PHYS 512 – A1, Question 4

Figure 1: Polynomial, Cubic Spline and Rational Interpolation of Cos(x)

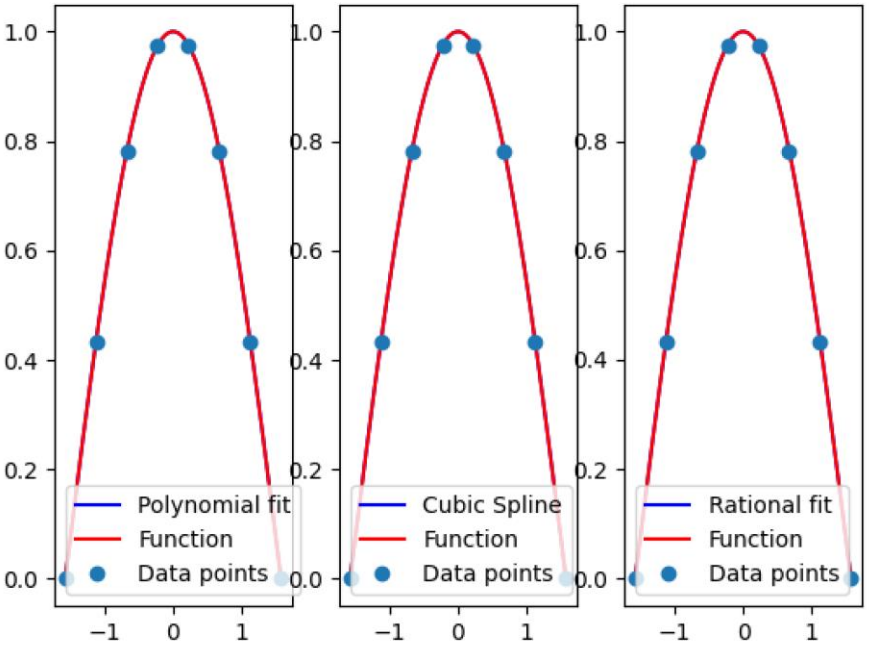
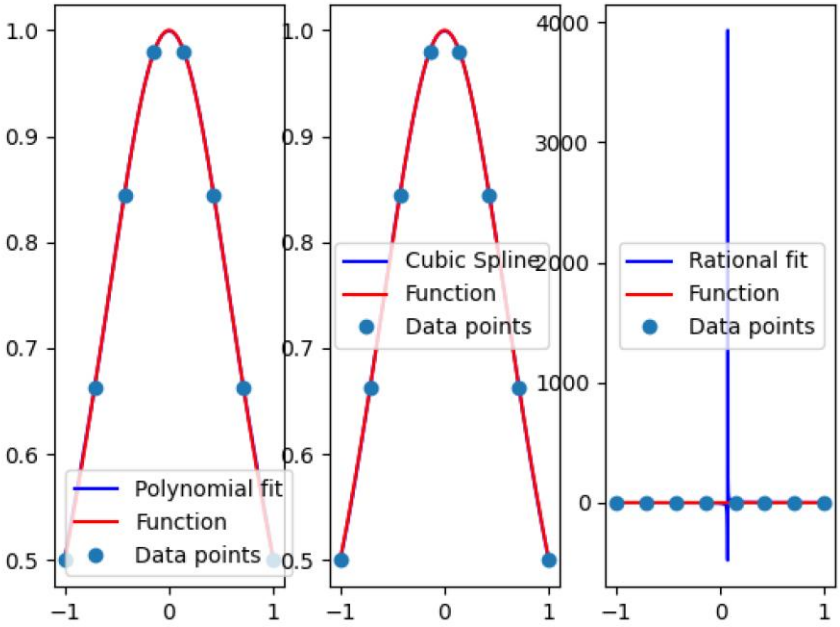


Figure 2: Polynomial, Cubic Spline and Rational Interpolation of the Lorentzian



4.1) I would expect the rational fit of the Lorentzian to have an error of approximately zero. This is because the Lorentzian is itself a rational function, so the rational function interpolation should be exact.

4.2) I used a high order rational function (n = 4, m = 5), and it did not meet my expectations at all. Here are the errors produced by each fit:

```
C:\Users\Owner\Documents\Classes\F2021\PHYS 512\emacs>python lorentian.py
Polynomial error = 0.00045380108142265116, Spline error = -0.00018953641952221712 and Rational error = 1.567479619055471
```

Clearly, the rational function interpolation has a very large error compared to the other methods of interpolation.

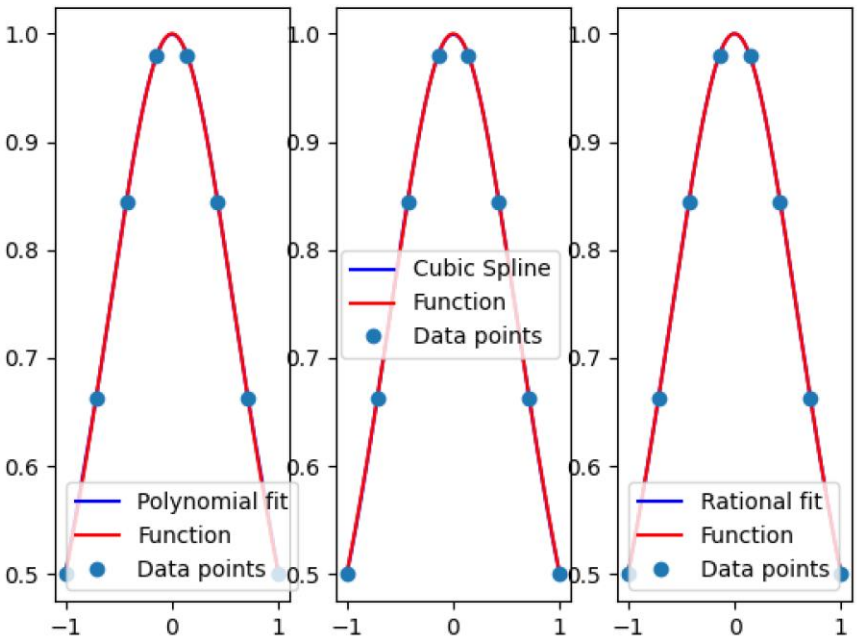
4.3) When switching from np.linalg.inv() to np.linalg.pinv(), I get a much better fit. Here is the error using this method:

```
C:\Users\Owner\Documents\Classes\F2021\PHYS 512\emacs>python lorentian.py
Polynomial error = 0.00045380108142265116, Spline error = -0.00018953641952221712 and Rational error = -2.8976820942716584e-16
```

In this case, the error has improved by 16 orders of magnitude.

Here is the plot of the new fit:

Figure 3: Polynomial, Cubic Spline and Rational Interpolation of the Lorentzian w/ pinv



4.4) I unfortunately do not understand what happened. ☹