9.1: Computing the Area Under a Curve

This chapter essentially deals with the problem of computing the area under a curve. First, we will employ a basic approach and form trapezoids under a curve. From these trapezoids, we can calculate the total area under a given curve. This method can be tedious and is prone to errors, so in the second half of the chapter, we will utilize a built-in MATLAB function to carry out numerical integration.

A Basic Approach

There are various methods to calculating the area under a curve, for example, Rectangle Method, Trapezoidal Rule and Simpson's Rule. The following procedure is a simplified method.

Consider the curve below:

![Figure](https://eng.libretexts.org/Bookshelves/Computer_Science/Book%3A_A_Brief_Introduction_to_Engineering_Computation_with…)
Each segment under the curve can be calculated as follows:

\[
\frac{1}{2}
\left( y_0 + y_1 \right) \Delta x + \frac{1}{2}
\left( y_1 + y_2 \right) \Delta x + \frac{1}{2}
\left( y_2 + y_3 \right) \Delta x
\]

Therefore, if we take the sum of the area of each trapezoid, given the limits, we calculate the total area under a curve. Consider the following example.

Given the following data, plot an x-y graph and determine the area under a curve between x=3 and x=30

<table>
<thead>
<tr>
<th>Index</th>
<th>x [m]</th>
<th>y [N]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>27.00</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>14.50</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>9.40</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>6.70</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>5.30</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>4.50</td>
</tr>
</tbody>
</table>

Data Set

First, let us enter the data set. For x, issue the following command:

\[ x = [3, 10, 15, 20, 25, 30]; \]

For y,

\[ y = [27, 14.5, 9.4, 6.7, 5.3, 4.5]; \]

If you type in \[ [x', y'] \], you will see the following tabulated result. Here we transpose row vectors with ' and displaying them as columns:

\[
\begin{array}{ccc}
3.0000 & 27.0000 \\
10.0000 & 14.5000 \\
15.0000 & 9.4000 \\
20.0000 & 6.7000 \\
25.0000 & 5.3000 \\
30.0000 & 4.5000
\end{array}
\]

Compare the data set above with the given information in the question.

To plot the data type the following:

ParseError: invalid XmlNode (click for details)

The following figure is generated:
To compute $dx$ for consecutive $x$ values, we will use the index for each $x$ value, see the given data in the question:

$$dx=[x(2)-x(1), x(3)-x(2), x(4)-x(3), x(5)-x(4), x(6)-x(5)];$$

$dy$ is computed by the following command:

$$dy=[0.5*(y(2)+y(1)), 0.5*(y(3)+y(2)), 0.5*(y(4)+y(3)), 0.5*(y(5)+y(4)), 0.5*(y(6)+y(5))];$$

$dx$ and $dy$ can be displayed with the following command: $[dx', dy']$. The result will look like this: ParseError: invalid XmlNode

Our results so far are shown below:

<table>
<thead>
<tr>
<th>$x$ [m]</th>
<th>$y$ [N]</th>
<th>$dx$ [m]</th>
<th>$dy$ [N]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>27.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>14.50</td>
<td>7.00</td>
<td>20.75</td>
</tr>
<tr>
<td>15</td>
<td>9.40</td>
<td>5.00</td>
<td>11.95</td>
</tr>
<tr>
<td>20</td>
<td>6.70</td>
<td>5.00</td>
<td>8.05</td>
</tr>
<tr>
<td>25</td>
<td>5.30</td>
<td>5.00</td>
<td>6.00</td>
</tr>
<tr>
<td>30</td>
<td>4.50</td>
<td>5.00</td>
<td>4.90</td>
</tr>
</tbody>
</table>

$x$, $y$ and corresponding differential elements
If we multiply \( dx \) by \( dy \), we find \( da \) for each element under the curve. The differential area \( da = dx \cdot dy \), can be computed using the 'term by term multiplication' technique in MATLAB as follows:

\[
\begin{align*}
da &= dx .* dy \\
da &= 145.2500 \quad 59.7500 \quad 40.2500 \quad 30.0000 \quad 24.5000
\end{align*}
\]

Each value above represents an element under the curve or the area of trapezoid. By taking the sum of array elements, we find the total area under the curve.

\[
\text{sum}(da) \\
\text{ans} = 299.7500
\]

The following illustrates all the steps and results of our MATLAB computation.

<table>
<thead>
<tr>
<th>( x ) [m]</th>
<th>( y ) [N]</th>
<th>( dx ) [m]</th>
<th>( dy ) [N]</th>
<th>( dA ) [Nm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>27.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>14.50</td>
<td>7.00</td>
<td>20.75</td>
<td>145.25</td>
</tr>
<tr>
<td>15</td>
<td>9.40</td>
<td>5.00</td>
<td>11.95</td>
<td>59.75</td>
</tr>
<tr>
<td>20</td>
<td>6.70</td>
<td>5.00</td>
<td>8.05</td>
<td>40.25</td>
</tr>
<tr>
<td>25</td>
<td>5.30</td>
<td>5.00</td>
<td>6.00</td>
<td>30.00</td>
</tr>
<tr>
<td>30</td>
<td>4.50</td>
<td>5.00</td>
<td>4.90</td>
<td>24.50</td>
</tr>
</tbody>
</table>

Computation of the approximate area under a curve

### The Trapezoidal Rule

Sometimes it is rather convenient to use a numerical approach to solve a definite integral. The trapezoid rule allows us to approximate a definite integral using trapezoids.

The

\[
\text{trapz}
\]
Command

\[ \text{Z} = \text{trapz}(Y) \] computes an approximation of the integral of \( Y \) using the trapezoidal method.

Now, let us see a typical problem.

Given \( \text{Area} = \int_{2}^{5} x^2 \, dx \), an analytical solution would produce 39. Use \text{trapz} command and solve it

1. Initialize variable \( x \) as a row vector, from 2 with increments of 0.1 to 5: \( x = 2 : 0.1 : 5 \);
2. Declare variable \( y \) as \( y = x^2 \). Note the following error prompt: ```??? Error using ==> mpower Inputs must be a scalar and a square matrix. ``` This is because \( x \) is a vector quantity and MATLAB is expecting a scalar input for \( y \). Because of that, we need to compute \( y \) as a vector and to do that we will use the dot operator as follows: \( y = x .^2 \). This tells MATLAB to create vector \( y \) by taking each \( x \) value and raising its power to 2.
3. Now we can issue the following command to calculate the first area, the output will be as follows:

   ```matlab
   area1 = trapz(x, y)
   area1 =
   39.0050
   ```

   Notice that this numerical value is slightly off. So let us increase the number of increments and calculate the area again:

   ```matlab
   x = 2 : 0.01 : 5;
   y = x .^2;
   area2 = trapz(x, y)
   area2 =
   39.0001
   ```

   Yet another increase in the number of increments:

   ```matlab
   x = 2 : 0.001 : 5;
   y = x .^2;
   area3 = trapz(x, y)
   area3 =
   39.0000
   ```

Determine the value of the following integral:

\[ \int_{0}^{\pi} \sin(x) \, dx \]

1. Initialize variable \( x \) as a row vector, from 0 with increments of \( \pi/100 \) to \( \pi \): \( x = 0 : \pi/100 : \pi \);
2. Declare variable \( y \) as \( y = \sin(x) \);
3. Issue the following command to calculate the first area, the output will be as follows:

```matlab
area1 = trapz(x,y)
area1 =
    1.9998
```

Let us increase the increments as above:

```matlab
x=0:pi/1000:pi;
y=sin(x);
area2 = trapz(x,y)
area2 =
    2.0000
```

A gas expands according to the law, \( PV^{1.4} = c \). Initially, the pressure is 100 kPa when the volume is 1 m\(^3\). Write a script to compute the work done by the gas in expanding to three times its original volume\(^1\).

Recall that PV diagrams can be used to estimate the net work performed by a thermodynamic cycle, see Wikipedia or we can use definite integral to compute the work done (WD) as follows:

\[
\text{WD} = \int p \, dv
\]

If we rearrange the expression pressure as a function of volume, we get:

\[
P = \frac{c}{V^{1.4}}
\]

By considering the initial state, we can determine the value of \( c \):

\[
c = 100 \times 1^{1.4} = 100
\]

From the equation and the equation above, we can write:

\[
P = \frac{100}{V^{1.4}}
\]

By inserting \( P \) in WD, we get:

\[
\text{WD} = \int_{1}^{3} \frac{100}{v^{1.4}} \, dv
\]

For MATLAB solution, we will consider \( P \) as a function of \( V \) and WD. Now, let us apply the three-step approach we have used earlier:

1. Initialize variable volume as a row vector, from 1 with increments of 0.001 to 3: \( v = 1:0.001:3; \)
2. Declare variable pressure as \( p = 100./v.^{1.4}; \)
3. Use the \texttt{trapz} function to calculate the work done, the output will be as follows:
A body moves from rest under the action of a direct force given by \( F = \frac{15}{x+3} \) where \( x \) is the distance in meters from the starting point. Write a script to compute the total work done in moving a distance 10 m. \(^2\)

Recall that the general definition of mechanical work is given by the following integral, see Wikipedia:

\[
\text{WD} = \int F \, dx
\]

Therefore we can write:

\[
\text{WD} = \int_{0}^{10} \frac{15}{x+3} \, dx
\]

Applying the steps we followed in the previous examples, we write:

\[
\text{WD} = \int_{0}^{10} \frac{15}{x+3} \, dx
\]

The output of the above code is:

A body moves from rest under the action of a direct force given by \( F = \frac{15}{x+3} \) where \( x \) is the distance in meters from the starting point. Compute the total work done in moving a distance 10 m.

\[
\text{WorkDone} =
\]

21.9951
Function

As we have seen earlier, `trapz` gives a good approximation for definite integrals. The `integral` function streamlines numerical integration even further. Before we learn about `integral` function, first we will look at anonymous functions.

Anonymous Functions

An anonymous function is a function that can be defined in the command window (i.e. it does not need to be stored in a program file). Anonymous functions can accept inputs and return outputs, just as standard functions do such as `sqrt(X)` or `log(X)`.

To define an anonymous function, first we create a handle with `@(x)` and type in the function: `myfunction=@(x) x^2+1`.

If you want to evaluate `myfunction` at 1, just type in `a=myfunction(1)` at the command window and you get the result of 2.

Syntax for `integral` To evaluate an integral from a minimum to a maximum value, we specify a function and its minimum and maximum `Z = integral(fun,xmin,xmax)`.

Given `y=x^2`, evaluate the integral from `x=2` to `x=5` as we have done it with `trapz` command.

1. Define function `myfunction=@(x) x.^2;`
2. Apply the syntax to `myfunction` as follows `Z = integral(myfunction,2,5)`
3. You should get a result of `Z = 39`. aside

Notice that, unlike in

```
      trapz
```

example, we did not need to define a vector and change the increments to get an accurate result.

Summary of Key Points

1. In its simplest form, numerical integration involves calculating the areas of segments that make up the area under a curve,
2. MATLAB has built-in functions to perform numerical integration,
4. Anonymous functions are inline statements that we can define with `@(x),`
5. `Z = integral(fun,xmin,xmax)` numerically integrates function `fun` from `xmin` to `xmax`. 
Footnotes