Computational Methods in Physics (PHY 365)

FA23

Dr. Muhammad Kamran

Department of Physics

COMSATS University Islamabad

Lab 23

Second order ODE

- $\frac{d^2y}{dx^2} + a\frac{dy}{dx} + b\,y^2 = x; \quad y(0) = 0, y'(0) = 1$

Second order ODE

- $\frac{d^2y}{dx^2} + a\frac{dy}{dx} + b\, y^2 = x; \quad y(0) = 0, y'(0) = 1$

- Initial and final x

```
x_i = 0;

x_f = 5 * pi;

x = [x_i , x_f];
```

Second order ODE

- $\frac{d^2y}{dx^2} + a\frac{dy}{dx} + b\,y^2 = x; \quad y(0) = 0, y'(0) = 1$

- Initial and final x

```
x_i = 0;

x_f = 5 * pi;

x = [x_i , x_f];
```

- The parameters a and b

```
a = 10  ^  -2;

b = 1;
```

Second order ODE

- Initial conditions

  IC = [0 , 1];

Second order ODE

- Initial conditions

  IC = [0 , 1];

- Calling the ODE solver

  [X , Y] = ode45 (@ func_7 , x , IC);

Second order ODE

- Initial conditions

  IC = [0 , 1];

- Calling the ODE solver

  [X , Y] = ode45 (@ func_7 , x , IC);

- Extracting y from Y

  y = Y(: , 1);

Second order ODE

- Initial conditions

  IC = [0 , 1];

- Calling the ODE solver

  [X , Y] = ode45 (@ func_7 , x , IC);

- Extracting y from Y

  y = Y(: , 1);

- Plotting the result

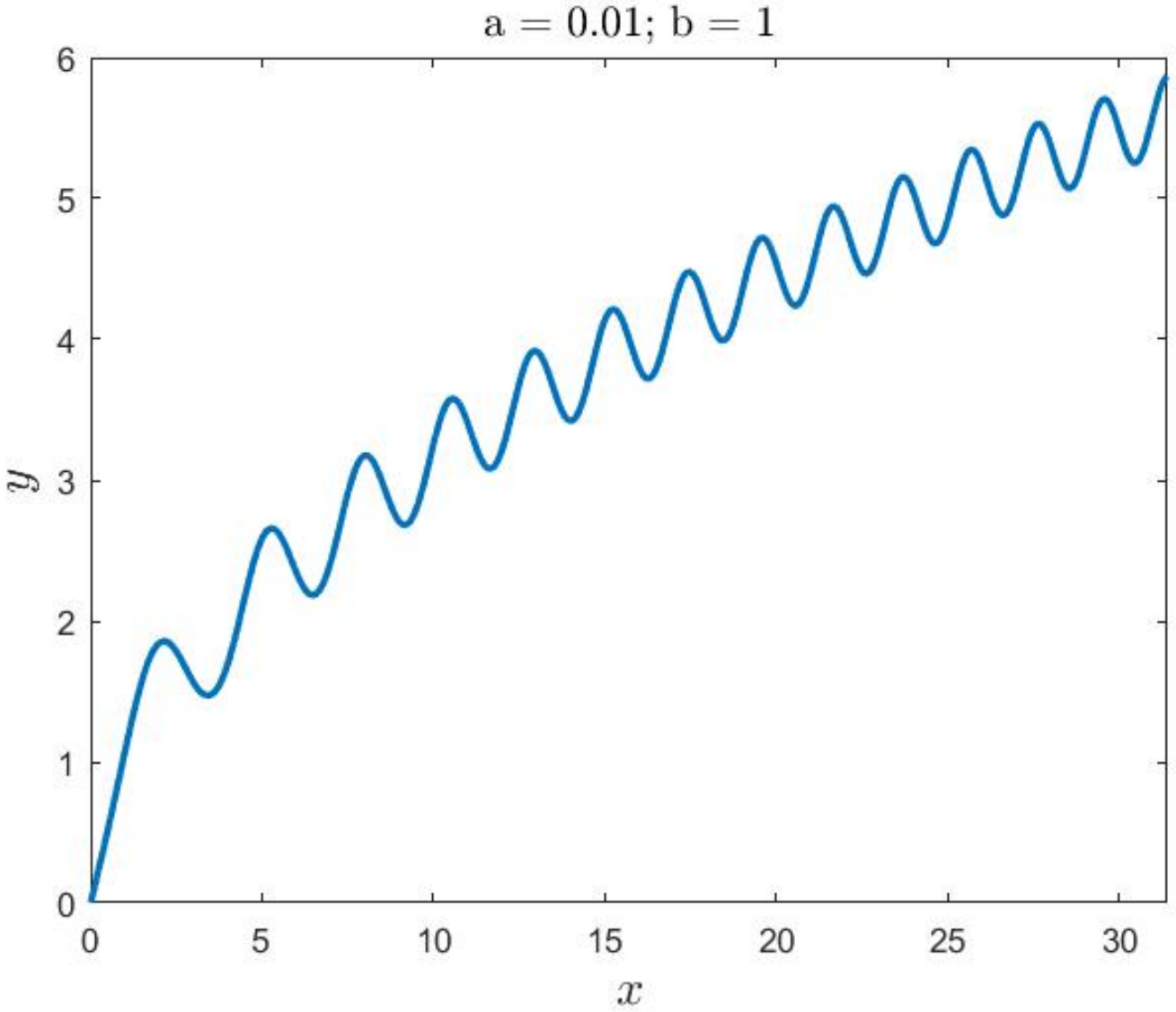
  plot(X , y)

Second order ODE

- The function file

```
function dy = func_7(x , y)
a = evalin('base' , 'a');
b = evalin('base' , 'b');
dy(1) = y(2);
dy(2) = x - a * y(2) - b * y(1) .^2;
dy = dy';
```

Second order ODE



MATLAB dsolve function

- S = dsolve(eqn) solves the differential equation "eqn".
  - eqn is a symbolic equation.

MATLAB dsolve function

- S = dsolve(eqn) solves the differential equation "eqn".
  - eqn is a symbolic equation.
- S = dsolve(eqn , cond) solves eqn with the initial or boundary condition "cond".

MATLAB dsolve function

- S = dsolve(eqn) solves the differential equation "eqn".
  - eqn is a symbolic equation.
- S = dsolve(eqn , cond) solves eqn with the initial or boundary condition "cond".
- S = dsolve(eqn , cond , Name , Value) uses additional options specified by one or more Name,Value pair arguments.

MATLAB dsolve function

- S = dsolve(eqn) solves the differential equation "eqn".
  - eqn is a symbolic equation.
- S = dsolve(eqn , cond) solves eqn with the initial or boundary condition "cond".
- S = dsolve(eqn , cond , Name , Value) uses additional options specified by one or more Name,Value pair arguments.
- [y1 , . . . , yN] = dsolve(___) assigns the solutions to the variables y1,. . . ,yN.

First order ODE

- $\frac{dx}{dt} = 0; \qquad x(0) = 2.$

First order ODE

- $\frac{dx}{dt} = 0; \qquad x(0) = 2.$

- The differential equation

```
syms x(t);
my_eqn = diff(x , t) == 0;
```

First order ODE

- $\frac{dx}{dt} = 0; \qquad x(0) = 2.$

- The differential equation

```
syms x(t);
my_eqn = diff(x , t) == 0;
```

- Initial condition

```
IC = x(0) == 2;
```

First order ODE

- $\frac{dx}{dt} = 0; \qquad x(0) = 2.$

- The differential equation

```
syms x(t);
my_eqn = diff(x , t) == 0;
```

- Initial condition

```
IC = x(0) == 2;
```

- Solving the equation

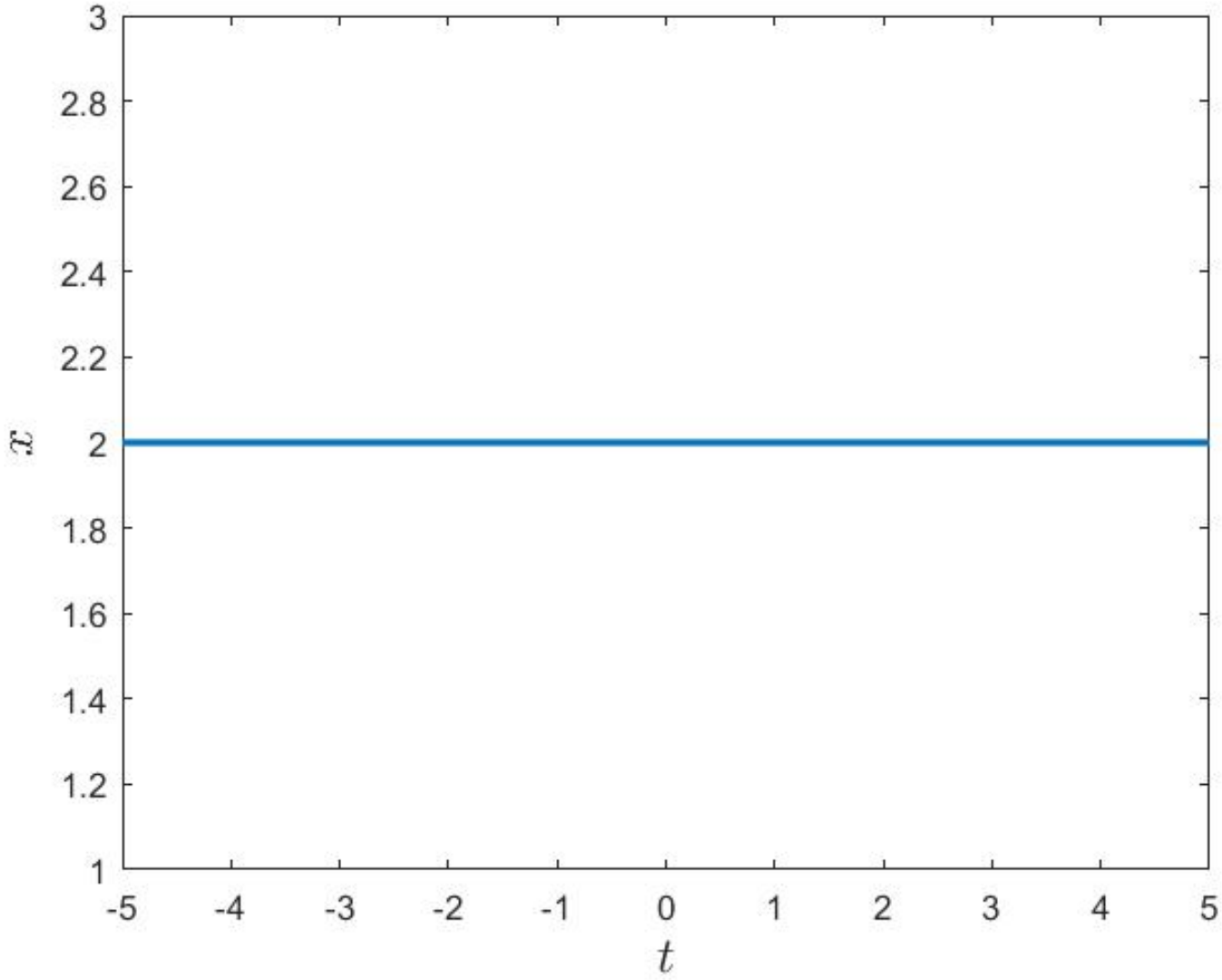
```
my_sol = dsolve(my_eqn , IC , 't');
```

First order ODE

- Plotting the result

```
fplot(my_sol)
```

First order ODE



First order ODE

- $\frac{dx}{dt} = \sin(x); \qquad x(0) = 1.$

First order ODE

- $\frac{dx}{dt} = \sin(x); \qquad x(0) = 1.$

- The differential equation

```
syms x(t);
my_eqn = diff(x , t) == sin (x);
```

First order ODE

- $\frac{dx}{dt} = \sin(x); \qquad x(0) = 1.$

- The differential equation

```
syms x(t);
my_eqn = diff(x , t) == sin (x);
```

- Initial condition

```
IC = x(0) == 1;
```

First order ODE

- $\frac{dx}{dt} = \sin(x); \qquad x(0) = 1.$

- The differential equation

```
syms x(t);
my_eqn = diff(x , t) == sin (x);
```

- Initial condition

```
IC = x(0) == 1;
```

- Solving the equation

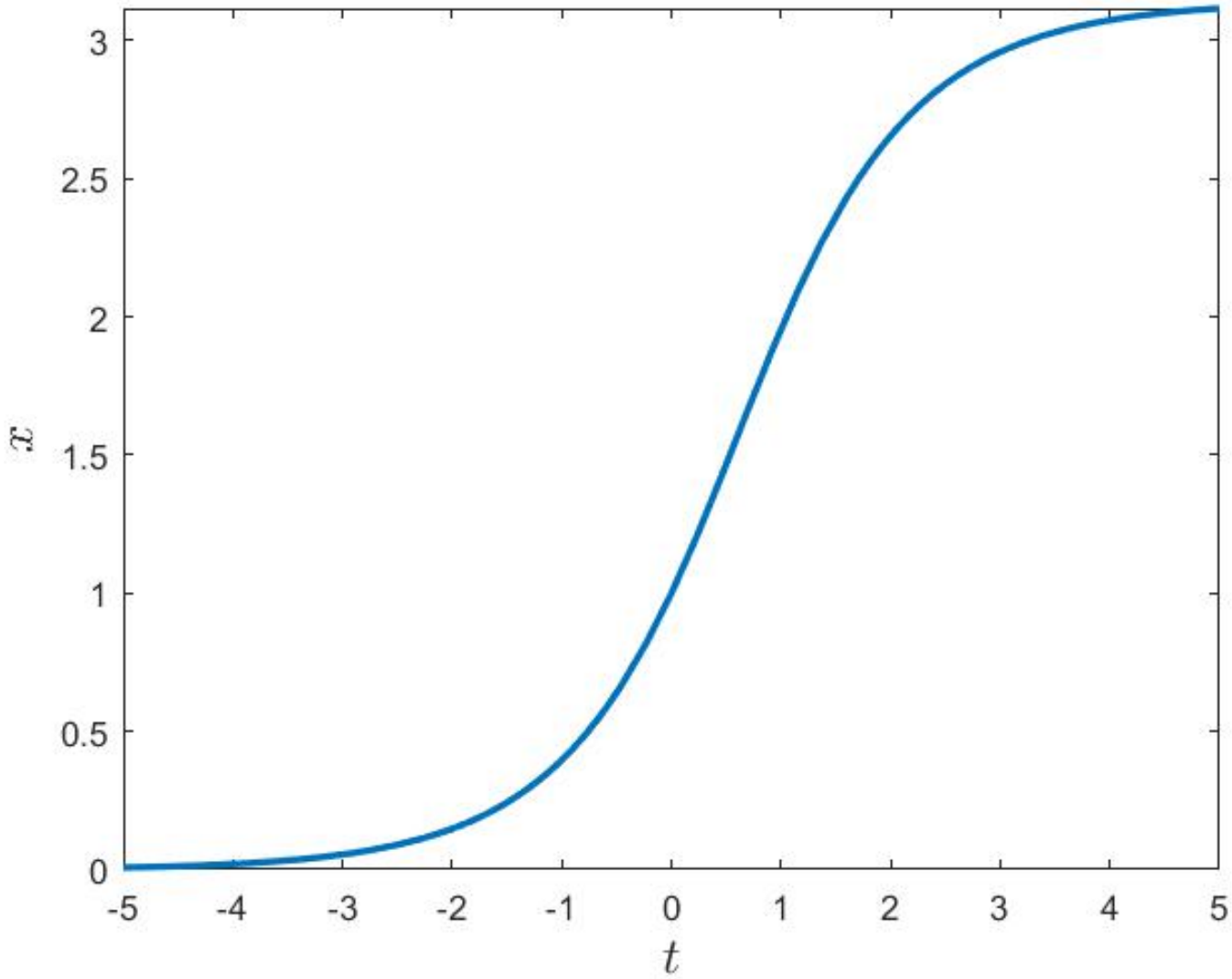
```
my_sol = dsolve(my_eqn , IC , 't');
```

First order ODE

- Plotting the result

```
fplot(my_sol)
```

First order ODE



Second order ODE

- $\frac{d^2x}{dt^2} = t + \exp(1-t)\,; \qquad x(0) = 0, x'(0) = 1.$

Second order ODE

- $\frac{d^2x}{dt^2} = t + \exp(1-t)$; $x(0) = 0, x'(0) = 1$.

- The differential equation

```
syms x(t);

my_eqn = diff(x , t , 2) == t + exp(1 - t);
```

Second order ODE

- $\frac{d^2x}{dt^2} = t + \exp(1-t)$; $x(0) = 0, x'(0) = 1$.

- The differential equation

```
syms x(t);

my_eqn = diff(x , t , 2) == t + exp(1 - t);
```

- Initial conditions

```
Dx = diff(x , t);

IC = [x(0) == 0, Dx(0) == 1];
```

Second order ODE

- Solving the equation

```
my_sol = dsolve(my_eqn , IC , 't');
```

Second order ODE

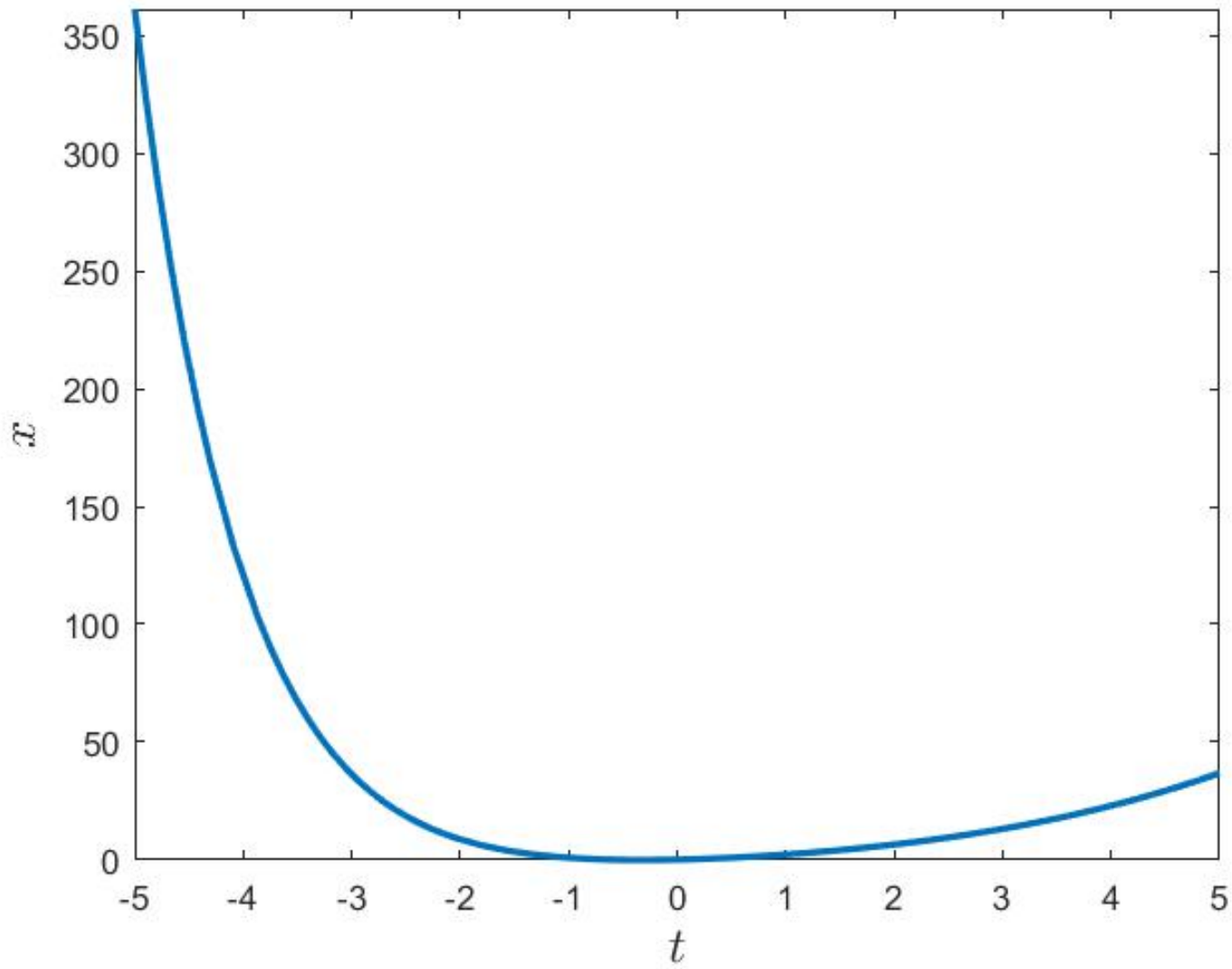
- Solving the equation

  my_sol = dsolve(my_eqn , IC , 't');

- Plotting the result

  fplot(my_sol)

Second order ODE



References

- https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html
- http://www.mathworks.com/company/newsletters/articles/stiff-differential-equations.html
- http://www.mathworks.com/help/matlab/math/ordinary-differential-equations.html
- http://www.mathworks.com/help/matlab/ref/ode45.html
- https://www.mathworks.com/help/symbolic/dsolve.html
- http://www.math.umd.edu/~immortal/206tutorial/Ch12_SolvingDE.pdf