Today we introduce discrete convolution for solving the zero-state response for arbitrary input $x$. This turns out to be much more practically useful than convolution in continuous systems because it leads to a simple algorithm. After today you should be able to:

1. use convolution to determine the zero-state response using a convergent series
2. use convolution to determine the zero-state response using a table
3. use convolution to determine the zero-state response using a graphical method
4. express convolution in algorithmic form for finite duration signals

**Discrete Convolution**

Consider an arbitrary DT signal $x[n]$. We can write this as a sum over all samples picked out by a delta

$$x[n] = \sum_{m=-\infty}^{\infty} x[m] \delta[n-m]$$

Suppose a system is characterized by its impulse response $h[n]$. Then by linearity

$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} x[m] h[n-m]$$

This is called the discrete convolution sum.

- if the system and the input are causal then $x[m]$ and $h[n-m]$ overlap only for $0 < m < n$ and the convolution simplifies to

  $$y[n] = \sum_{m=0}^{n} x[m] h[n-m]$$

- similarly if $x \neq 0$ only for some range $n^l_x \leq n \leq n^u_x$ and $h \neq 0$ only for some range $n^l_h \leq n \leq n^u_h$, define the width of $x$ as $W_x = n^u_x - n^l_x$ and the width of $h$ as $W_h = n^u_h - n^l_h$, and the width of the result is $W_x + W_h$ with

  $$y[n] = \sum_{m=n^l_x}^{n} x[m] h[n-m]$$
• convolution is commutative \( x_1[n] * x_2[n] = x_2[n] * x_1[n] \)

• convolution is distributive
\[
x_1[n] * (x_2[n] + x_3[n]) = x_1[n] * x_2[n] + x_1[n] * x_3[n]
\]

• convolution is associative
\[
x_1[n] * (x_2[n] * x_3[n]) = (x_1[n] * x_2[n]) * x_3[n]
\]

• shifting
\[
y[n] = x_1[n] * x_2[n] \text{ implies } x_1[n-m] * x_2[n-p] = y[n-m-p]
\]

• for \( x[n] = x_r[n] + jx_i[n] \) and \( h[n] \) real \( y[n] = x[n] * h[n] = (x_r[n] * h[n]) + j(x_i[n] * h[n]) \)

**Zero-State Response**

The above identities and a table of convolutions\(^1\) allow us to determine the zero-state response for a large class of useful inputs. Thus the total solution of a difference equation is

\[
y[n] = y_0[n] + h[n] * x[n]
\]

\(^1\) See Lathi Table 3.1 pg. 291.