CHAPTER 12

Solving Ordinary Differential Equations Using *Maple*

Computer software packages for symbolic computations, such as *Maple* and *Mathematica*, are very useful in mathematical analysis. In this chapter, *Maple* commands commonly used in solving ordinary differential equations are introduced.

`dsolve` is a general ordinary differential equation solver, which can handle various types of ordinary differential equation problems, including

- compute *closed form solutions* for a single or a system of ordinary differential equations, with or without initial (boundary) conditions;
- compute solutions using *integral transforms*, such as the Laplace transform;
- compute *series* and *numerical solutions* for a single or a system of ordinary differential equations.

Calling Sequence

✍ Solve `ODE` for function `y(x)`, in which `x` is the independent variable and `y` is the dependent variable. In using `dsolve`, `options` may be specified to instruct *Maple* how to solve the equation, such as expressing the solution in implicit equation or using the Laplace transform.

```
dsolve(ODE,y(x),options)
```

✍ Solve `ODE` together with initial conditions `IC` for function `y(x)`.

```
dsolve({ODE,IC},y(x),options)
```

In the following, various examples are solved using `dsolve` to illustrate the procedures. The output from *Maple* is formatted somewhat for better presentation.

12.1 Closed-Form Solutions of Differential Equations

12.1.1 Simple Ordinary Differential Equations

First-order and simple higher-order ordinary differential equations studied in Chapter 2 can be easily solved using *Maple*.

Example 12.1 (Example 2.9)

Solve $y' = (x+y)^2$.

```
>ODE:=diff(y(x),x)=(x+y(x))^2;
```
✍ Define the ODE.

$$\text{ODE} := \frac{\mathrm{d}}{\mathrm{d}x}\, y(x) = (x + y(x))^2$$

```
>sol:=dsolve(ODE,y(x));
```
✍ Solve the ODE using `dsolve`.

$$\text{sol} := y(x) = -x - \tan(-x + \_C1)$$

✍ $\_C1$ is an arbitrary constant.

Example 12.2 (Example 2.8)

Solve $\dfrac{\mathrm{d}y}{\mathrm{d}x} = \dfrac{x-y+5}{2x-2y-2}$.

```
>ODE:=diff(y(x),x)=(x-y(x)+5)/(2*x-2*y(x)-2):
```
✍ Define the ODE.

```
>sol:=dsolve(ODE,y(x));
```
✍ Solve the ODE using `dsolve` without options.

$$\text{sol} := y(x) = x - 6\,\text{LambertW}\left(-\tfrac{1}{6}\, \mathrm{e}^{x/12}\, \_C1\, \mathrm{e}^{-7/6}\right) - 7$$

✍ The result is in terms of a special function, because *Maple* tries to solve a nonlinear equation to obtain an explicit solution.

✍ To overcome this problem, use the `implicit` option to force the solution in the form of an implicit equation.

```
>sol_implicit:=dsolve(ODE,y(x),implicit);
```

$$\text{sol\_implicit} := -x + 2\,y(x) - 12\ln((y(x) - x + 7) - \_C1 = 0$$

Example 12.3 (Example 2.45)

Solve $yy'' = y'^2(1 - y'\sin y - yy'\cos y)$.

✍ In *Maple*, the nth-order derivative of $y(x)$ with respect to x is `diff(y(x),x$n)`.

```
>ODE:=y(x)*diff(y(x),x$2)=diff(y(x),x)^2*(1-diff(y(x),x)*sin(y(x))
  -y(x)*diff(y(x),x)*cos(y(x))):
```
✍ Define the ODE.

```
>sol:=dsolve(ODE,y(x));
```
✍ Solve the ODE using `dsolve` without options.

$$\text{sol} := y(x) = \_C1, \quad -\cos(y(x)) + \_C1\ln(y(x)) - x - \_C2 = 0$$

✍ An explicit or implicit solution can be tested using `odetest`, which checks the validity of the solution by substituting it into the ODE. If the solution is valid, `odetest` returns 0. When there are many solutions, use `map` to apply `odetest` to each element of `sol`.

```
>map(odetest,[sol],ODE);
```

$$[0, 0]$$

✍ The first 0 indicates that the first solution `sol[1]` is valid; the second 0 indicates that the second solution `sol[2]` is valid.

Example 12.4 (Example 2.33)

Solve $2xyy' = y^2 - 2x^3, \quad y(1) = 2.$

```
>ODE:=2*x*y(x)*diff(y(x),x)=y(x)^2-2*x^3:
```
✍ Define the ODE.

```
>IC:=y(1)=2;
```
✍ Define the initial condition IC.

$$\text{IC} := y(1) = 2$$

✍ Solve the ODE and IC using `dsolve` without options.

```
>sol:=dsolve({ODE,IC},y(x));
```

$$\text{sol} := y(x) = \sqrt{-x^3 + 5x}$$ ✍ Explicit solution.

✍ Use `odetest` to check if the result satisfies the ODE and IC.

```
>odetest(sol, [ODE, IC]);
```

$$[0, 0]$$

✍ The first 0 indicates that the solution satisfies the ODE; the second 0 indicates that the solution satisfies the IC.

✍ Solve the ODE and IC using `dsolve` with `implicit` option.

```
>sol_implicit:=dsolve({ODE,IC},y(x),implicit);
```

$$\text{sol\_implicit} := y(x)^2 + x^3 - 5x = 0$$ ✍ Implicit solution.

```
>odetest(sol_implicit, [ODE, IC]);
```

$$[0, 0]$$

Example 12.5 (Example 2.1)

Solve $\dfrac{dy}{dx} + \dfrac{1}{y}\,e^{y^2+3x} = 0.$

```
>ODE:=diff(y(x),x)+1/y(x)*exp(y(x)^2+3*x)=0:
```
✍ Define the ODE.

```
>sol:=dsolve(ODE,y(x));
```
✍ Solve the ODE using `dsolve` without options.

$$\text{sol} := y(x) = \sqrt{\ln\left(\frac{3}{2}\,\frac{1}{e^{3x}+3\_C1}\right)}, \quad y(x) = -\sqrt{\ln\left(\frac{3}{2}\,\frac{1}{e^{3x}+3\_C1}\right)}$$

✍ Solve the ODE using `dsolve` with `implicit` option.

```
>sol_implicit:=dsolve(ODE,y(x),implicit);
```

$$\text{sol\_implicit} := \frac{1}{3}\,e^{3x} - \frac{1}{2}\,e^{-y(x)^2} + \_C1 = 0$$

Example 12.6 (Example 2.24)

Solve $y^2\,dx + (xy + y^2 - 1)\,dy = 0.$

```
>ODE:=y(x)^2+(x*y(x)+y(x)^2-1)*diff(y(x),x)=0:
```
✍ Define the ODE.

```
>sol:=dsolve(ODE,y(x));
```
✍ Solve the ODE using `dsolve` without options.

$$\text{sol} := y(x) = e^{\text{RootOf}(2\_Z - 2x e^{-\_Z} - (e^{-\_Z})^2 + 2\_C1)}$$

✍ The solution is in terms of RootOf(· · ·), because *Maple* tries to solve a nonlinear equation to obtain an explicit expression for $y(x)$.

✍ Solve the ODE using `dsolve` with `implicit` option.

```
>sol_implicit:=dsolve(ODE,y(x),implicit);
```

$$\text{sol\_implicit} := x - \frac{-\frac{1}{2}y(x)^2 + \ln(y(x)) + \_C1}{y(x)} = 0$$

✍ Simplify the result selecting the left-hand side of the equation using `lhs` and extracting the numerator using `numer`.

```
>sol_implicit:=numer(lhs(sol_implicit))=0;
```

$$\text{sol\_implicit} := 2xy(x) + y(x)^2 - 2\ln(y(x)) - 2\_C1 = 0$$

Example 12.7 (Problem 2.92)

Solve $2xy' - y = \ln y'.$

```
>ODE:=2*x*diff(y(x),x)-y(x)=ln(diff(y(x),x)):
```
✍ Define the ODE.

✍ Solve the ODE using `dsolve` with `implicit` option.

```
>sol_implicit:=dsolve(ODE,y(x),implicit);
```

$$\text{sol\_implicit} := \left[x(\_T) = \frac{\_T + \_C1}{\_T^2},\quad y(\_T) = \frac{2(\_T + \_C1)}{\_T} - \ln(\_T)\right]$$

✍ This ODE is of the type solvable for variable x or y. The solution is in the form of parametric equations with $\_T$ being the parameter.

Example 12.8 (Example 2.37)

Solve $y = x\left\{\dfrac{dy}{dx} + \sqrt{1 + \left(\dfrac{dy}{dx}\right)^2}\right\}.$

```
>ODE:=y(x)=x*(diff(y(x),x)+sqrt(1+diff(y(x),x)^2)):
```
✍ Define the ODE.

```
>sol:=dsolve(ODE,y(x));
```
✍ Solve the ODE using `dsolve` without options.

$$\text{sol} := \frac{\_C1}{\sqrt{\dfrac{(x^2 + y(x)^2)^2}{y(x)^2 x^2}}\left(-\dfrac{1}{2}\dfrac{-y(x)^2 + x^2}{x\,y(x)} + \dfrac{1}{2}\sqrt{\dfrac{(x^2 + y(x)^2)^2}{y(x)^2 x^2}}\right)} + x = 0$$

✍ To simplify the result, use `simplify` with option `sqrt` and assume that both x and $y(x)$ are positive.

```
> sol:=simplify(sol,sqrt) assuming x::positive, y(x)::positive;
```

$$\frac{x\left(\_C1\,x + x^2 + y(x)^2\right)}{x^2 + y(x)^2} = 0$$

✍ Simplify the result further: take the left-hand side of the equation using `lhs`; extract the numerator using `numer`; and divide by x.

```
> sol:=numer(lhs(sol))/x=0;
```

$$\_C1\,x + x^2 + y(x)^2 = 0$$

✍ Solve the ODE using `dsolve` with `implicit` option.

```
> sol_implicit:=dsolve(ODE,y(x),implicit);
```

$$\text{sol\_implicit} := \left[x(T) = \frac{\_C1}{\sqrt{1+T^2}\,e^{\operatorname{arcsinh}(T)}},\quad y(T) = \frac{\_C1\left(T + \sqrt{1+T^2}\right)}{\sqrt{1+T^2}\,e^{\operatorname{arcsinh}(T)}}\right]$$

```
> simplify(sol_implicit);
```
✍ Use `simplify` to simplify the result.

$$\left[x(T) = \frac{\_C1}{\sqrt{1+T^2}\left(T + \sqrt{1+T^2}\right)},\quad y(T) = \frac{\_C1}{\sqrt{1+T^2}}\right]$$

✍ For this ODE, `dsolve` with option `implicit` yields solution in the parametric form.

Example 12.9 (Example 2.16)

Solve $\left(\frac{1}{y}\sin\frac{x}{y} - \frac{y}{x^2}\cos\frac{y}{x} + 1\right)\mathrm{d}x + \left(\frac{1}{x}\cos\frac{y}{x} - \frac{x}{y^2}\sin\frac{x}{y} + \frac{1}{y^2}\right)\mathrm{d}y = 0.$

```
> ODE:=1/y(x)*sin(x/y(x))-y(x)/x^2*cos(y(x)/x)+1+(1/x*cos(y(x)/x)
  -x/y(x)^2*sin(x/y(x))+1/y(x)^2)*diff(y(x),x)=0:
```
✍ Define the ODE.

✍ Solve the ODE using `dsolve` and simplify the result using `simplify`.

```
> sol:=simplify(dsolve(ODE,y(x)));
```

$$\text{sol} := -\frac{1+2y(x)\cos^2\frac{x}{2y(x)} - 2y(x) - \_C1\,y(x) - 2y(x)\sin\frac{y(x)}{2x}\cos\frac{y(x)}{2x} - x\,y(x)}{y(x)} = 0$$

✍ Simplify the result further: take the left-hand side of the equation using `lhs`; extract the numerator using `numer`; simplify the trigonometric functions using `combine`; and collect terms of $y(x)$ using `collect`.

```
> sol:=collect(combine(numer(lhs(sol))),y(x))=0;
```

$$\text{sol} := -1 + \left(\sin\frac{y(x)}{x} - \cos\frac{x}{y(x)} + x + 1 + \_C1\right)y(x) = 0$$

✍ The "1" before $\_C1$ can be absorbed in the constant $\_C1$ using `algsubs`.

```
>sol:=algsubs(1+_C1=C[1],sol);
```

$$\text{sol} := -1 + \left(\sin\frac{y(x)}{x} - \cos\frac{x}{y(x)} + x + C_1\right)y(x) = 0$$

Example 12.10 (Example 2.4)

Solve $\dfrac{\mathrm{d}y}{\mathrm{d}x} + \dfrac{x}{y} + 2 = 0, \quad y(0) = 1.$

`>ODE:=diff(y(x),x)+x/y(x)+2=0:` ✍ Define the ODE.

`>IC:=y(0)=1:` ✍ Define the IC.

`>sol:=dsolve({ODE,IC},y(x));` ✍ Solve the ODE and IC using `dsolve`.

$$\text{sol} := y(x) = -\frac{x\,(1 + \text{LambertW}(-x))}{\text{LambertW}(-x)}$$

✍ Without the `implicit` option, the solution is in terms of a special function.

✍ Now solve the ODE using `dsolve` with the `implicit` option.

```
>sol_implicit:=dsolve({ODE,IC},y(x),implicit);
```

$$\text{sol\_implicit} :=$$

✍ With the `implicit` option, *Maple* does not render a solution if the initial condition is included.

✍ Solve the ODE using the `implicit` option but without the initial condition.

```
>sol_implicit:=dsolve(ODE,y(x),implicit);
```

$$\text{sol\_implicit} := -\_C1 + \frac{\ln\left(\frac{y(x)+x}{x}\right)y(x) + \ln\left(\frac{y(x)+x}{x}\right)x + x}{y(x)+x} + \ln(x) = 0$$

✍ The solution contains terms with x appearing in the denominator; hence, the initial condition at $x=0$ cannot be applied directly. This is the reason that *Maple* does not give a solution when both the `implicit` option and the initial condition `IC` are imposed.

✍ To simplify the implicit solution, take the left-hand side of the solution using `lhs`, and extract the numerator using `numer`.

```
>SOL1:=numer(lhs(sol_implicit));
```

$$\begin{aligned}\text{SOL1} := {} & -\_C1\,y(x) - \_C1\,x + \ln\left(\frac{y(x)+x}{x}\right)y(x) + \ln\left(\frac{y(x)+x}{x}\right)x + x \\ & + \ln(x)\,y(x) + \ln(x)\,x\end{aligned}$$

✍ The solution can be further simplified by combining the logarithmic terms using `combine` and assuming that x is positive.

```
> SOL2:=combine(SOL1) assuming x::positive;
```

$$SOL2 := -\_C1\, y(x) - \_C1\, x + y(x)\ln(y(x)+x) + x\ln(y(x)+x) + x$$

✍ The initial condition $y(0)=1$ can now be applied using subs.

```
> eqn:=subs({x=0,y(x)=1},SOL2);
```

$$eqn := -\_C1 + \ln(1)$$

✍ The constant $\_C1$ is solved from the resulting algebraic equation using solve.

```
> _C1:=solve(eqn,_C1);
```

$$\_C1 := 0$$

✍ The particular solution in the implicit form is obtained.

```
> sol_implicit:=collect(SOL2,ln);
```

$$sol\_implicit := (y(x)+x)\ln(y(x)+x) + x$$

✍ Verify that the solution obtained satisfies both the ODE and IC using odetest.

```
> odetest(sol_implicit,[ODE,IC]);
```

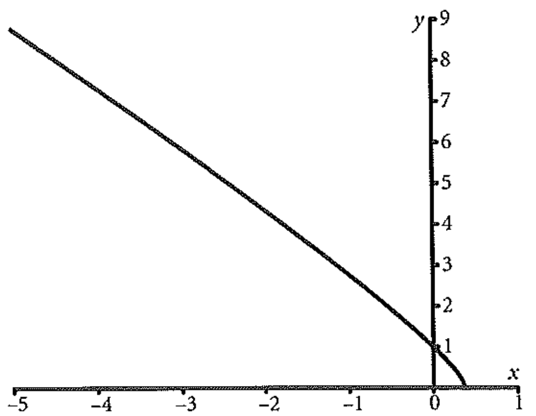
$$[0, 0]$$

✍ To plot an implicit equation, use implicitplot in the plots package.

```
> with(plots):
```

✍ Load the plots package.

```
> implicitplot(sol_implicit,x=-5..5,y(x)=-0..10,view=[-5..1,0..9],
  numpoints=100000,thickness=3,labels=["x","y"],
  tickmarks=[[-5,-4,-3,-2,-1,0,1],[0,1,2,3,4,5,6,7,8,9]]);
```



Example 12.11 (Example 2.35)

Solve $x = \dfrac{dy}{dx} + \left(\dfrac{dy}{dx}\right)^4$.

```
> ODE:=x=diff(y(x),x)+diff(y(x),x)^4:
```

✍ Define the ODE.

`>sol:=dsolve(ODE,y(x));` Solve the ODE using `dsolve`.

$$sol := y(x) = \frac{4}{5}\,x\,\text{RootOf}(-x+\_Z+\_Z^4) - \frac{3}{10}\,\text{RootOf}(-x+\_Z+\_Z^4)^2 + \_C1$$

The solution is expressed in terms of a root of a polynomial equation of degree four, i.e., $\_Z^4+\_Z-x=0$, in which $\_Z$ is the unknown. Using the option `implicit` in `dsolve` does not help.

Extract the value of $y(x)$ from the solution `sol` using `eval`.

`>eval(y(x),sol);`

$$\frac{4}{5}\,x\,\text{RootOf}(-x+\_Z+\_Z^4) - \frac{3}{10}\,\text{RootOf}(-x+\_Z+\_Z^4)^2 + \_C1$$

This result is then assigned to y, defined as a function of x, using `unapply`. The ditto operator, `%`, is a shorthand notation for the previous result.

`>y:=unapply(%,x);`

$$y := x \to \frac{4}{5}\,x\,\text{RootOf}(-x+\_Z+\_Z^4) - \frac{3}{10}\,\text{RootOf}(-x+\_Z+\_Z^4)^2 + \_C1$$

These two lines can be combined as

`>y:=unapply(eval(y(x),sol),x);`

This task can also be accomplished by using

`>y:=unapply(rhs(sol),x);`

The ODE is of the type solvable for variable x; the solution can be expressed in the form of parametric equations. Rewrite the fourth-degree polynomial equation and solve for x in terms of the parameter $\_Z$.

`>eqn:=RootOf(-x+_Z+_Z^4,_Z)=_Z:`

`>xp:=solve(eqn,x);`

$$xp := \_Z + \_Z^4$$

Variable y is also expressed in terms of the parameter $\_Z$.

`>yp:=simplify(eval(subs(x=xp,subs(eqn,y(x)))));`

$$yp := \frac{1}{2}\,\_Z^2 + \frac{4}{5}\,\_Z^5 + \_C1$$

For a better presentation, the parameter $\_Z$ is replaced by p.

`>xp:=subs(_Z=p,xp); yp:=subs(_Z=p,yp);`

$$xp := p + p^4, \qquad yp := \frac{1}{2}\,p^2 + \frac{4}{5}\,p^5 + \_C1$$

Remarks: Most of the first-order and simple higher-order ODEs studied in Chapter 2 are nonlinear equations with closed-form solutions in the form of nonlinear equations or parametric equations; `implicit` is one of the most important options in obtaining a solution in terms of elementary functions (exponential, logarithmic, and trigonometric functions). As illustrated in the examples, in many cases, one has to direct *Maple* interactively in order to obtain an useful result.

12.1.2 Linear Ordinary Differential Equations

Linear ordinary differential equations studied in Chapter 4 can be easily solved using *Maple* as shown in the following examples.

Example 12.12

Solve $y''' - y'' + y' - y = 8x\sin x + 5e^{2x} + x^2$, $y(0) = 2$, $y'(0) = 9$, $y''(0) = 5$.

✍ In *Maple*, $y'(a) = b \Longrightarrow$ `D(y)(a)=b`, and $y^{(n)}(a) = b \Longrightarrow$ `(D@@n)(y)(a)=b`.

✍ Define the ODE and initial conditions ICs.

```
>ODE:=diff(y(x),x$3)-diff(y(x),x$2)+diff(y(x),x)-y(x)=8*x*sin(x)
 +5*exp(2*x)+x^2:
```

```
>ICs:=y(0)=2, D(y)(0)=9, (D@@2)(y)(0)=5:
```

✍ Solve the ODE with ICs using `dsolve`. Sort the result by collecting the sine terms and then the cosine terms.

```
>sol:=collect(collect(dsolve({ODE,ICs},y(x)),sin),cos);
```

$$\text{sol} := y(x) = (-x^2 - 3x + 6)\sin x + (x^2 - x - 3)\cos x + 4e^x + e^{2x} - x^2 - 2x$$

Example 12.13 (Example 4.29)

Solve $y''' - y' = \dfrac{e^x}{1 + e^x}$.

```
>ODE:=diff(y(x),x$3)-diff(y(x),x)=exp(x)/(1+exp(x)):
```
✍ Define the ODE.

✍ Solve the ODE using `dsolve`. Sort the result by collecting exponential terms.

```
>sol:=collect(dsolve(ODE,y(x)),exp);
```

$$\text{sol} := y(x) = \left(\_C1 - \tfrac{1}{2}\ln(1+e^x) + \tfrac{1}{2}\ln(e^x)\right)e^x - \left(\_C2 + \tfrac{1}{2}\ln(1+e^x)\right)e^{-x} + \_C3 + \tfrac{1}{2} - \ln(1+e^x)$$

✍ The $\frac{1}{2}$ after $\_C3$ can be absorbed in the arbitrary constant $\_C3$.

Example 12.14

Solve $x^2y'' - xy' + 2y = x(\ln x)^4 + 4x\sin(\ln x)$.

✍ Define the ODE. This is an Euler differential equation.

```
>ODE:=x^2*diff(y(x),x$2)-x*diff(y(x),x)+2*y(x)=x*(ln(x))^4
 +4*x*sin(ln(x)):
```

```
>sol:=dsolve(ODE,y(x));
```
✍ Solve the ODE using `dsolve`.

$$\text{sol} := y(x) = \_C1\,x\cos(\ln x) + \_C2\,x\sin(\ln x) + x\left((\ln x)^4 - 12(\ln x)^2 + 24 + \sin(\ln x) - 2\ln x\cos(\ln x)\right)$$

12.1.3 The Laplace Transform

Integral transforms, such as the Laplace transform and the Fourier transform, are available by loading the `inttrans` package using `with(inttrans)`. In *Maple*, the Heaviside step function $H(t-a)$ is `Heaviside(t-a)`, and the Dirac delta function $\delta(t-a)$ is `Dirac(t-a)`.

`>with(inttrans):` ✍ Load the `inttrans` package.

✍ Given a function $f(t)$.

```
>f:=t*cosh(2*t)+t^2*sin(5*t)+t^3+sin(t)*Heaviside(t-Pi);
```

$$f := t\cosh 2t + t^2 \sin 5t + t^3 + \sin t\, \mathrm{Heaviside}(t-\pi)$$

`>F:=laplace(f,t,s);` ✍ Evaluate the Laplace transform using `laplace`.

$$F := \frac{1}{2(s-2)^2} + \frac{1}{2(s+2)^2} + \frac{10(3s^2-25)}{(s^2+25)^3} + \frac{6}{s^4} + \frac{e^{-\pi s}}{s^2+1}$$

✍ Given the Laplace transform of a function $G(s)$.

```
>G:=(s-3)/(s^2-6*s+25)+2/(s+2)^3+exp(-2*s)*(2+1/(s^2+1));
```

$$G := \frac{s-3}{s^2-6s+25} + \frac{2}{(s+2)^3} + e^{-2s}\left(2 + \frac{1}{s^2+1}\right)$$

✍ Evaluate inverse Laplace transform using `invlaplace`.

```
>g:=invlaplace(G,s,t);
```

$$g := e^{3t}\cos 4t + t^2 e^{-2t} + 2\,\mathrm{Dirac}(t-2) + \mathrm{Heaviside}(t-2)\sin(t-2)$$

✍ When evaluating the inverse Laplace transform by hand, one frequently needs to perform partial fraction decomposition, which can be easily done using *Maple*.

`>F:=8*(s+2)/(s-1)/(s+1)^2/(s^2+1)/(s^2+9);` ✍ Define a fraction.

$$F := \frac{8(s+2)}{(s-1)(s+1)^2(s^2+1)(s^2+9)}$$

✍ Perform partial fraction decomposition using `convert` with option `parfrac`, in which s is the variable.

```
>convert(F,parfrac,s);
```

$$\frac{3}{10(s-1)} - \frac{1}{5(s+1)^2} - \frac{27}{50(s+1)} + \frac{s-3}{4(s^2+1)} + \frac{s-11}{100(s^2+9)}$$

When an ODE is solved using `dsolve` with the option `method=laplace`, *Maple* forces the equation to be solved by the method of Laplace transform.

Example 12.15 (Example 6.18)

Solve $y''' - y'' + 4y' - 4y = 40(t^2 + t + 1)H(t-2)$, $y(0) = 5$, $y'(0) = 0$, $y''(0) = 10$.

```
>ODE:=diff(y(t),t$3)-diff(y(t),t$2)+4*diff(y(t),t)-4*y(t)
 =40*(t^2+t+1)*Heaviside(t-2):
```
✍: Define the ODE.

```
>ICs:=y(0)=5, D(y)(0)=0, (D@@2)(y)(0)=10:
```
✍: Define the ICs.

✍: Solve the ODE and ICs using `dsolve` with the option `method=laplace`. The trigonometric terms are simplified using `combine`. The result is simplified by first collecting terms with `Heaviside(t-2)` and then exponential terms using `collect`.

```
>sol:=collect(collect(combine(dsolve({ODE,ICs},y(t),method=laplace)),
 Heaviside(t-2)),exp);
```

$$\text{sol} := y(t) = 6e^t + 112\big(1 - \text{Heaviside}(2-t)\big)e^{t-2} + \big(23\cos(2t-4) - 21\sin(2t-4) - 10t^2 - 30t - 35\big)\,\text{Heaviside}(t-2) - \cos 2t - 3\sin 2t$$

Example 12.16

Solve $y''' - y'' + 4y' - 4y = 10e^{-t}$.

✍: Define the ODE.

```
>ODE:=diff(y(t),t$3)-diff(y(t),t$2)+4*diff(y(t),t)-4*y(t)=10*exp(-t):
```

✍: Solve the ODE using `dsolve` with the option `method=laplace`. Simplified the result by collecting sine, cosine, and then exponential terms using `collect`.

```
>collect(collect(collect(dsolve(ODE,y(t),method=laplace),sin),cos),exp);
```

$$y(t) = -e^{-t} + \big(1 + \tfrac{4}{5}y(0) + \tfrac{1}{5}y''(0)\big)e^t + \tfrac{1}{5}\big(y(0) - y''(0)\big)\cos 2t + \big(-1 - \tfrac{2}{5}y(0) + \tfrac{1}{2}y'(0) - \tfrac{1}{10}y''(0)\big)\sin 2t$$

```
>dsolve(ODE,y(t));
```
✍: Solve the ODE using `dsolve` without any option.

$$y(t) = -e^{-t} + \_C1\,e^t + \_C2\cos 2t + \_C3\sin 2t$$

Remarks: Using the option `method=laplace` to solve an ODE by the method of Laplace transform is done "behind-the-scene." When the initial conditions are specified, there is no difference between the particular solutions obtained with and without the `method=laplace` option.

However, when the initial conditions are not specified, the general solution obtained with the `method=laplace` option is expressed in terms of the unknown initial conditions $y(0)$, $y'(0)$, $y''(0)$, ...; whereas the general solution obtained without the `method=laplace` option is given in terms of arbitrary constants $\_C1$, $\_C2$, ...

12.1.4 Systems of Ordinary Differential Equations

Example 12.17

Solve $y_1' - y_2 = 0, \quad y_2' - y_3 = 2H(x-1), \quad 6y_1 + 11y_2 + y_3' + 6y_3 = \mathrm{e}^{-x}$.

```
>ODE[1]:=diff(y[1](x),x)-y[2](x)=0:
```
✍ Define the ODEs.
```
>ODE[2]:=diff(y[2](x),x)-y[3](x)=2*Heaviside(x-1):
>ODE[3]:=6*y[1](x)+11*y[2](x)+diff(y[3](x),x)+6*y[3](x)=exp(-x):
```

✍ Solve the ODEs using `dsolve`. The result is simplified by collecting first the exponential terms and then terms involving `Heaviside(x-1)` using `collect`.

```
>sol:=collect(collect(dsolve({ODE[1],ODE[2],ODE[3]},{y[1](x),y[2](x),
 y[3](x)}),exp),Heaviside(x-1));
```

$$\begin{aligned}
\text{sol} := \Big\{ y_1(x) &= \left(-\mathrm{e}^{-3(x-1)} + 4\,\mathrm{e}^{-2(x-1)} - 5\,\mathrm{e}^{-(x-1)} + 2\right)\text{Heaviside}(x-1) \\
&+ \left(_C1 + \frac{3_C2}{2} + \frac{_C3}{2}\right)\mathrm{e}^{-3x} - (3_C1 + 4_C2 + _C3)\,\mathrm{e}^{-2x} \\
&+ \left(3_C1 + \frac{5_C2}{2} + \frac{_C3}{2} + \frac{x}{2} - \frac{3}{4}\right)\mathrm{e}^{-x}, \\
y_2(x) &= \left(3\,\mathrm{e}^{-3(x-1)} - 8\,\mathrm{e}^{-2(x-1)} + 5\,\mathrm{e}^{-(x-1)}\right)\text{Heaviside}(x-1) \\
&+ \left(-3_C1 - \frac{9_C2}{2} - \frac{3_C3}{2}\right)\mathrm{e}^{-3x} + (6_C1 + 2_C2 + 8_C3)\,\mathrm{e}^{-2x} \\
&+ \left(-3_C1 - \frac{5_C2}{2} - \frac{_C3}{2} - \frac{x}{2} + \frac{5}{4}\right)\mathrm{e}^{-x}, \\
y_3(x) &= \left(-9\,\mathrm{e}^{-3(x-1)} + 16\,\mathrm{e}^{-2(x-1)} - 5\,\mathrm{e}^{-(x-1)-2}\right)\text{Heaviside}(x-1) \\
&+ \left(9_C1 + \frac{27_C2}{2} + \frac{9_C3}{2}\right)\mathrm{e}^{-3x} - (12_C1 + 16_C2 + 4_C3)\,\mathrm{e}^{-2x} \\
&+ \left(3_C1 + \frac{5_C2}{2} + \frac{_C3}{2} - \frac{7}{4}\right)\mathrm{e}^{-x}\Big\}
\end{aligned}$$

Example 12.18

Solve $x'' + 3x' + 2x + y' + y = 4t^2 + 2 + 40t\cos 2t$,

$x' + 2x + y' - y = 192t\,\mathrm{e}^{2t} + 5\sin 2t, \quad x(0) = 0,\ x'(0) = 0,\ y(0) = 0.$

```
>ODE[1]:=diff(x(t),t$2)+3*diff(x(t),t)+2*x(t)+diff(y(t),t)+y(t)
 =4*t^2+2+40*t*cos(2*t):
```
✍ Define the ODEs.
```
>ODE[2]:=diff(x(t),t)+2*x(t)+diff(y(t),t)-y(t)=192*t*exp(2*t)
 +5*sin(2*t):
>ICs:=x(0)=0, D(x)(0)=0, y(0)=0:
```
✍ Define the initial conditions ICs.

✍ Solve the ODEs and ICs using `dsolve`. The result is simplified by collecting the exponential terms, the cosine terms, and the sine terms using `collect`.

```
>sol:=collect(collect(collect(dsolve({ODE[1],ODE[2],ICs},{x(t),y(t)}),
  exp),cos),sin);
```

$$\mathrm{sol} := \Big\{x(t) = \left(4t+\tfrac{21}{40}\right)\sin 2t + \left(-3t+\tfrac{16}{5}\right)\cos 2t + \tfrac{77}{16}\,\mathrm{e}^{-2t} - \tfrac{148}{15}\,\mathrm{e}^{-t} + \left(-24t^2+12t-\tfrac{151}{48}\right)\mathrm{e}^{2t} + t^2 - 4t + 5,$$
$$y(t) = \left(2t-\tfrac{101}{20}\right)\sin 2t + \left(6t+\tfrac{7}{20}\right)\cos 2t - \tfrac{74}{15}\,\mathrm{e}^{-t} + \left(96t^2+\tfrac{7}{12}\right)\mathrm{e}^{2t} + 2t^2 - 2t + 4\Big\}$$

✍ Use `odetest` to check if the solution satisfies the ODEs and ICs.

```
>odetest(sol,[ODE[1],ODE[2],ICs]);
```

$[0, 0, 0, 0, 0]$ ✍ All of the two ODEs and three ICs are satisfied.

Eigenvalues and Eigenvectors of a Matrix

In solving systems of linear ordinary differential equations using the matrix method, as studied in Chapter 7, one needs to find the eigenvalues and the corresponding eigenvectors (sometimes the generalized eigenvectors if there are multiple eigenvalues) of the system matrix **A**. This task can be easily accomplished using *Maple*.

`>with(LinearAlgebra):` ✍ Load the `LinearAlgebra` package.

`>A:=Matrix([[1,-1],[2,-1]]);` ✍ Define matrix **A**.

$$\mathbf{A} := \begin{bmatrix} 1 & -1 \\ 2 & -1 \end{bmatrix}$$

✍ Evaluate the eigenvalues and eigenvectors using `Eigenvectors`, stored in λ and $\mathbf{u}$, respectively. In *Maple*, i is denoted as I.

```
>(lambda,v):=Eigenvectors(A);
```

$$\lambda,\ \mathbf{v} := \begin{bmatrix} I \\ -I \end{bmatrix},\ \begin{bmatrix} \frac{1}{2}+\frac{1}{2}I & \frac{1}{2}-\frac{1}{2}I \\ 1 & -1 \end{bmatrix}$$

As studied in Chapter 7, if matrix **A** has a multiple eigenvalue λ with multiplicity $m > 1$ and if there are fewer than m linearly independent eigenvectors corresponding to λ, then matrix **A** is defective. In this case, a complete basis of eigenvectors is obtained by including generalized eigenvectors.

In matrix theory, if matrix **A** of dimension $n \times n$ has n linearly independent eigenvectors, it can be reduced to a diagonal matrix **D**, i.e., $\mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} = \mathbf{D}$, in which the diagonal elements of D are the eigenvalues of A and the columns of Q are the corresponding eigenvectors.

However, if matrix A is defective, it can be reduced to the Jordan form J, i.e., $Q^{-1}AQ = J$, in which the diagonal elements of the Jordan form J are the eigenvalues of A and the columns of Q are the eigenvectors and generalized eigenvectors of A.

`>with(LinearAlgebra):` ✍: Load the `LinearAlgebra` package.

`>A:=Matrix([[1,0,1],[0,1,-1],[0,0,2]]);` ✍: Define matrix A.

$$A := \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

`>(lambda,v):=Eigenvectors(A);` ✍: Evaluate the eigenvalues and eigenvectors.

$$\lambda, v := \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

✍: Although matrix A has an eigenvalue $\lambda = 1$ of multiplicity 2, it does have two linearly independent eigenvectors.

`>A:=Matrix([[4,-1,0],[3,1,-1],[1,0,1]]);` ✍: Define matrix A.

$$A := \begin{bmatrix} 4 & -1 & 0 \\ 3 & 1 & -1 \\ 1 & 0 & 1 \end{bmatrix}$$

`>(lambda,v):=Eigenvectors(A);` ✍: Evaluate the eigenvalues and eigenvectors.

$$\lambda, v := \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

✍: Matrix A has an eigenvalue $\lambda = 2$ of multiplicity 3; but it has only one eigenvector, because the last two columns of v are zero vector.

✍: Use `JordanForm` to determine the Jordan form J and matrix Q.

`>(J,Q):=JordanForm(A,output=['J','Q']);`

$$J, Q := \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

✍: In matrix Q, the first column is the eigenvector and the last two columns are the generalized eigenvectors.

12.2 Series Solutions of Differential Equations

Special Functions

✍ *Maple* can be used to evaluate special functions effectively.

`>convert(GAMMA(n+1),factorial);` ✍ Convert Gamma function to factorial.

$$n! \qquad ✍\ \Gamma(n+1) = n!$$

`>GAMMA(1/2);` ✍ Evaluate $\Gamma\left(\frac{1}{2}\right)$.

$$\sqrt{\pi} \qquad ✍\ \Gamma\left(\tfrac{1}{2}\right) = \sqrt{\pi}$$

`>evalf(GAMMA(1/3));` ✍ Evaluate $\Gamma\left(\frac{1}{3}\right)$ using floating-point arithmetic.

$$2.678938537 \qquad ✍\ \Gamma\left(\tfrac{1}{3}\right) = 2.678938537$$

`>evalf(BesselJ(1/3,1));` ✍ Evaluate $J_{\frac{1}{3}}(1)$ using floating-point arithmetic.

$$0.7308764022 \qquad ✍\ J_{\frac{1}{3}}(1) = 0.7308764022$$

✍ Expand $J_0(x)$ in series about $x=0$ to the order $O(x^{12})$.

`>J[0]:=series(BesselJ(0,x),x,12);`

$$J_0 := 1 - \frac{x^2}{4} + \frac{x^4}{64} - \frac{x^6}{2304} + \frac{x^8}{147456} - \frac{x^{10}}{14745600} + O(x^{12})$$

✍ Convert J_0 to polynomial by dropping the order term $O(x^{12})$.

`>J[0]:=convert(J[0],polynom);`

$$J_0 := 1 - \frac{x^2}{4} + \frac{x^4}{64} - \frac{x^6}{2304} + \frac{x^8}{147456} - \frac{x^{10}}{14745600}$$

✍ Expand $Y_0(x)$ in series about $x=0$ to the order $O(x^{12})$ and then convert to polynomial. Unfortunately, the result is not presented in a very clear format.

`>Y[0]:=convert(series(BesselY(0,x),x,12),polynom);`

$$Y_0 := \frac{2(-\ln 2 + \ln x)}{\pi} + \frac{2\gamma}{\pi} + \left(-\frac{1}{2}\,\frac{-\ln 2 + \ln x}{\pi} - \frac{-\frac{1}{2} + \frac{\gamma}{2}}{\pi}\right)x^2 + \cdots$$

✍ To factor out $2/\pi$, divide Y_0 by $2/\pi$ first (and then multiply by $2/\pi$ later).

✍ Since *Maple* does not allow collecting in terms of $(\ln x - \ln 2 + \gamma)$, replace $(\ln x - \ln 2 + \gamma)$ by T and then `collect` in terms of T.

`>Y[0]:=collect(simplify(subs(ln(x)=T+ln(2)-gamma,Y[0]/(2/Pi))),T);`

$$Y_0 := \left(1 - \frac{x^2}{4} + \frac{x^4}{64} - \frac{x^6}{2304} + \frac{x^8}{147456} - \frac{x^{10}}{14745600}\right)T$$
$$+ \frac{x^2}{4} - \frac{3x^4}{128} + \frac{11x^6}{13824} - \frac{25x^8}{1769472} + \frac{137x^{10}}{884736000}$$

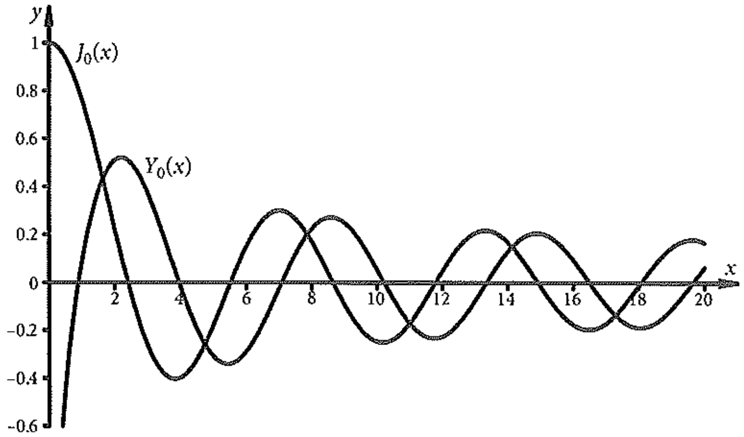
✍ Replace T with $\ln(x/2)+\gamma$ and multiply by $2/\pi$ to obtain final result.

```
>Y[0]:=2/Pi*subs(T=ln(x/2)+gamma,Y[0]);
```

$$Y_0 := \frac{2}{\pi}\left[\left(1-\frac{x^2}{4}+\frac{x^4}{64}-\frac{x^6}{2304}+\frac{x^8}{147456}-\frac{x^{10}}{14745600}\right)\left(\ln\frac{x}{2}+\gamma\right) + \frac{x^2}{4}-\frac{3x^4}{128}+\frac{11x^6}{13824}-\frac{25x^8}{1769472}+\frac{137x^{10}}{884736000}\right]$$

✍ Plot $J_0(x)$ and $Y_0(x)$.

```
>plot({BesselJ(0,x),BesselY(0,x)},x=0..20,y=-1..1,numpoints=1000);
```



Example 12.19 — Buckling of a Tapered Column (Example 9.9)

Evaluate the first three buckling parameters p_n, $n=1,2,3$, of the fixed-free tapered column ($r_1=0.5\,r_0$) with circular cross-section considered in Section 9.3.2.

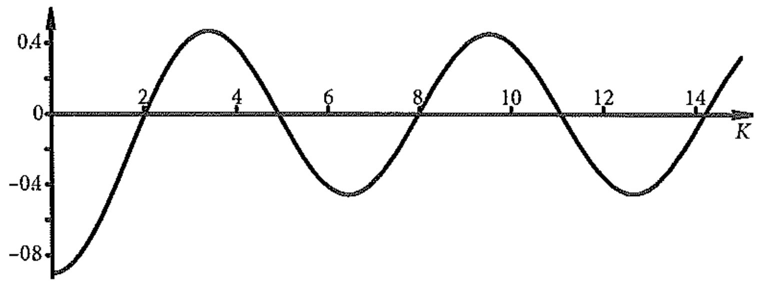
✍ Define the buckling equation.

```
>eqn:=BesselJ(-1/2,K/(1-kappa))*(BesselJ(1/2,K)-K*BesselJ(-1/2,K))
  -BesselJ(1/2,K/(1-kappa))*(BesselJ(-1/2,K)+K*BesselJ(1/2,K)):
```

`>kappa:=(r[0]-r[1])/r[0]:` ✍ $\kappa=(r_0-r_1)/r_0$

`>r[1]:=0.5*r[0]:` ✍ $r_1=0.5\,r_0$

`>plot(eqn,K=0..15);` ✍ Plot the buckling equation for $0\leqslant K\leqslant 15$.



```
>K[1]:=fsolve(eqn,K=1..3);
```
✍ Find the first root K_1 between 1 and 3.

$$K_1 := 2.028757838$$

```
>p[1]:=evalf(kappa*K[1]/Pi);
```
✍ The first buckling load p_1.

$$p_1 := 0.3228868382$$

```
>K[2]:=fsolve(eqn,K=4..6);
```
✍ Find the second root K_2 between 4 and 6.

$$K_2 := 4.913180439$$

```
>p[2]:=evalf(kappa*K[2]/Pi);
```
✍ The second buckling load p_2.

$$p_2 := 0.7819569531$$

✍ Find the third root between 6 and 10 and evaluate the third buckling load p_3.

```
>p[3]:=evalf(kappa*fsolve(eqn,K=6..10)/Pi);
```

$$p_3 := 1.269844087$$

Series Solutions of Differential Equations

Series solution of an ordinary differential equation can be obtained using `dsolve` with the option `series` and `'point'`$=x_0$ to expand the series about point $x=x_0$.

Example 12.20 (Problem 9.1)

Solve the Airy equation $y''(x) - x\,y(x) = 0.$

```
>ODE:=diff(y(x),x$2)-x*y(x)=0:
```
✍ Define the ODE.

✍ Solve the ODE without any option. The solution is given in terms of the Airy functions.

```
>dsolve(ODE,y(x));
```

$$y(x) = \_C1\,\mathrm{AiryAi}(x) + \_C2\,\mathrm{AiryBi}(x)$$

✍ Solve the ODE with the `series` option. The series is expanded about `'point'` $x=0$. The default order of the series expansion is 6, i.e., $O(x^6)$.

```
>sol:=dsolve(ODE,y(x),series,'point'=0);
```

$$\mathrm{sol} := y(x) = y(0) + \mathrm{D}(y)(0)\,x + \tfrac{1}{6}y(0)\,x^3 + \tfrac{1}{12}\mathrm{D}(y)(0)\,x^4 + O(x^6)$$

✍ Convert the series into a polynomial by dropping the order term $O(x^6)$.

```
>sol:=convert(sol,polynom);
```

$$\mathrm{sol} := y(x) = y(0) + \mathrm{D}(y)(0)\,x + \tfrac{1}{6}y(0)\,x^3 + \tfrac{1}{12}\mathrm{D}(y)(0)\,x^4$$

✍ Rearranging the solution by collecting terms involving $y(0)$ and $\mathrm{D}(y)(0)$.

```
>collect(collect(sol,D(y)(0)),y(0));
```

$$y(x) = \left(1+\frac{x^3}{6}\right)y(0) + \left(x+\frac{x^4}{12}\right)\mathrm{D}(y)(0)$$

✍ Expand the series about $x=1$ using 'point'=1. Convert to polynomial and then rearrange the solution by collecting terms involving $y(1)$ and $D(y)(1)$.

```
>sol:=dsolve(ODE,y(x),series,'point'=1):
>sol:=convert(sol,polynom):
>collect(collect(sol,D(y)(1)),y(1));
```

$$y(x) = \left[1+\frac{(x-1)^2}{2}+\frac{(x-1)^3}{6}+\frac{(x-1)^4}{24}+\frac{(x-1)^5}{30}\right]y(1) + \left[(x-1)+\frac{(x-1)^3}{6}+\frac{(x-1)^4}{12}+\frac{(x-1)^5}{120}\right]D(y)(1)$$

✍ The order of the series expansion can be changed using Order.

```
>Order:=11:
```
✍ Change the order of series expansion to $O(x^{11})$.

```
>sol:=dsolve(ODE,y(x),series,'point'=0):
>sol:=convert(sol,polynom):
>collect(collect(sol,D(y)(0)),y(0));
```

$$y(x) = \left(1+\frac{x^3}{6}+\frac{x^6}{180}+\frac{x^9}{12960}\right)y(0) + \left(x+\frac{x^4}{12}+\frac{x^7}{504}+\frac{x^{10}}{45360}\right)D(y)(0)$$

✍ When the ODE is a homogeneous linear ordinary differential equation with polynomial coefficients, dsolve with option 'formal_solution' gives a set of formal solutions with the specified coefficients at the given 'point'. In the following, the result is formatted for better presentation.

```
>Formal_Solution:=dsolve(ODE,y(x),'formal_solution','point'=0);
```

$$\text{Formal\_Solution} := y(x) = C_1\,\Gamma\left(\tfrac{2}{3}\right)\sum_{n=0}^{\infty}\frac{9^{-n}x^{3n}}{\Gamma(n+1)\,\Gamma\left(n+\frac{2}{3}\right)} + C_2\,\frac{2\pi}{\Gamma\left(\frac{2}{3}\right)}\sum_{n=0}^{\infty}\frac{3^{-\frac{3}{2}-2n}x^{3n+1}}{\Gamma(n+1)\,\Gamma\left(n+\frac{4}{3}\right)}$$

Example 12.21

Solve the Riccati-Bessel equation $x^2y''(x)+\left[x^2-k(k+1)\right]y(x)=0$, for $k=-\frac{1}{2}$.

✍ x is a regular singular point. The roots of the indicial equation are $\alpha_1=\alpha_2=\frac{1}{2}$.

```
>ODE:=x^2*diff(y(x),x$2)+(x^2-k*(k+1))*y(x)=0:
```
✍ Define the ODE.

```
>k:=-1/2:
```
✍ Set the value of k to $-\frac{1}{2}$.

✍ Solve the ODE without any option. The solution is given in terms of the Bessel functions $J_0(x)$ and $Y_0(x)$.

```
>dsolve(ODE,y(x));
```

$$y(x) = \_C1\,\sqrt{x}\,\text{BesselJ}(0,x) + \_C2\,\sqrt{x}\,\text{BesselY}(0,x)$$

✍ Obtain formal solutions using `dsolve` with option `'formal_solution'`.

```
>Formal_Solution:=dsolve(ODE,y(x),'formal_solution','point'=0);
```

$$\text{Formal\_Solution} := y(x) = C_1\sqrt{x}\sum_{n=0}^{\infty}\frac{(-1)^n 4^{-n} x^{2n}}{\Gamma(n+1)^2}$$

✍ For this ODE, only one formal solution is given. It seems that, in many cases, *Maple* is not able to give all the formal solutions.

`>Order:=11:` ✍ Set the order of series expansion to $O(x^{11})$.

`>sol:=dsolve(ODE,y(x),series,'point'=0):` ✍ Obtain series solution.

```
>sol:=convert(sol,polynom);
```

$$\begin{aligned}\text{sol} := y(x) = \_C1\,\sqrt{x}\left(1-\frac{x^2}{4}+\frac{x^4}{64}-\frac{x^6}{2304}+\frac{x^8}{147456}-\frac{x^{10}}{14745600}\right)\\ +\_C2\,\sqrt{x}\left[\left(1-\frac{x^2}{4}+\frac{x^4}{64}-\frac{x^6}{2304}+\frac{x^8}{147456}-\frac{x^{10}}{14745600}\right)\ln x\right.\\ \left.+\left(\frac{x^2}{4}-\frac{3x^4}{128}+\frac{11x^6}{13824}-\frac{25x^8}{1769472}+\frac{137x^{10}}{884736000}\right)\right]\end{aligned}$$

Example 12.22 (Problem 9.11)

Solve the Laguerre equation $xy''(x)+(1-x)y'(x)+ky(x)=0$, for $k=4$.

✍ x is a regular singular point. The roots of the indicial equation are $\alpha_1=\alpha_2=0$.

`>ODE:=x*diff(y(x),x$2)+(1-x)*diff(y(x),x)+k*y(x)=0:` ✍ Define the ODE.

`>k:=4:` ✍ Set the value of k to 4.

✍ Solve the ODE without any option. The solution is given in terms of the exponential integral $\text{Ei}(1,-x)$.

```
>dsolve(ODE,y(x));
```

$$\begin{aligned}y(x) = \_C1\,(24-96x+72x^2-16x^3+x^4) + \frac{\_C2}{576}\left[(24-96x+72x^2-16x^3\right.\\ \left.+x^4)\,\text{Ei}(1,-x)+e^x(-50+58x-15x^2+x^3)\right]\end{aligned}$$

✍ Obtain formal solutions using `dsolve` with option `'formal_solution'`.

```
>Formal_Solution:=dsolve(ODE,y(x),'formal_solution','point'=0);
```

$$\begin{aligned}\text{Formal\_Solution} := y(x) = \_C1\left(1-4x+3x^2-\frac{2x^3}{3}+\frac{x^4}{24}\right)+\_C2\left[\left(600\right.\right.\\ \left.-2400x+1800x^2-400x^3+25x^4\right)\ln x+5400x-6450x^2+1900x^3-\frac{625x^4}{4}\\ \left.+14400\sum_{n=5}^{\infty}\frac{x^n}{\Gamma(n+1)\,n(n-1)(n-2)(n-3)(n-4)}\right]\end{aligned}$$

✍ For this ODE, *Maple* does give two formal solutions. The second solution can be simplified by converting the Gamma function to factorial, and absorbing the constant 600 into $\_C2$ by using $\_C2 = \_C3/600$. This makes the polynomial in front of $\ln x$ the same as the first solution, consistent with Fuchs' Theorem.

```
>collect(collect(collect(simplify(subs(_C2=_C3/600,
 convert(Formal_Solution,factorial))),ln(x)),_C3),_C1);
```

$$y(x) = \_C1\left(1-4x+3x^2-\frac{2x^3}{3}+\frac{x^4}{24}\right)+\_C3\left[\left(1-4x+3x^2-\frac{2x^3}{3}+\frac{x^4}{24}\right)\ln x + 9x-\frac{43x^2}{4}+\frac{19x^3}{6}-\frac{25x^4}{96}+24\sum_{n=5}^{\infty}\frac{x^n}{n!\,n(n-1)(n-2)(n-3)(n-4)}\right]$$

`>Order:=10:` ✍ Set the order of series expansion to $O(x^{10})$.

`>sol:=dsolve(ODE,y(x),series,'point'=0):` ✍ Obtain series solution.

`>convert(sol,polynom);`

$$y(x) = \_C1\left(1-4x+3x^2-\frac{2x^3}{3}+\frac{x^4}{24}\right)+\_C2\left[\left(1-4x+3x^2-\frac{2x^3}{3}+\frac{x^4}{24}\right)\ln x + 9x-\frac{43x^2}{4}+\frac{19x^3}{6}-\frac{25x^4}{96}+\frac{x^5}{600}-\frac{x^6}{21600}+\frac{x^7}{529200}-\frac{x^8}{11289600}+\frac{x^9}{228614400}\right]$$

12.3 Numerical Solutions of Differential Equations

Numerical solution of an ordinary differential equation is accomplished by `dsolve` with the option `numeric` or `type=numeric`. The default method for initial value problems is a Runge-Kutta-Fehlberg method that produces a fifth-order accurate solution. If the option `stiff=true` is specified, then the differential equation is regarded as a stiff equation, and the default method is a Rosenbrock method.

For most initial value problems, the default approach is sufficient. However, if an equation is known to be stiff, it is more efficient to specify the `stiff=true` option as illustrated in the following example.

`>with(LinearAlgebra):` ✍ Load the `LinearAlgebra` package.

`>A:=Matrix([[99998,99999],[-199998,-199999]]);` ✍ Define matrix A.

$$\mathrm{A} := \begin{bmatrix} 99998 & 99999 \\ -199998 & -199999 \end{bmatrix}$$

`>(lambda,v):=Eigenvectors(A);` ✍ Evaluate the eigenvalues and eigenvectors.

$$\lambda,\ \mathbf{v} := \begin{bmatrix} -100000 \\ -1 \end{bmatrix},\ \begin{bmatrix} -1/2 & -1 \\ 1 & 1 \end{bmatrix}$$

✍ Define the ODEs as $\mathbf{z}'(x)=\mathbf{A}\mathbf{z}(x)$ with ICs $z_1(0)=2$, $z_2(0)=-3$.

```
>ODE[1]:=diff(z[1](x),x)=A[1,1]*z[1](x)+A[1,2]*z[2](x):
>ODE[2]:=diff(z[2](x),x)=A[2,1]*z[1](x)+A[2,2]*z[2](x):
>ICs := z[1](0)=2, z[2](0)=-3:
```

✍ Solve the system of ODEs with ICs using `dsolve`.

```
>sol:=dsolve({ODE[1],ODE[2],ICs},{z[1](x),z[2](x)});
```

$$\text{sol} := \left\{z_1(x) = \mathrm{e}^{-100000x} + \mathrm{e}^{-x},\ \ z_2(x) = -2\mathrm{e}^{-100000x} - \mathrm{e}^{-x}\right\}$$

✍ $z_1(x)$ is assigned to y_1 as a function of x.

```
>y[1]:=unapply(eval(z[1](x),sol[1]),x);
```

$$y_1 := x \rightarrow \mathrm{e}^{-100000x} + \mathrm{e}^{-x}$$

✍ Solve the system of ODEs with ICs numerically using `dsolve` with the option `numeric`. The system is solved using the default Runge-Kutta-Fehlberg method. The maximum number of evaluating the right-hand side functions is set to `maxfun=1000000` with the default being 30000.

```
>yn_nonstiff:=dsolve({ODE[1],ODE[2],ICs},{z[1](x),z[2](x)},numeric,
 maxfun=1000000);
```

yn_nonstiff := proc(x_rkf45) ... end proc

✍ Solve the system numerically again using `dsolve` with the option `numeric`. The option `stiffness=true` is specified so that the system is regarded as stiff and solved using the default Rosenbrock method.

```
>yn_stiff:=dsolve({ODE[1],ODE[2],ICs},{z[1](x),z[2](x)},numeric,
 stiff=true);
```

yn_stiff := proc(x_rosenbrock) ... end proc

✍ Evaluate the value of $z_1(x)=y_1(x)$ at $x=10.0$ using three approaches.

`>y[1](10.0);` ✍ Numerical value evaluated from the analytical expression

0.00004539992976

✍ Numerical value for which the system is not regarded as stiff.

```
>yn_nonstiff(10.0);
  Error, (in yn_nonstiff) cannot evaluate the solution further right
  of 6.1284837, maxfun limit exceeded (see ?dsolve,maxfun for details)
```

✍ Due to the extremely small stepsizes that the Runge-Kutta-Fehlberg method has to take, the solution cannot go beyond $x=6.1284837$ when `maxfun` $=10^6$. When `maxfun` $=10^7$, the solution cannot go beyond $x=61.294082$. The Runge-Kutta-Fehlberg method is not efficient for this stiff equation.

`>yn_stiff(10.0);` ✍ Numerical value for which the system is regarded as stiff.

$$\left[x = 10.0,\ z_1(x) = 0.0000454805058997\,53,\ z_2(x) = -0.000045480505899753\right]$$

Example 12.23 — Dynamical Response of Parametrically Excited System

Consider the parametrically excited nonlinear system given by

$$\ddot{x} + \beta\dot{x} - (1+\mu\cos\Omega t)x + \alpha x^3 = 0.$$

Examples of this equation are found in many applications of mechanics, especially in problems of dynamic stability of elastic systems. In particular, the transverse vibration of a buckled column under the excitation of a periodic end displacement is described by this equation. The system is called parametrically excited because the forcing term $\mu\cos\Omega t$ appears in the coefficient (parameter) of the equation.

✍ It is a good practice to put `restart` at the beginning of each program so that *Maple* can start fresh if the program has to be rerun.

```
>restart:
```

```
>with(plots):
```
✍ Load the `plots` package.

```
>ODE:=diff(x(t),t$2)+beta*diff(x(t),t)-(1+mu*cos(Omega*t))*x(t)
 +alpha*x(t)^3=0:
```
✍ Define the ODE.

```
>ICs:=x(0)=0,D(x)(0)=0.1:
```
✍ Define the ICs: $x(0)=0$, $\dot{x}(0)=0.1$.

Periodic Motion ($\mu=0.3$)

```
>alpha:=1.0: beta:=0.2: Omega:=1.0: mu:=0.3:
```
✍ Assign the parameters.

✍ Solve the system numerically using `dsolve` with option `numeric`.

```
>sol:=dsolve({ODE,ICs},x(t),numeric,maxfun=1000000):
```

✍ Plot the time series $x(t)$ versus t, $(x_1=x)$. ✍ Figure 12.1(a)

```
>odeplot(sol,[t,x(t)],t=0..500,numpoints=10000,labels=["t","x1"],
 tickmarks=[[0,100,200,300,400,500],[-1.5,-1,-0.5,0,0.5,1,1.5]]);
```

✍ Plot the time series $\dot{x}(t)$ versus t, $(x_2=\dot{x})$. ✍ Figure 12.1(b)

```
>odeplot(sol,[t,D(x)(t)],t=0..500,numpoints=10000,labels=["t","x2"],
 tickmarks=[[0,100,200,300,400,500],[-0.8,-0.6,-0.4,-0.2,0,0.2,0.4,
 0.6,0.8]]);
```

✍ Plot the phase portrait $\dot{x}(t)$ versus $x(t)$. ✍ Figure 12.1(c)

```
>odeplot(sol,[x(t),D(x)(t)],t=0..500,numpoints=10000,view=[-1.8..1.8,
 -1.0..1.0], tickmarks=[[-1.8,-1.2,-0.6,0.6,1.2,1.8],[-1,-0.75,-0.5,
 -0.25,0.25,0.5,0.75,1]],axes=normal,labels=["x1","x2"]);
```

✍ When $\mu=0.3$, after some transient part, the response of the system will settle down to periodic motion.

Chaotic Motion ($\mu=0.4$)

```
>alpha:=1.0: beta:=0.2: Omega:=1.0: mu:=0.4:
```
✍ Assign the parameters.

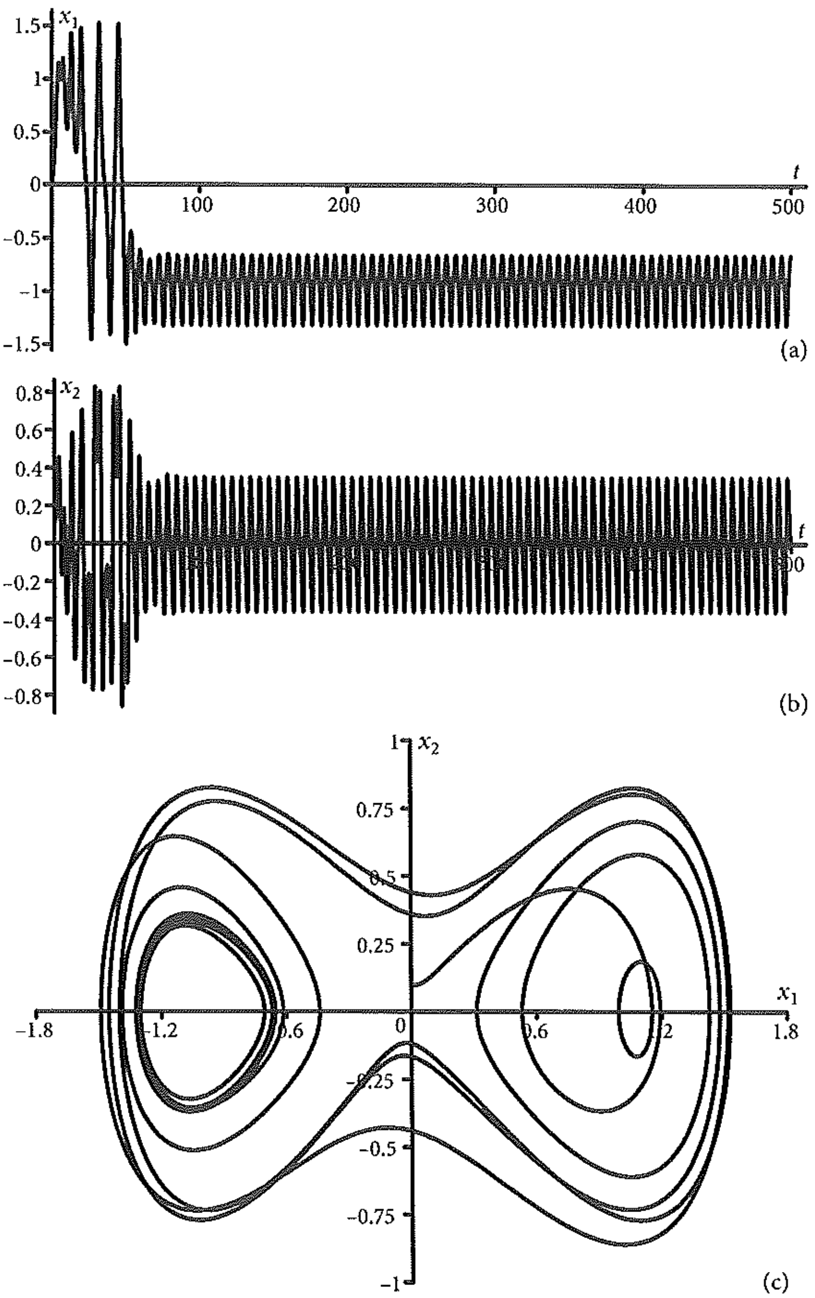


Figure 12.1 Periodic motion.

✍ Solve the system numerically using dsolve with option numeric.

```
>sol:=dsolve({ODE,ICs},x(t),numeric,maxfun=1000000):
```

✍ Plot the time series $x(t)$ versus t. ✍ Figure 12.2(a)

```
>odeplot(sol,[t,x(t)],t=0..500,numpoints=10000,labels=["t","x1"],
  tickmarks=[[0,100,200,300,400,500],[-1.5,-1,-0.5,0,0.5,1,1.5]]);
```

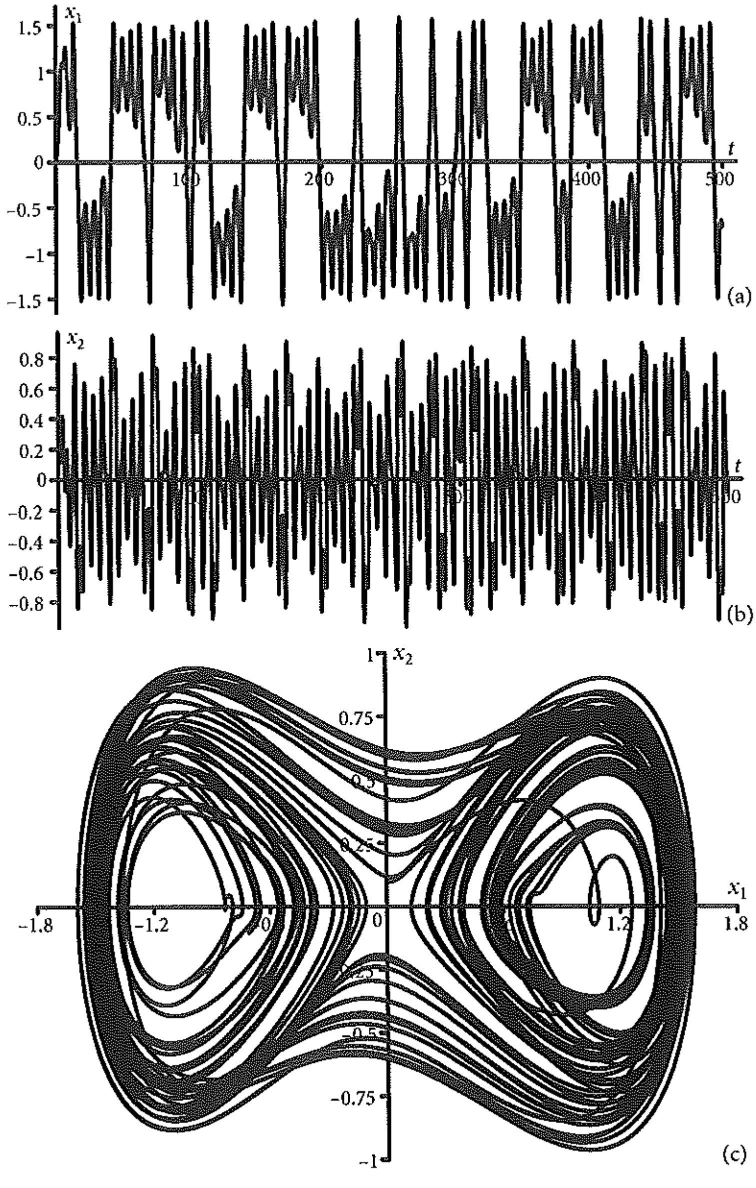


Figure 12.2 Chaotic motion.

Plot the time series $\dot{x}(t)$ versus t. Figure 12.2(b)

```
>odeplot(sol,[t,D(x)(t)],t=0..500,numpoints=10000,labels=["t","x2"],
 tickmarks=[[0,100,200,300,400,500],[-0.8,-0.6,-0.4,-0.2,0,0.2,0.4,
 0.6,0.8]]);
```

Plot the phase portrait $\dot{x}(t)$ versus $x(t)$. Figure 12.2(c)

```
>odeplot(sol,[x(t),D(x)(t)],t=0..500,numpoints=10000,view=[-1.8..1.8,
 -1.0..1.0], tickmarks=[[-1.8,-1.2,-0.6,0.6,1.2,1.8],[-1,-0.75,-0.5,
 -0.25,0.25,0.5,0.75,1]],axes=normal,labels=["x1","x2"]);
```

✍ When $\mu = 0.4$, the values of $x(t)$ or $\dot{x}(t)$ change from positive to negative or vice versa in an obviously random manner; this seemingly random motion is called chaotic motion.

Example 12.24 — Lorenz System

In 1963, Edward N. Lorenz, a meteorologist and mathematician from MIT, presented an analysis of a coupled set of three quadratic ordinary differential equations representing three modes (one in velocity and two in temperature) for fluid convection in a two-dimensional layer heated from below. The equations are

$$\dot{x} = \sigma(y - x), \quad \dot{y} = \rho x - y - xz, \quad \dot{z} = -\beta x + xy,$$

where σ is the Prandtl number, ρ is the Rayleigh number, and β is a geometric factor. All σ, ρ, $\beta > 0$, but usually $\sigma = 10$, $\beta = 8/3$ and ρ is varied. Study the behavior of the system for $\rho = 28$.

```
>restart:
>with(plots):
```
✍ Load the `plots` package.
```
>ODE[1]:=diff(x(t),t)=sigma*(y(t)-x(t)):
```
✍ Define the ODEs.
```
>ODE[2]:=diff(y(t),t)=rho*x(t)-y(t)-x(t)*z(t):
>ODE[3]:=diff(z(t),t)=-beta*z(t)+x(t)*y(t):
>ICs:=x(0)=-8,y(0)=8,z(0)=27:
```
✍ Define the ICs.
```
>beta:=8/3: sigma:=10: rho:=28:
```
✍ Assign the parameters.
```
>sol:=dsolve({ODE[1],ODE[2],ODE[3],ICs},{x(t),y(t),z(t)},numeric,
 maxfun=1000000):
```
✍ Solve the system numerically using `dsolve` with `numeric`.

✍ To understand the structure of `sol`, the solution `sol(1.0)` at time $t = 1.0$ is displayed. $t = 1.0$ is the first element of `sol(1.0)`. $x(1.0)$, $y(1.0)$, $z(1.0)$ are the second, third, and fourth elements of `sol(1.0)`, respectively.
```
>sol(1.0);
```
$[t = 1.0,\ x(t) = 9.057106766983,\ y(t) = 14.55890078678,\ z(t) = 18.41514672820]$

✍ Plot the time series $x(t)$, $y(t)$, and $z(t)$ versus t. ✍ Figure 12.3
```
>odeplot(sol,[t,x(t)],t=0..100,numpoints=10000,labels=["t","x"],
 tickmarks=[[0,20,40,60,80,100],[-15,-10,-5,0,5,10,15]]);
>odeplot(sol,[t,y(t)],t=0..100,numpoints=10000,labels=["t","y"],
 tickmarks=[[0,20,40,60,80,100],[-20,-10,0,10,20]]);
>odeplot(sol,[t,z(t)],t=0..100,numpoints=10000,labels=["t","z"],
 tickmarks=[[0,20,40,60,80,100],[0,10,20,30,40]]);
```

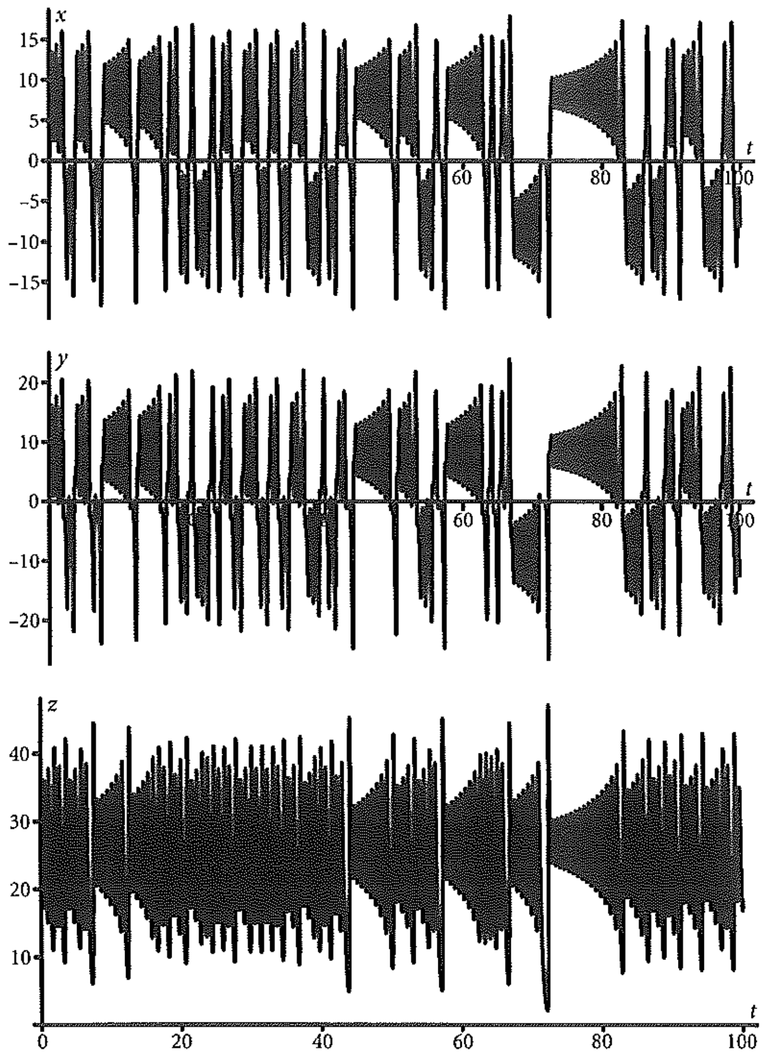


Figure 12.3 Lorenz system.

✍ Plot $z(t)$ versus $x(t)$. ✍ Figure 12.4(a)

```
>odeplot(sol,[x(t),z(t)],t=0..150,numpoints=100000,labels=["x","z"],
  view=[-20..20,0..50],tickmarks=[[-20,-10,0,10,20],[0,10,20,30,40,50]]);
```

✍ 3D plot of the variations of $x(t)$, $y(t)$, $z(t)$ with time t. ✍ Figure 12.4(b)

```
>odeplot(sol,[x(t),y(t),z(t)],t=0..150,numpoints=100000,
  orientation=[-45,65],view=[-30..30,-30..30,0..50],thickness=0,
  tickmarks=[[-30,-20,-10,0,10,20,30],[-30,-20,-10,0,10,20,30],
  [0,10,20,30,40,50]],axes=normal,labels=["x","y","z"]);
```

Remarks: When the solution is plotted in the xz-plane, the butterfly pattern is traced in "real time": the moving solution point $P(x(t), y(t), z(t))$ appears to undergo a random number of oscillations on the right followed by a random

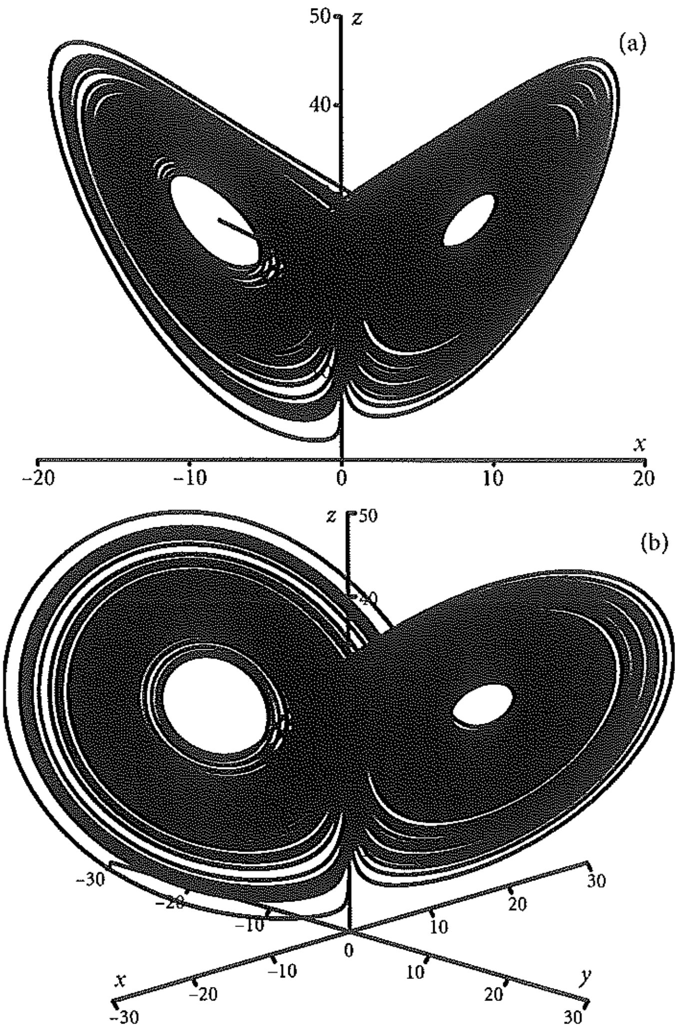


Figure 12.4 Lorenz system.

number of oscillations on the left, then a random number of oscillations on the right followed by random number of oscillations on the left, and so on. Given the meteorological origin of the Lorenz system, one naturally thinks of a random number of clear days followed by a random number of rainy days, then a random number of clear days followed by a random number of rainy days, and so on.

Solutions of Lorenz system are sensitive to the initial conditions—small variations of the initial condition may produce large variations in the long-term behavior of the system. This notion of sensitive dependence on initial conditions is the so-called butterfly effect in chaos theory—the flap of a butterfly's wings in Brazil may set off a tornado in Texas.

The difference between the solutions with only small difference in the initial conditions can be plotted to demonstrate sensitive dependence of the solutions of

Lorenz system to the initial conditions. Initially, the two solutions seem coincident. After a period of time, the difference between the two solutions is as large as the value of the solution.

✍ Define a new set of ICs. $x(0)$ is changed from -8 to -8.00001.

```
>ICs2:=x(0)=-8.00001,y(0)=8,z(0)=27:
>sol2:=dsolve({ODE[1],ODE[2],ODE[3],ICs2},{x(t),y(t),z(t)},numeric,
  maxfun=1000000):
```
✍ Solve the system again with new initial conditions `ICs2`.

```
>N:=3000:
```
✍ Number of points in the plot.

```
>for n from 0 by 1 to N do
>     T[n]:=n/100.:
```
✍ $t = 0.00, 0.01, 0.02, \ldots, 29.99, 30.00$.

```
>     delta[n]:=rhs(sol(T[n])[2])-rhs(sol2(T[n])[2]):
>end do:
```

✍ `delta[n]` is the difference between the solutions of $x(t)$ solved with two different sets of ICs at time `T[n]`.

```
>T_list:=[seq(T[n],n=0..N)];
```
✍ Create a `list` of times.

$$T\_list := [0.00, 0.01, \ldots, 29.99, 30.00]$$

```
>X_list:=[seq(delta[n],n=0..N)];
```
✍ Create a `list` of the differences of $x(t)$.

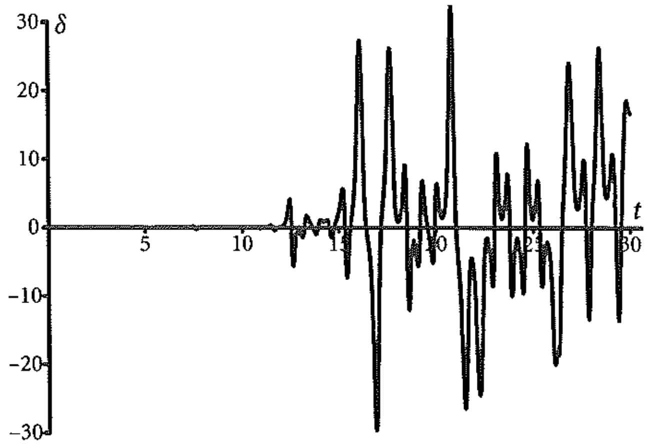
$$X\_list := [0.00001, 0.9056\times 10^{-5}, \ldots, 16.81918330, 16.71847501]$$

```
>P:=zip(`[]`,T_list,X_list);
```
✍ Merge the lists T_list and X_list using `zip`.

$$P := [[0., 0.00001], [0.01, 0.9056\times 10^{-5}], \ldots, [29.99, 16.81918330], [30., 16.71847501]]$$

✍ Plot data points using `pointplot` in the `plots` package with the data in a list of lists of the form $[[t_0, x_0], [t_1, x_1], \ldots, [t_n, x_n]]$.

```
>pointplot(P,style=line,color=black,tickmarks=[[0,5,10,15,20,25,30],
  [-30,-20,-10,0,10,20,30]],labels=["t","delta"],
  labelfont=[TIMES,ITALIC,14],axesfont=[TIMES,ROMAN,12]);
```



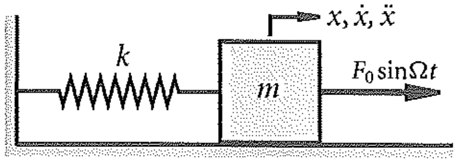
Problems

Most of the exercise and example problems presented in this book can serve as practicing problems using *Maple*.

12.1 For the example of *dynamical response of parametrically excited system*, study the dynamics of the system for $\mu = 0.34$, 0.35, and 0.57 by plotting the time series $x(t)$, $\dot{x}(t)$, and phase portraits $\dot{x}$-x using *Maple*. Discuss the results obtained.

12.2 Coulomb Dry Friction

Consider the mass-spring system moving on a rough surface as shown.



The equation of motion is given by

$$m\ddot{x} + \mu F_n \operatorname{sgn}(\dot{x}) + kx = F_0 \sin \Omega t,$$

where $\operatorname{sgn}(\cdot)$ is the signum function defined as

$$\operatorname{sgn}(\dot{x}) = \begin{cases} +1, & \dot{x} > 0; \\ 0, & \dot{x} = 0; \\ -1, & \dot{x} < 0. \end{cases}$$

The term $\mu F_n \operatorname{sgn}(\dot{x})$ is the dry friction between the mass and the rough surface, and is called Coulomb damping. However, it is not proportional to the velocity, but rather depends only on the algebraic sign (direction) of the velocity. The equation of motion is consequently nonlinear.

Rewrite the equation of motion as

$$\ddot{x} + \xi \operatorname{sgn}(\dot{x}) + \omega_0^2 x = f_0 \sin \Omega t, \qquad \xi = \frac{\mu F_n}{m}, \quad \omega_0^2 = \frac{k}{m}, \quad f_0 = \frac{F_0}{m}.$$

Suppose the parameters $\xi = 0.1$, $\omega_0 = 2$, $f_0 = 1$, and the system is at rest at time $t = 0$. Plot the response $x(t)$ of the system, for $\Omega = 1.5$, 1.9, 1.97, 2, 2.1, 2.3, 2.5, and 4, using *Maple*. Discuss the results obtained.

✍ Note that $\operatorname{sgn}(x)$ is `signum(x)` in *Maple*.

12.3 The Damped Mathieu Equation

Consider the damped Mathieu equation of the form

$$\ddot{x}(t) + 2\varepsilon\zeta\omega_0\dot{x}(t) + \omega_0^2(1 - 2\varepsilon\mu \cos \nu t)x(t) = 0,$$

which is a parametrically excited single degree-of-freedom system. The system is parametrically excited because the forcing term $2\varepsilon\mu \cos \nu t$ appears in the coefficient (parameter) of the differential equation.

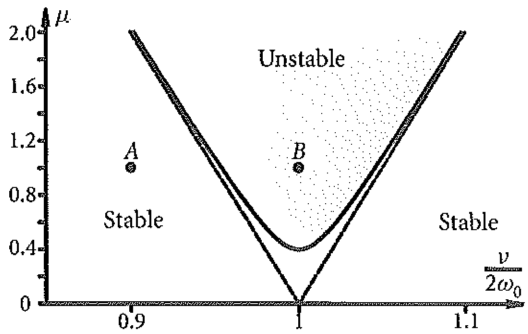
The Mathieu equation finds many applications in engineering applications. In particular, the transverse vibration of an elastic column subjected to harmonic axial load $P(t) = p_0 \cos \nu t$ is governed by the Mathieu equation.

It can be shown that, approximately, when the excitation frequency ν is in the following region

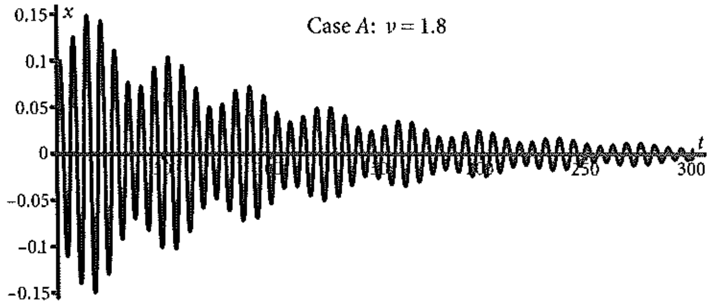
$$1 - \varepsilon\left(\frac{\mu^2}{4} - \zeta^2\right)^{1/2} < \frac{\nu}{2\omega_0} < 1 + \varepsilon\left(\frac{\mu^2}{4} - \zeta^2\right)^{1/2},$$

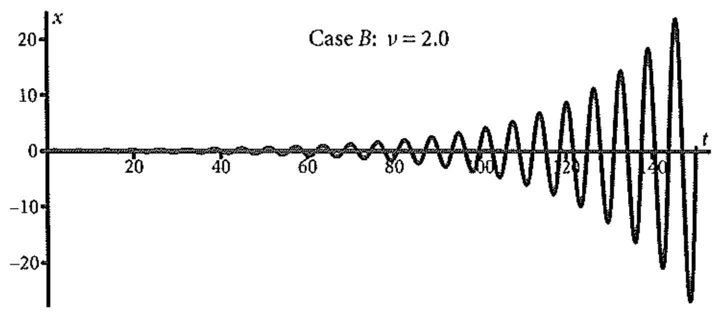
where $\varepsilon > 0$ is a small parameter, parametric resonance occurs, in which the response $x(t)$ grows exponentially.

For the parameters $\varepsilon = 0.1$, $\zeta = 0.1$, $\mu = 1$, $\omega_0 = 1$, the instability region is shown in the following figure



- For Case A, $\nu = 1.8$ and the system is in the *stable* region, in which the amplitude of the response *decays* exponentially.
- For Case B, $\nu = 2.0$ and the system is in the *unstable* region, in which the amplitude of the response *grows* exponentially.





1. For these two cases, verify the stability of the response by numerically solving the equation of motion and plotting the responses using *Maple*. Use the initial conditions $x(0) = 0$ and $\dot{x}(0) = 0.1$.
2. Numerically determine by trial-and-error (accurate to the thousandth decimal place) the instability region, i.e., the value of ν_L and ν_U such that the system is in parametric resonance when $\nu_L \leqslant \nu \leqslant \nu_U$.

12.4 The van der Pol Oscillator

The van der Pol oscillator is the second-order ordinary equation with nonlinear damping given by

$$\ddot{x} + \beta\,(x^2 - 1)\,\dot{x} + x = a\cos\Omega t,$$

where $\beta > 0$ is a parameter indicating the strength of nonlinear damping. This model was proposed by the Dutch electrical engineer and physicist Balthasar van der Pol (1889–1959) in 1920s to study circuits containing vacuum tubes. He found that these circuits have stable oscillations or *limit cycles*. When they are driven with a signal with frequency near that of the limit cycle, the resulting periodic response shifts its frequency to that of the driving signal, i.e., the circuit becomes "entrained" to the driving signal. However, the waveform or signal shape can be quite complicated and contain a rich structure of harmonics and subharmonics.

Van der Pol built a number of electronic circuit models of the human heart to study the stability of heart dynamics. A real heart driven by a pacemaker is modeled by the circuit driven by an external signal. He was interested in finding out, using his entrainment work, how to stabilize a heart's irregular beating.

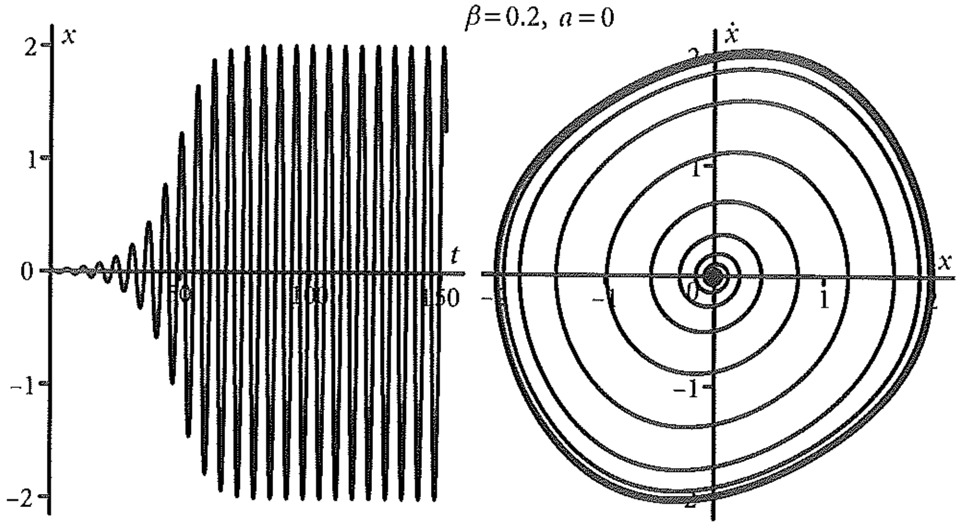
Van der Pol and his colleague van der Mark reported that an "irregular noise" was heard at certain driving frequencies between the natural entrainment frequencies. This is probably one of the first experimental reports of deterministic chaos.

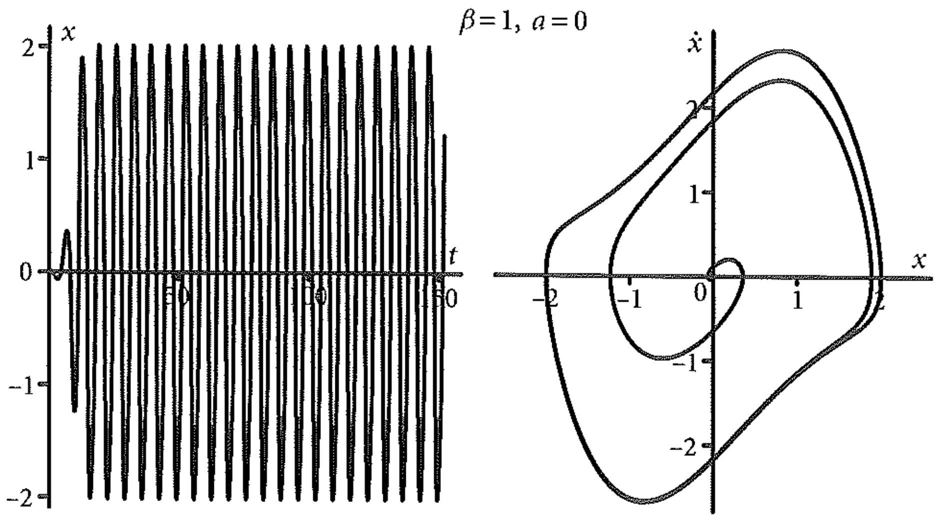
The Unforced van der Pol Oscillator

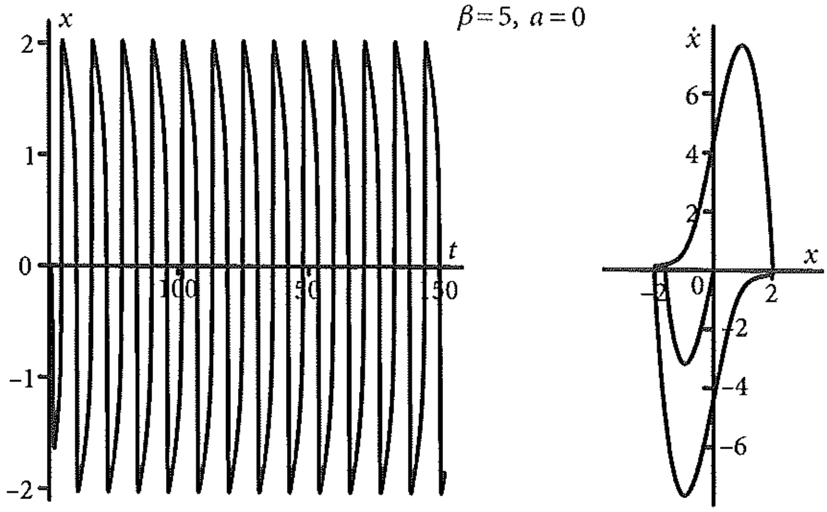
When $a = 0$, the van der Pol system is not externally driven. Using the initial conditions $x(0) = 0.01$ and $\dot{x}(0) = 0$, the wave forms (x-t plots) and phase portraits ($\dot{x}$-x plots) of the system are shown in the following figures for $\beta = 0.2$, 1, and 5. The existence of limit cycle is clearly seen. Use *Maple* to numerically solve the van der Pol equation and reproduce these results.

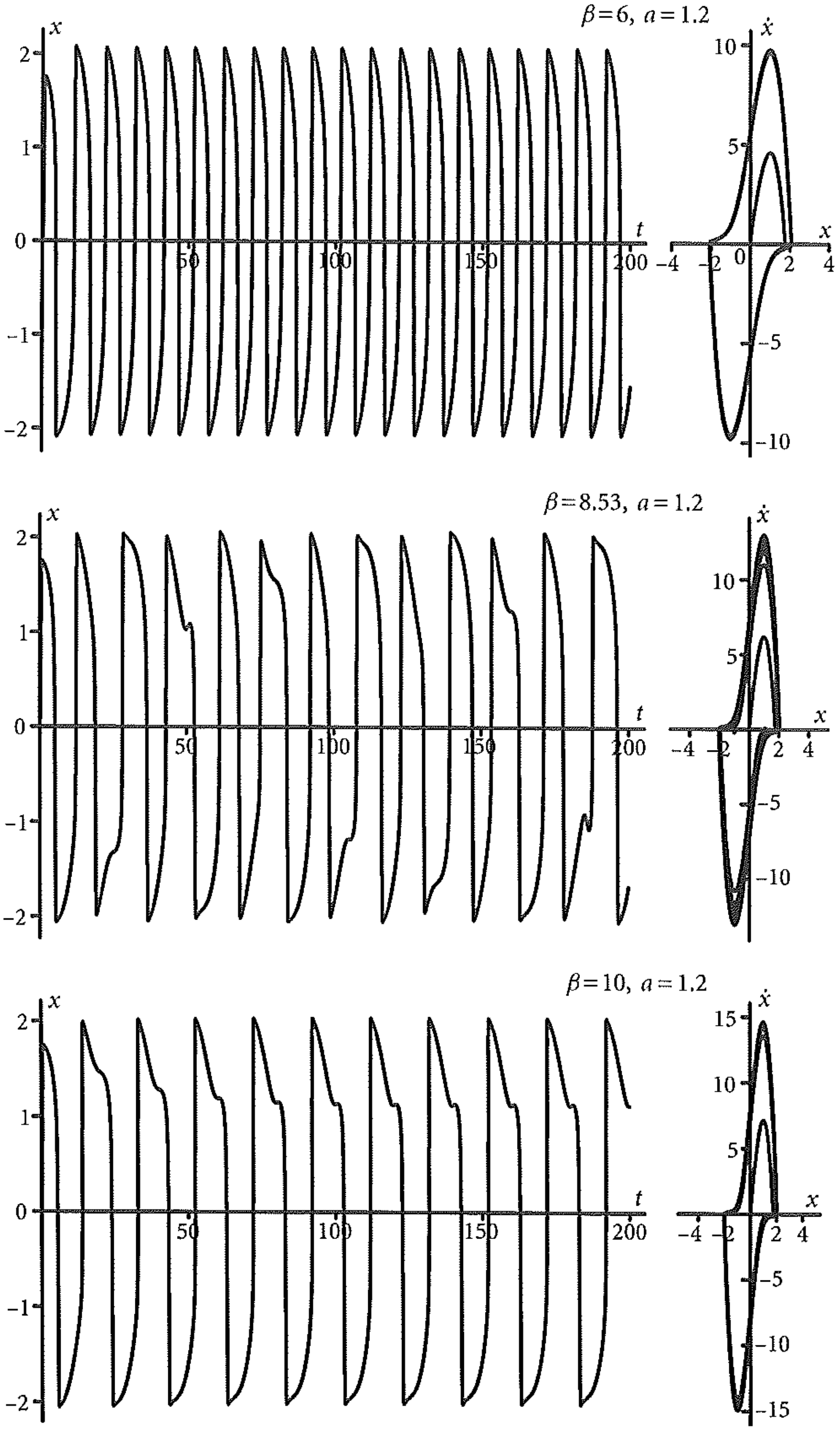
The Forced van der Pol Oscillator

When $a \neq 0$, the van der Pol system is externally driven by a sinusoidal signal. Using the initial conditions $x(0) = 0.01$ and $\dot{x}(0) = 0$, the wave forms (x-t plots) and phase portraits ($\dot{x}$-x plots) of the system are shown in the following figure for $a = 1.2$ and $\beta = 6$, 8.53, and 10. The phenomenon of entrainment can be observed. For $\beta = 8.53$, the system exhibits chaotic behavior. Use *Maple* to numerically solve the van der Pol equation and reproduce these results.


β=0.2, a=0
x
t
ẋ
x


β=1, a=0
x
t
ẋ
x


β=5, a=0
x
t
ẋ
x


$\beta=6,\ a=1.2$
$\beta=8.53,\ a=1.2$
$\beta=10,\ a=1.2$
x
t
$\dot{x}$