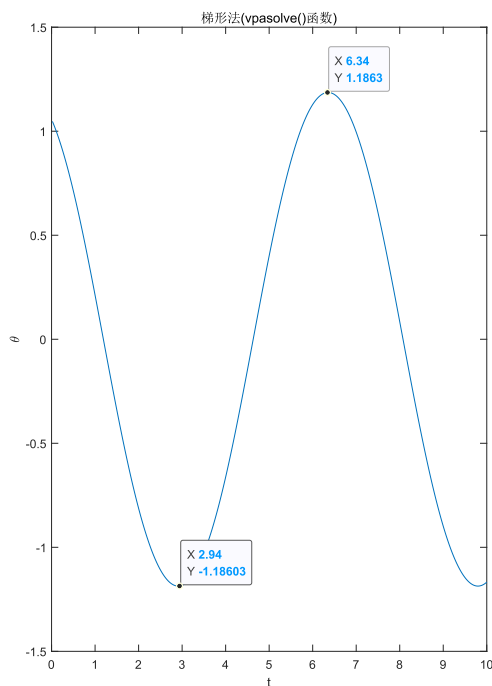
将初始迭代值代入梯形公式, 即: $\theta_{1(n+1)}^{(1)} = \theta_{1(n)} + \frac{h}{2}\theta_{2(n)} + \frac{h}{2}\theta_{2(n+1)}^{(0)}$, $\theta_{2(n+1)}^{(1)} = \theta_{2(n)} - \frac{h}{2} (\sin[\theta_{1(n)}] + \sin[\theta_{1(n+1)}^{(0)}])$

$\therefore \theta_{1(n+1)}^{(k+1)} = \theta_{1(n)} + \frac{h}{2}\theta_{2(n)} + \frac{h}{2}\theta_{2(n+1)}^{(k)}$, $\theta_{2(n+1)}^{(k+1)} = \theta_{2(n)} - \frac{h}{2} (\sin[\theta_{1(n)}] + \sin[\theta_{1(n+1)}^{(k)}])$

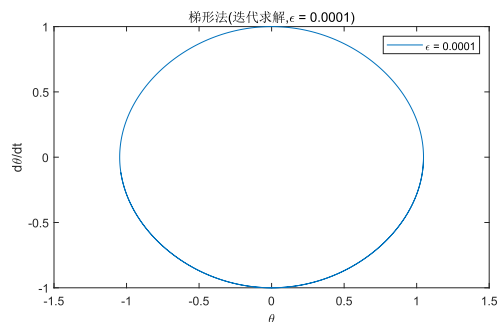
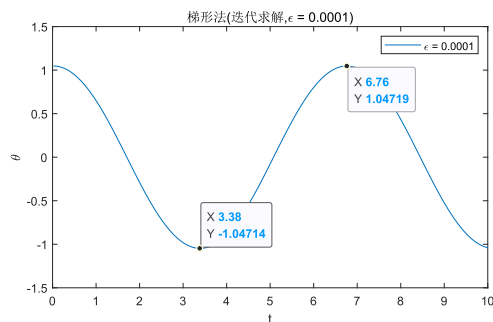
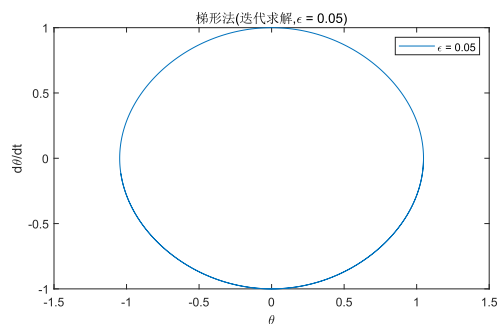
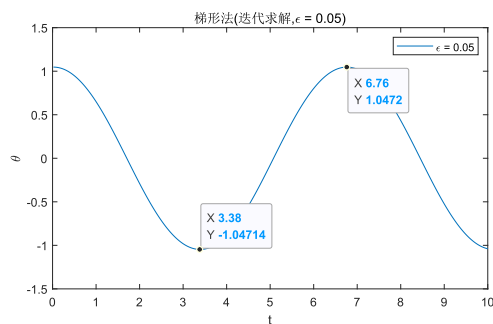
当 $|\theta_{1(n+1)}^{(k+1)} - \theta_{1(n+1)}^{(k)}| < \epsilon$, $|\theta_{2(n+1)}^{(k+1)} - \theta_{2(n+1)}^{(k)}| < \epsilon$ 时迭代终止, 然后计算 $\theta_{1(n+2)}$ 和 $\theta_{2(n+2)}$

(2). 试验结果

- [方法一]



- [方法二]



(3). 结果分析

- (i) [方法二 $\epsilon = 0.05$] 和 [方法二 $\epsilon = 0.0001$] 的极值点和极值几乎相同
- (ii) [方法一] 与 [方法二 $\epsilon = 0.05$] (或 [方法二 $\epsilon = 0.0001$]) 的极值点与极值有较明显的差别, 其原因是 [方法一] 的 `vpasolve()` 函数的原理是直接求数值解, 与 [方法二] 存在精度上的差异
- (iii) 与隐式欧拉法相比, 梯形法 [方法二 $\epsilon = 0.05$] 和 [方法二 $\epsilon = 0.0001$] 的极值的差异更小 :
 隐式欧拉法极值差异为 : $t = 3.34$ 处相差 $-1.0146 - (-1.01461) = 0.00001$; $t = 6.68$ 处相差 $0.983137 - 0.983153 = -0.000015$
 梯形法极值差异为 : $t = 3.38$ 处相差 $-1.04714 - (-1.04714) = 0$; $t = 6.76$ 处相差 $1.0472 - 1.04719 = 0.00001$
 推测梯形公式的迭代收敛速率大于隐式欧拉公式的迭代收敛速率
- (iv) 梯形法 [方法一], [方法二] 保结构

(4). 代码

- [方法一]

```

1  clf;
2  h = 0.02;
3  n = 10/h;
4  syms x;
5  theta1 = zeros(1,n);
6  theta2 = theta1;
7  theta1(1) = pi/3;
8  theta2(1) = -1/2;
9  for i = 2:1:n
10     equ = (x + h^2*sin(x)/4 == theta1(i-1) + h*theta2(i-1) - h^2*sin(theta1(i-1))/4);
11     ans = vpasolve(equ);
12     theta1(i) = ans;
13     theta2(i) = theta2(i-1) - (h/2)*(sin(theta1(i-1)) + sin(ans));
14 end
15 subplot(1,2,2);
16 plot(theta1,theta2);
17 xlabel('\theta');
18 ylabel('d\theta/dt');
19 title('梯形法(vpasolve()函数)');
20 t = h:n*h;
21 subplot(1,2,1);
22 plot(t,theta1);
23 xlabel('t');
24 ylabel('\theta');
25 title('梯形法(vpasolve()函数)');

```

- [方法二] 以 $\epsilon = 0.05$ 为例：

```
27 h = 0.02;
28 n = 10/h;
29 t = h:h:n*h;
30 theta1 = zeros(1,n);
31 theta2 = theta1;
32 epsilon = 0.05;
33 theta1(1) = pi/3;
34 for i = 2:1:n
35     the1_0 = theta1(i-1) + h*theta2(i-1); % 初始迭代值
36     the2_0 = theta2(i-1) - h*sin(theta1(i-1)); % 初始迭代值
37     the1_ = theta1(i-1) + (h/2)*theta2(i-1) + (h/2)*the2_0; % 第一次迭代值
38     the2_ = theta2(i-1) - (h/2)*(sin(the1_0)+sin(theta1(i-1))); % 第一次迭代值
39     while (abs(the1_1 - the1_) > epsilon) && (abs(the2_2 - the2_) > epsilon)
40         the1_1 = the1_; % 保留前一次迭代的值
41         the2_2 = the2_; % 保留前一次迭代的值
42         the1_ = theta1(i-1) + (h/2)*theta2(i-1) + (h/2)*the2_2; % 第2,3,...次迭代值
43         the2_ = theta2(i-1) - (h/2)*(sin(the1_1) + sin(theta1(i-1))); % 第2,3,...次迭代值
44     end
45     theta1(i) = the1_;
46     theta2(i) = the2_;
47 end
48 subplot(2,2,1);
49 plot(t,theta1);
50 xlabel('t');
51 ylabel('\theta');
52 legend('\epsilon = 0.05')
53 title('梯形法(迭代求解,\epsilon = 0.05)')
54 subplot(2,2,2);
55 plot(theta1,theta2);
56 xlabel('\theta');
57 ylabel('d\theta/dt');
58 legend('\epsilon = 0.05')
59 title('梯形法(迭代求解,\epsilon = 0.05)')
```