PRACTICE 1 Report
From Xu Ziyang, 22320607

Overview:

1.Equations and graph corresponding to each system
2.Matlab Things
3.Calculate code in Wolfram Language

Part 1:

Equation1:

$$y'(t) + ky(t) = 0$$

Which means:

$$y'(t) + 7y(t) = 0$$

Character Equation:

$$\lambda + 7 = 0$$

Character Roots:

$$\{\lambda = -7\}$$

General Solution:

$$y = c_1 e^{-7t}$$

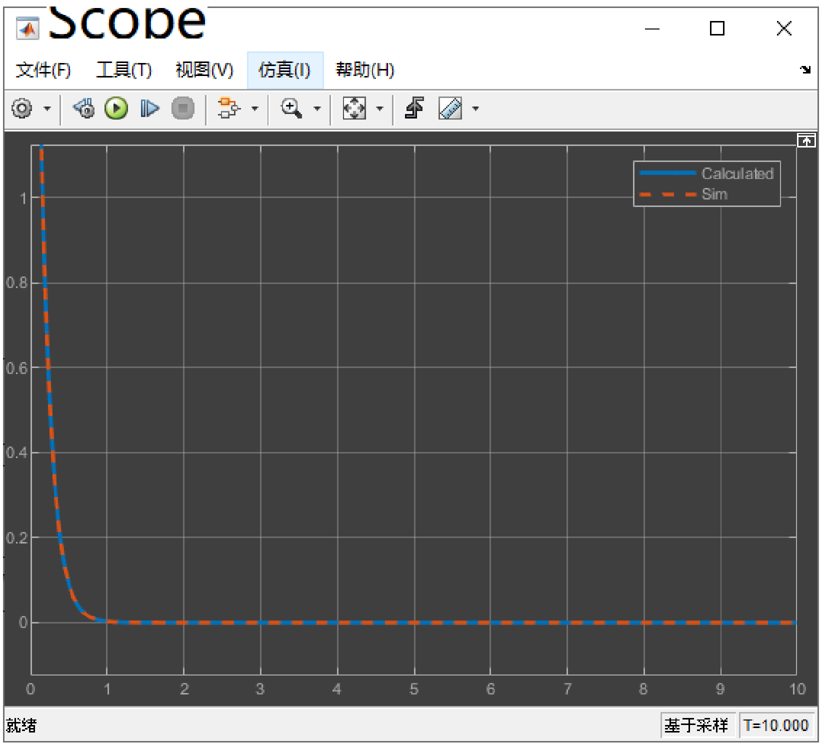
Initial Condition:

$$y(0) = 3$$

Solution:

$$y = 3e^{-7t}$$

Graph:**(You can see bugs with Windows, just forgive me and Microsoft…)**



Equation2:

$$y''(t) + (k+1)y'(t) + ky(t) = 0$$

Which means:

$$y''(t) + 8y'(t) + 7y(t) = 0$$

Character Equation:

$$\lambda^2 + 8\lambda + 7 = 0$$

Character Roots:

$$\{\lambda = -7, \lambda = -1\}$$

General Solution:

$$y = c_1 e^{-7t} + c_2 e^{-t}$$

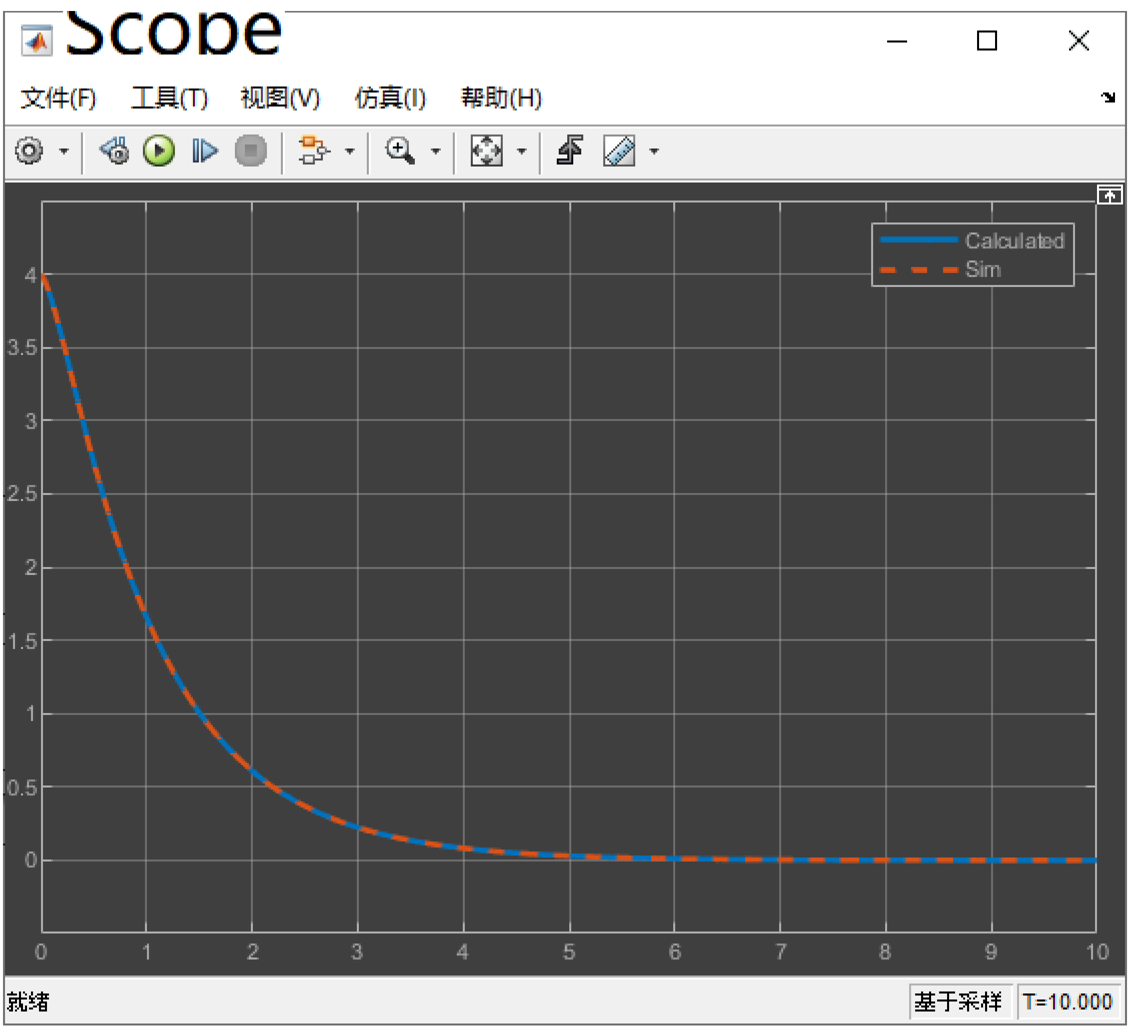
Initial Condition:

$$y(0) = 4, y'(0) = -1$$

Solution:

$$y = 1/2 * e^{-7t} (9e^{6t} - 1)$$

Graph:



Equation3:

$$y''(t) + k^2 y(t) = 0$$

Which means:

$$y''(t) + 49\, y(t) = 0$$

Character Equation:

$$\lambda^2 + 49 = 0$$

Character Roots:

$$\{\lambda = -7i,\ \lambda = 7i\}$$

General Solution:

$$y = c_1 \sin(7t) + c_2 \cos(7t)$$

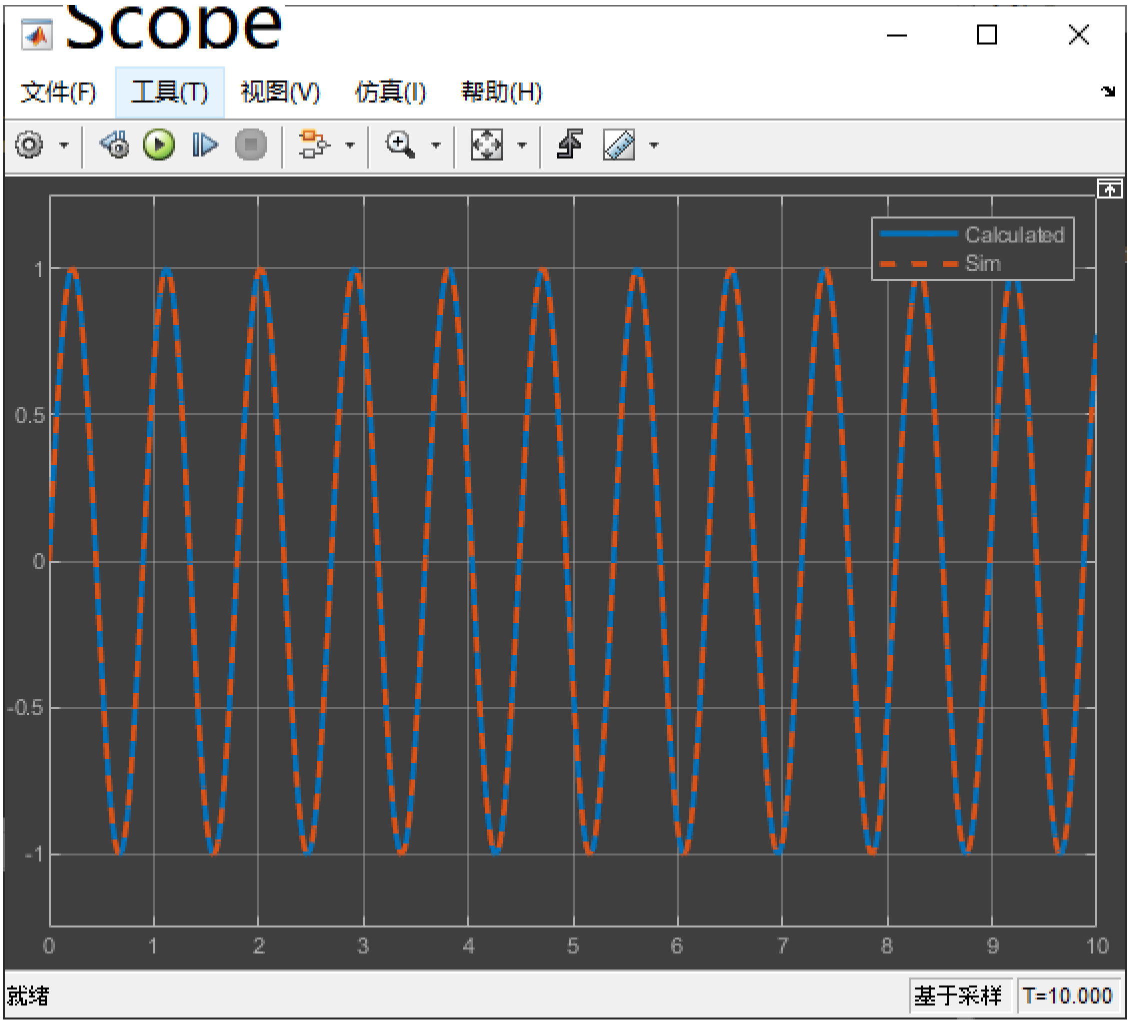
Initial Condition:

$$y(0) = 0,\ y'(0) = -7$$

Solution:

$$y = \sin(7t)$$

Graph:



Equation4:

$$y''(t) + 4y'(t) + 13y(t) = 0$$

Character Equation:

$$\lambda^2 + 4\lambda + 13 = 0$$

Character Roots:

$$\{\lambda = -2-3i,\ \lambda = -2+3i\}$$

General Solution:

$$y = e^{-2t}(c_1\sin(3t) + c_2\cos(3t))$$

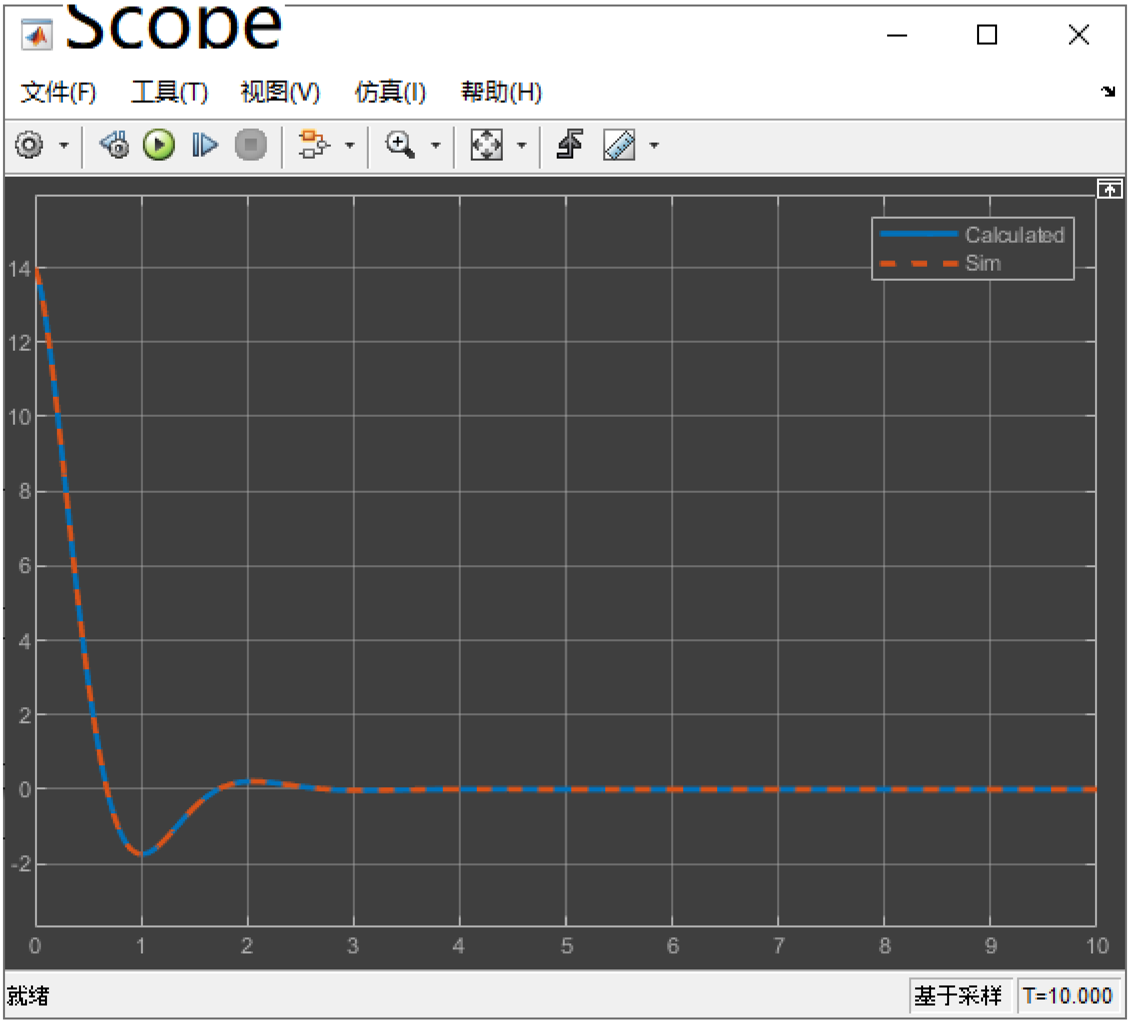
Initial Condition:

$$y(0) = 14,\ y'(0) = -7$$

Solution:

$$y = 7e^{-2t}(\sin(3t) + 2\cos(3t))$$

Graph:



Equation5:

Notice that y(t) is not 0(t) since y(0) is not 0.

$$y''(t) + ay'(t) + ky(t) = 0$$

Which means:

$$y''(t) + ay'(t) + 7y(t) = 0$$

Character Equation:

$$\lambda^2 + a\lambda + 7 = 0$$

Character Roots:

$$\left\{\text{lambda} \to \frac{1}{2}\left(-\sqrt{a^2-28} - a\right)\right\}, \left\{\text{lambda} \to \frac{1}{2}\left(\sqrt{a^2-28} - a\right)\right\}$$

General Solution:

$$\left\{\left\{y \to \left(\{t\} \mapsto c_1 e^{\frac{1}{2}\left(-\sqrt{a^2-28} - a\right)t} + c_2 e^{\frac{1}{2}\left(\sqrt{a^2-28} - a\right)t}\right)\right\}\right\}$$

Taking ***a* = 2*sqrt(7)(Let delta be zero)**:

Taking Initial Condition:

$$y(0) = 1, y'(0) = 0$$

Solution:

$$\left\{\left\{y \to \left(\{t\} \mapsto e^{-\sqrt{7}\,t}\left(\sqrt{7}\,t + 1\right)\right)\right\}\right\}$$

Taking ***a* = 2*sqrt(5)**:

Taking Initial Condition:

$$y(0) = 1, y'(0) = 0$$

Solution:

$$\left\{\left\{y \to \left(\{t\} \mapsto \frac{1}{2} e^{-\sqrt{5}\,t}\left(\sqrt{10}\sin\left(\sqrt{2}\,t\right) + 2\cos\left(\sqrt{2}\,t\right)\right)\right)\right\}\right\}$$

Taking ***a* = 6**:

Taking Initial Condition:

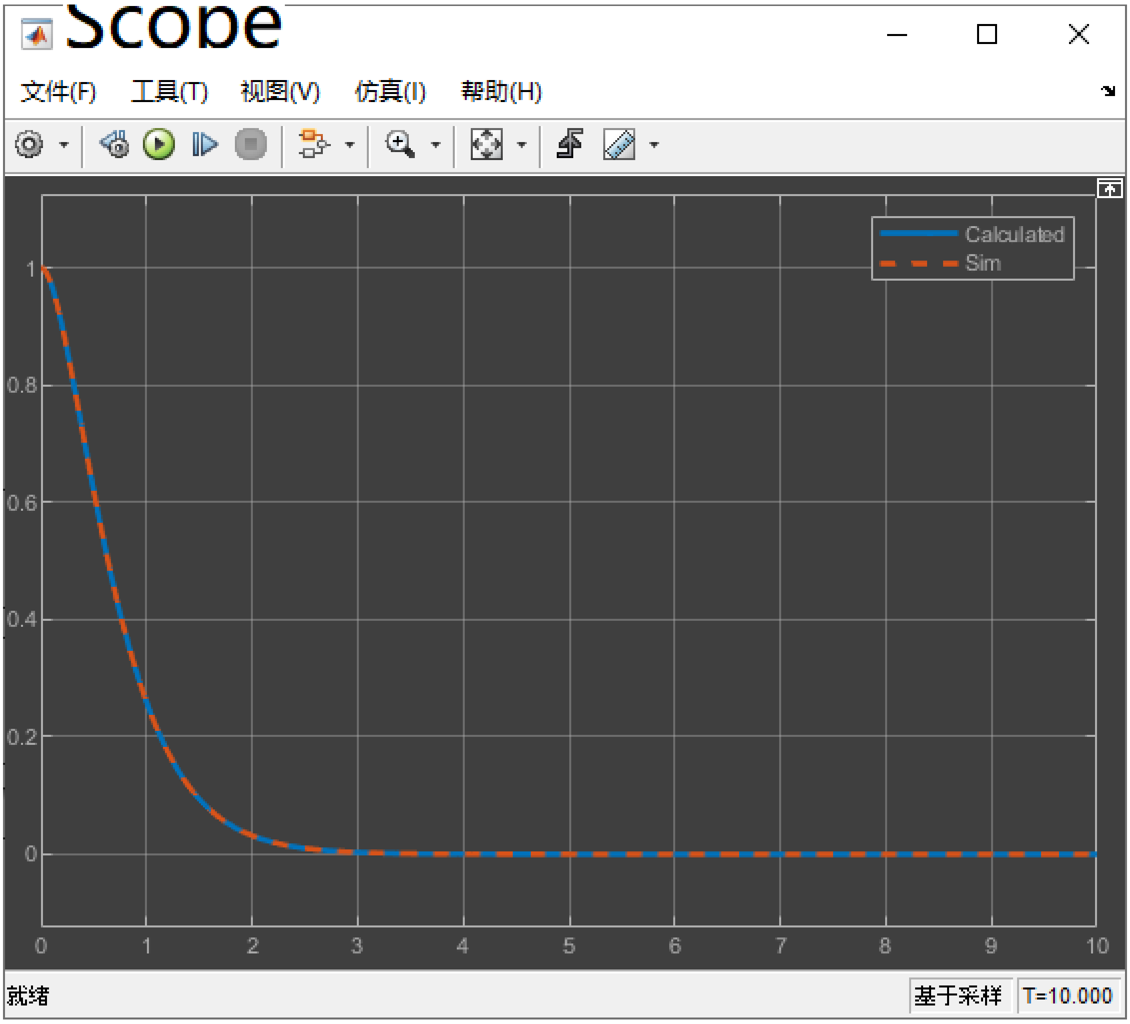
$$y(0) = 1, y'(0) = 0$$

Solution:

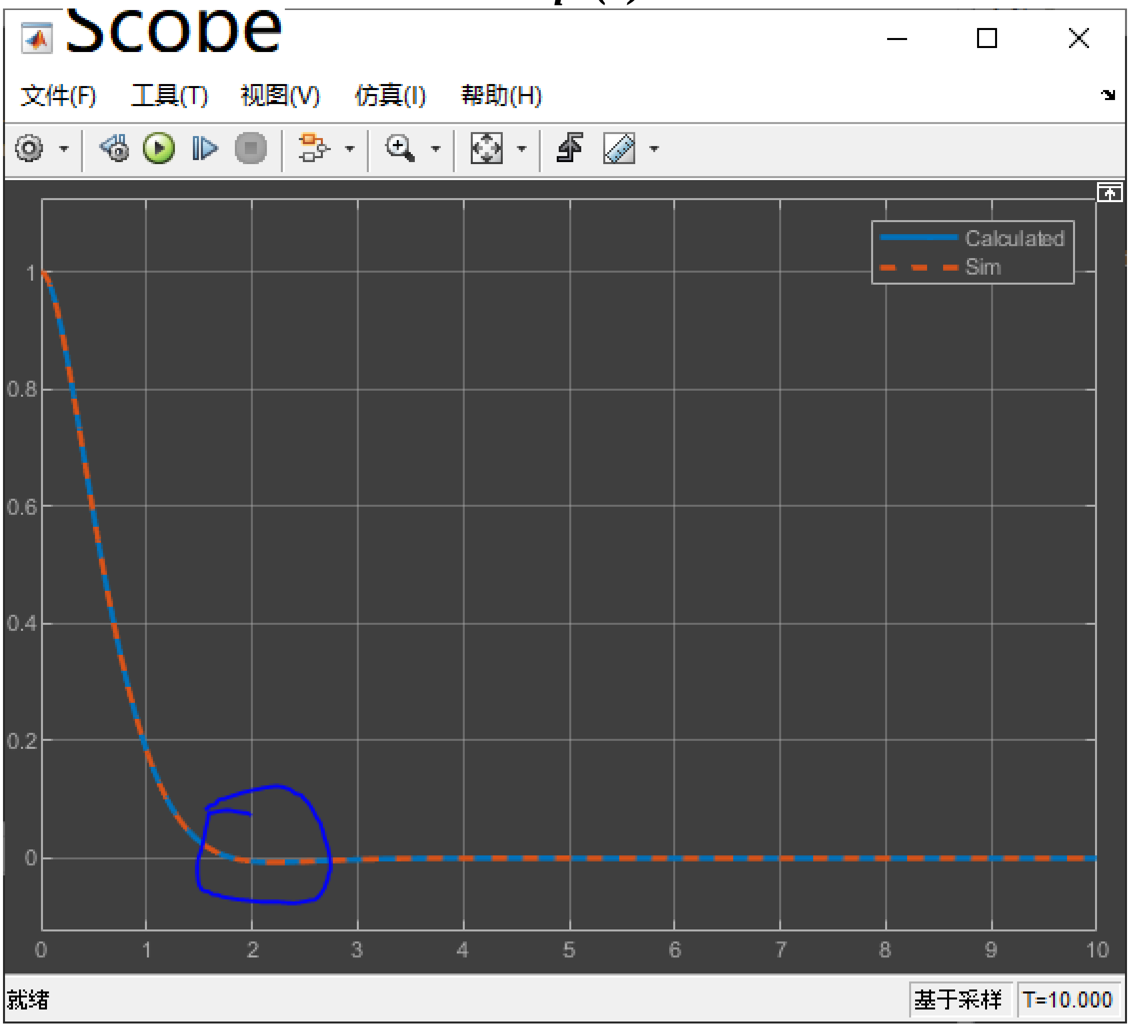
$$\left\{\left\{y \to \left(\{t\} \mapsto \frac{1}{4}\left(2 e^{\left(-3-\sqrt{2}\right)t} - 3\sqrt{2}\, e^{\left(-3-\sqrt{2}\right)t} + 2 e^{\left(\sqrt{2}-3\right)t} + 3\sqrt{2}\, e^{\left(\sqrt{2}-3\right)t}\right)\right)\right\}\right\}$$

Graphs:

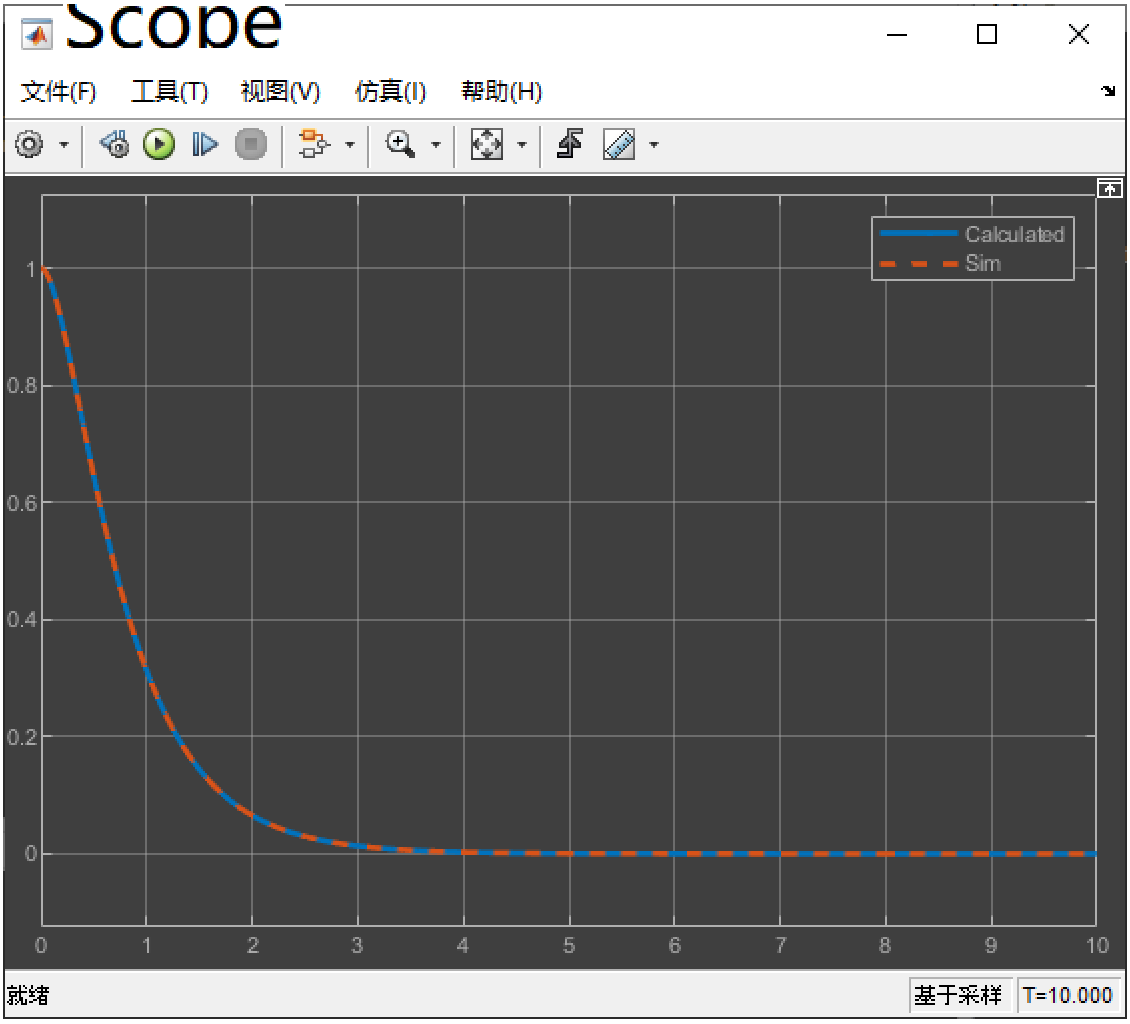
a=2*sqrt(7):



a=2*sqrt(5):


Scope
文件(F)
工具(T)
视图(V)
仿真(I)
帮助(H)
Calculated
Sim
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
10
就绪
基于采样
T=10.000

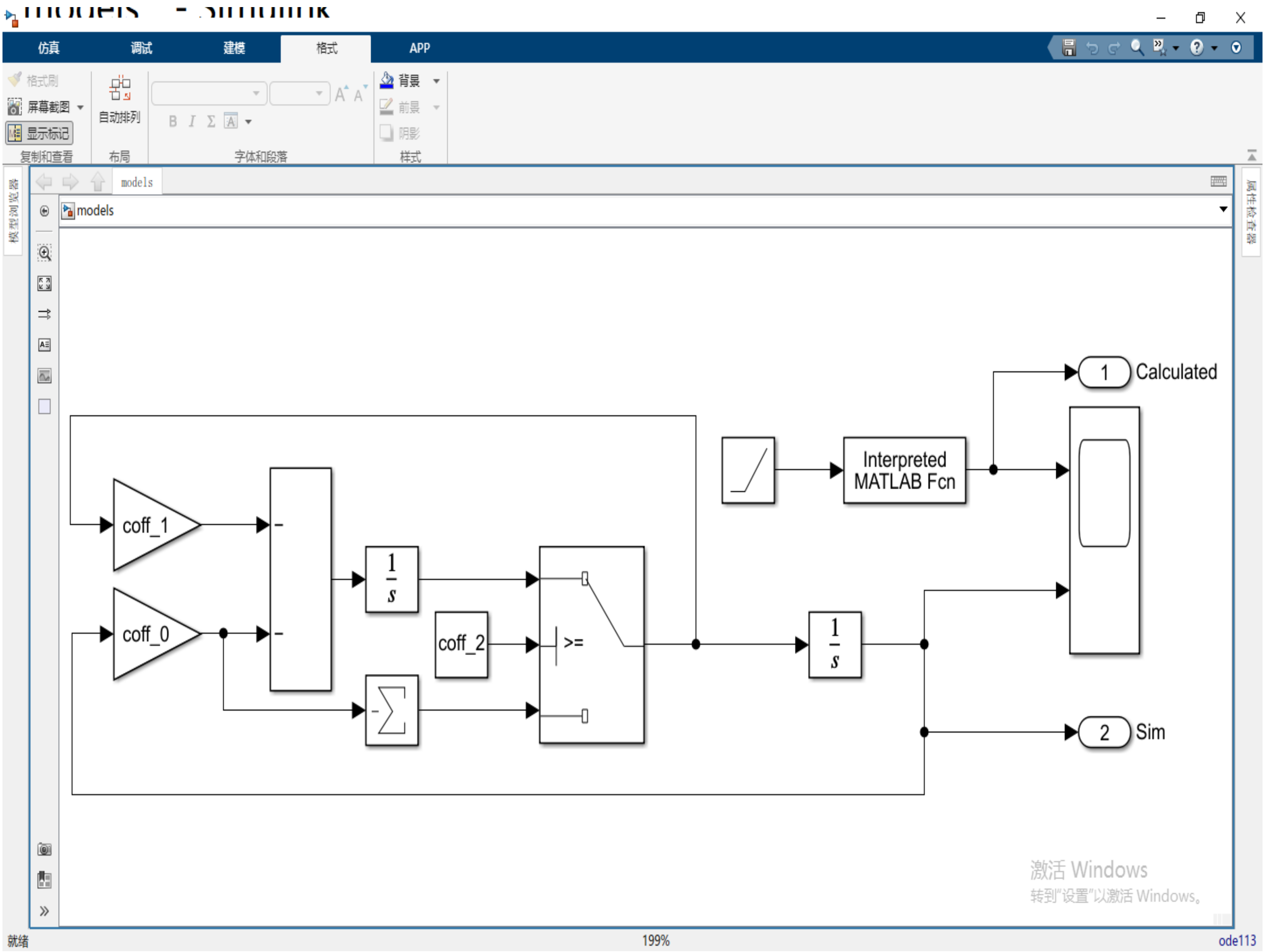
a=6:



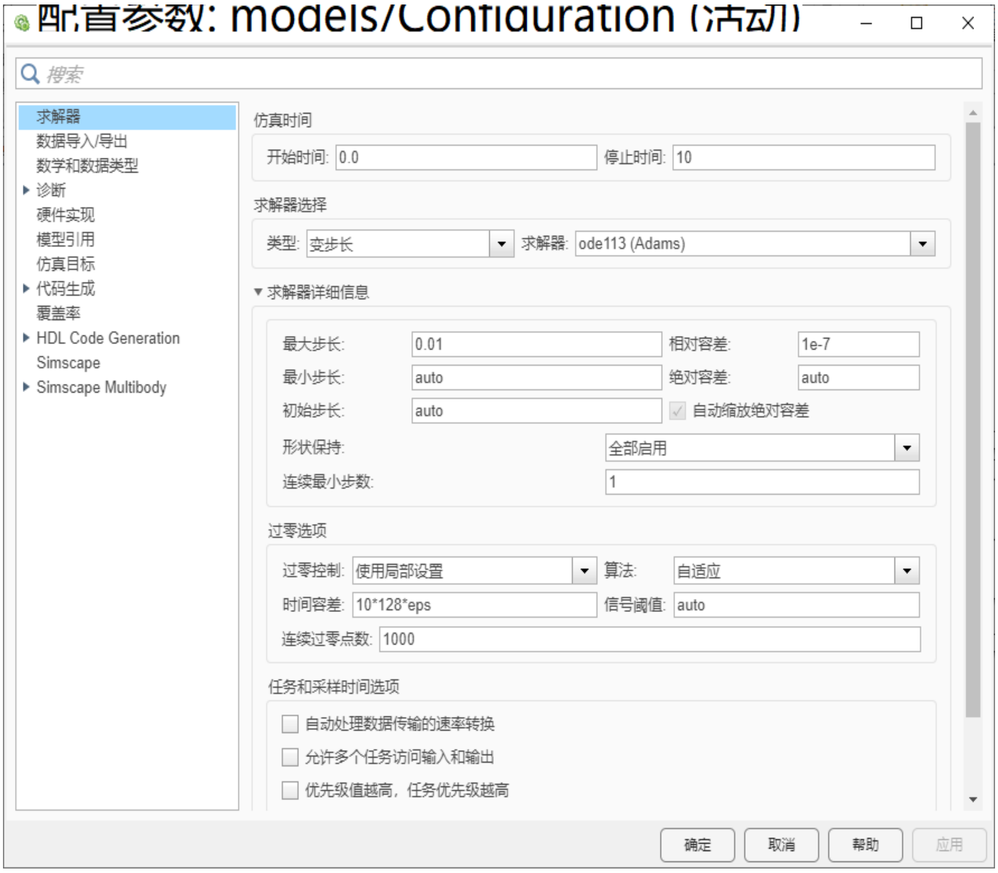
Scope
文件(F)
工具(T)
视图(V)
仿真(I)
帮助(H)
Calculated
Sim
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
10
就绪
基于采样
T=10.000

Part 2:

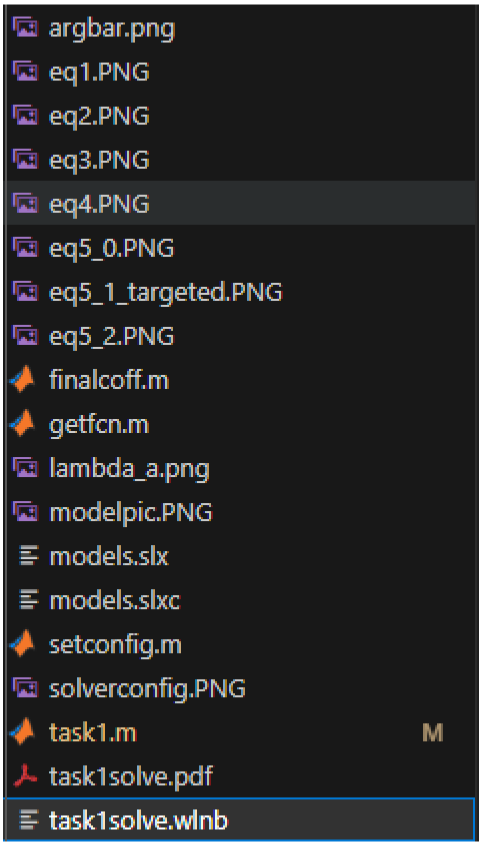
Simulink Model:



Solver Configure:

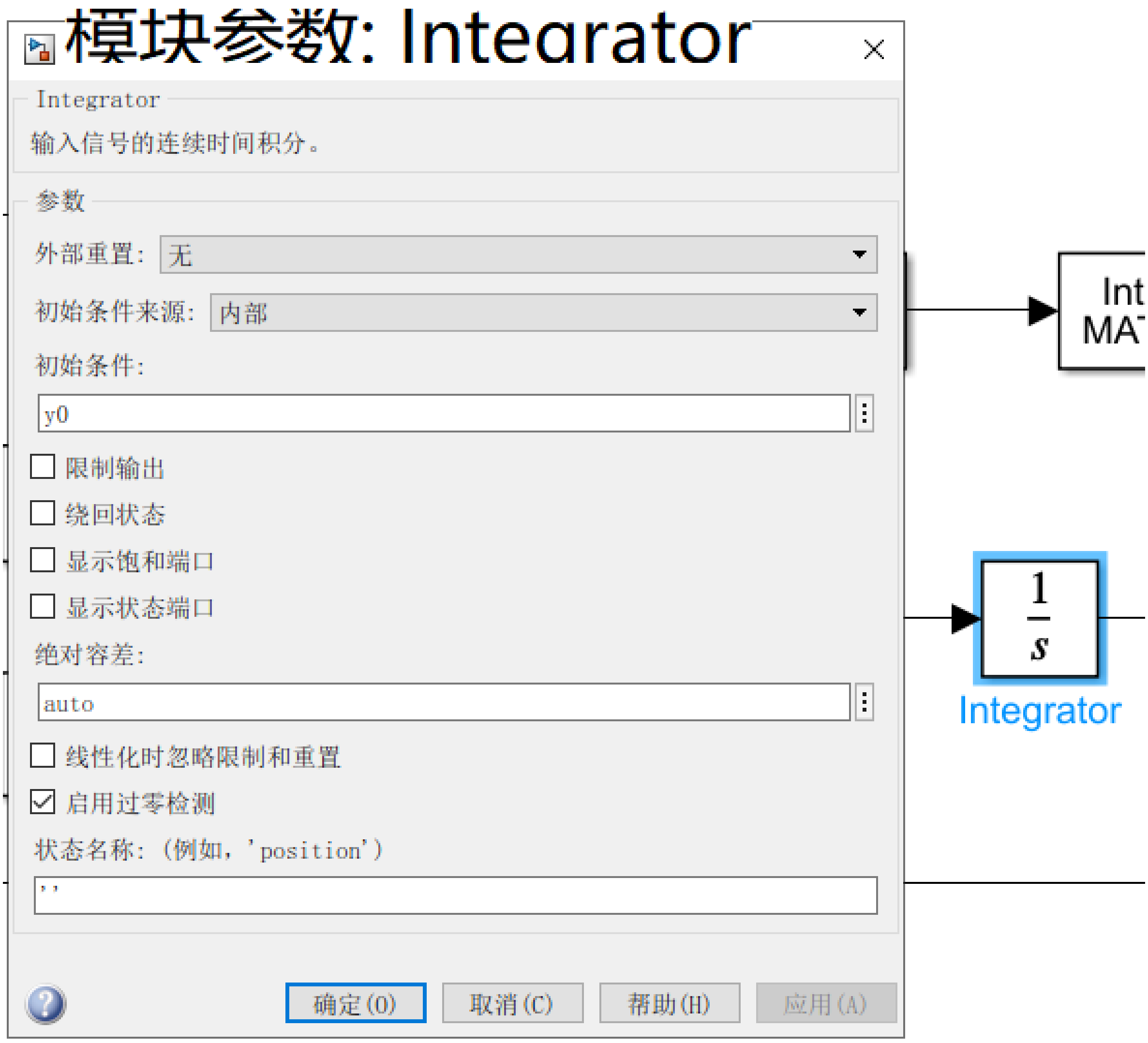

配置参数: models/Configuration (活动)
搜索
求解器
数据导入/导出
数学和数据类型
诊断
硬件实现
模型引用
仿真目标
代码生成
覆盖率
HDL Code Generation
Simscape
Simscape Multibody
仿真时间
开始时间: 0.0
停止时间: 10
求解器选择
类型: 变步长
求解器: ode113 (Adams)
求解器详细信息
最大步长: 0.01
相对容差: 1e-7
最小步长: auto
绝对容差: auto
初始步长: auto
自动缩放绝对容差
形状保持: 全部启用
连续最小步数: 1
过零选项
过零控制: 使用局部设置
算法: 自适应
时间容差: 10*128*eps
信号阈值: auto
连续过零点数: 1000
任务和采样时间选项
自动处理数据传输的速率转换
允许多个任务访问输入和输出
优先级值越高，任务优先级越高
确定
取消
帮助
应用

Repository:


argbar.png
eq1.PNG
eq2.PNG
eq3.PNG
eq4.PNG
eq5_0.PNG
eq5_1_targeted.PNG
eq5_2.PNG
finalcoff.m
getfcn.m
lambda_a.png
modelpic.PNG
models.slx
models.slxc
setconfig.m
solverconfig.PNG
task1.m
M
task1solve.pdf
task1solve.wlnb

Model info:


模块参数: Interpret...
Interpreted MATLAB Function
将输入值传递给 MATLAB 函数进行计算。该函数必须返回单个值，该值具有由 '输出维度' 和 '将二维结果折叠为一维' 指定的维度。
示例: sin、sin(u)、foo(u(1), u(2))
参数
MATLAB 函数:
fcn(u)
输出维度:
-1
输出信号类型: 自动
将二维结果降为一维
确定(O)
取消(C)
帮助(H)
应用(A)


模块参数: Integrator
Integrator
输入信号的连续时间积分。
参数
外部重置: 无
初始条件来源: 内部
初始条件:
y0
限制输出
绕回状态
显示饱和端口
显示状态端口
绝对容差:
auto
线性化时忽略限制和重置
启用过零检测
状态名称: (例如，'position')
''
确定(O)
取消(C)
帮助(H)
应用(A)
1
s
Integrator


模块参数: Integrator1
Integrator
输入信号的连续时间积分。
参数
外部重置: 无
初始条件来源: 内部
初始条件:
y0dot1
限制输出
绕回状态
显示饱和端口
显示状态端口
绝对容差:
auto
线性化时忽略限制和重置
启用过零检测
状态名称: (例如, 'position')
''
确定(O)
取消(C)
帮助(H)
应用(A)
Integrator1

Codes(Why LibreOffice cannot copy format correctly?):

finalcoff.m:

```
function [coff_dot2,coff_dot1,coff_nodot] = finalcoff(modeltesting,k,a)
    switch modeltesting
        case 1
            coff_dot1 = 1;
            coff_dot2 = 0;
            coff_nodot = k;
        case 2
            coff_dot1 = k+1;
            coff_dot2 = 1;
            coff_nodot = k;
        case 3
            coff_dot1 = 0;
            coff_dot2 = 1;
            coff_nodot = k^2;
        case 4
            coff_dot1 = 4;
            coff_dot2 = 1;
            coff_nodot = 13;
        case 5+0
            coff_dot1 = a;
            coff_dot2 = 1;
            coff_nodot = 7;
        case 5+1
            coff_dot1 = a;
            coff_dot2 = 1;
            coff_nodot = 7;
        case 5+2
            coff_dot1 = a;
            coff_dot2 = 1;
            coff_nodot = k;
    end
end
```

getfcn.m:

```
function f = getfcn(modeltesting)
    switch modeltesting
        case 1
            f = @(t)3*exp(-7*t);
        case 2
            f = @(t)exp(-7*t)/2*(9*exp(6*t)-1);
        case 3
            f = @(t)sin(7*t);
        case 4
            f = @(t)7*exp(-2*t)*(sin(3*t)+2*cos(3*t));
        case (5+0)
            f = @(t)exp(-sqrt(7)*t)*(sqrt(7)*t+1);
        case (5+1)
            f = @(t)1/2*exp(-sqrt(5)*t)*(sqrt(10)*sin(t*sqrt(2))+2*cos(t*sqrt(2)));
        case (5+2)
            f = @(t)1/4*(2*exp((-3-sqrt(2))*t)-3*sqrt(2)*exp((-3-sqrt(2))*t)
+2*exp((sqrt(2)-3)*t)+3*sqrt(2)*exp((sqrt(2)-3)*t));
    end
end
```

setconfig.m:

```
function [y0,y0dot1,a] = setconfig(modeltesting,k)
    switch modeltesting
        case 1
            y0 = 3;
            y0dot1= 0;
            a = 0;
        case 2
            y0 = 4;
            y0dot1 = -1;
            a = 0;
        case 3
            y0 = 0;
            y0dot1 = k;
            a = 0;
        case 4
            y0 = 2*k;
            y0dot1 = -k;
            a = 0;
        case 5+0
            y0 = 1;
            y0dot1 = 0;
            a=sqrt(28);
        case 5+1
            y0 = 1;
            y0dot1 = 0;
            a=sqrt(20);
        case 5+2
            y0 = 1;
            y0dot1 = 0;
            a=sqrt(36);
    end
end
```

task1.m(No tabs so it’s working):

```
k = 7;
modeltesting = 7;%changes for each model
[y0,y0dot1,a]=setconfig(modeltesting,k);
[coff_2,coff_1,coff_0]=finalcoff(modeltesting,k,a);
fcn = getfcn(modeltesting);
```

Part 3:

按 Esc 退出全屏

In[•]:=
```
Remove[a, x, y, lambda]
```

In[•]:=
```
k = 7
TraditionalForm[equation1 = {y'[t] + y[t] * k == 0}]
TraditionalForm[charaEq1 = {x + k == 0}]
TraditionalForm[Solve[charaEq1, x]]
TraditionalForm[DSolve[equation1, y, t]]
TraditionalForm[init1 = {y[0] == 3}]
TraditionalForm[DSolve[{equation1, init1}, y, t]]
```

Out[•]=

7

Out[•]//TraditionalForm=

$\{y'(t) + 7\, y(t) = 0\}$

Out[•]//TraditionalForm=

$\{x + 7 = 0\}$

Out[•]//TraditionalForm=

$\{\{x \to -7\}\}$

Out[•]//TraditionalForm=

$\{\{y \to (\{t\} \mapsto c_1\, e^{-7\,t})\}\}$

Out[•]//TraditionalForm=

$\{y(0) = 3\}$

Out[•]//TraditionalForm=

$\{\{y \to (\{t\} \mapsto 3\, e^{-7\,t})\}\}$

In[•]:=
```
TraditionalForm[equation2 = {y''[t] + (k + 1) * y'[t] + k * y[t] == 0}]
TraditionalForm[charaEq2 = {x^2 + (k + 1) * x + k == 0}]
TraditionalForm[Solve[charaEq2]]
TraditionalForm[DSolve[equation2, y, t]]
TraditionalForm[init2 = {y[0] == 4, y'[0] == -1}]
TraditionalForm[DSolve[{equation2, init2}, y, t]]
```

Out[•]//TraditionalForm=

$\{y''(t) + 8\, y'(t) + 7\, y(t) = 0\}$

Out[•]//TraditionalForm=

$\{x^2 + 8\, x + 7 = 0\}$

Out[•]//TraditionalForm=

$\{\{x \to -7\}, \{x \to -1\}\}$

Out[•]//TraditionalForm=

$\{\{y \to (\{t\} \mapsto c_1\, e^{-7\,t} + c_2\, e^{-t})\}\}$

Out[•]//TraditionalForm=

$\{y(0) = 4,\ y'(0) = -1\}$

Out[•]//TraditionalForm=

$$\left\{\left\{y \to \left(\{t\} \mapsto \frac{1}{2}\, e^{-7\,t}\,(9\, e^{6\,t} - 1)\right)\right\}\right\}$$

```
In[•]:= TraditionalForm[equation3 = {y''[t] + k^2 * y[t] == 0}]
TraditionalForm[charaEq3 = {x^2 + k^2 == 0}]
TraditionalForm[Solve[charaEq3, x]]
TraditionalForm[DSolve[equation3, y, t]]
TraditionalForm[init3 = {y[0] == 0, y'[0] == k}]
TraditionalForm[DSolve[{equation3, init3}, y, t]]
```

Out[•]//TraditionalForm=
$\{y''(t) + 49\, y(t) = 0\}$

Out[•]//TraditionalForm=
$\{x^2 + 49 = 0\}$

Out[•]//TraditionalForm=
$\{\{x \to -7\, i\}, \{x \to 7\, i\}\}$

Out[•]//TraditionalForm=
$\{\{y \to (\{t\} \mapsto c_1 \cos(7\, t) + c_2 \sin(7\, t))\}\}$

Out[•]//TraditionalForm=
$\{y(0) = 0,\ y'(0) = 7\}$

Out[•]//TraditionalForm=
$\{\{y \to (\{t\} \mapsto \sin(7\, t))\}\}$

```
In[•]:= TraditionalForm[equation4 = {y''[t] + 4 * y'[t] + 13 * y[t] == 0}]
TraditionalForm[charaEq4 = {x^2 + 4 * x + 13 == 0}]
TraditionalForm[Solve[charaEq4, x]]
TraditionalForm[DSolve[equation4, y, t]]
TraditionalForm[init4 = {y[0] == 2 * k, y'[0] == -k}]
TraditionalForm[DSolve[{equation4, init4}, y, t]]
```

Out[•]//TraditionalForm=
$\{y''(t) + 4\, y'(t) + 13\, y(t) = 0\}$

Out[•]//TraditionalForm=
$\{x^2 + 4\, x + 13 = 0\}$

Out[•]//TraditionalForm=
$\{\{x \to -2 - 3\, i\}, \{x \to -2 + 3\, i\}\}$

Out[•]//TraditionalForm=
$\{\{y \to (\{t\} \mapsto c_2\, e^{-2\, t} \cos(3\, t) + c_1\, e^{-2\, t} \sin(3\, t))\}\}$

Out[•]//TraditionalForm=
$\{y(0) = 14,\ y'(0) = -7\}$

Out[•]//TraditionalForm=
$\{\{y \to (\{t\} \mapsto 7\, e^{-2\, t} (\sin(3\, t) + 2 \cos(3\, t)))\}\}$

In[•]:=
```
TraditionalForm[Solve[lambda^2 + a * lambda + k == 0, lambda]]
TraditionalForm[Solve[lambda^2 + a * lambda + k == 0, a]]
ComplexPlot3D[(-p^2 - 8) / p, {p, -10 - 10 * Sqrt[2] I, 10 + 10 I}, PlotLegends -> Automatic]
TraditionalForm[Solve[- (lambda^2 + 8) == 0]]
TraditionalForm[DSolve[y''[t] + a * y'[t] + k * y[t] == 0, y, t]]
TraditionalForm[DSolve[y''[t] + k * y[t] == 0, y, t]]
a = Sqrt[28]
TraditionalForm[DSolve[{y''[t] + a * y'[t] + k * y[t] == 0, y[0] == 1, y'[0] == 0}, y, t]]
a = Sqrt[20]
TraditionalForm[DSolve[{y''[t] + a * y'[t] + k * y[t] == 0, y[0] == 1, y'[0] == 0}, y, t]]
a = Sqrt[36]
TraditionalForm[DSolve[{y''[t] + a * y'[t] + k * y[t] == 0, y[0] == 1, y'[0] == 0}, y, t]]
```

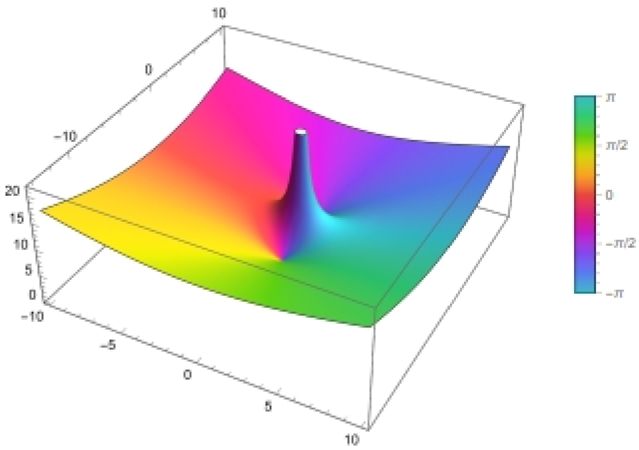
Out[•]//TraditionalForm=

$$\left\{\left\{\text{lambda} \to \frac{1}{2}\left(-\sqrt{a^2-28}-a\right)\right\}, \left\{\text{lambda} \to \frac{1}{2}\left(\sqrt{a^2-28}-a\right)\right\}\right\}$$

Out[•]//TraditionalForm=

$$\left\{\left\{a \to \frac{-\text{lambda}^2-7}{\text{lambda}}\right\}\right\}$$

Out[•]=



Out[•]//TraditionalForm=

$$\{\{\text{lambda} \to -2\, i\, \sqrt{2}\}, \{\text{lambda} \to 2\, i\, \sqrt{2}\}\}$$

Out[•]//TraditionalForm=

$$\left\{\left\{y \to \left(\{t\} \mapsto c_1\, e^{\frac{1}{2}\left(-\sqrt{a^2-28}-a\right)t} + c_2\, e^{\frac{1}{2}\left(\sqrt{a^2-28}-a\right)t}\right)\right\}\right\}$$

Out[•]//TraditionalForm=

$$\{\{y \to (\{t\} \mapsto c_1 \cos(\sqrt{7}\, t) + c_2 \sin(\sqrt{7}\, t))\}\}$$

Out[•]=

$2\sqrt{7}$

Out[•]//TraditionalForm=

$$\left\{\left\{y \to \left(\{t\} \mapsto e^{-\sqrt{7}\, t}\left(\sqrt{7}\, t + 1\right)\right)\right\}\right\}$$

Out[∘]=

$2\sqrt{5}$

Out[∘]//TraditionalForm=

$$\left\{\left\{y \to \left(\{t\} \longmapsto \frac{1}{2} e^{-\sqrt{5}\, t} \left(\sqrt{10} \sin(\sqrt{2}\, t) + 2 \cos(\sqrt{2}\, t)\right)\right)\right\}\right\}$$

Out[∘]=

6

Out[∘]//TraditionalForm=

$$\left\{\left\{y \to \left(\{t\} \longmapsto \frac{1}{4} \left(2 e^{(-3-\sqrt{2})t} - 3\sqrt{2}\, e^{(-3-\sqrt{2})t} + 2 e^{(\sqrt{2}-3)t} + 3\sqrt{2}\, e^{(\sqrt{2}-3)t}\right)\right)\right\}\right\}$$