Lagrange Interpolation

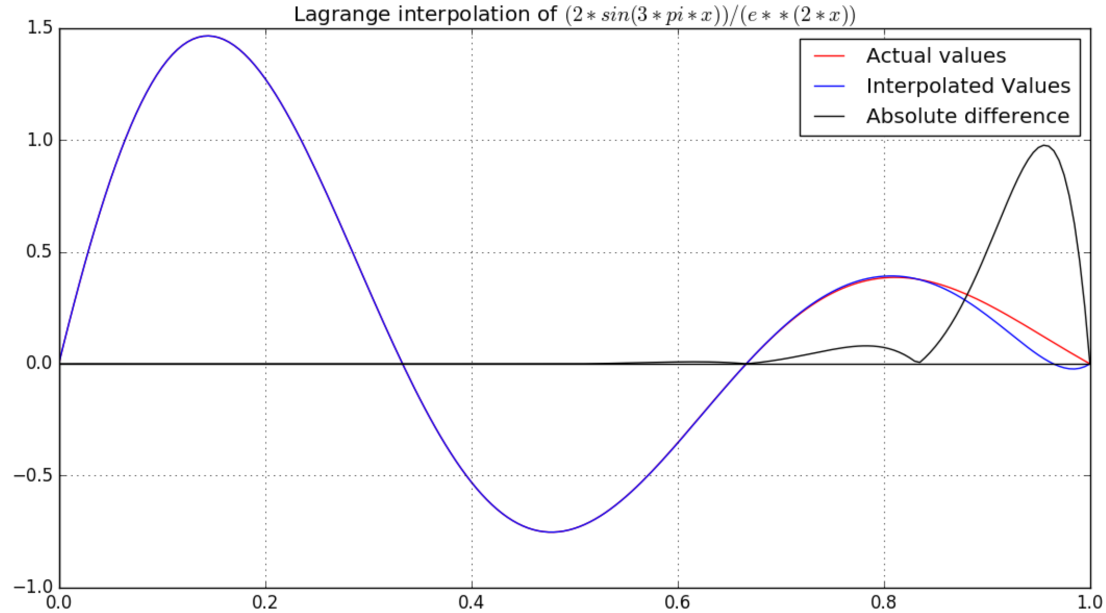

Figure 1. Approximating *g(x)* with a Lagrange interpolating polynomial using 11 nodes with different spacing between them. The difference curve, scaled by 10, shows that the error in the approximation increases for successive nodes that are spread farther apart.

The Lagrange interpolant, for the most part, does a good job. We can see that it fails due to its significant error when $x > 0.8$. In this portion of the graph, *g(x)* remains concave down, but the interpolant changes concavity from concave down to concave up. The difference curve is scaled by a factor of 10, which allows us to see the point at which the error begins. The curve is flat at *0* until *x* is close to *0.6* (a hump in the curve can be seen starting from $x = 0.5$). If we look at the differences between the nodes, we can find a clue as to what is going on.

Successive nodes	**Difference**
1/12 – 1/24	.0416
1/6 – 1/12	.0833
1/4 – 1/6	.0833
1/3 – 1/4	.0833
5/12 – 1/3	.0833
1/2 – 5/12	.0833
2/3 – 1/2	.1666
5/6 – 2/3	.1666
1 – 5/6	.1666

Table 1. Spacing between successive nodes.

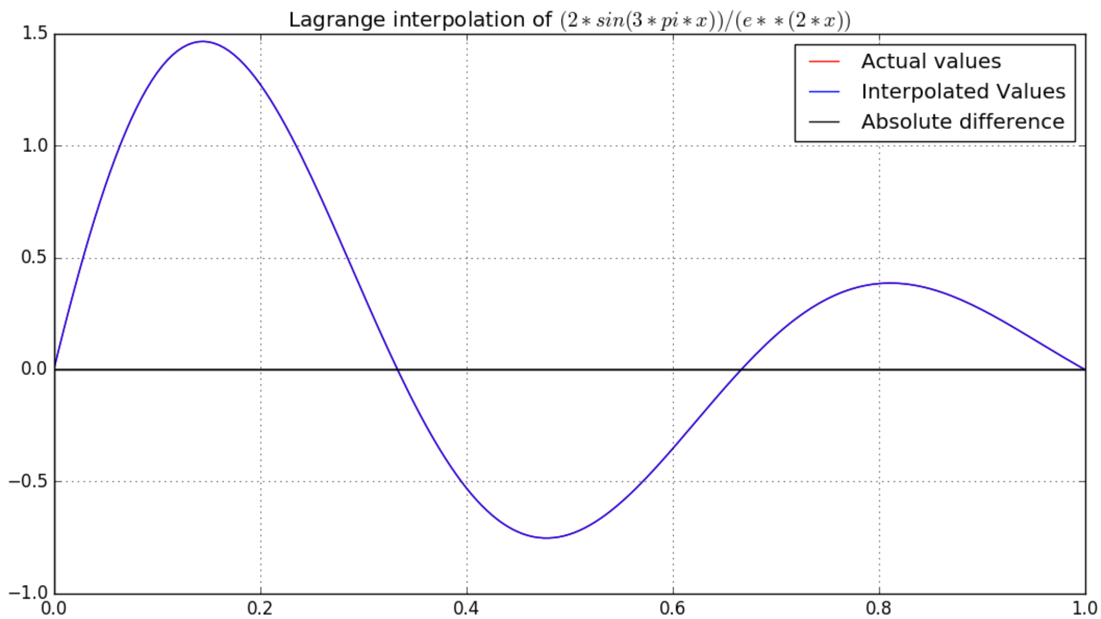

Figure 2. Approximating *g(x)* with a Lagrange interpolating polynomial using nodes with equal spacing of *0.0833*. The difference curve, scaled by a factor of 10, is not visible. The error that appears in figure 1 has been corrected by using more nodes that are closer together.

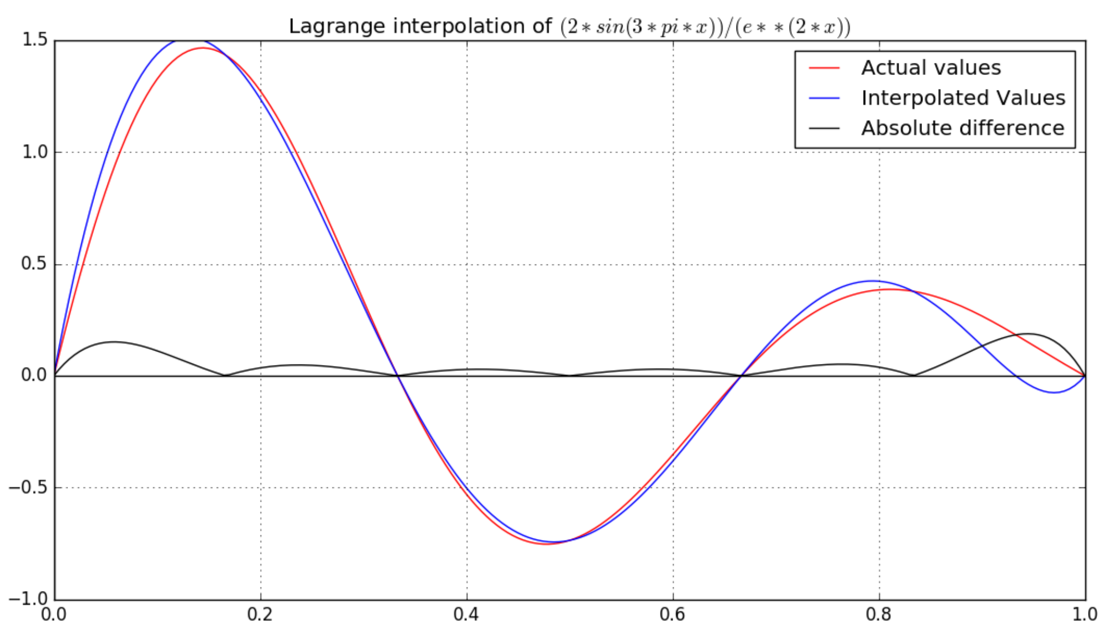

Figure 3. Approximating *g(x)* with a Lagrange interpolating polynomial using nodes with equal spacing of *0.1666*. This time, the difference curve (which has not been scaled) is not necessary; we can immediately see large differences between the interpolant and the exact curve. Using fewer nodes that are spaced far apart is disasterous for Lagrange interpolation.

The spacing between nodes increases to *0.1666* when $x = 0.5$, which is the point at which the error curve deviates from *0*. From this, we can hypothesize that *0.1666* is too large of a distance to separate successive nodes. Figures 2 and 3 agree with this hypothesis. In figure 2, successive nodes are separated by a distance of *0.0833*. The Lagrange interpolant approximates $g(x)$ wonderfully, just as in the first half of figure 1. In figure 3, successive nodes are separated by a distance of *0.1666*. We can clearly see that the Lagrange interpolant is not at all a good approximation to $g(x)$. *0.1666* is simply too much separation to get accurate approximations. This is a great example that shows how sensitive Lagrange interpolation is to the spacing of nodes. Using nodes that are too spaced out will produce abysmal approximations.

Hermite Interpolation

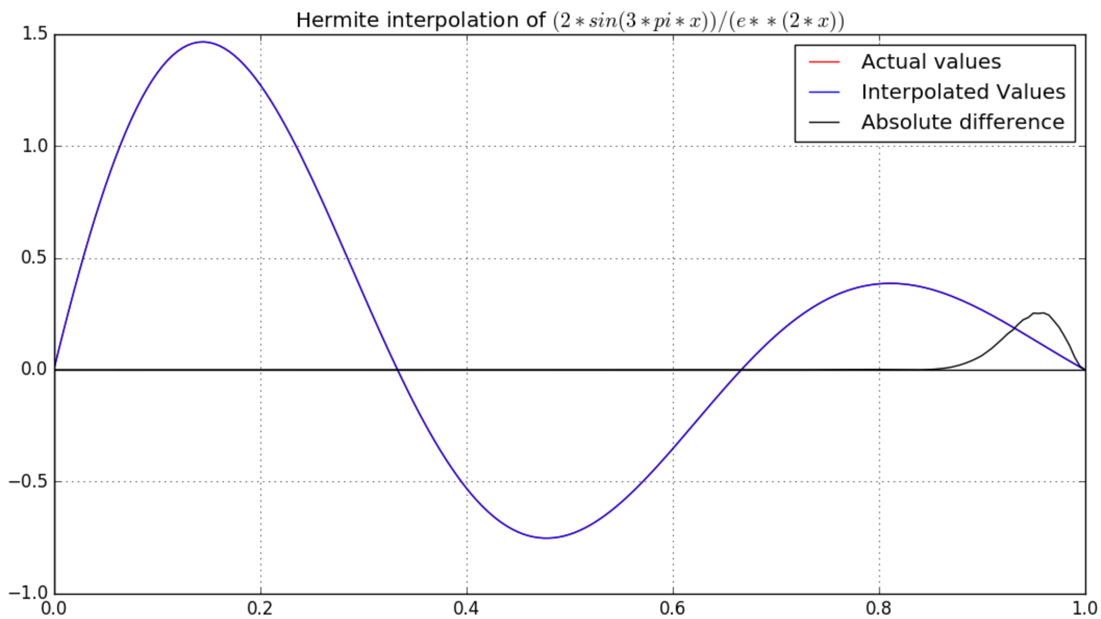

Figure 4. Approximating $g(x)$ with an Hermite interpolating polynomial using the same nodes as in figure 1. The difference curve, scaled by *10e7*, shows that there is almost no error in the approximations.

The Hermite interpolant is extremely good. Out of the four interpolants, this approximates $g(x)$ the best. We had to scale the difference curve by *10e7* to see anything of significance. Like the Lagrange interpolant, the Hermite one has a significant error when $x > 0.8$, relative to the rest of the curve. Unlike the Lagrange, this does not seem to be due to spacing between successive nodes. Regardless, Hermite interpolation produces stunningly accurate approximations to $g(x)$ everywhere on the interval.

Linear Spline Interpolation

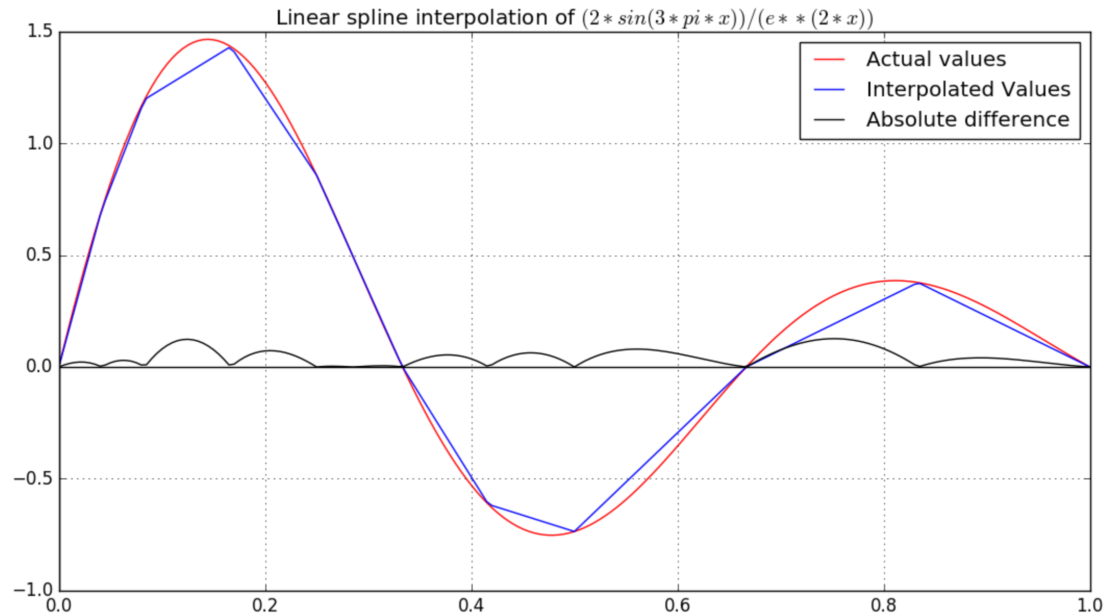

Figure 5. Approximating *g(x)* using a linear spline interpolant and the same nodes as in figure 1. Unsurpsingly, the approximations are not good where *g(x)* changes slope quickly, but not bad when *g(x)*'s slope is close to linear. The difference curve is not scaled.

As expected, the linear spline interpolant was the worst. The difference curve was not scaled. The approximations veer off the true value primarily during intervals where the slope changes quickly.

As figure 5 shows, the difference curve spikes in the intervals that contain local extrema. However, the interpolant does a fine job during the intervals where the slope is close to constant. This shows that linear spline interpolation is not a good choice for curves with slopes that change frequently and rapidly, but it is a good choice for, not surprisingly, linear or close to linear curves.

Cubic Spline Interpolation

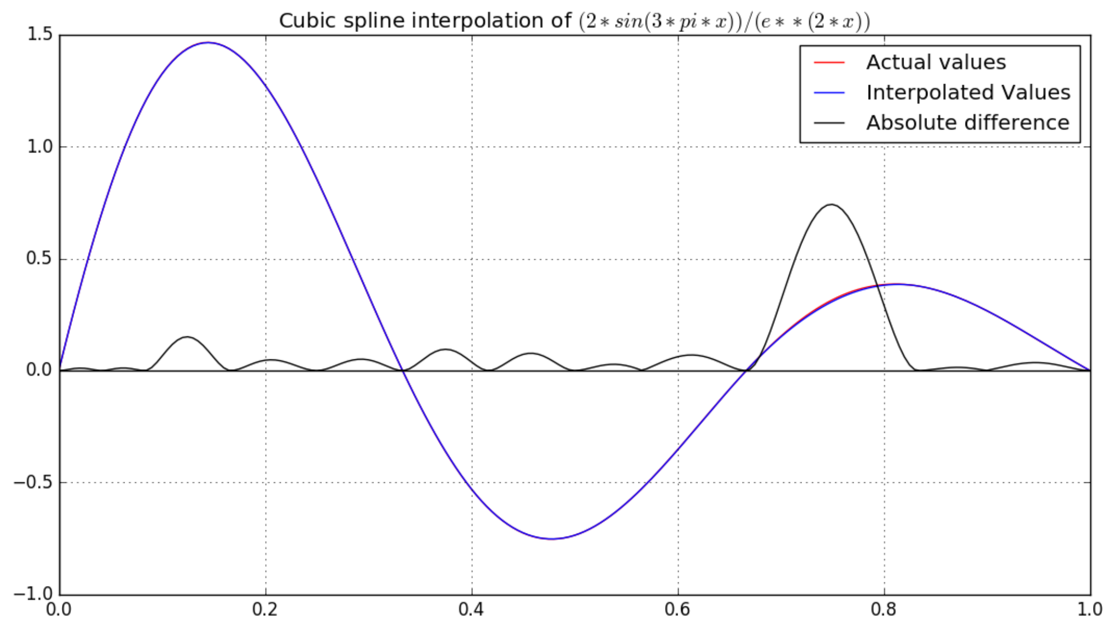

Figure 6. Approximating *g(x)* using a cubic spline interpolant and the same nodes as in figure 1. The difference curve has been scaled by 100. It shows some small errors within every interval; however, the errors do not propogate to the other intervals, as was seen in figure 1.

Cubic spline interpolation does a fairly goodjob at approximating *g(x)*. It does a better job than the Lagrange interpolant when $x > 0.5$, but is worse when $x < 0.5$. The difference curve is scaled by 100, so the errors of the cubic spline interpolant when $x < 0.5$ are by no means large, however, in the same interval, the Lagrange interpolant's errors are much closer to zero. In the interval from $x > 0.5$, the cubic spline interpolant is significantly better than the Lagrange, however, it is significantly worse than the Hermite. Curiously, we can see that the difference curve is zero at regular intervals, except at the large spike when x is between *0.7* and *0.8*. Judging by the curve before *0.7*, we would expect the it to be zero precisely when it spikes. We would like to attribute this anomoly to the spacing of nodes, but this behavior is not seen in other intervals of equal length. This can actually be seen as a strength of cubic spline interpolation—local errors do not affect global approximations. That is, inaccurate results in one interval do not affect the accuracy in another. This keeps the errors low when $x > 0.5$, while in Lagrange interpolation, the errors mostly kept getting worse.