CS/Math 3414 Assignment 3
Solution Sketches

---

1. If we take $p(x) = ax^2 + bx + c$, it is clear that $c = 0$ and $b = 1 - a$. Further, since $p'(\alpha) = 2a\alpha + (1 - a) = 2$, this gives us $a = 1/(2\alpha - 1)$. This means we can represent the desired polynomial as:

$$\begin{aligned} p(x) &= \frac{1}{2\alpha - 1}x^2 + (1 - \frac{1}{2\alpha - 1})x \\ &= \frac{x^2 + 2\alpha x - 2x}{2\alpha - 1} \end{aligned}$$

Since the book requires that we give the polynomial in terms of α, the above is the right format for the answer. It is easy to verify that

$$p'(x) = \frac{2x + (2\alpha - 2)}{2\alpha - 1}$$

which depends on both x and α. When $x = \alpha$,

$$\begin{aligned} p'(\alpha) &= \frac{2\alpha + (2\alpha - 2)}{2\alpha - 1} \\ &= \frac{4\alpha - 2}{2\alpha - 1} \\ &= 2, \text{whenever } \alpha \neq 1/2 \end{aligned}$$

Hence, the α_0 that the question is referring to is $1/2$. Intuitively, $p(x)$ defines a family of polynomials, all of which have a slope of 2 at $x = \alpha$, except at 0.5. To gain more understanding into this problem, you can experiment with the following MATLAB code:

```
clf;
clear;
hold on;
vals = [0,0.1,0.2,0.3,0.4,0.6,0.7,0.8,0.9,1.0];
for alpha = vals
        % calculate and store the coefficients
        % of the polynomial of degree <= 2
        coeffs(1) = 1/(2*alpha - 1);
        coeffs(2) = ((2*alpha - 2)/(2*alpha - 1));
        coeffs(3) = 0;
        x = 0:0.01:1;
        y = polyval(coeffs,x);
        plot (x,y);
end
```

As can be seen from the plot, every value of α leads to a polynomial and if you measure the slope of the function at the specific value of α (for each of the curves), it is 2.

2. The divided difference table for this problem is given by:

2	1.5713			
		0.0006		
3	1.5719		0.00011667	
		0.00095		0
5	1.5738		0.00011667	
		0.0013		
6	1.5751			

So, using a third-degree polynomial should give you no advantage to a second-degree polynomial. The estimate is $f(4) = 1.5727$.

3. Computing the divided difference table using the given information is trivial. We give it below. To interpolate for $\log(1.2)$ and $\log(2.4)$ you have to be careful about the choice of coefficients and the 'centers' around which you express the Newton form of the polynomial. $\log(1.2)$ can be evaluated by using the coefficients from the leading diagonal, whereas $\log(2.4)$ is better evaluated by using coefficients from the next-to-leading or next-to-next-to-leading diagonal (why?).

1	0.0					
		0.35218				
1.5	0.17609		-0.10230			
		0.24988		0.0265533		
2	0.30103		-0.0491933		-0.0064081	
		0.17609		0.0105333		0.001412
3	0.47713		-0.0281266		-0.002172	
		0.13390		0.0051033		
3.5	0.54407		-0.01792			
		0.11598				
4	0.60206					

Using the leading diagonal, $\log(1.2) = 0.077848$ and using the next diagonal, $\log(2.4) = 0.38099$ (these are what the book records as answers).

4. This was worked out in class (on the board) and is also introduced in Section 4.2 of your book (as Rolle's function). The bottom line is that you should get a really poor approximation to $f(x)$ using a polynomial of *any* degree with equally spaced nodes in $[-5, 5]$. Your graphs should show the wide discrepancy between the interpolating polynomial and $f(x)$, with 41 nodes. This is most visible near the ends of the interval $[-5, 5]$.

5. This problem should lead to the conclusion that Chebyshev nodes are better. Using equally spaced nodes, you should get a good approximation with five nodes (for a fourth-degree polynomial). This is the point at which one of the equally spaced nodes falls in the region where the curve changes character (notice that it is not linear, but piecewise linear). After this point, by adding more nodes, the performance of the fit should actually become worse. With Chebyshev nodes, on the other hand, you should get a fit that monotonically improves with the increase in the degree of the interpolating polynomial.