Implementation of Newton Interpolant

- Let $d_{ij}$ denote the $(i, j)$-entry in the following table where the indexing begins with $d_{00} = f[x_0]$.

$$
\begin{align*}
  f[x_0] &= f_0 \\
  f[x_1] &= f_1 \\
  f[x_2] &= f_2 \\
  f[x_3] &= f_3 \\
  \vdots & \quad \vdots & \quad \vdots & \quad \vdots
\end{align*}
$$

- The array can be built up columnwise.

$$
  d_{ij} = d_{i,j-1} - d_{i-1,j-1} / (x_i - x_{i-j+1}).
$$

- The diagonal elements are the coefficients of the Newton interpolant.

- It is not necessary to store the entire 2-dimensional table. Suppose the values of $f_1, \ldots, f_n$ have been stored in the array $c_1, \ldots, c_n$ (For convenience of indexing, only $n$ support data are marked in this example.) Then

  ```matlab
  for j=2:1:n
    for i=n:-1:j
      c[i]=(c[i]-c[i-1])/(x[i]-x[i-j+1]);
    end
  end
  ```

  - Entries of the resulting array $c$ are the desirable coefficients.
  - The columns are generated from the bottom up to avoid premature overwriting of values of $c$.
  - The operation counts is $n^2$ additions and $\frac{1}{2}n^2$ divisions.