Homework8

Wednesday, October 25, 2023 3:21 PM

Tony Samour
APPM 4600

1.) 1. Consider the task of interpolating the function $f(x) = \frac{1}{1+x^2}$ on the interval $[-5, 5]$. Using equispaced nodes with $n = 5, 10, 15$ and 20, interpolate the function using the methods below:

(a) Lagrange interpolation.

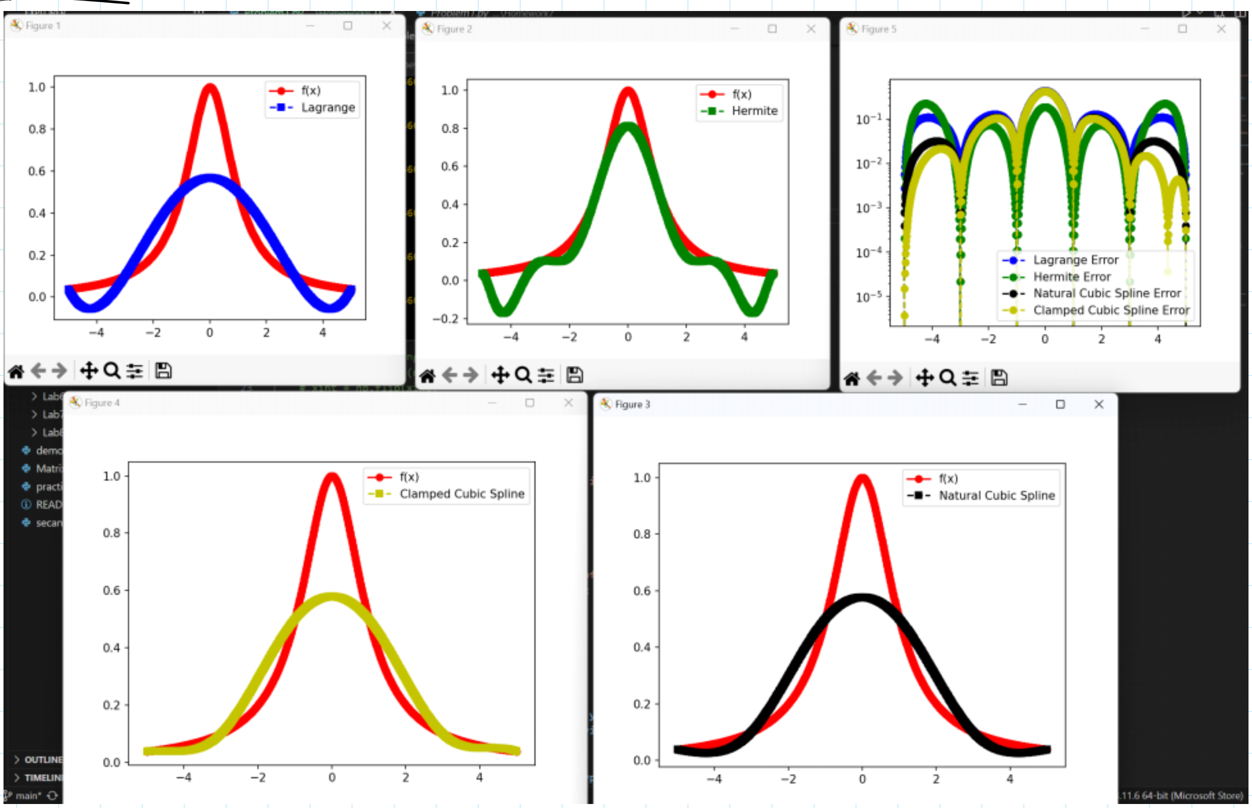
(b) Hermite interpolation.

(c) Natural Cubic spline.

(d) Clamped Cubic spline.

Which method performs best? Do you have an intiution why?

N=5



N=10

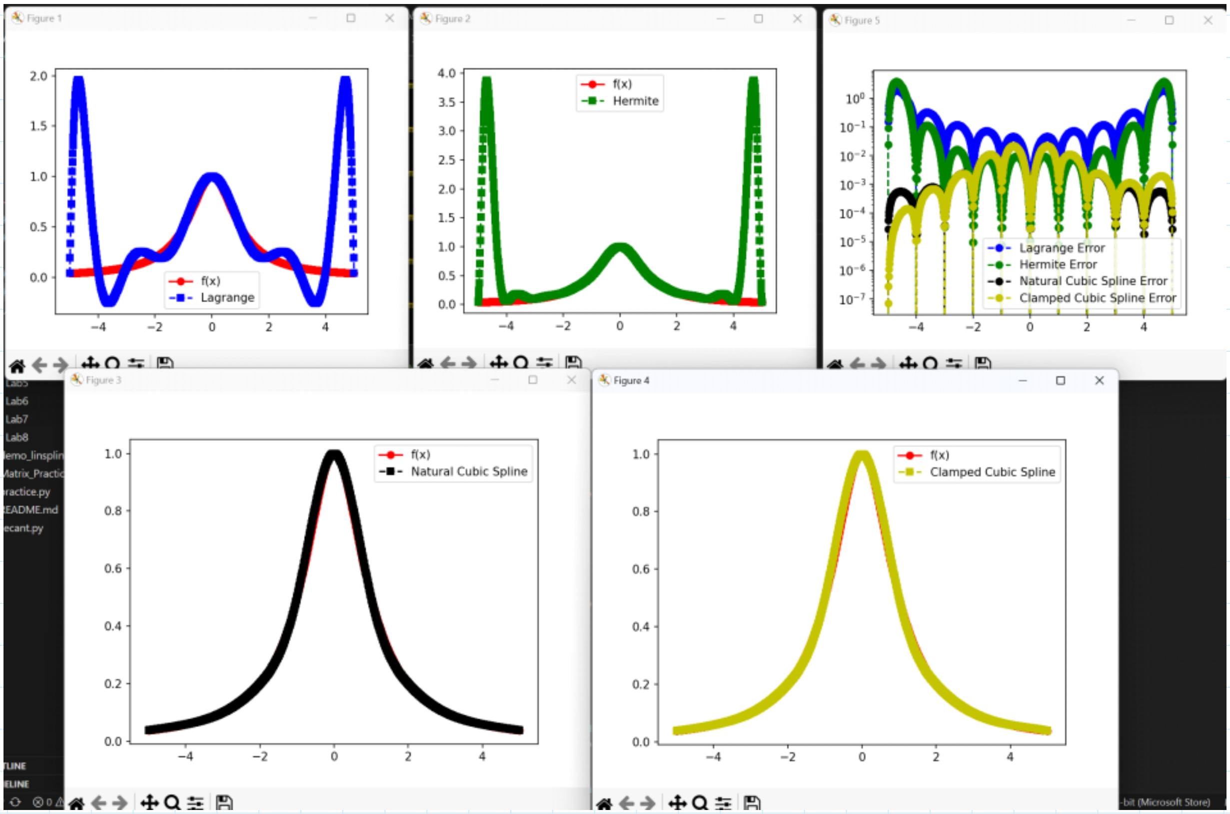


Figure 1
f(x)
Lagrange
Figure 2
f(x)
Hermite
Figure 5
Lagrange Error
Hermite Error
Natural Cubic Spline Error
Clamped Cubic Spline Error
Figure 3
f(x)
Natural Cubic Spline
Figure 4
f(x)
Clamped Cubic Spline

N=15

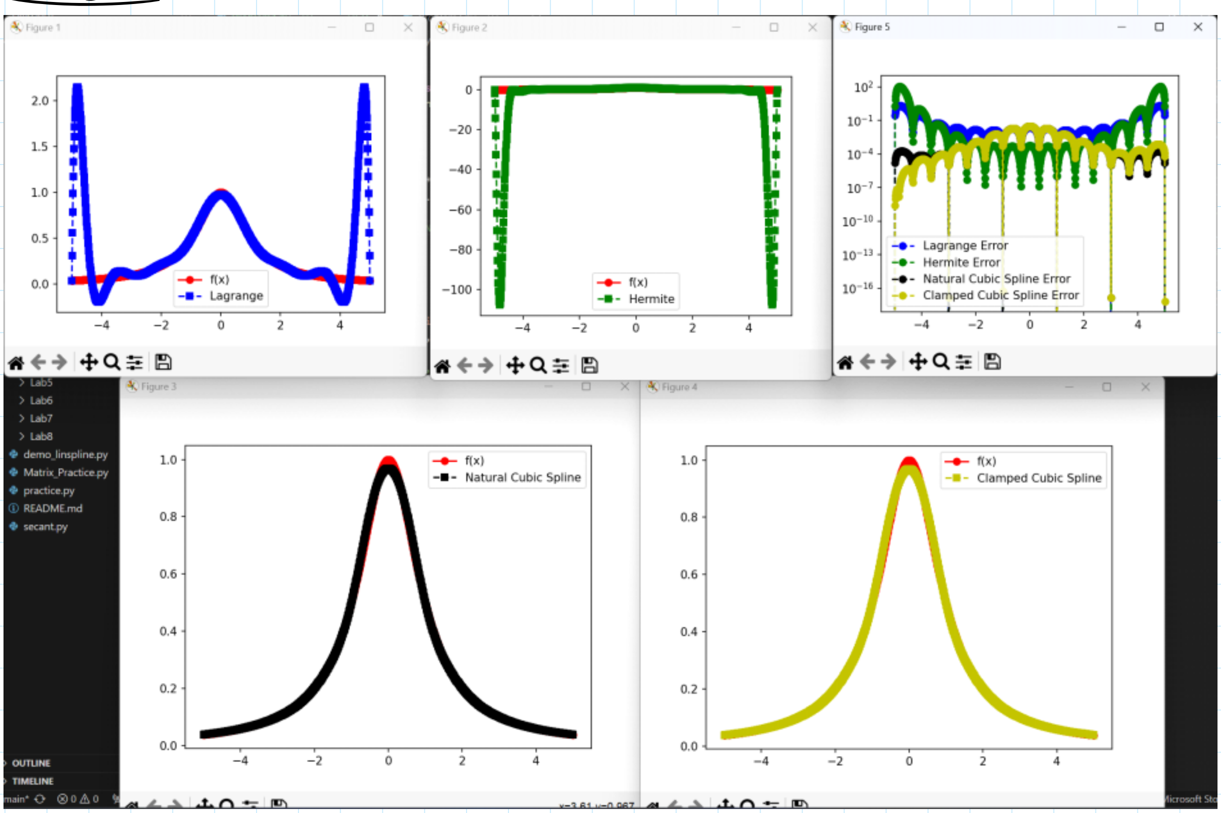


Figure 1
f(x)
Lagrange
Figure 2
f(x)
Hermite
Figure 5
Lagrange Error
Hermite Error
Natural Cubic Spline Error
Clamped Cubic Spline Error
Figure 3
f(x)
Natural Cubic Spline
Figure 4
f(x)
Clamped Cubic Spline

N=20

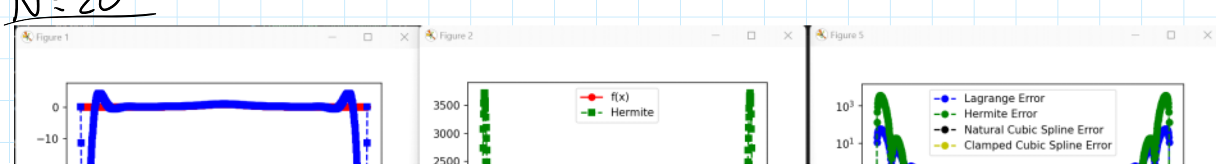
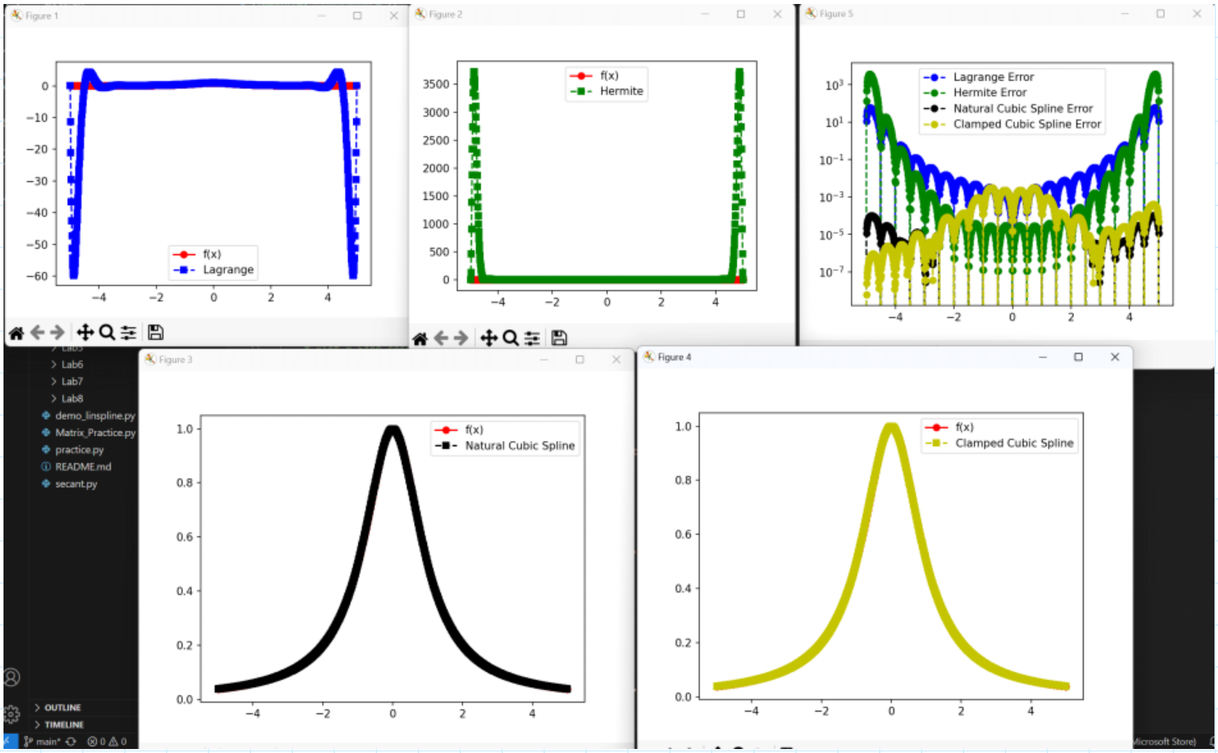


Figure 1
Figure 2
f(x)
Hermite
Figure 5
Lagrange Error
Hermite Error
Natural Cubic Spline Error
Clamped Cubic Spline Error



As we noticed in the last homework, the Lagrange and Hermite approximations experience high error near the endpoints. Also, it appears that for small N values, the Hermite approximation most closely matches the shape of f(x).

2.) 2. Repeat the experiment from the previous problem but with Chebychev nodes. How does this impact the performance of the different interpolation techniques?

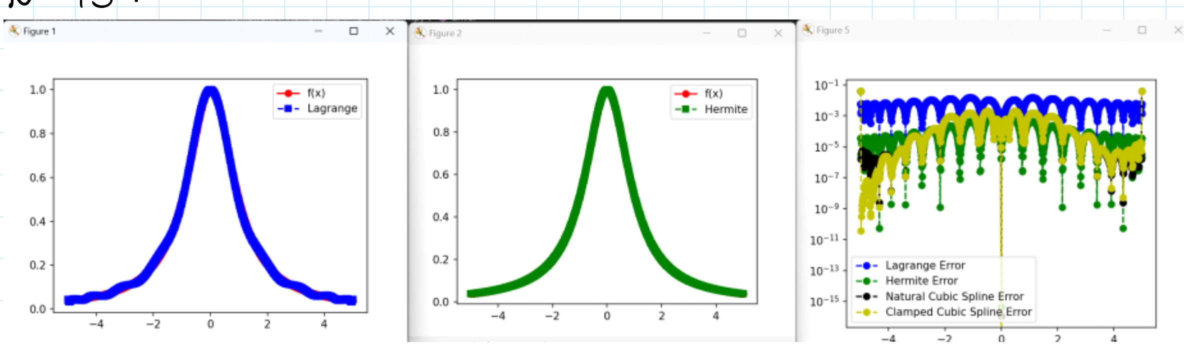
On an interval [a,b], the Chebyshev nodes are given by:

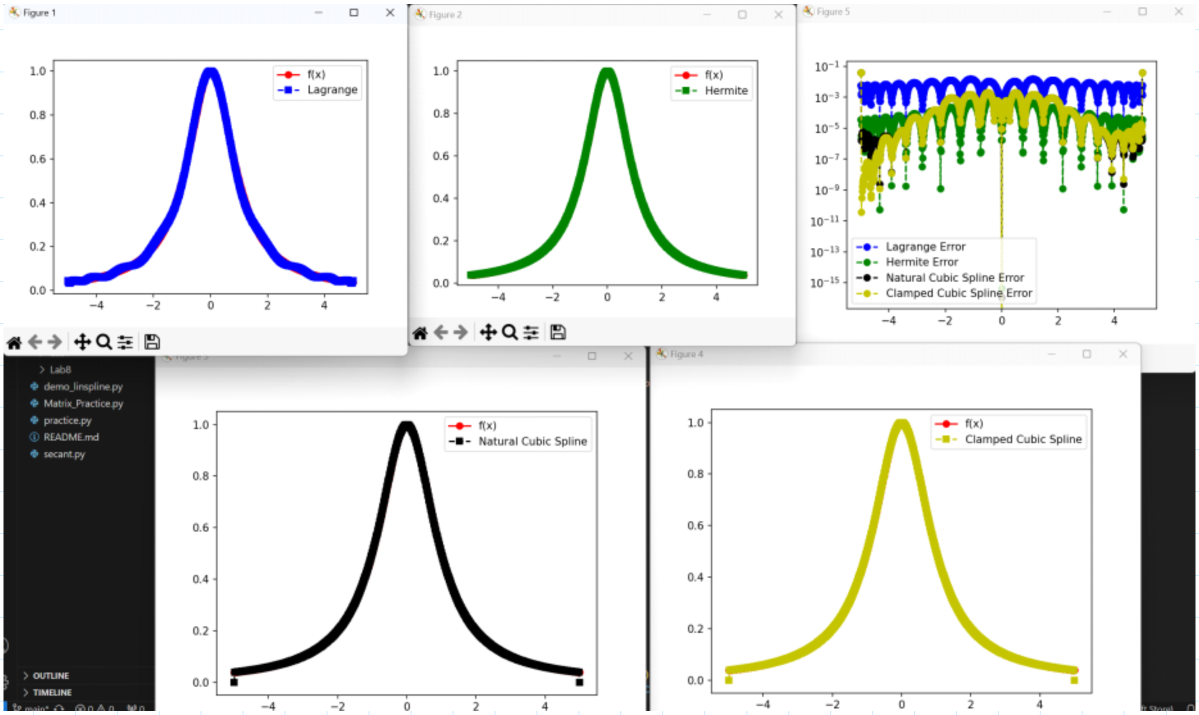
$$x_j = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cdot\cos\left(\frac{2j-1}{2n}\pi\right), \text{ for } j=1,\dots,n$$

In Python, over the interval $[-5,5]$, the nodes are given as follows:
Note: as mentioned in the class announcement, the ordering of the Chebyshev nodes need to be flipped for the cubic spline interpolations.

```
j = np.array(range(1,N+2))
xint = 5*np.cos(np.pi*(2*j -1)/(2*(N+1)))
xint = np.flip(xint)
```

Using the above nodes and the same code as used above, I found the following for $N = 15$:





As we found before, the use of the Chebyshev nodes helped to decrease the error at the endpoints for the Lagrange and Hermite Approximations. It appears that it did add one point with larger error right at the end of the cubic spline approximations. This could be a result of the conditioning of the splines and how the endpoints are defined.

3.) 3. Consider the task of approximating a periodic function such as $f(x) = \sin(10x)$ on the interval $[0, 2\pi]$ using the cubic spline. How do you modify the end point conditions on the coefficients so that the spline is naturally periodic?

To ensure that the spline will properly approximate any function $f(x)$, we already specify that the spline is continuous, and that the first two derivatives of the spline are continuous. That is, we need to satisfy the following conditions:

$$S_i(x_i) = f(x_i) \qquad \text{and} \qquad S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$$
$$S_i(x_{i+1}) = f(x_{i+1}) \qquad\qquad\quad S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$$

To ensure that the spline is naturally periodic, we must also satisfy the condition that the endpoints, and their derivatives, are also equivalent. That is to say:

For some S_i evaluated at x_i, where $i = 0, 1, 2, \ldots, n$

$$S_0(x_0) = S_n(x_n), \quad S_0'(x_0) = S_n'(x_n), \quad \text{and} \quad S_0''(x_0) = S_n''(x_n)$$