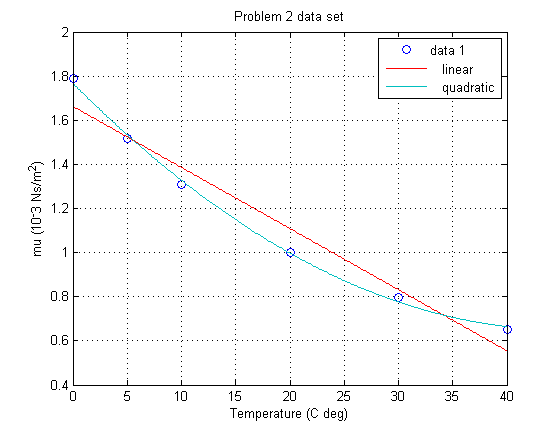
Problem 2) b.

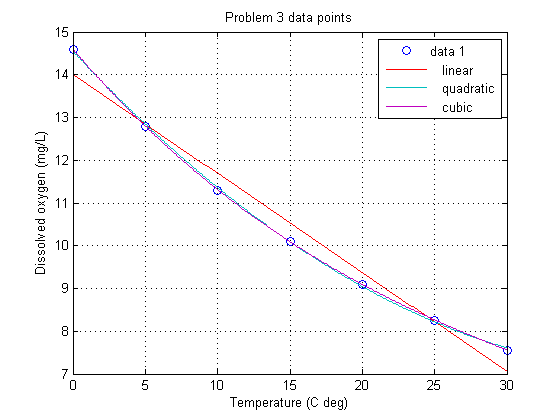


Problem 3) a.

Linear) c = -0.2315T + 14.0025

Quad) c = 0.0044T2 - 0.3634T + 14.5519

Cubic) c = -0.0001T3 + 0.0073T2 - 0.3956T + 14.6002



b.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Total Error | Standard Error | Average Error | Relative Error |
| Linear Fit | 1.0284 | 1.0141 | 0.1449 | 0.0355 |
| Quadratic Fit | 0.0142 | 0.1193 | 0.0170 | 0.0042 |
| Cubic Fit | 2.0714e-004 | 0.0144 | 0.0021 | 5.0424e-004 |

c. It looks like the cubic polynomial fit the best to the data. This makes since because the data is not quite linear and has a changing slope; this causes the 2nd order factor to be affect which can only be characterized by a cubic fit.

d. Yes, it does matter which error estimator people look at to decide which the best method is. In my opinion, the best error estimator would be the total error estimator. This is because it would find the total error in the end result.

fits.m

t = [0 5 10 15 20 25 30];

c = [14.6 12.8 11.3 10.1 9.09 8.26 7.56];

linear = polyfit(t,c,1)

quad = polyfit(t,c,2)

cubic = polyfit(t,c,3)

totalerrLin = sum((polyval(linear,t) - c).^2)

totalerrQuad = sum((polyval(quad,t) - c).^2)

totalerrCubic = sum((polyval(cubic,t) - c).^2)

standerrLin = sqrt(sum((polyval(linear,t) - c).^2))

standerrQuad = sqrt(sum((polyval(quad,t) - c).^2))

standerrCubic = sqrt(sum((polyval(cubic,t) - c).^2))

avgerrLin = standerrLin/length(t)

avgerrQuad = standerrQuad/length(t)

avgerrCubic = standerrCubic/length(t)

relativeerrLin = standerrLin/sqrt(sum(c.^2))

relativeerrQuad = standerrQuad/sqrt(sum(c.^2))

relativeerrCubic = standerrCubic/sqrt(sum(c.^2))