Zachary Job

4/5/15

P1 - Interpolation

1

Natural Cubic Spline)

y1 = (58 + 1.21 .* (xrange1 - (1)) + 0 .* (xrange1 - (1)).^2 + -0.20996 .* (xrange1 - (1)).^3)

y2 = (59 + 0.58007 .* (xrange2 - (2)) + -0.62989 .* (xrange2 - (2)).^2 + 0.049823 .* (xrange2 - (2)).^3)

y3 = (59 + -0.53025 .* (xrange3 - (3)) + -0.48042 .* (xrange3 - (3)).^2 + 0.010673 .* (xrange3 - (3)).^3)

y4 = (58 + -1.4591 .* (xrange4 - (4)) + -0.44841 .* (xrange4 - (4)).^2 + 0.90749 .* (xrange4 - (4)).^3)

y5 = (57 + 0.36657 .* (xrange5 - (5)) + 2.2741 .* (xrange5 - (5)).^2 + -1.6406 .* (xrange5 - (5)).^3)

y6 = (58 + -0.0071859 .* (xrange6 - (6)) + -2.6478 .* (xrange6 - (6)).^2 + 1.655 .* (xrange6 - (6)).^3)

y7 = (57 + -0.33782 .* (xrange7 - (7)) + 2.3172 .* (xrange7 - (7)).^2 + -0.97935 .* (xrange7 - (7)).^3)

y8 = (58 + 1.3585 .* (xrange8 - (8)) + -0.62087 .* (xrange8 - (8)).^2 + 1.2624 .* (xrange8 - (8)).^3)

y9 = (60 + 3.9039 .* (xrange9 - (9)) + 3.1663 .* (xrange9 - (9)).^2 + -3.0702 .* (xrange9 - (9)).^3)

y10 = (64 + 1.0259 .* (xrange10 - (10)) + -6.0443 .* (xrange10 - (10)).^2 + 4.0185 .* (xrange10 - (10)).^3)

y11 = (63 + 0.99261 .* (xrange11 - (11)) + 6.0111 .* (xrange11 - (11)).^2 + -2.0037 .* (xrange11 - (11)).^3)

Newton Polynomial Interpolation)

y = (58

+ 1 .* (x - (1))

+ -0.5 .* (x - (1)) .* (x - (2))

+ 0 .* (x - (1)) .* (x - (2)) .* (x - (3))

+ 0.041667 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4))

+ 6.9389e-18 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4)) .* (x - (5))

+ -0.0097222 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4)) .* (x - (5)) .* (x - (6))

+ 0.0055556 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4)) .* (x - (5)) .* (x - (6)) .* (x - (7))

+ -0.0018849 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4)) .* (x - (5)) .* (x - (6)) .* (x - (7)) .* (x - (8))

+ 0.00047123 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4)) .* (x - (5)) .* (x - (6)) .* (x - (7)) .* (x - (8)) .* (x - (9))

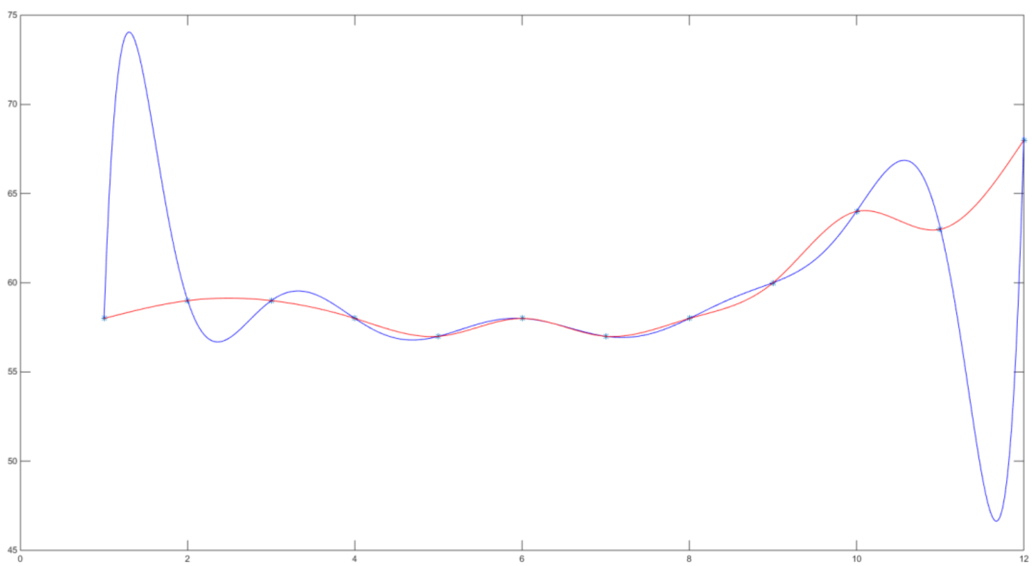
+ -9.6451e-05 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4)) .* (x - (5)) .* (x - (6)) .* (x - (7)) .* (x - (8)) .* (x - (9)) .* (x - (10))

+ 1.7612e-05 .* (x - (1)) .* (x - (2)) .* (x - (3)) .* (x - (4)) .* (x - (5)) .* (x - (6)) .* (x - (7)) .* (x - (8)) .* (x - (9)) .* (x - (10)) .* (x - (11))

)

2

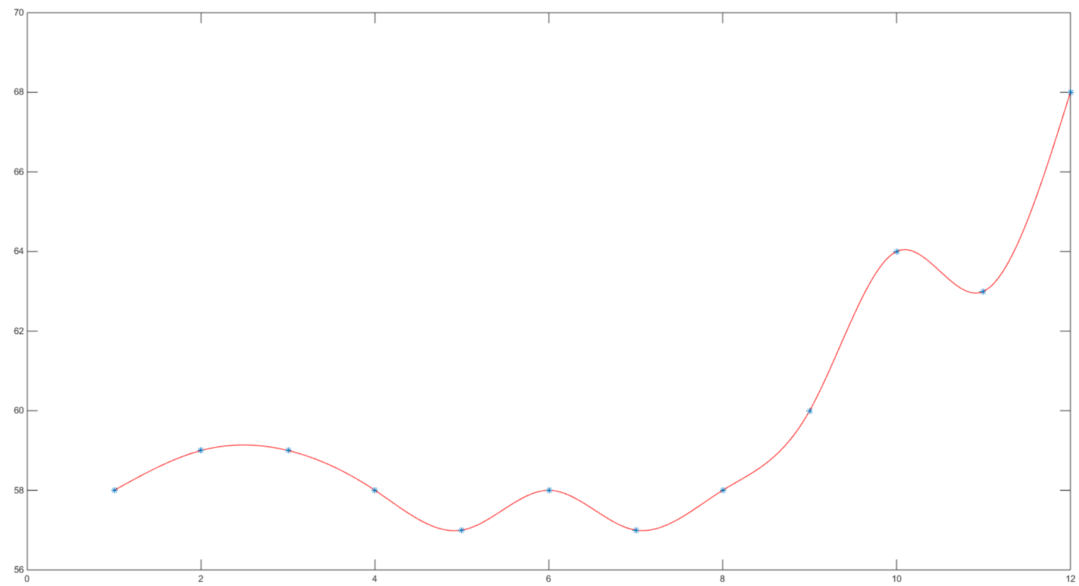
Comparison)



While the spline appears to cling rather well to the points, the newton interpolation oscillates heavily making the approximation at any point other than those defined likely to be inaccurate. However near the middle the function approximates the points decently, it is mostly the ends that begin to uncontrollably oscillate. For example, if I pick point 1.5, instead of receiving what could be guessed to be 58.5, I receive something closer to 74. Near the middle, if I pick 6.5, I expect around 57.5. I receive a similar result.

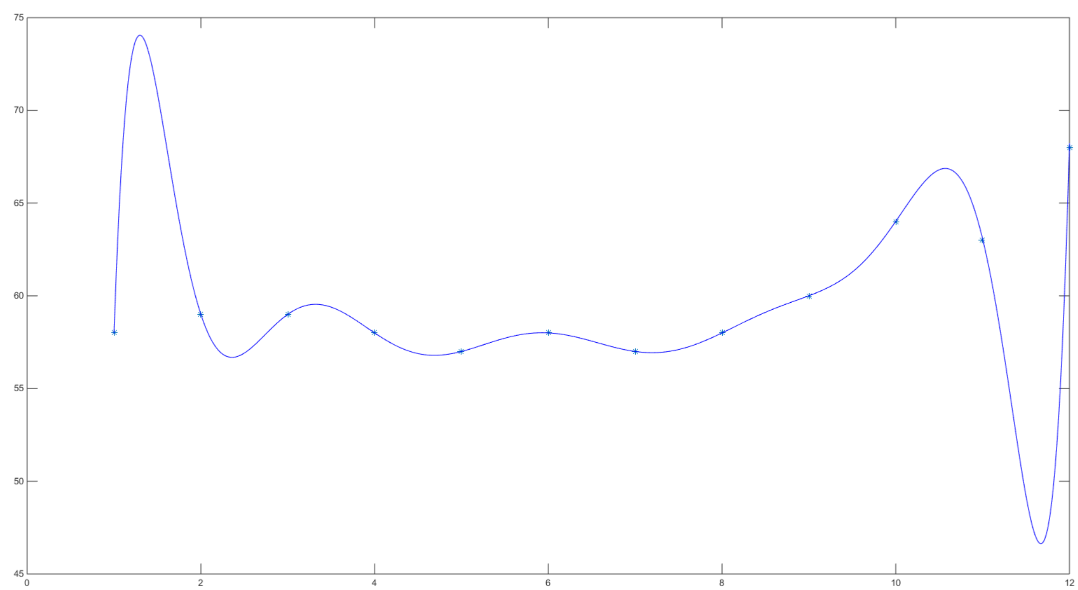
3

Natural Cubic Spline)



Here, the function gradually connects the points. Seeing the temperature can be affected by a multitude of factors causing randomness (possible oscillation between points), this function well approximates the given points. Pairs should not typically exhibit very linear behavior given this randomness as exhibited. This is a great approximation.

Newton Interpolating Polynomial)



Even including most of the decent approximations near the middle of the data, this approximation is not optimal. In order for such a large function to work, the function resembles an oscillation manipulated by contributing parts to best fit points at a location. However, with larger sets the power of x being the input to the n size increases greatly. This will cause larger and larger oscillations that deviate near the function borders.

Comparison)

The choice is clear. Since the spline is not limited by being bound to one function, smaller more accurate functions best represent the y action of the given data.

4

The script outputs the average over a range of 12 to be 59.5925. This seems like a reasonable approximation as the average temperature is 59.9167.