插值法编程作业结果分析

李丰杰-3019244196

本次实验完成了对范德蒙多项式插值、拉格朗日插值、牛顿插值、分段线性插值、分段三次Hermite插值算法的代码实现。

本次实验所选用的默认参数为：

1. [a,b] = [-2,8]
2. f(x) = 10\*sinx + 10 \* cosx
3. n+1 = 8
4. m = 1000

Part I 平均误差

下面给出各种方法相对于标准函数值的平均误差表：

*注：均使用上述默认参数*

可以看到根据平均误差而言，发现分段三次Hermite插值最小，而分段线性插值误差最大。而其他三种插值方式平均误差几乎相等。

推测原因是因为插值结点个数为8导致分段线性插值拟合度很低，于是我将插值结点数提升到100,结果发现平均误差为如下表所示：

可以看到此时拉格朗日插值和牛顿插值都出现了严重的**Runge现象**，误差极大。而分段线性插值和分段三次Hermite插值仍保持着较小的误差，这就是教材上所说的**分段插值的优势所在**。且分段三次Hermite插值的误差还是要比分段线性插值小很多的。

就平均误差方面而言，分段线性插值、分段三次Hermite插值与范德蒙德插值是要优于拉格朗日插值与牛顿插值法的，其中又以**分段三次Hermite插值法最佳。**

Part II 函数曲线与比较

1. 范德蒙德多项式插值法

根据使用前述德默认参数值，我们可以得到范德蒙德多项式插值法的函数图像与标准函数图像的对比图：

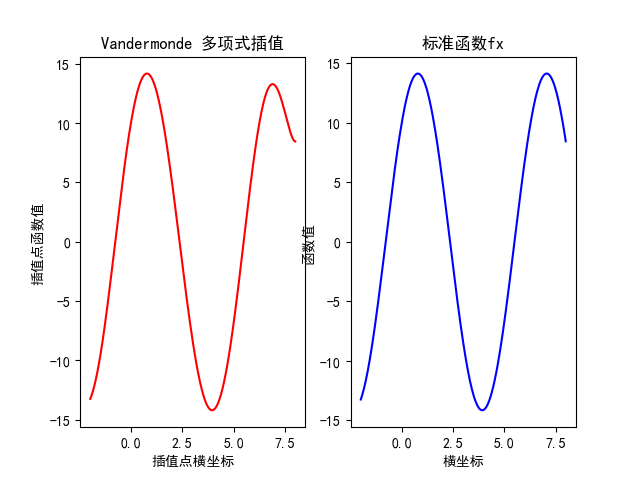


图1:范德蒙德多项式插值函数对比图

可以看到，范德蒙德多项式插值的图像与标准函数fx十分接近，唯一的不足是在**端点处存在些许的震荡现象**。

1. Lagrange插值法

根据使用前述德默认参数值，我们可以得到Lagrange插值法的函数图像与标准函数图像的对比图：

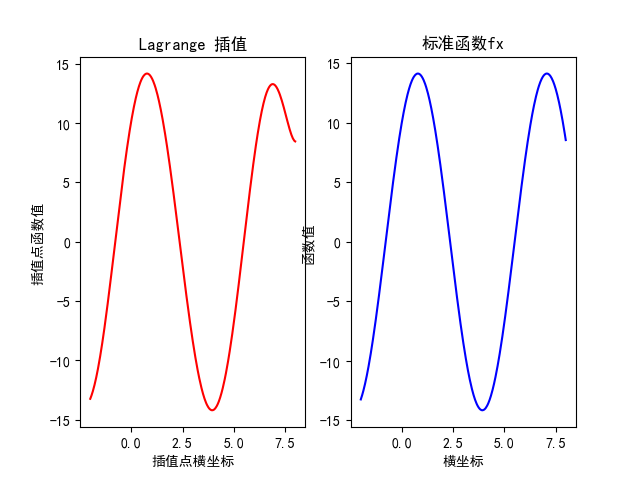


图2:Lagrange插值函数对比图

可以看到，其与标准函数的相差也不太大，但在端点处的震荡幅度略大于范德蒙德多项式插值法。

接下来我们将插值节点个数改为100，发现其出现明显的**Runge现象：**

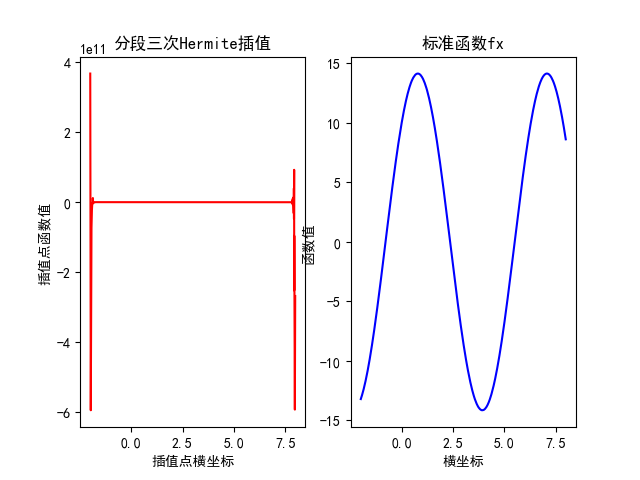


图3:Lagrange插值法的Runge现象

1. Newton插值法

根据使用前述德默认参数值，我们可以得到Newton插值法的函数图像与标准函数图像的对比图：

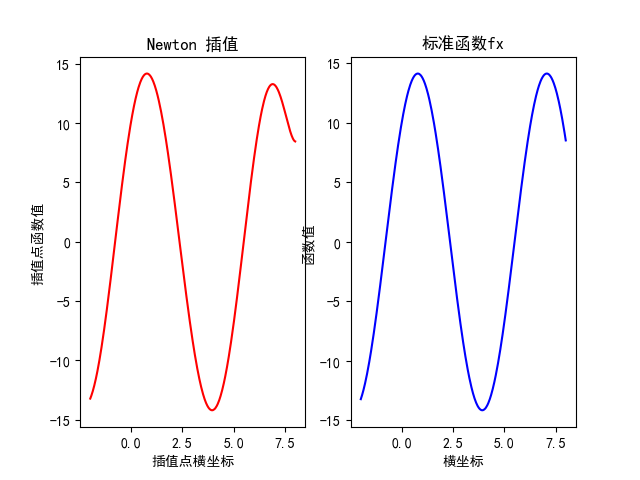


图4:Newton插值函数对比图

结果与Lagrange插值相似，同样也会产生Runge现象，不再赘述。

1. 分段线性插值

根据使用前述德默认参数值，我们可以得到分段线性插值法的函数图像与标准函数图像的对比图：

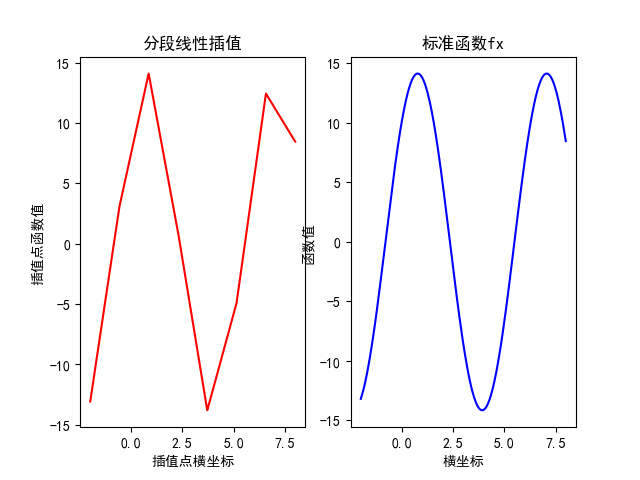


图5:分段线性插值函数对比图

可以看到，相比于前面几个插值法，他的拟合度其实不太好，但这是因为**插值结点数目过少的原因，**下面我们给出插值结点数目为100时其函数图像：

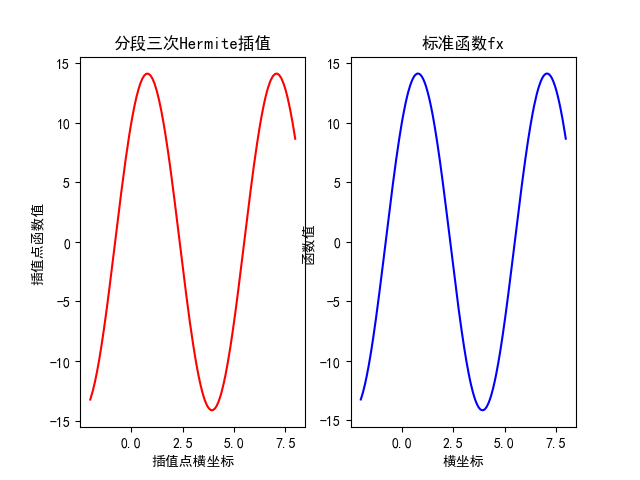


图6:插值点增加后分段线性插值函数对比图

可以看到其拟合程度仍然保持的非常好，而且完全没有出现端点处的振荡现象，这就是分段插值所具有的优势：可以在高次情况下仍保持很好的拟合度。

1. 分段三次Hermite插值

根据使用前述德默认参数值，我们可以得到分段三次Hermite插值法的函数图像与标准函数图像的对比图：

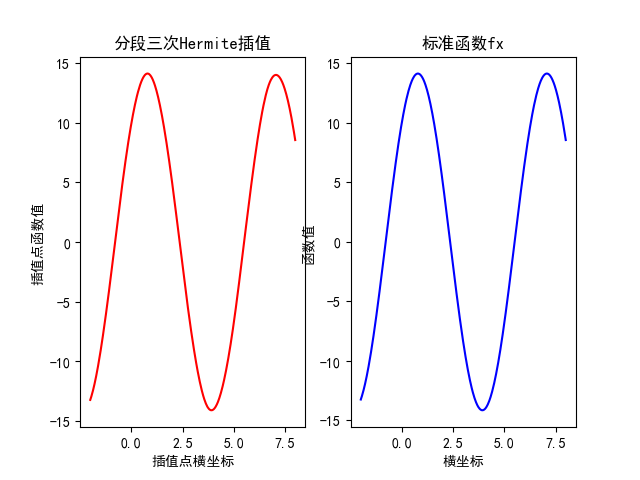


图7:分段三次Hermite插值函数对比图

可以看到，即使在插值结点数目较小的情况下，该插值法也可以保证很高的拟合度，且避免端点处的振荡现象。且在插值结点数目很多的情况下依然可以避免Runge现象。

Part III 总结

通过本次实验，我更加了解了各个插值法的原理，不仅仅是只会手写题或者记住公式，而是能够在计算机上编写相关程序，将其优劣通过数据进行对比、并对Runge现象进行了较为深入的讨论，收益很多。

此外我也学习了python3相关语法与函数库的使用，增强了我的编码能力。在实验中遇到的一些困难也能够及时在网络中找到解决方法，lab用时一天，整体实现还算简单。

本报告的内容还有很多不足，望老师批评指正。