Software to Support Differential Equations Classes

Guihong Fan

Department of Mathematics and Philosophy

Columbus State University

Presentation at Teaching and RPG Seminar at CSU

Outline

- Introduction to Matlab (**Matrix laboratory**).
- Student Project Lab 1: Functions and plotting.
- Student Project Lab 2: Solving linear system.
- Student Project Lab 3: Solving differential equations
- Create animation with Matlab; Other free software.

Why do we need Matlab?

- Matlab is used heavily in industry in some areas.
- Engineering departments frequently rely on Matlab.
- Matlab has definitely useful things for some applied mathematicians.

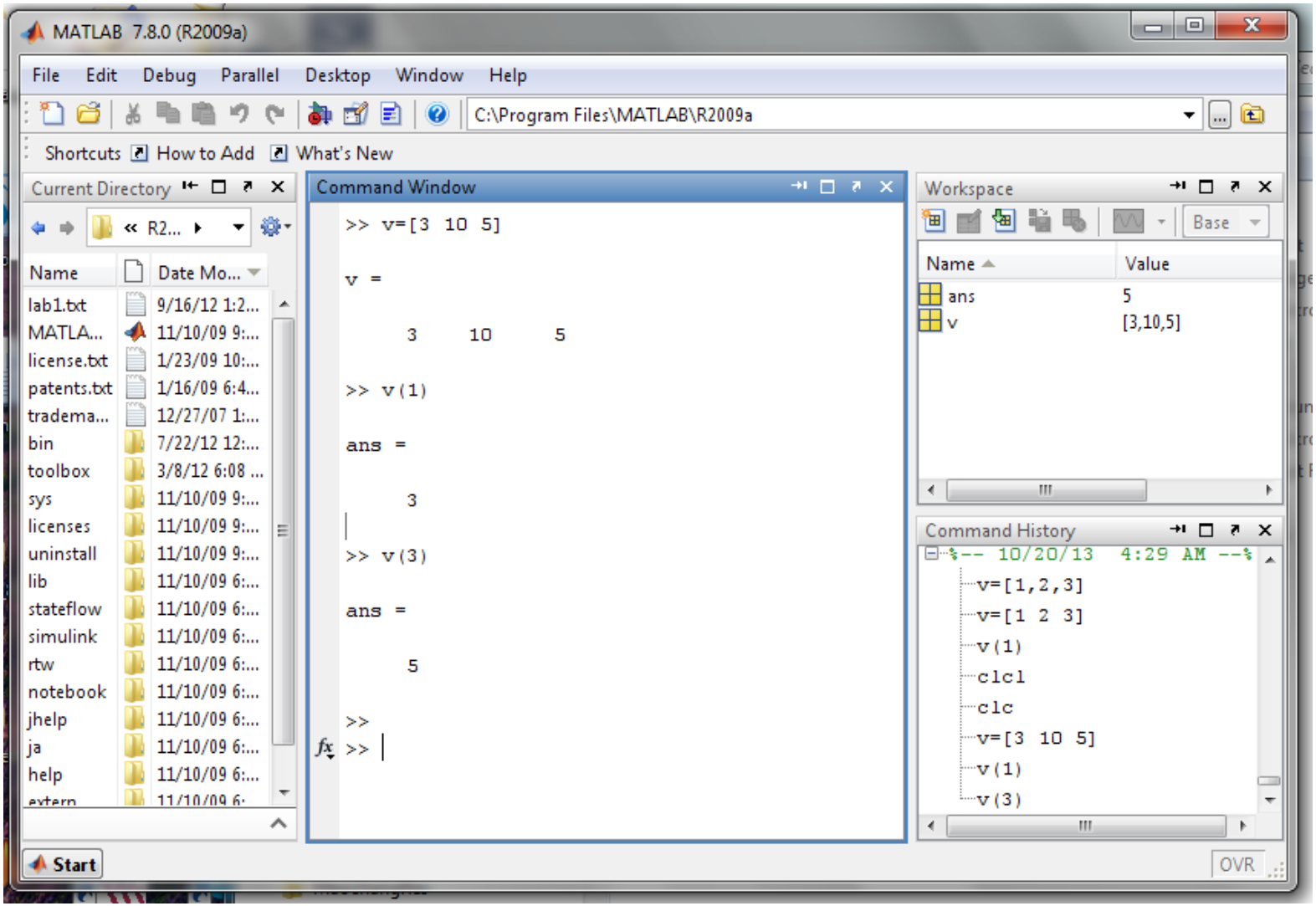
Matlab's advantages

- Extremely fast coding: It has lots of build-in functions.
- Debugging is easy and fast
- Powerful plotting tools: easy to edit figures.
- Documentation: clear and easily accessible.
- Matlab has a wide range of "toolboxes": Maple toolbox, DDeBiftool, MatCont etc.

What Matlab can do?

- Matlab is good at numerical computations and graphics.
- Matlab produces approximate rather than exact solutions
- Matlab is designed for matrix computations:
  - Solving systems of linear equations
  - Computing eigenvalues and eigenvectors
  - Factoring matrices, and so forth

Starting Matlab



Lab 1: Vectors

Basics And Help

Commands are entered in the Command Window.

★ Basic operations are `+`, `-`, `*`, and `/`. The sequence

```
>> a=2; b=3; a+b, a*b
ans =
     5
ans =
     6
```

★ Standard functions can be invoked using their usual mathematical notations.

```
>> theta=pi/5;
>> cos(theta)^2+sin(theta)^2
ans =
     1
```

Help in Matlab

- A list of elementary math functions:

```
>> help elfun
```

- To obtain a description of the use of a particular function:

```
>> help cosh
```

- To get a list of other groups of MATLAB programs

```
>> help
```

Plotting

- The function

$$y = \frac{x^2 - \sin(\pi x) + e^x}{x - 1}$$

- Input variable:

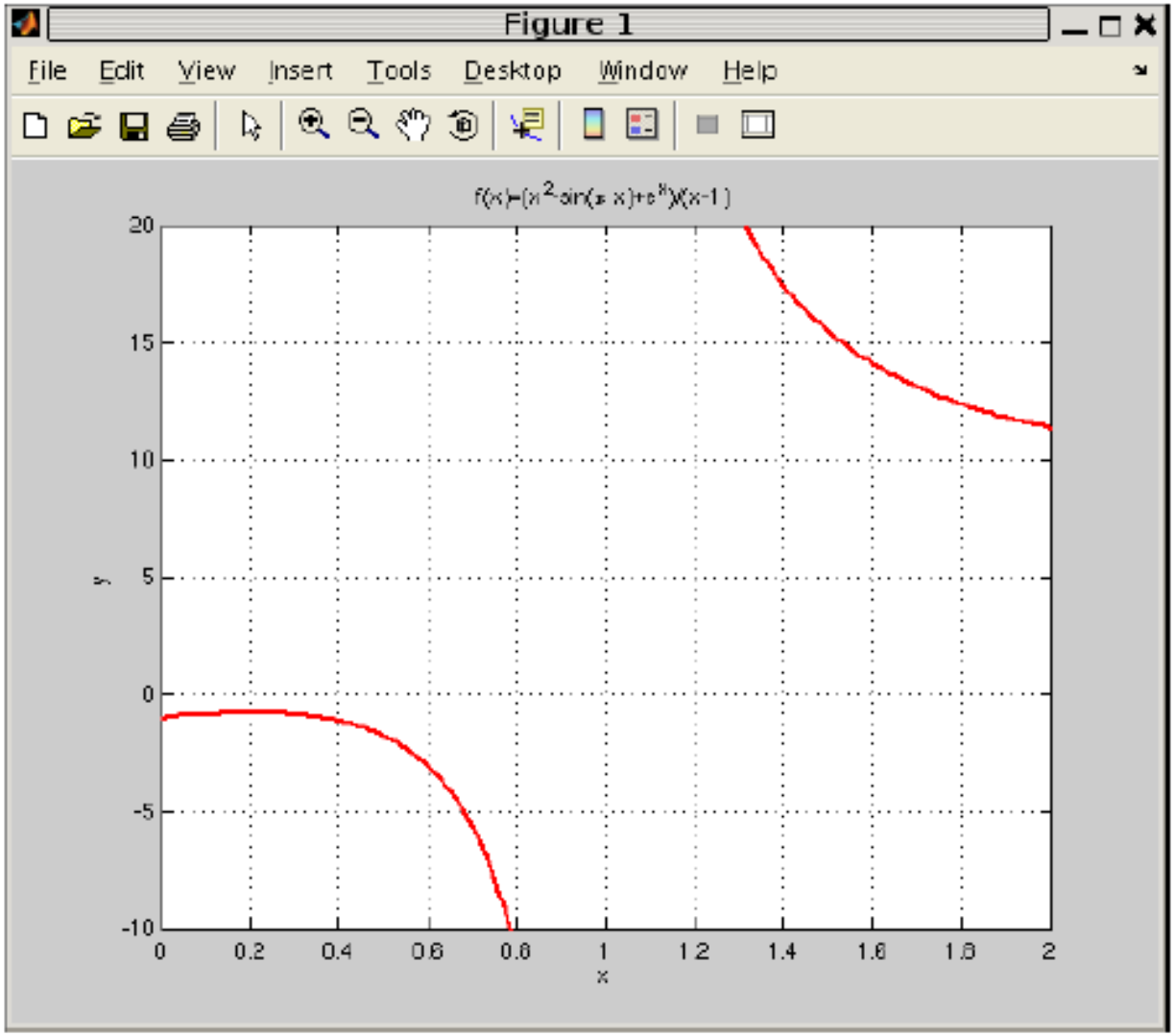
```
>> x=0:.1:2
```

```
x =
 Columns 1 through 7
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
 Columns 8 through 14
    0.7000    0.8000    0.9000    1.0000    1.1000    1.2000    1.3000
 Columns 15 through 21
    1.4000    1.5000    1.6000    1.7000    1.8000    1.9000    2.0000
```

- Input function

```
>> y=(x.^2-sin(pi.*x)+exp(x))./(x-1)
```

Plotting: ≫plot(x,y)



Generate Lab report easily

- Type "diary lab1.txt"
- Type any commands for your computation
- Then type "diary off"
- Matlab generate your lab report lab1.txt:

```
x=0:0.2:2

x =

  Columns 1 through 7

         0    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000

  Columns 8 through 11

    1.4000    1.6000    1.8000    2.0000

y=sin(x);
plot(x,y)
diary off
```

- Easy to access any element of a matrix

Lab 2: Matrix

★ Matrices can be constructed in MATLAB in different ways

$A = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$ can be entered as

```
>> A=[8,1,6;3,5,7;4,9,2]
A =
     8     1     6
     3     5     7
     4     9     2
```

Lab 2: Matrix

- Easy to access any element of a matrix

```
>> A(2,3)  % coefficient of A in 2nd row, 3rd column
ans =
    7
>> A(1,:)  % 1st row of A
ans =
    8     1     6
>> A(:,3)  % 3rd column of A
ans =
    6
    7
    2
```

Matrix manipulation

- Matrices can be manipulated very easily in MATLAB

```
>> A+B      % sum of A and B
ans =
    16     4    10
     4    10    16
    10    16     4
>> A*B      % standard linear algebra matrix multiplication
ans =
   101    71    53
    71    83    71
    53    71   101
>> A.*B     % coefficientwise multiplication
ans =
    64     3    24
     3    25    63
    24    63     4
```

Matrix manipulation

- Eigenvalues and eigenvectors:

```
>> [S,D]=eig(A)
S =
  -0.5774   -0.8131   -0.3416
  -0.5774    0.4714   -0.4714
  -0.5774    0.3416    0.8131
D =
  15.0000         0         0
         0    4.8990         0
         0         0   -4.8990
```

Easy to solve $A * x = b$

```
>> A=magic(3)
A =
     8     1     6
     3     5     7
     4     9     2
```

- Column vector $b = [28; 34; 28]$.

Easy to solve $A * x = b$

- Solution $x =$

```
>> inv(A)*b

ans =

    1.0000

    2.0000

    3.0000
```

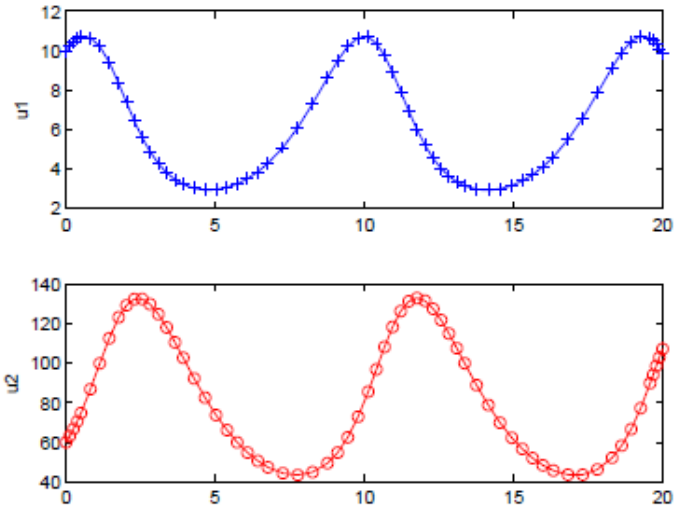
Lab 3: Solving Differential Equations

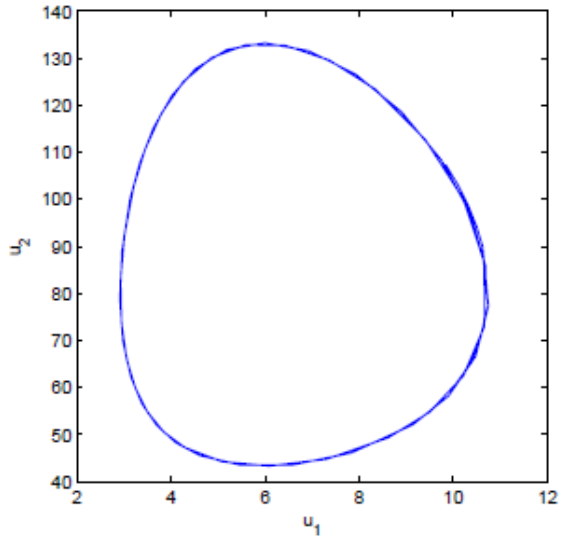
- A predator prey system:

$$\begin{aligned} \frac{\mathrm{d}u_1}{\mathrm{d}t} &= au_1(t) - bu_2(t) \\ \frac{\mathrm{d}u_1}{\mathrm{d}t} &= -cu_1(t) + du_2(t) \end{aligned}$$

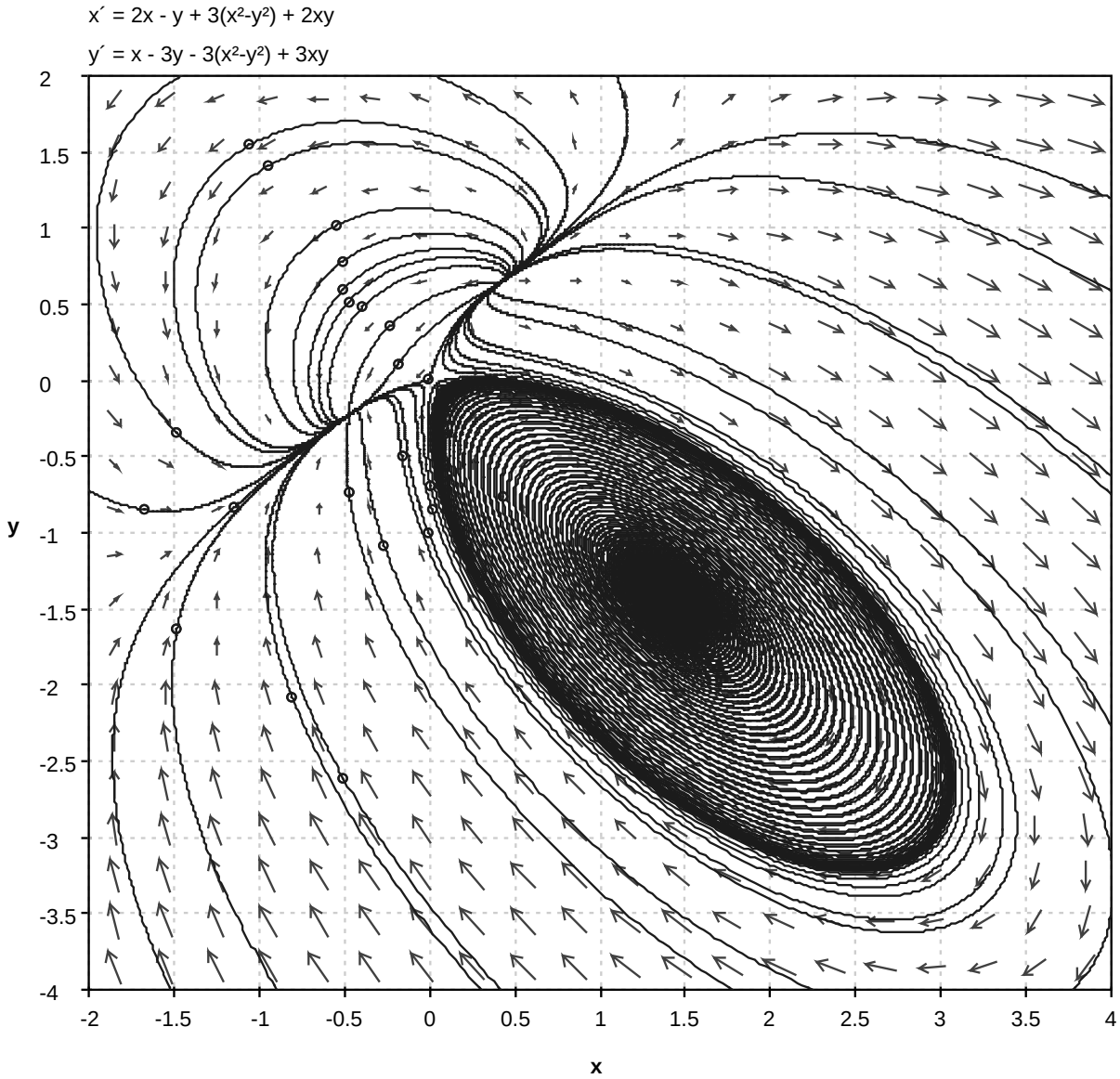
```
 1 function ex_with_2eqs
 2 t0 = 0; tf = 20; y0 = [10;60];
 3 a = .8; b = .01; c = .6; d = .1;
 4 [t,y] = ode45(@f,[t0,tf],y0,[],a,b,c,d);
 5 u1 = y(:,1); u2 = y(:,2);    % y in output has 2 columns corresponding to u1 and u2
 6 figure(1);
 7 subplot(2,1,1); plot(t,u1,'b-+'); ylabel('u1');
 8 subplot(2,1,2); plot(t,u2,'ro-');  ylabel('u2');
 9 figure(2)
10 plot(u1,u2); axis square; xlabel('u_1'); ylabel('u_2');  % plot the phase plot
11 %-----------------------------------------------------------------------
12 function dydt = f(t,y,a,b,c,d)
13 u1 = y(1); u2 = y(2);
14 dydt = [ a*u1-b*u1*u2 ; -c*u2+d*u1*u2 ];
```

Lab 3: Differential Equations



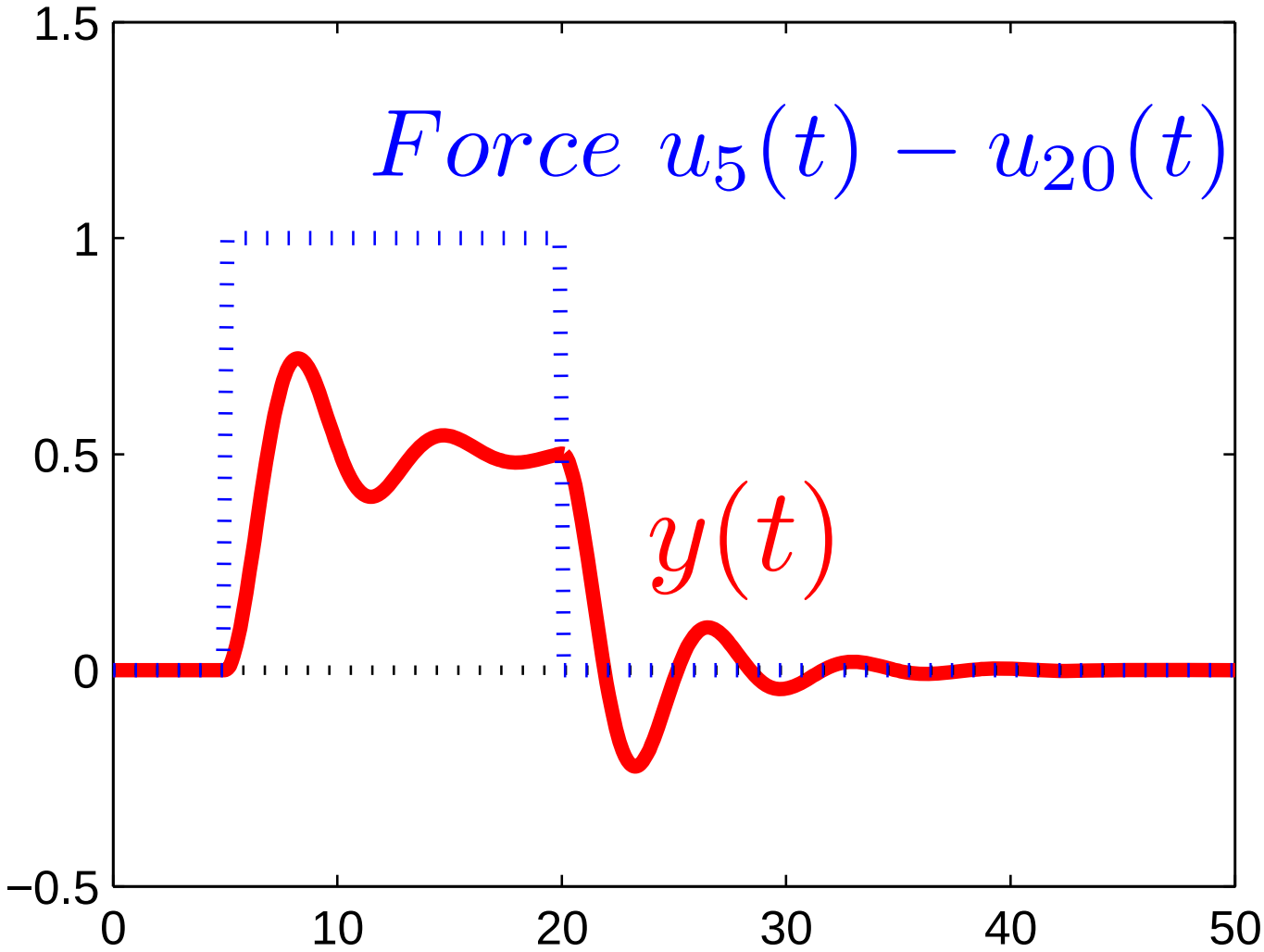


Other free software:Dfields and PPlane



Easy to produce Animation

Easy to edit your figures



Good in bifurcation analysis:MatCont