Although the convolution operation is completely defined by equation (7.4) it is helpful to explore some graphical concepts that aid in actually performing convolution. The two functions that are multiplied and then summed over $-\infty < m < \infty$ are $x[m]$ and $h[n-m]$. To illustrate the idea of graphical convolution, let the two functions $x[n]$ and $h[n]$ be the simple functions illustrated in figure 7.6.

Since the summation index in (7.4) is $m$, the function $h[n-m]$ should be considered a function of $m$ for purposes of performing the summation in (7.4). With that point of view, we can imagine that $h[n-m]$ is created by two transformations, $m \rightarrow -m$, which changes $h[m]$ to $h[-m]$, and then $m \rightarrow m - n$, which changes $h[-m]$ to $h[-(m - n)] = h[n-m]$. The first transformation $m \rightarrow -m$ forms the discrete-time inverse of $h[m]$ and the second transformation $m \rightarrow m - n$ shifts the already-time-inverted function $n$ units to the right (fig. 7.7).

Now, realizing that the convolution result is $y[n] = \sum_{m=-\infty}^{\infty} x[m] h[n-m]$, the process of graphing the convolution result $y[n]$ versus $n$ is to pick a value of $n$ and do the operation $\sum_{m=-\infty}^{\infty} x[m] h[n-m]$ for that $n$, plot that single numerical result for $y[n]$ at that $n$ and then repeat the whole process for each $n$. Every time a new $n$ is chosen, the function $h[n-m]$ shifts to a new position, $x[m]$ stays right where it is because there is no $n$ in $x[m]$ and the summation $\sum_{m=-\infty}^{\infty} x[m] h[n-m]$ is simply the sum of the products of the values of $x[m]$ and $h[n-m]$ for that choice of $n$. Figure 7.8 is an illustration of this process. For all values of $n$ not represented in figure 7.8 $y[n] = 0$, so we can now graph $y[n]$ as illustrated in figure 7.9.

It is very common in engineering practice for both signals being convolved to be zero before some finite time. Let $x$ be zero before time $n = n_x$, and let $h$ be zero before time $n = n_h$. The convolution sum is

$$x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m] h[n-m]$$
Chapter 7  The Convolution Sum

Chapter 7  The Convolution Sum

Figure 7.8
y[n] for n = -1, 0, 1, and 2.

Since x is zero before time n = nx, all the terms in the summation for m < nx are zero and

\[ x[n] \ast h[n] = \sum_{m=nx}^{\infty} x[m] h[n-m] \]

Also, when n - m < nh, the h terms are zero. That puts an upper limit on m of n - nh and

\[ x[n] \ast h[n] = \sum_{m=nx}^{n-nh} x[m] h[n-m] \]

For those n’s for which n - nh < nx, the lower summation limit is greater than the upper summation limit and the convolution result is zero. So it would be more complete and accurate to say that the convolution result is

\[ x[n] \ast h[n] = \begin{cases} \sum_{m=nx}^{n-nh} x[m] h[n-m], & n-nh \geq nx \\ 0, & n-nh < nx \end{cases} \]

In this case, another method of doing a convolution sum is sometimes convenient. We can form an array of numbers and compute the convolution sum directly from them in a systematic way (fig. 7.10).
Figure 7.10
Array method of computing a convolution sum.

Let the first row of numbers in the array be the values of \( x \) from \( n_x \) upward and let the second row of numbers in the array be the values of \( h \) from \( n_h \) upward. Each entry in the third row is the product of the value of \( x \) in that column times the value of \( h \) in the first column. Each entry in the fourth row is the value of \( x \) in the previous column times the value of \( h \) in the second column. This pattern is repeated for as many values of the convolution sum as desired. The value of the convolution sum at \( n = n_x + n_h \) is the sum of the values in the first column from the third row down (which is just one number in the first column). The value of the convolution sum at \( n = n_x + n_h + 1 \) is the sum of the values in the second column from the third row down. All further values of the convolution are computed by an extension of this algorithm.

**Example 7.3**

### Convolving Two Signals Using the Array Method

Let \( x \) be zero before time \( n = -2 \) and let \( h \) be zero before time \( n = 3 \) and let the first few nonzero values of \( x \) and \( h \) be

\[
\begin{align*}
 n & \quad -2 & -1 & 0 & 1 & \quad n & \quad 3 & 4 & 5 & 6 \\
x[n] & \quad 4 & 1 & -3 & -1 & \quad h[n] & \quad 1 & 6 & -2 & 3
\end{align*}
\]

Using the array method, find the first four nonzero values of \( y[n] = x[n] \ast h[n] \).

**Solution**

The array is shown in figure 7.11.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>-3</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>-2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-3</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>6</td>
<td>-18</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8</td>
<td>-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>-5</td>
<td>-9</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.11
Array for computing values of \( y \) from \( n = 1 \) to \( n = 4 \).