# 数值计算报告

16340150

刘俊峰

软件工程教务2班

## 问题描述

1. 已知 sin(0.32)=0.314567，sin(0.34)=0.333487，sin(0.36)=0.352274， sin(0.38)=0.370920。请采用线性插值、二次插值、三次插值分别计算 sin(0.35)的值。
2. 请采用下述方法计算 115 的平方根，精确到小数点后六位。

（1）二分法。选取求根区间为[10, 11]。

（2）牛顿法。

（3）简化牛顿法。

（4）弦截法。

绘出横坐标分别为计算时间、迭代步数时的收敛精度曲线。

1. 请采用递推最小二乘法求解超定线性方程组 Ax=b，其中 A 为 mⅹn 维的已知矩阵，b 为 m 维的已知向量，x 为 n 维的未知向量，其中 n=10，m=10000。A 与 b 中的元素服从独立同分 布的正态分布。绘出横坐标为迭代步数时的收敛精度曲线。
2. 请编写 1024 点快速傅里叶变换的算法。自行生成一段混杂若干不同频率正弦的信号， 测试所编写的快速傅里叶变换算法。
3. 请采用复合梯形公式与复合辛普森公式，计算 sin(x)/x 在[0, 1]范围内的积分。采样 点数目为 5、9、17、33。
4. 请采用下述方法，求解常微分方程初值问题 y’=y-2x/y，y(0)=1，计算区间为[0, 1]， 步长为 0.1。

（1）前向欧拉法。

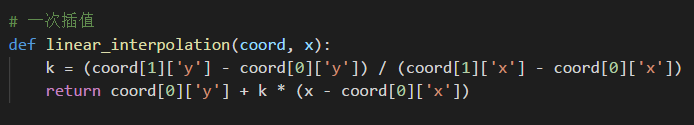
（2）后向欧拉法。

（3）梯形方法。

（4）改进欧拉方法。

## 算法设计

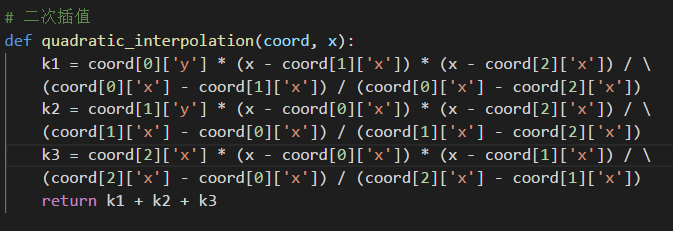
1. **插值计算sin(0.35)**
2. 线性插值



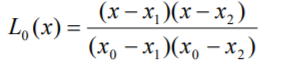
如图。采用了线性插值公式的点斜式

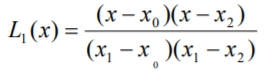
斜率：k=(y1-y0)/(x1-x0), Lx = k(x-x0)+y0

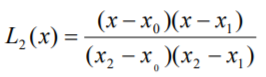
1. 二次插值

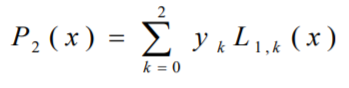


二次插值公式：







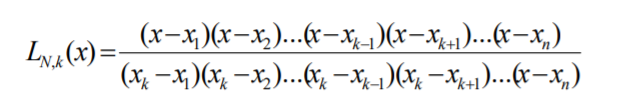


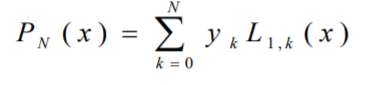
代码中分别用k1,k2,k3表示L0,L1,L2

1. 三次插值

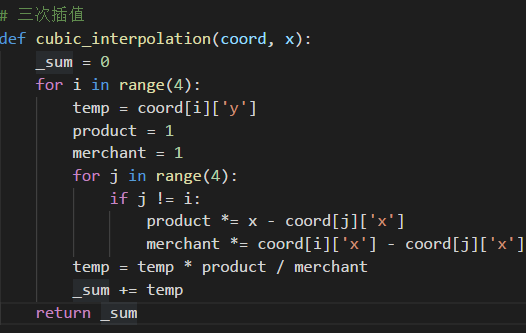
书本上没有现成的三次插值的公式，采用的是N次的拉格朗日插值

即：





二次插值需要三个点，三次插值则需要四个点

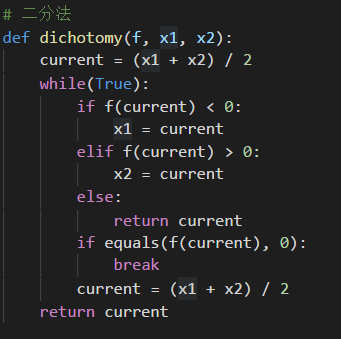


Merchant是Lx的分母，product是Lx的分子

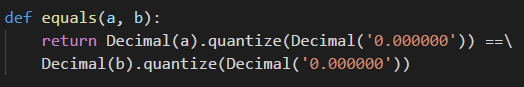
1. **请采用下述方法计算 115 的平方根，精确到小数点后六位。**
2. 二分法，区间为[10,11]

基本设计思路：

1. 找到区间[a,b]的中点x=(a+b)/2
2. 计算fx的值
3. 若fx>0, b=x
4. 若fx<0, a=x
5. 若fx=0, 返回x
6. 判断fx是否有小于E-6精度的误差，小于返回x；否则回到1）



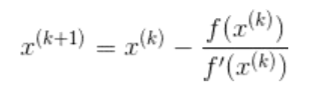
Equals函数用于判断精度是否符合要求



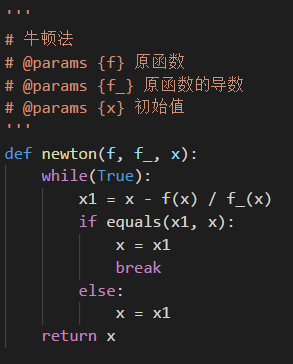
1. 牛顿法

牛顿法是一种迭代算法，当结果精度满足时可以完成迭代

迭代公式：

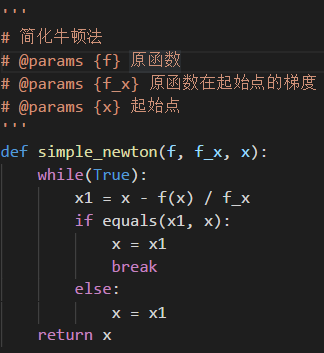


在设计算法的时候，需要自己输入fx的导函数f\_x



1. 简化牛顿法

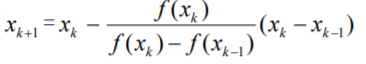
和牛顿法类似，不过为了方便计算，将f\_x的值固定为初始点的值

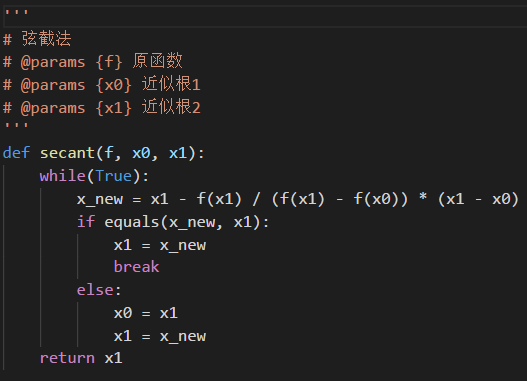


同样需要自己输入f\_x

1. 弦截法

可以看作是牛顿法的变种，用点斜式取代牛顿法中的f\_x，同样有简化计算的效果



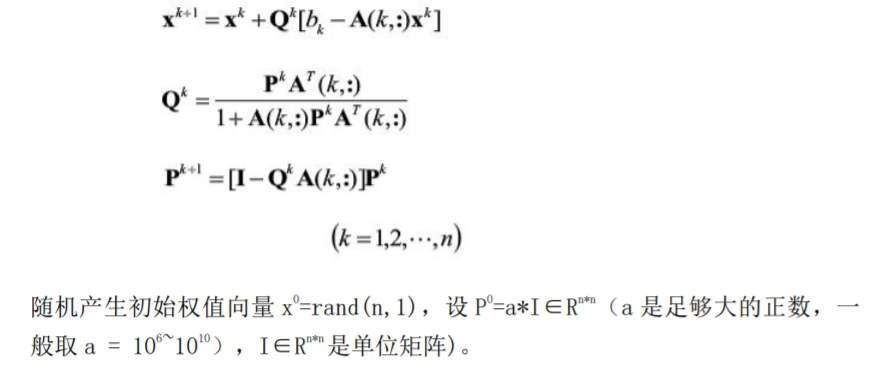


1. **请采用递推最小二乘法求解超定线性方程组 Ax=b，其中 A 为 mⅹn 维 的已知矩阵，b 为 m 维的已知向量，x 为 n 维的未知向量，其中 n=10， m=10000。A 与 b 中的元素服从独立同分布的正态分布。绘出横坐标为迭代步 数时的收敛精度曲线。**

由于Ax=b中，A不是一个方阵，因此用消元法和迭代法都不能有效的解出它的结果。

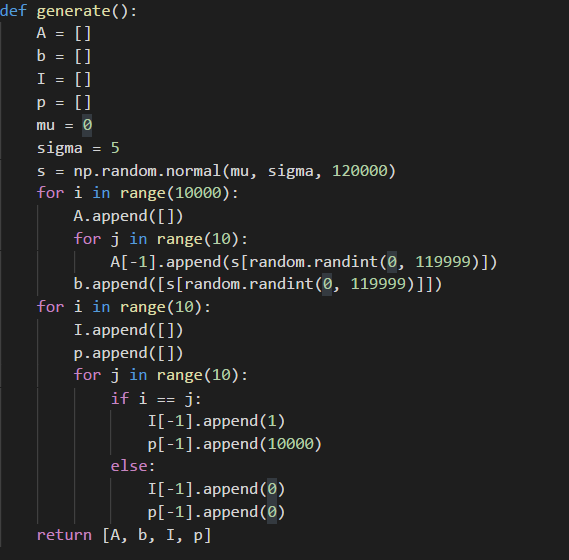
利用公式：

ATAx=ATb，将原方程转化为可解的普通线性方程，再利用最小二乘法：



迭代解出x的值

根据题目要求，要先生成一个矩阵A，向量b, 单位阵I, 以及初始化P

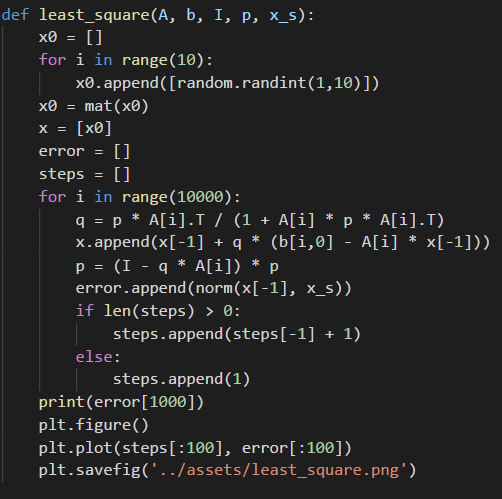


另外要算相对误差，就必须算出标准解

这里利用了scipy库的solve方法，解出ATAx=ATb的值



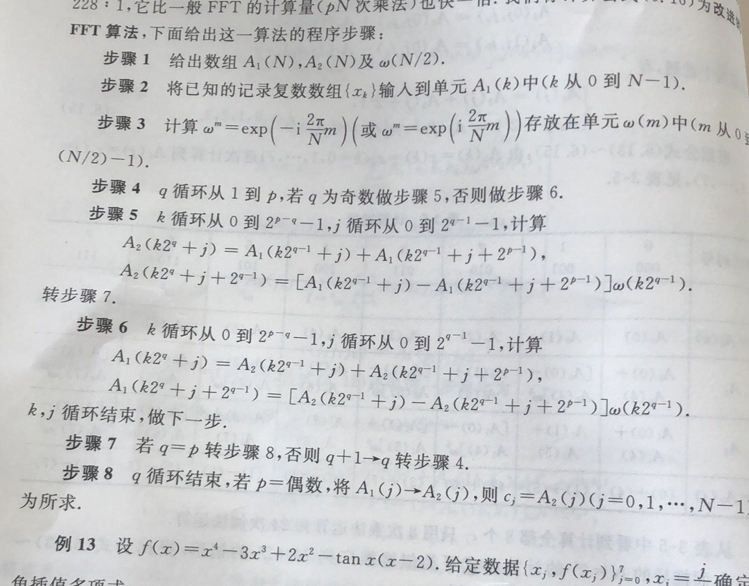
然后设计最小二乘算法



X\_s是标准解，误差用x-x\_s的二范数表示

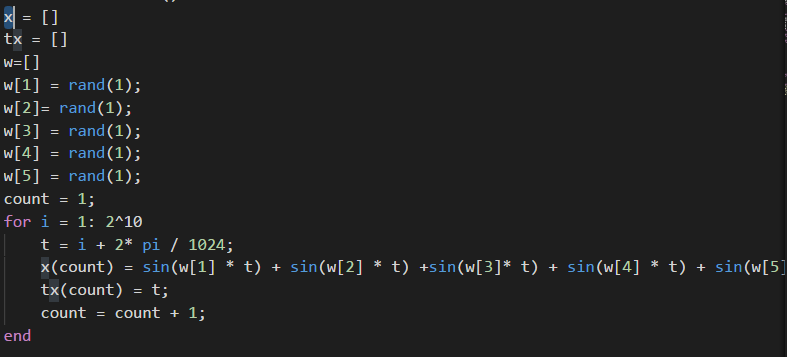
1. **请编写 1024 点快速傅里叶变换的算法。自行生成一段混杂若干不同频率正弦的信号， 测试所编写的快速傅里叶变换算法。**

算法设计如下：



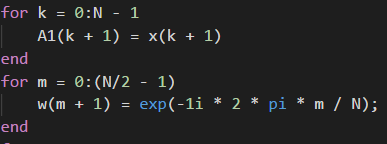
源码：

生成正弦波：

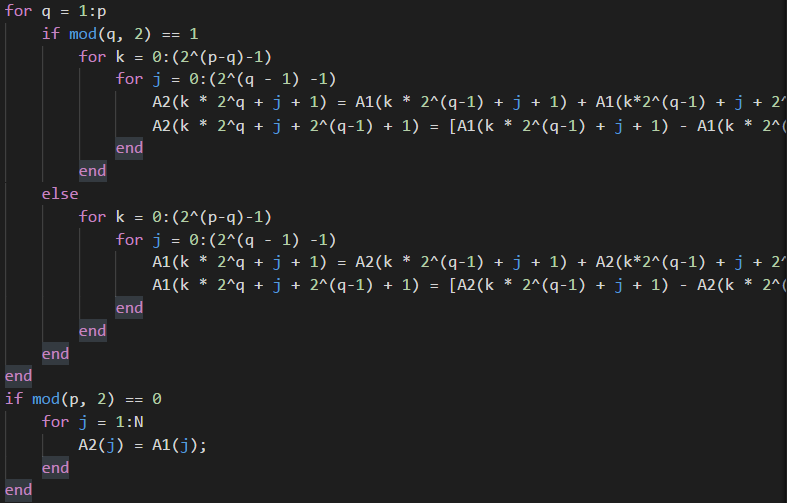


用了5段正弦波形叠加

初始化A1和w

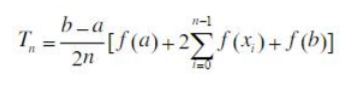


计算A2

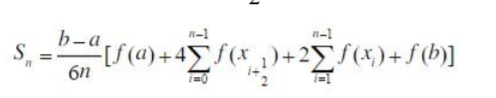


1. **请采用复合梯形公式与复合辛普森公式，计算 sin(x)/x 在[0, 1]范围 内的积分。采样点数目为 5、9、17、33。**

梯形公式：

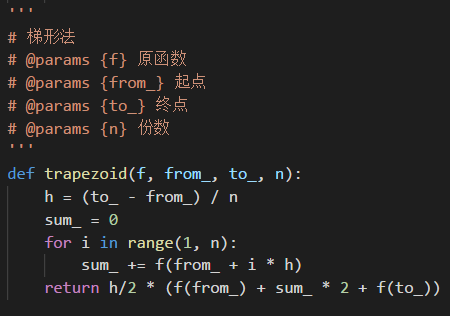


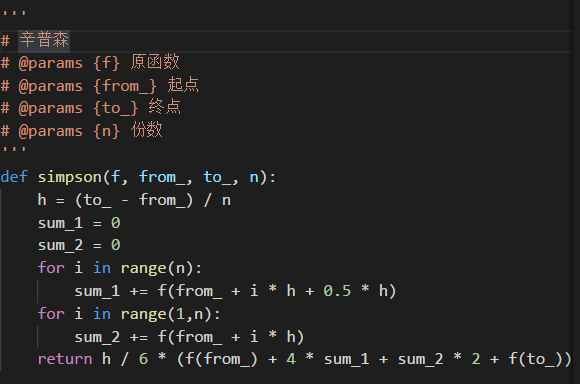
辛普森公式：



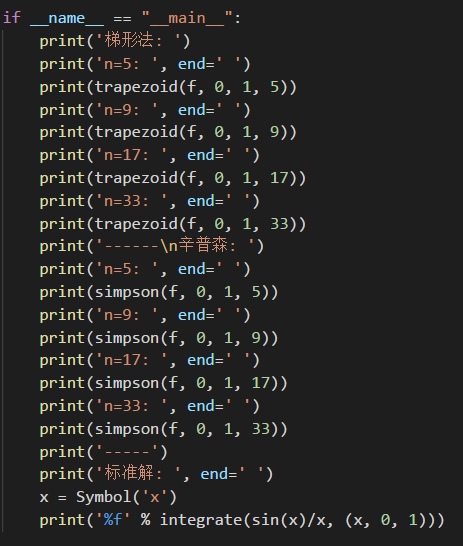
其中







测试：



1. **请采用下述方法，求解常微分方程初值问题 y’=y-2x/y，y(0)=1，计算区间为 [0, 1]， 步长为 0.1。**

**（1）前向欧拉法。**

**（2）后向欧拉法。**

**（3）梯形方法。**

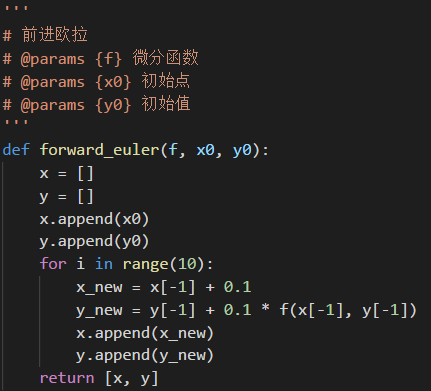
**（4）改进欧拉方法。**

1. 前向欧拉法

公式：

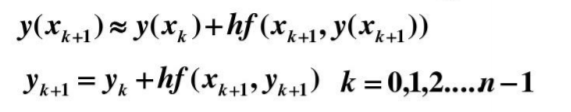


算法设计：

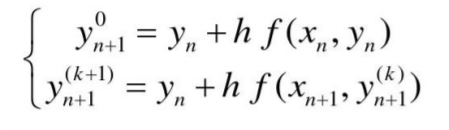


1. 后向欧拉法：

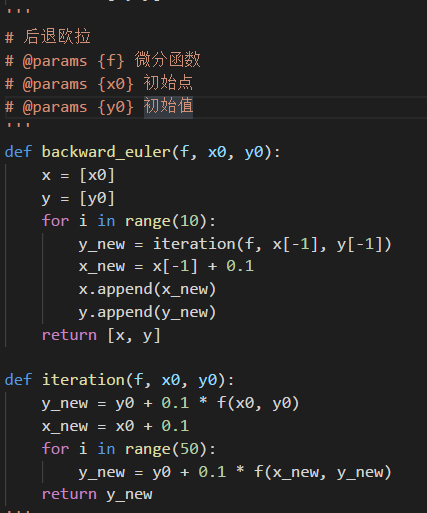
公式



后向欧拉法是隐式迭代公式，要算出yk+1可以用这样的迭代方式：



具体设计：

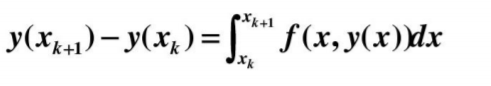


后面的iteration用来迭代yk+1

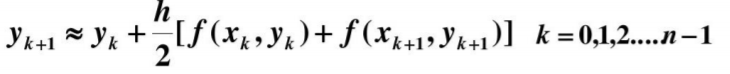
1. 梯形方法

跟后向欧拉方法类似，是隐式迭代方法

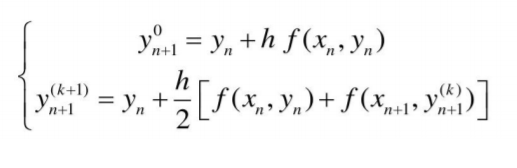
公式



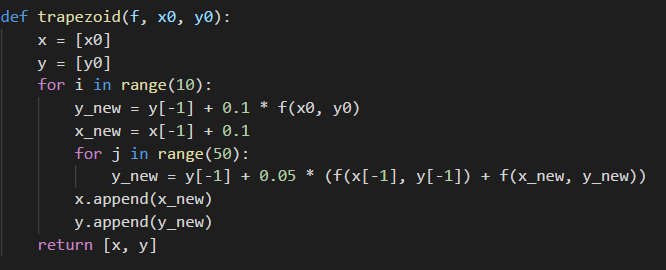
即：



递推公式为：

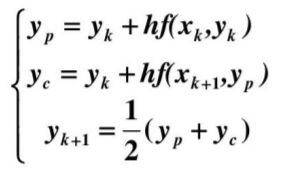


具体实现：



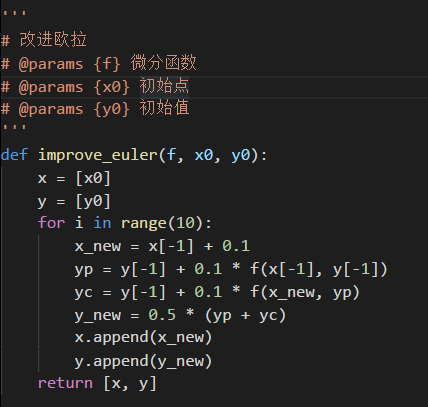
进行50次迭代逼近yk+1

1. 改进欧拉



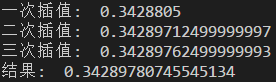
先用欧拉方法求出近似值，然后用梯形方法校正

不需要隐式迭代，比梯形方法的效率高



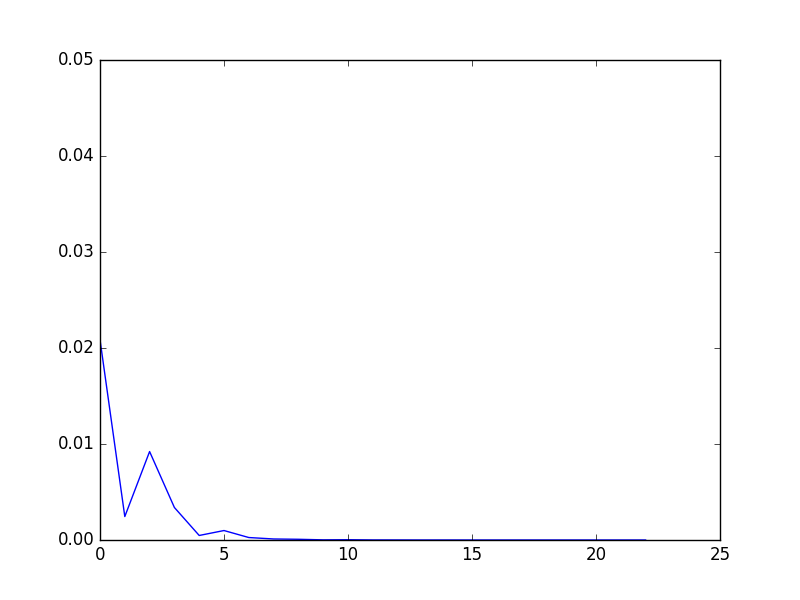
## 数值实验

1. 三种插值

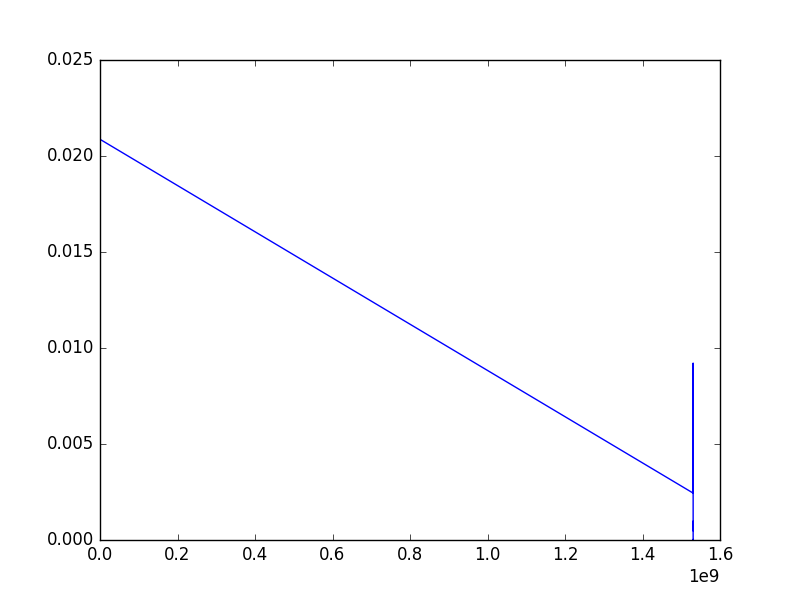


1. 二分法

收敛速度较慢

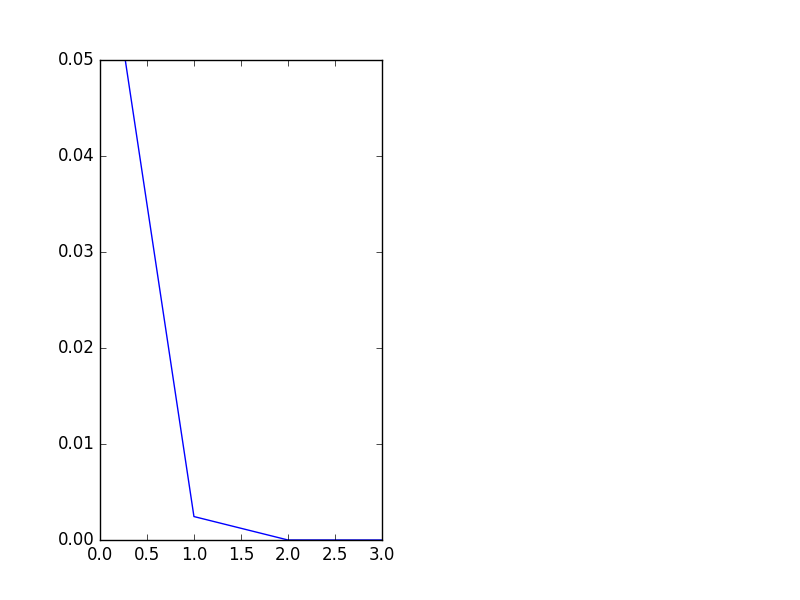


Time

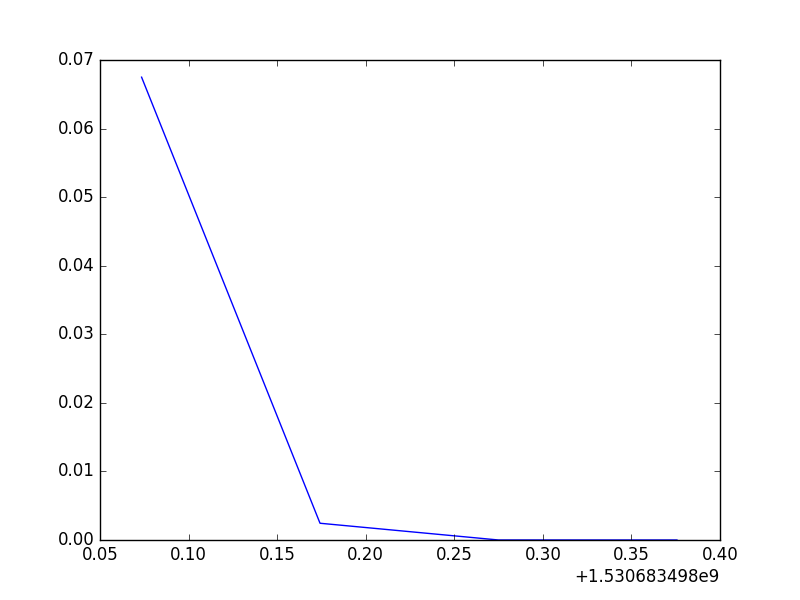


1. 牛顿法

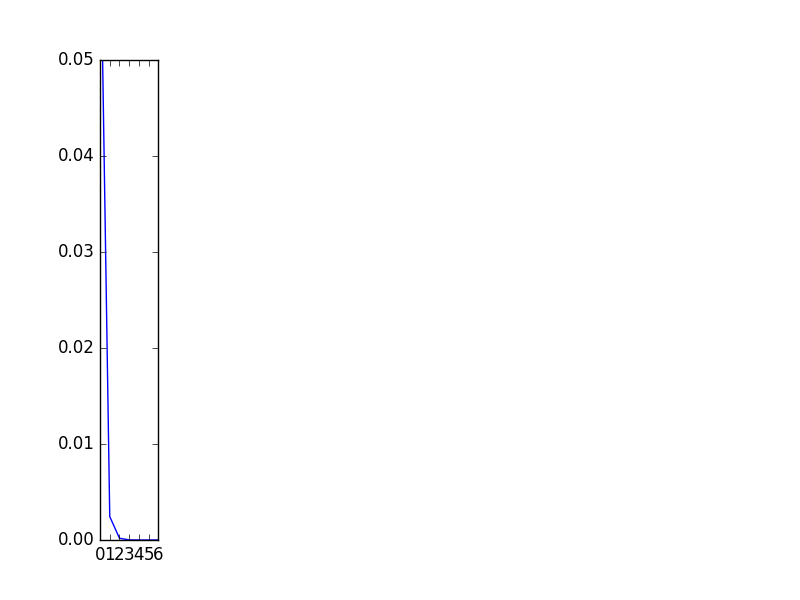
收敛速度非常快



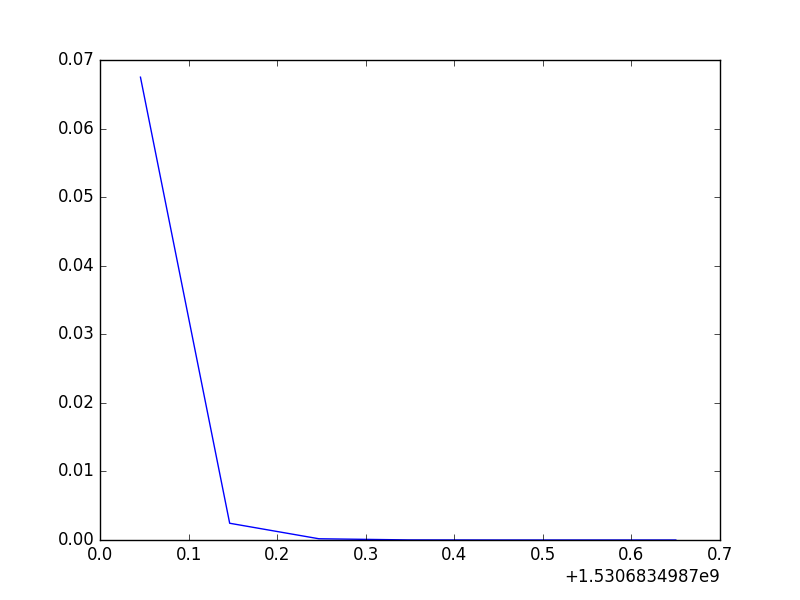
Time



1. 简单牛顿



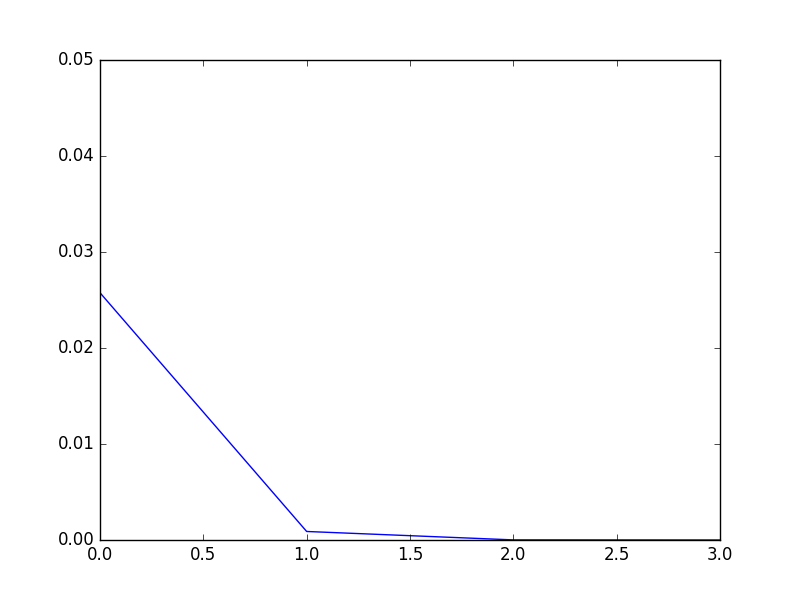
Time



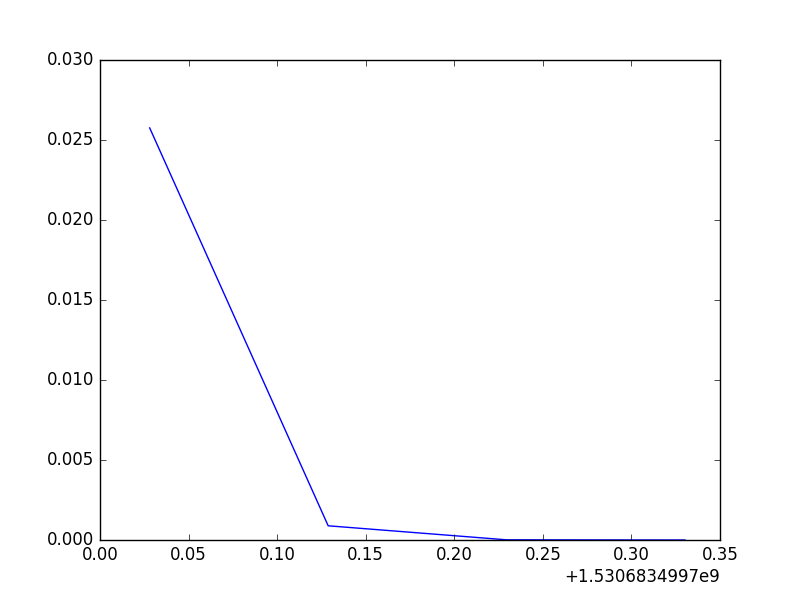
收敛速度比牛顿稍慢

1. 弦截法

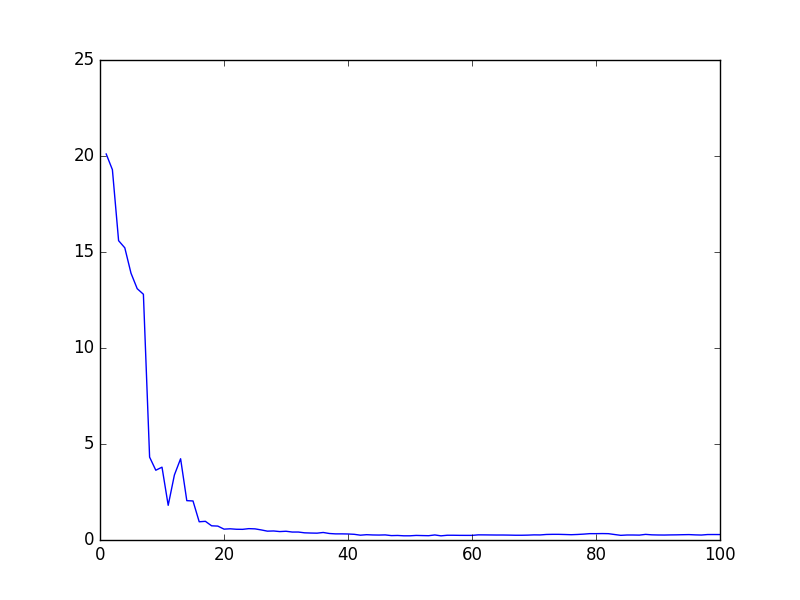
速度仅次于牛顿法



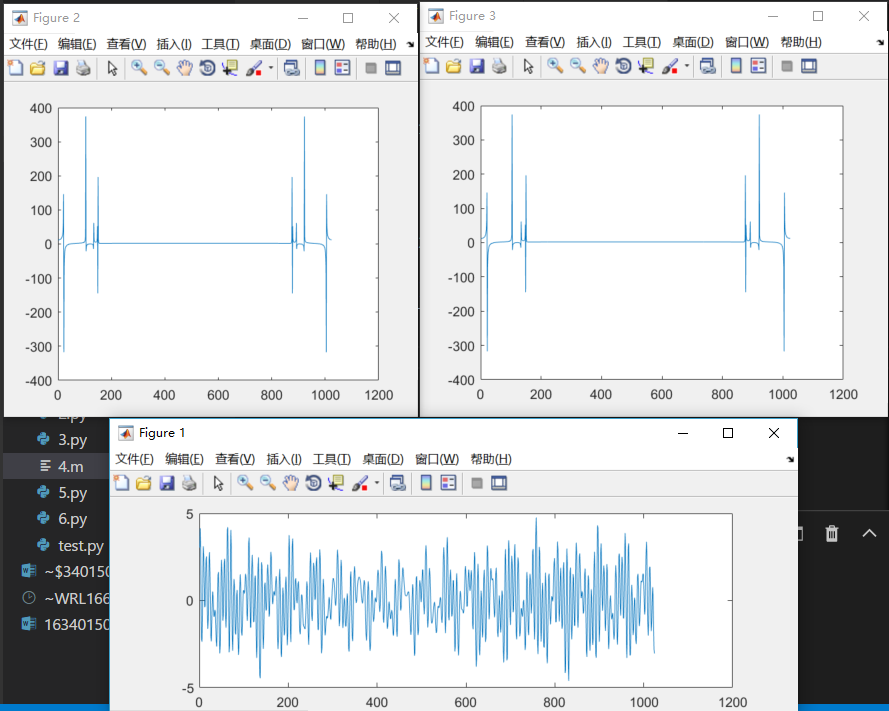
Time



1. RLS解超定线性方程

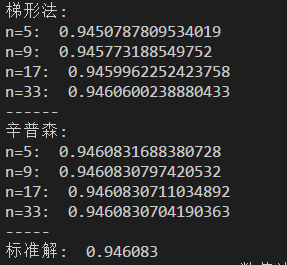


1. 快速傅里叶变换

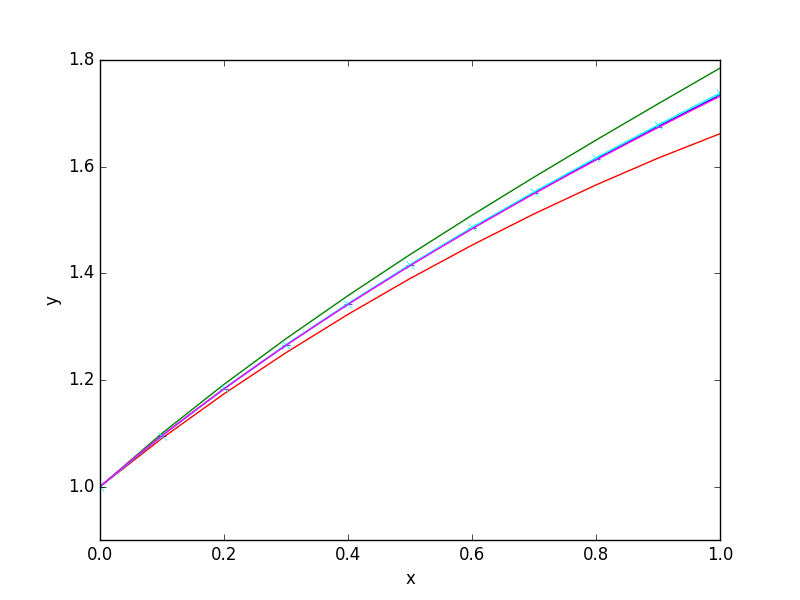


左上图为fft，右上图为内置fft，下图为正弦波

1. 梯形公式和辛普森公式



1. 解常微分方程



绿色是向前欧拉，红色是向后欧拉。蓝色线是梯形公式，浅蓝色线是改进欧拉，紫色线是标准解。

## 结果分析

1. 三种插值法的结果分析

显然，三次插值的结果是离标准解最近的，二次插值次之。三种插值都离标准解较近，说明拉格朗日插值方法的正确

1. 四种迭代算法的误差分析

二分法的迭代步数是四种方法里面最多的，进行到20+步时才收敛到精确值附近

牛顿法的收敛速度很快，两步之内收敛到标准解

简化牛顿法的收敛速度比牛顿法稍微慢一点，但计算上简单，而且，只要6步就收敛

弦截法速度是四种方法里第二快的算法，要4步收敛，而且计算上并不复杂（不需要求导），因此也是一种优秀的迭代方法。

在时间图上，由于后三种方法收敛速度较快，在时间上几乎体现不出区别。（通过sleep函数提高耗时画图）

1. 由上图可得，最小二乘法的收敛速度并不是十分优秀，在100+步之后仍然没有达到很高的精度。
2. 上图为测试结果，可以看出混杂若干不同频率正弦的信号在经过傅里叶变换后，可以明显看出 ck的值。并且在比较内置fft算法后发现差别不大（误差在E-15）
3. 梯形公式和辛普森公式。对于两个公式而言，取的点越多越逼近准确值。而比较两种算法可以发现，在取相同点的时候，辛普森算法更加接近真实值，比梯形公式要更加优秀
4. 常微分方程求解。由图可以发现，前向和后向欧拉方法的误差会较大。而改进欧拉方法和梯形方法与真实值的差别较小。