1. Sec. 6.1 of the textbook by Kincard and Cheney: 13, 15, 16, 22.
5. Sec. 6.1 of the textbook by Kincard and Cheney: 27.

I. Section 6.1

13. Prove that if we take any set of 23 nodes in the interval $[1, 1]$ and interpolate the function $f(x) = \cosh x$ with a polynomial $p$ of degree 22, then the relative error $\|p(x) - f(x)\|/\|f(x)\|$ is no greater than $5 \times 10^6$ on $[-1, 1]$.

Proof:
It is obvious $f(x) = \cosh x = \frac{e^x + e^{-x}}{2} \in C^{23}[-1, 1]$. Let $x_0, x_1, \ldots, x_{22}$ denote 23 nodes in the interval $[1, 1]$ and the interpolating polynomial is $p(x)$, then we can have the error by Theorem 2.

$$f(x) - p(x) = \frac{1}{23!} f^{(23)}(\xi_x) \prod_{i=0}^{22} (x - x_i), \text{ where } \xi_x, x \in [-1, 1].$$

We know that $f^{(23)}(\xi_x) = \sinh(\xi_x) = \frac{e^{\xi_x} - e^{-(\xi_x)}}{2}$ is a monotonous increasing function,

$$|f^{(23)}(\xi_x)| \leq \sinh(1) < \frac{3}{2}, \xi_x \in [-1, 1].$$

And we know $\prod_{i=0}^{22} (x - x_i) < 2^{23}$ as $x \in [-1, 1]$.

Combine all these inequality, we have $|f(x) - p(x)| < \frac{1}{23!} \cdot \frac{3}{2} \cdot 2^{23}$. And $|f(x)| \geq 1$, then

$$\frac{|p(x) - f(x)|}{|f(x)|} < \frac{\frac{1}{23!} \cdot \frac{3}{2} \cdot 2^{23}}{1} = \frac{2^{22} \cdot 3}{23!} = 4.8673 \times 10^{-16} < 5 \times 10^{-16}.$$ 

15. What is the final value of $v$ in the algorithm shown?

```plaintext
v ← c_{i-1}
for j = i to n do
    v ← vx + c_j
end do
```

What is the number of additions and subtractions involved in this algorithm?

Solution:

$v = \left( (c_{i-1}x + c_i)x + c_{i+1}x + \cdots + c_{n-1}x \right) x + c_n$

$= c_{i-1}x^{n-i+1} + c_i x^{n-i} + \cdots + c_{n-1}x + c_n$
In this algorithm, there are \( n - i + 1 \) times of additions and no subtractions.

16. Write an efficient algorithm for evaluating

\[
u = \sum_{i=1}^{n-1} d_j.\]

Solution:
The algorithm

\[
u \leftarrow d_n
\]

\[
\text{for } i = n - 1 \text{ to } 1 \text{ step -1 do}
\]

\[
u \leftarrow (u + 1)d_i
\]

\[
\text{end do}
\]

22. Find the Lagrange and Newton forms of the interpolating polynomial for these data: Write

<table>
<thead>
<tr>
<th>( x )</th>
<th>(-2)</th>
<th>( 0 )</th>
<th>( 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(x) )</td>
<td>( 0 )</td>
<td>( 1 )</td>
<td>(-1)</td>
</tr>
</tbody>
</table>

both polynomials in the form \( a + bx + cx^2 \) in order to verify that they are identical as functions.

Solution:
Lagrange form:

\[
l_0(x) = \frac{x(x-1)}{(-2)(-2-1)} = \frac{x(x-1)}{6},
\]

\[
l_1(x) = \frac{(x+2)(x-1)}{2(-1)} = \frac{(x+2)(x-1)}{2}
\]

\[
l_2(x) = \frac{(x+2)x}{(1+2)1} = \frac{(x+2)x}{3}
\]

\[
p_L(x) = 0l_0(x) + l_1(x) - l_2(x)
\]

\[
= -\frac{(x+2)(x-1)}{2} - \frac{(x+2)x}{3}
\]

\[
= 1 - \frac{7}{6}x - \frac{5}{6}x^2.
\]
Newton:

\[ p_0(x) = c_0 \Rightarrow c_0 = f(x_0) = 0, \]
\[ p_1(x) = c_0 + c_1(x + 2) \Rightarrow c_1 = \frac{1}{2}, \]
\[ p_2(x) = c_0 + c_1(x + 2) + c_2(x + 2)x \Rightarrow c_2 = -\frac{5}{6}. \]
\[ p_N(x) = c_0 + c_1(x + 2) + c_2(x + 2)x \]
\[ = \frac{1}{2}(x + 2) - \frac{5}{6}(x + 2)x \]
\[ = 1 - \frac{7}{6}x - \frac{5}{6}x^2. \]

We can find the Lagrange form and Newton form are the same.

III. Section 6.2

4. Prove that if \( f \) is a polynomial of degree \( k \), then for \( n > k \)
\[ f[x_0, x_1, \ldots, x_n] = 0. \]

Proof:
Suppose \( p(x) \) is the interpolation polynomial of at most degree \( n \) for \( f \), then
\[ p(x_i) = f(x_i). \quad i = 0, 1, \ldots, n. \]
Let \( q(x) = p(x) - f(x) \) be a polynomial of at most degree \( n \). From above, we know that \( q(x) \) at least has \( n + 1 \) roots. Hence
\[ q(x) = 0. \Rightarrow p(x) - f(x) = 0 \Leftrightarrow p(x) = f(x). \]
Which means \( p(x) \) is a polynomial of degree \( k \). By divided difference, we have
\[ p(x) = \sum_{i=0}^{n} f[x_0, x_1, \ldots, x_i] \cdot \prod_{j=0}^{i-1} (x - x_j). \]
Then \( f[x_0, x_1, \ldots, x_n] \) is the coefficient of \( x^n \). Hence, \( f[x_0, x_1, \ldots, x_n] = 0 \) when \( n > k \).

10. Compare the efficiency of the divided difference algorithm to the procedure described in Section 6.1 for computing the coefficients in a Newton interpolating polynomial.

Solution:
Algorithm in Section 6.1:

\[ c_0 \leftarrow y_0 \]
\[ \text{for } k = 1 \text{ to } n \text{ do} \]
\[ d \leftarrow x_k - x_{k-1} \]
\[ u \leftarrow c_{k-1} \]
\[ \text{for } i = k - 2 \text{ to } 0 \text{ step } -1 \text{ do} \]
\[ u \leftarrow u(x_k - x_i) + c_i \]
\[
d d \leftarrow d(x_k - x_i) 
\end{align*}
end do
\[
c_k \leftarrow (y_k - u)/d
\end{align*}
end do

We just need to calculate the number of operations in inner loop. There are 1 addition, 2 subtractions and 2 multiplies. Then the number of operations is:

\[
\begin{align*}
\sum_{k=1}^{n} \sum_{i=k-2}^{0} 5 &= \sum_{k=1}^{n} 5(k-1) = \frac{5n^2}{2} - \frac{5}{2}n \sim \frac{5}{2}n^2.
\end{align*}
\]

Algorithm in section 6.2:

\[
\begin{align*}
\text{for } i = 0 \text{ to } n \text{ do} \\
d_i \leftarrow f_{x_i} \\
\text{end do} \\
\text{for } j = 1 \text{ to } n \text{ do} \\
\text{for } i = j \text{ to } n \text{ do} \\
d_i \leftarrow (d_i - d_{i-1})/(x_i - x_{i-j}) \\
\text{end do} \\
\text{end do}
\end{align*}
\]

We just need to calculate the number of operations in inner loop. There are 2 multiplies and 1 dividings. Then the number of operations is:

\[
\begin{align*}
\sum_{j=1}^{n} \sum_{i=j}^{n} 3 &= \sum_{j=1}^{n} 3(n - j + 1) = \frac{3}{2}n^2 + \frac{3}{2}n \sim \frac{3}{2}n^2.
\end{align*}
\]

Hence, the algorithm in section 6.2 is more efficient than the algorithm in section 6.1.

24. Write the Newton interpolating polynomial by divided differences for these data:

\[
\begin{array}{c|cccc}
  x & 4 & 2 & 0 & 3 \\
  f(x) & 63 & 11 & 7 & 28 \\
\end{array}
\]

Solution:

By divided differences, we have table as following:

\[
\begin{array}{c|cccc}
  4 & 63 & 26 & 6 & 1 \\
  2 & 11 & 2 & 5 \\
  0 & 7 & 7 \\
  3 & 28 \\
\end{array}
\]

Then the Newton interpolating polynomial is:

\[
p(x) = 63 + 26(x - 4) + 6(x - 4)(x - 2) + (x - 4)(x - 2)x.
\]
IV. Section 6.1:

27. If we interpolate the function \( f(x) = e^{x-1} \) with a polynomial \( p \) of degree 12 using 13 nodes in \([-1, 1]\), what is a good upper bound for \( |f(x) - p(x)| \) on \([-1, 1]\)?

Solution:
As \( f(x) = e^{x-1} \) is smooth enough, we can choose Chebyshev points to interpolate \( f(x) \). Therefore, we can get the error formula

\[
|f(x) - p(x)| \leq \frac{1}{2^{12}(12 + 1)!} \max_{||t|| \leq 1} |f^{13}(t)|
\]

\[
= \frac{1}{2^{12} \cdot 13!} \max_{||t|| \leq 1} |e^{t-1}|
\]

\[
= \frac{1}{2^{12} \cdot 13!}.
\]

Hence, the good upper bound is \( \frac{1}{2^{12} \cdot 13!} \approx 3.9207 \times 10^{-14} \).