CAS Exercise Examples for Chapter 9: Further Applications of Integration

Section 9.1 Slope Fields and Separable Differential Equations

Slope Fields

For Exercises 25 - 36, you are asked to draw slope fields for the differential equations given. A package called PlotField, contained in a file called Graphics, contains a command called PlotVectorField which can be used to plot the slope field of a differential equation. This must be loaded first

```
<< Graphics`PlotField`
```

The Mathematica command

```
PlotVectorField[{1,f[x,y]},{x,xmin,xmax},{ymin,ymax}]
```

will plot the slope field of the differentiable equation \( y' = f(x,y) \). Think of the change in \( x \) as 1 while the change in \( y \) is \( y' \). The option AspectRatio \( \rightarrow 1 \) makes the graph height to width ratio equal to one.

Example: Obtain a slope field for the differential equation \( y' = \sqrt{xy} \).
Clear[x, y]

dirfield = PlotVectorField[\{1, Sqrt[x y], \{x, 0, 2\}, \{y, 0, 5\},
Axes \to True, Asp\{c\}tRatio \to 1, AxesLabel \to \{x, y\},
ScaleFactor \to .3\};

Solutions with Separable Variables

Now, let us look at a graph of the solution curve passing through \((0, 1)\) together with the slope field graph. First, the solution is found by first noting that the differential equation is equivalent to \(\frac{1}{\sqrt{y}} \frac{dy}{dx} = \sqrt{x} \, dx\). So you can begin by assigning an arbitrary name (such as eq) to the equation after the antiderivative of each side has been computed.

\[
eq = \int \frac{1}{\sqrt{y}} \, dy = \int \sqrt{x} \, dx + c
\]

\[
2 \sqrt{y} = c + \frac{2 x^{3/2}}{3}
\]

Now the value of \(c\) is obtained using the \texttt{Solve} and \texttt{/.} commands.

\[
cval = \texttt{Solve[eq/.\{x \rightarrow 0, y \rightarrow 1\}, c]}
\]

\[
\{\{c \rightarrow 2\}\}
\]
\begin{verbatim}
cval[[1, 1, 2]]
2
The Solve command is used again to find the solution for \( y \) in terms of \( x \).
\[
sol = \text{Solve}[eq \rightarrow \text{cval}[[1, 1, 2]], \ y]
\[
\{ \{ y \rightarrow \frac{1}{9} (9 + 6 \ x^{3/2} + \ x^3) \} \}
\]
Now you can create a function \( y = f(x) \) representing the solution just found.
\[
\text{Clear}[f];
\text{f}[x_] = \text{sol}[[1, 1, 2]]
\[
\frac{1}{9} (9 + 6 \ x^{3/2} + \ x^3)
\]
The option \text{PlotStyle} -> \{\text{Thickness}[.015], \text{RGBColor}[0, 0, 1]\} is used to increase the thickness of the solution curve and to display the curve in color blue.
\[
solgraph = \text{Plot}[\text{f}[x], \{x, 0, 2\}, \text{AspectRatio} \rightarrow 1, \text{PlotStyle} \rightarrow \{\text{Thickness}[.015], \text{RGBColor}[0, 0, 1]\}];
\end{verbatim}
Solutions Curves with Direction Fields

Show[{solgraph, dirfield}];

If you wish to find and plot more solution curves with the direction field, simply re-execute the following set of commands, changing the terms in red as necessary. To show all solutions found with direction field, add additional names, e.g., solgraph3, to the SHO command.

cval = Solve[eq /. {x → 0, y → 2}, c]
sol = Solve[eq /. {c → cval[[1, 1, 2]]}, y]
f2[x_] = sol[[1, 1, 2]]
solgraph2 = Plot[f2[x], {x, 0, 2}, AspectRatio -> 1,
               PlotStyle -> {Thickness[.015], RGBColor[0, 1, 0]}];
Show[{solgraph, solgraph2, dirfield}, PlotRange -> {0, 5}];

\{c \to 2 \sqrt{2}\}

\{y \to \frac{1}{9} \left(18 + 6 \sqrt{2} \ x^{3/2} + x^3\right)\}

\frac{1}{9} \left(18 + 6 \sqrt{2} \ x^{3/2} + x^3\right)
Exploring Solutions Graphically Using Slope Fields

Consider Exercise 33 where the variables cannot be separated. Can you see a pattern to the solutions? Could you trace a solution curve starting at the initial point (0, 2)? You will explore other methods for finding solutions in later sections.

\[
\text{Clear}[x, y]\\
\text{df33} = \text{PlotVectorField}[[1, \cos[2 x - y]], \{x, 0, 5\}, \{y, 0, 5\},\\
\text{Axes} \rightarrow \text{True}, \text{AspectRatio} \rightarrow 1, \text{AxesLabel} \rightarrow \{x, y\},\\
\text{ScaleFactor} \rightarrow .3];
\]

When plotting the direction field for Exercise 34, you must start y at a value slightly larger than 0, since the logarithm of 0 is not defined. Do the solution curves seem to be approaching a certain value? Consider the values of y where y' is zero.
Clear[x, y]

def34 = PlotVectorField[{1, y (1/2 - Log[y])}, {x, 0, 4}, 
   {y, .1, 3}, Axes -> True, AspectRatio -> 1, AxesLabel -> {x, y}, 
   ScaleFactor -> .4];

Using Mathematica to Find General Solutions to Differential Equations

The command `DSolve[eqn, y[x], x]` will solve a differential equation involving $y'(x)$ for $y$ in terms of $x$. For example, the following input command is used to solve the differential equation $y' = x \sqrt{(x - 1)^2}$ for $y$ in terms of $x$. It is critical here that the dependent variable be referred to as $y[x]$, not simply as $y$. Note also the double inequality used for the equation. Try this with other functions by changing the terms in red.

$$\text{sol} = \text{DSolve} \left[ y'[x] = x \sqrt{(x - 1)^2}, y[x], x \right]$$

$$\left\{ \left\{ y[x] \to (-1 + x)^2 \left( \frac{9}{40} - \frac{3 x}{20} + \frac{3 x^2}{8} \right) + C[1] \right\} \right\}$$

The solution is extracted using the command `sol [[1, 1, 2]]`. The value $C[1]$ in the solution represents an arbitrary constant. In the following input cell, the command `/.` is used to replace $C[1]$ with specific values and the solution is then plotted for these specific values of $C[1]$. 
Now suppose you want to find and plot the solution passing through \((\frac{1}{2}, -2)\). Begin by letting \(y\) equal the solution found earlier with \(x\) replaced with \(\frac{1}{2}\).

\[
y = \text{sol}[[1, 1, 2]] \rightarrow \frac{1}{2}
\]

\[
-\frac{33}{160} \ 2^{2/3} + C[1]
\]

The command \(\text{Solve} [eqn, \ var]\) will attempt to find the solutions to the variable \(\var\) in the equation \(eqn\). Therefore you can use the \(\text{Solve}\) command to find the value of the constant \(C[1]\) for which \(y=-2\).

\[
\text{const} = \text{Solve}[y == -2, C[1]]
\]

\[
\{\{C[1] \rightarrow \frac{1}{320} \ (-640 + 33 \ 2^{1/3})\}\}\]

Now the specific solution is plotted by replacing \(C[1]\) with the constant just obtained.
Using *Mathematica* to Find Particular Solutions to Differential Equations

The same **DSolve** command used above works when an initial condition is given, except that the initial condition becomes part of the equation system (enclosed in braces). Consider Exercise 33 again. The warning messages appear here due to the use of the inverse trigonometric functions in the solution process. As before, you can check out other functions by changing the terms in red.

\[
\text{sol33} = \text{DSolve}\left[\{y'[x] = \cos(2x - y[x]), y[0] = 2\}, y[x], x\right]
\]

Solve::ifun:
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. More...

Solve::ifun:
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. More...

\[
\{\{y[x] \to 2 \left( x - \arctan\left[ \frac{\frac{1}{4} \left( 2 \sqrt{3} x - 4 \arctan\left[ \frac{\sqrt{3} \tan[1]}{3} \right] \right)}{\sqrt{3}} \right] \right) \}\}
\]

This problem with inverse functions can be visualized with the graphs.

\[
\text{pl33} = \text{Plot}[y[x] / . \text{sol33}, \{x, 0, 5\}, \text{AspectRatio} \to 1, \\
\text{PlotStyle} \to \{\text{Thickness}[.015], \text{RGBColor}[0, 0, 1]\}];
\]
\[
\text{Show}[\text{pl33}, \text{df33}, \text{PlotRange} \to \{0, 5\}];
\]
CAS Exercise Examples for Chapter 9
Using *Mathematica* to Find Numerical Solutions to Differential Equations

Many differential equations cannot be solve analytically. This is the way a numerical solution is found. Note the use `N` before the `DSolve` and also the fact that a domain must be given for the independent variable. An initial condition is required here, and the domain starts with that initial value of the independent variable. We will re-do Exercise 33 and plot the results. You may ignore warnings about possible spelling errors.

```mathematica
nsol33 = NDSolve[{y'[x] == Cos[2 x - y[x]], y[0] == 2}, y[x], {x, 0, 5}]
npl33 = Plot[y[x] /. nsol33, {x, 0, 5}, AspectRatio -> 1,
   PlotStyle -> {Thickness[.015], RGBColor[0, 1, 0]}];
Show[npl33, PlotRange -> {0, 5}];
{{y[x] -> InterpolatingFunction[{{0., 5.}}, <>][x]}}
```

![Graph of the solution](image)
Section 9.3 Euler's Method

Using Mathematica to Perform Euler's Method

Using Mathematica, you can define the functions \( x[i] \) to represent \( x_i \), \( y[i] \) to represent \( y_i \) and \( f[x, y] \) to equal \( f(x, y) \). Then, after defining \( x[0] \) and \( y[0] \) to equal the given values of \( x_0 \) and \( y_0 \) and setting \( dx \) to equal the step size, let

\[
x[i_] := x[i-1] + dx \quad \text{and} \quad y[i_] := y[i-1] + f[x[i-1],y[i-1]] \cdot dx
\]

Notice also that delayed equals (:=) are used for these two assignment statements since we do not want to evaluate the right hand side of the equations until a specific value of \( i \) is used.

When Mathematica is then asked to compute \( y[3] \) for example, it will first compute \( y[0], y[1], \) and \( y[2] \) using the assignment statements above. If you then have Mathematica compute \( y[4] \), it will first recompute all the previous values \( y[0] \) through \( y[3] \) again. To let Mathematica remember the previous values of \( x \) and \( y \), use the following assignment statements instead:

\[
x[i_] := x[i] = x[i-1] + dx \quad \text{and} \quad y[i_] := y[i] = y[i-1] + f[x[i-1],y[i-1]] \cdot dx
\]

Consider the following example.
Example: Use Euler's method to estimate the value of the solution to the initial value problem $y' = 1 - xe^y$, $y(0) = 1$ at the point $x^* = 2$ using the step size $dx = 0.1$. Compare the accuracy of your solution with the exact solution.

Here are the necessary assignment statements for Euler's method.

```math
Clear[f, x, y];
fx_, y_] = 1 + x^2 y;
x[0] = 1.; y[0] = 2.; dx = 0.1;
x[i_] := x[i] = x[i - 1] + dx
y[i_] := y[i] = y[i - 1] + f[x[i - 1], y[i - 1]] dx
```

The command `Table[{x[i], y[i]}, {i, 0, n}]` will form $\{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$.

```math
eulervals = Table[{x[i], y[i]}, {i, 0, 10}]

\{\{1., 2.\}, \{1.1, 2.3\}, \{1.2, 2.6783\}, \{1.3, 3.16398\},
\{1.4, 3.79869\}, \{1.5, 4.64323\}, \{1.6, 5.78796\},
\{1.7, 7.36967\}, \{1.8, 9.59951\}, \{1.9, 12.8097\}, \{2., 17.5341\}\}
```

A nice command for better visualization of the list of order pairs is the `TableForm` command and the additional option `TableHeadings`, appearing in the following input cell, will place appropriate headings on each column.

```math
TableForm[eulervals, TableHeadings -> {None, \"x_i\", \"y_i\"}]]

<table>
<thead>
<tr>
<th>x_i</th>
<th>y_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.</td>
</tr>
<tr>
<td>1.1</td>
<td>2.3</td>
</tr>
<tr>
<td>1.2</td>
<td>2.6783</td>
</tr>
<tr>
<td>1.3</td>
<td>3.16398</td>
</tr>
<tr>
<td>1.4</td>
<td>3.79869</td>
</tr>
<tr>
<td>1.5</td>
<td>4.64323</td>
</tr>
<tr>
<td>1.6</td>
<td>5.78796</td>
</tr>
<tr>
<td>1.7</td>
<td>7.36967</td>
</tr>
<tr>
<td>1.8</td>
<td>9.59951</td>
</tr>
<tr>
<td>1.9</td>
<td>12.8097</td>
</tr>
<tr>
<td>2.</td>
<td>17.5341</td>
</tr>
</tbody>
</table>
```
From the table, it appears that $y \approx 17.5341$ when $x = 2$.

If you want, you can now take the ordered pairs in the table and plot them to get an approximation of the solution curve.

```math
\text{eulerplot} = \text{ListPlot[eulervals, PlotJoined -> True, PlotStyle -> \{RGBColor[0, 0, 1]\}}];
```

The exact solution to $y' = f(x, y)$ where $y(x_0) = y_0$ can be found using the command

\[
\text{DSolve[\{y'[x] == f[x, y], y[x_0] == y_0\}, y[x], x].}
\]

```math
\text{Clear[x, y]; sol = \text{DSolve[\{y'[x] == 1 + x^2 y[x], y[1] == 2\}, y[x], x]} \}
\text{\{y[x] \to \frac{1}{3} e^{-\frac{x^3}{3}} \},}
\text{\left(6 + e^{1/3} \text{ExpIntegralE[\frac{2}{3}, \frac{1}{3}]} - e^{1/3} x \text{ExpIntegralE[\frac{2}{3}, \frac{x^3}{3}]} \right)\}}
```

Now the exact value of $y$ when $x = 2$ is found.

```math
\text{exactyval} = \text{sol[[1, 1, 2]] /. x -> 2.}
\text{25.6527}
```

To see why our approximation is so far off, a plot of the approximate solution and the exact solution are displayed in the following output cell. To obtain a better approximation, a smaller value of $dx$ should be used.
 Improved Euler's Method

If you replace the previous code for Euler's method with the following improved Euler's method, much better results will usually be obtained. Study the following input cell. As before, change the terms in red for different functions.

```
Clear[f, x, y];
f[x_, y_] = 1 + x^2 y;
x[0] = 1.; y[0] = 2.; dx = 0.1;
x[i_] := x[i] = x[i - 1] + dx;
z[i_] := z[i] = y[i - 1] + f[x[i - 1], y[i - 1]] dx;
y[i_] := y[i] = y[i - 1] + f[x[i - 1], y[i - 1]] + f[x[i], z[i]] dx/2

improvedEulervals = Table[{x[i], y[i]}, {i, 0, 10}]
{{1., 2.}, {1.1, 2.33915}, {1.2, 2.77667}, {1.3, 3.35345}, {1.4, 4.1308}, {1.5, 5.20266}, {1.6, 6.71654}, {1.7, 8.9097}, {1.8, 12.1739}, {1.9, 17.1734}, {2., 25.0678}}
```

From the table, it appears that y is approximately 25.0678 when x=2. Much better! The plot of the graphs show how good the approximate solution is compared to the exact solution.
improvedeplot = ListPlot[improvedEulervals,
   PlotStyle -> {PointSize[.04], RGBColor[0, 1, 0]}];

Show[{eulerplot, improvedeplot, exactplot}];