Answer to Homework Set No. 2

Problem 1

(a). In order to derive it, we expand the function \( f(x) \) at \( x = 0 \):

\[
f(x) = (1 + x)^n = f(0) + x \cdot f'(0) + \frac{x^2}{2!} f''(0) + \cdots + \frac{x^k}{k!} f^{(k)}(0) + \cdots
\]

Using that \( f(0) = 1 \), and

\[
f^{(k)}(x) = n(n-1) \cdots (n-k+1)(1 + x)^{n-k}, \quad f^{(k)}(0) = n(n-1) \cdots (n-k+1),
\]

we get

\[
(1 + x)^n = 1 + nx + \frac{n(n-1)}{2!} x^2 + \cdots + \frac{n(n-1) \cdots (n-k+1)}{k!} x^k + \cdots.
\]

For \( n = 2 \), we have

\[
(1 + x)^2 = 1 + 2x + x^2.
\]

For \( n = 3 \), we have

\[
(1 + x)^3 = 1 + 3x + 3x^2 + x^3.
\]

For \( n = 1/2 \), we get an infinite sum

\[
\sqrt{1 + x} = 1 + \frac{x}{2} - \frac{1}{8} x^2 + \frac{1}{16} x^3 + \cdots.
\]

When \( x = 0.0001 = 10^{-4} \), then \( x^4 = 10^{-16} \). So we only need to take 4 terms in order to get 15 decimal correct. This gives:

\[
\sqrt{1.0001} \approx 1 + 0.5 \cdot 10^{-4} - \frac{1}{8} \cdot 10^{-8} + \frac{1}{16} \cdot 10^{-12} = 1.000004999875006
\]

(b). For \( n = -1 \), we have

\[
(1 + x)^{-1} = 1 - x + \frac{2}{2} x^2 - \frac{3!}{3!} x^3 + \cdots + (-1)^k \frac{k!}{k!} x^k + \cdots
\]

\[
= 1 - x + x^2 - x^3 + \cdots + (-1)^k x^k + \cdots
\]

Set \( x = x^2 \) in the previous formula we get

\[
(1 + x^2)^{-1} = 1 - x^2 + x^4 - x^6 + x^8 + \cdots + (-1)^k x^{2k} + \cdots.
\]
Problem 2

(a). The 4 cardinal functions are:

\[ l_0(x) = \frac{(x - 2)(x - 3)(x - 4)}{(0 - 2)(0 - 3)(0 - 4)} = \frac{1}{24}(x - 2)(x - 3)(x - 4) \]

\[ l_1(x) = \frac{x(x - 3)(x - 4)}{(2 - 0)(2 - 3)(2 - 4)} = \frac{1}{4}x(x - 3)(x - 4) \]

\[ l_2(x) = \frac{x(x - 2)(x - 4)}{(3 - 0)(3 - 2)(3 - 4)} = -\frac{1}{3}x(x - 2)(x - 4) \]

\[ l_3(x) = \frac{x(x - 2)(x - 3)}{(4 - 0)(4 - 2)(4 - 3)} = \frac{1}{8}x(x - 2)(x - 3) \]

The Lagrange form of the polynomial is:

\[ p_3(x) = -\frac{7}{24}(x - 2)(x - 3)(x - 4) + \frac{11}{4}x(x - 3)(x - 4) - \frac{28}{3}x(x - 2)(x - 4) + \frac{63}{8}x(x - 2)(x - 3). \]

If one expands it and simplify, one should get

\[ p_3(x) = x^3 - 2x + 7. \]

(b). The table of divided differences is computed as

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( f[] )</th>
<th>( f[,] )</th>
<th>( f[,][] )</th>
<th>( f[,][,] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>9</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>63</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Then the interpolating polynomial is

\[ p_3(x) = 7 + 2x + 5x(x - 2) + x(x - 2)(x - 3) = x^3 - 2x + 7 \]

Note that this is the answer as part (a).

(c). One can think of the Lagrange forms of these two polynomials. All the cardinal functions are the same, and all the interpolating points are the same except the last point. Therefore, we have

\[ p(x) - q(x) = (61 - 30)l_5(x) \]

where \( l_5(x) \) is the 6-th cardinal function. This gives

\[ q(x) = p(x) - 31 \cdot l_5(x) = x^4 - x^3 + x^2 - x + 1 - 31 \frac{(x + 2)(x + 1)x(x - 1)(x - 2)}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} \]
which is
\[ q(x) = x^4 - x^3 + x^2 - x + 1 - \frac{31}{120}(x + 2)(x + 1)x(x - 1)(x - 2). \]

**Problem 3**

(a). By the error Theorem we get, for \( n = 1, \)
\[ e(x) = \frac{(x - x_0)(x - x_1)}{2} f''(\xi) \leq M \frac{(x - x_0)(x - x_1)}{2}(x - x_0)(x - x_1). \]
Using \((x - x_0)(x - x_1) \leq \frac{(x_1 - x_0)^2}{4}\) (prove it yourself!), we have
\[ e(x) \leq \frac{1}{8}(x_1 - x_0)^2 M. \]

(b). By the error Theorem we have, for \( n = 20, \)
\[ \max_{0 \leq x \leq 2} |(f^{(21)}(x)| = \max_{0 \leq x \leq 2} e^{-x} = 1 \]
and
\[ |(x - x_i)| \leq 2, \]
therefore the error bound is
\[ |e(x)| = |e^{-x} - p_{20}(x)| < \frac{1}{21!} \cdot 2^{21} \approx 4.1 \cdot 10^{-14} \]

**Problem 4**

a. This might be done more efficiently using Taylor expansions. Details on that could be found in the lecture notes, at the end of Chapter 1, where we discussed finite differences approximation for derivatives.

Assuming now we use the hint. Interpolating polynomial \( p_2(t) \) for the data
\[
\begin{array}{c|c|c|c}
 x & x + h & x + 2h \\
 f(x) & f(x + h) & f(x + 2h) \\
\end{array}
\]
is given by (using Lagrange form)
\[ p_2(t) = f(x)\frac{(t - x - h)(t - x - 2h)}{2h^2} - f(x + h)\frac{(t - x)(t - x - 2h)}{h^2} + f(x + 2h)\frac{(t - x)(t - x - h)}{2h^2}, \]
Note that \( x \) is used in the data set, so we use \( t \) as the variable for the function \( p_2. \)
We find an approximation to $f'(x)$ by deriving the interpolating polynomial and then compute the derivatives in $x$,

$$p'_2(t) \big|_{t=x} = \frac{-3f(x) + 4f(x + h) - f(x + 2h)}{2h} \approx f'(x)$$

Now we compute the error, using the error Theorem

$$f(t) - p_2(t) = \frac{1}{3!} f'''(\xi(t)) \cdot w(t), \quad w(t) = \prod_{i=0}^{2} (t - t_i)$$

with $t_0 = x, t_1 = x + h$ and $t_2 = x + 2h$, and with $\xi(t)$ in $(x, x + 2h)$. Then,

$$f'(t) - p'_2(t) = \frac{1}{3} \frac{df'''(\xi(t))}{dt} \cdot w(t) + \frac{1}{3} f'''(\xi(t)) \cdot \frac{dw(t)}{dt},$$

by the chain rule. But $w(t) = 0$ for $t = x$ while $\frac{dw(t)}{dt} \big|_{t=x} = 2h^2$. Set in $t = x$ we get

$$f'(x) - p'_2(x) = \frac{1}{3} f'''(\xi(x)) 2h^2 = \frac{1}{3} f'''(\xi(x)) h^2$$

b. When $f(x) = \tan(x)$, we have

$$f'(x) = \frac{1}{\cos^2(x)}, \quad f'(1) = \frac{1}{\cos^2(1)} \approx 3.4255188.$$

The datas are given in the next table. Note that we use $f'(x) = 1/\cos^2(x)$ to compute the error bound.

<table>
<thead>
<tr>
<th>h</th>
<th>Num. approx.</th>
<th>measured error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>3.0733192</td>
<td>0.3521997</td>
</tr>
<tr>
<td>0.01</td>
<td>3.4235</td>
<td>0.0019933</td>
</tr>
<tr>
<td>0.001</td>
<td>3.4255</td>
<td>0.0000190</td>
</tr>
</tbody>
</table>

We see that when $h$ reduces by 1/10, the error reduces by 1/100. This means that the error is of order $h^2$, which confirms the formula in error Theorem.

c. In Matlab:

% Make a sequence of interval length, h = 0.1, 0.05, 0.025, ....
h=0.1*2.^(0:-1:-25);
% Find the approximation to f'(x) with various h
app=(4*tan(1.0+h)-3*tan(1.0)-tan(1.0+2*h))./(2*h);
err=app-1./cos(1.0)^2;
loglog(h,abs(err))
grid
xlabel('h')
ylabel('Error')
Figure 1: Error in computing first derivatives.

The plot of the error is given in Figure 1.
We see that we get the minimum error by using \( h \approx 10^{-6} \). For \( h \)'s even smaller than this, we lose many significant digits in the subtraction, therefore the roundoff error (machine error) dominates.

Problem 5: Divided difference in Matlab.

Part (A): There is not a unique answer to the codes.
The file `divdiff.m` may look like this:

```matlab
function a = divdiff(x,y);
% compute the table of divided differences
n = length(x); % number of interpolating points
a = zeros(n); % make an nxn matrix with all 0 elements
% read in y values into the first column
for i=1:n
    a(i,1) = y(i);
end
% compute the rest of the table
for j=2:n
    for i=1:n-j+1
        a(i,j) = (a(i+1,j-1)-a(i,j-1))/(x(i+j-1)-x(i));
    end
end
```
And the file `polyvalue.m` could be:

```matlab
function v = polyvalue(a,x,t)
% compute the value of an interpolating polynomial
n = length(x);
v = a(1,n);
for i=n-1:-1:1
    v = v.*(t-x(i))+a(1,i);
end
```

**Parts (B) and (C):** For uniformly distributed nodes, you could do the following in Matlab:

```matlab
>> n = 20;
>> x = [-5:10/n:5];
>> y = 1./(x.^2+1);
>> a = divdiff(x,y);
>> t = [-5:0.005:5];
>> v = polyvalue(a,x,t);
>> plot(t,v,t,1./(t.^2+1),'r')
>> grid
>> title('21 uniform nodes')
```

The Chebyshev nodes that contains the boundary points are $x_i = 5 \cos(i \pi / n)$. You can do:

```matlab
>> i = [0:n];
>> x = 5*cos(i*pi/n);
```

with the rest the same as for uniform nodes.

For Chebyshev nodes that does not include the boundary points, they are $x_i = 5 \cos((2i+1)\pi/(2(n+1)))$. This is actually slightly better than the one that contains the boundary nodes. You can do in Matlab:

```matlab
>> i = [0:n];
>> x = 5*cos((2*i+1)*pi/(2*(n+1)));
```

with the rest the same as for uniform nodes.

The plots are given below. We see that Chebyshev nodes give evenly spread error, while the uniform nodes give catastrophic results near the end points of the interval.
Figure 2: With 21 uniformly distributed nodes, interpolation and error

Figure 3: Chebyshev method with nodes on the boundary, interpolation and error
Figure 4: Chebyshev method without nodes on the boundary, interpolation and error