

1、总体介绍

主要实现了一个计算器的功能，包括加减乘除、对数、三角函数等常用操作。

在此基础上，添加了求极值、积分、微分、作图等功能。作图具体包括作函数图像、作线性/非线性规划解的空间、作圆锥曲线等功能。

具有用户友好、功能实用全面、代码简洁、组织合理、健壮性强等特点。

2、配置环境说明

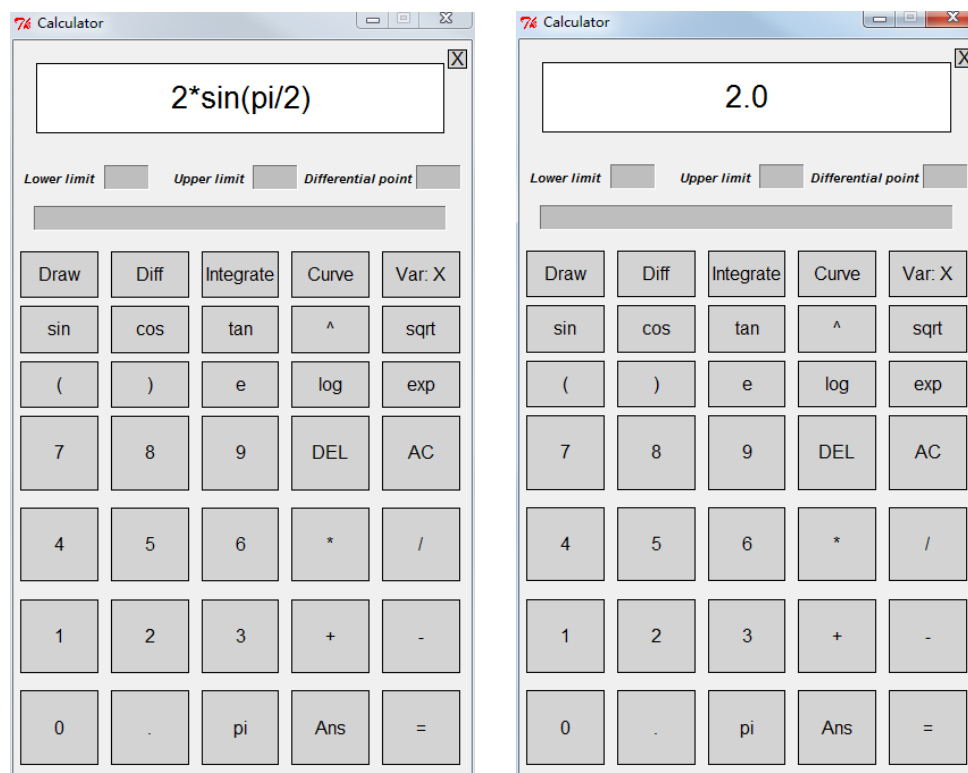
本程序使用了 python 知名的科学计算绘图库 matplotlib，我在机器上使用的是 Enthought 公司出品的 Canopy 软件。该软件集成了 matplotlib、scipy、numpy 等知名 python 科学计算库。我还使用了 graphics 库。

3、功能展示

➤ 计算器基本功能：

算式可在中间的长条黑框中输入或者通过按键输入。

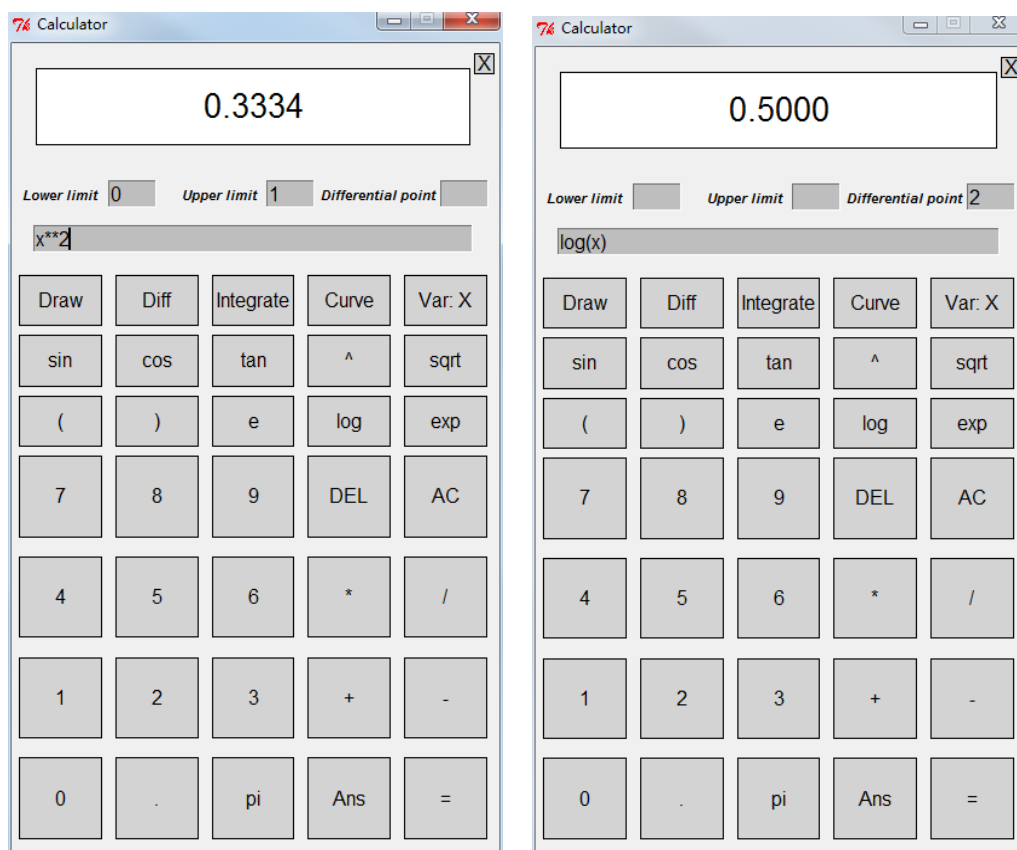
基本操作与普通计算器完全类似，不再赘述。



➤ 求解积分：

在 Lower limit 和 Upper limit 后中输入你积分的上界和下界。函数可在中间的长条黑框中输入或者通过按键输入。然后按下 Integrate 键。

保证前三位有效。



➤ **求解微分：**

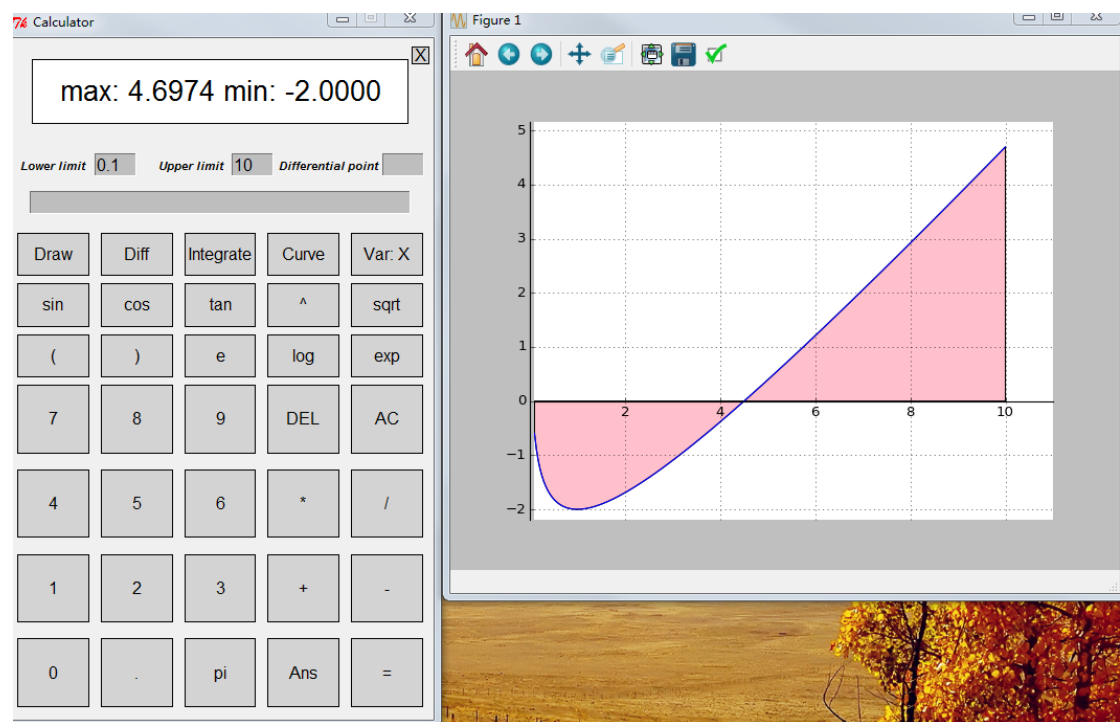
在 Differential point 中输入你求导的点的横坐标。函数可在中间的长条黑框中输入或者通过按键输入。按下 Diff 键。

保证前三位有效。

➤ **求解函数极值以及绘图：**

在 Lower limit 和 Upper limit 后中输入你绘图的 x 取值范围。函数通过按键输入。按下 Draw 键。如果不输取值范围那么默认为[-5,5]。

例如我们想求 $x - \ln(x) - 3$ 的在 [0.1,10] 上的极值并绘图，结果如下：



➤ 作线性规划/非线性规划解的空间：

在 Lower limit 和 Upper limit 后中输入你绘图的 x 取值范围。约束式在长条输入框中输入，整理成 y 在右边的不等式形式后输入并用分号分隔。按下 Draw 键。如果不输取值范围那么默认为[-5,5]。

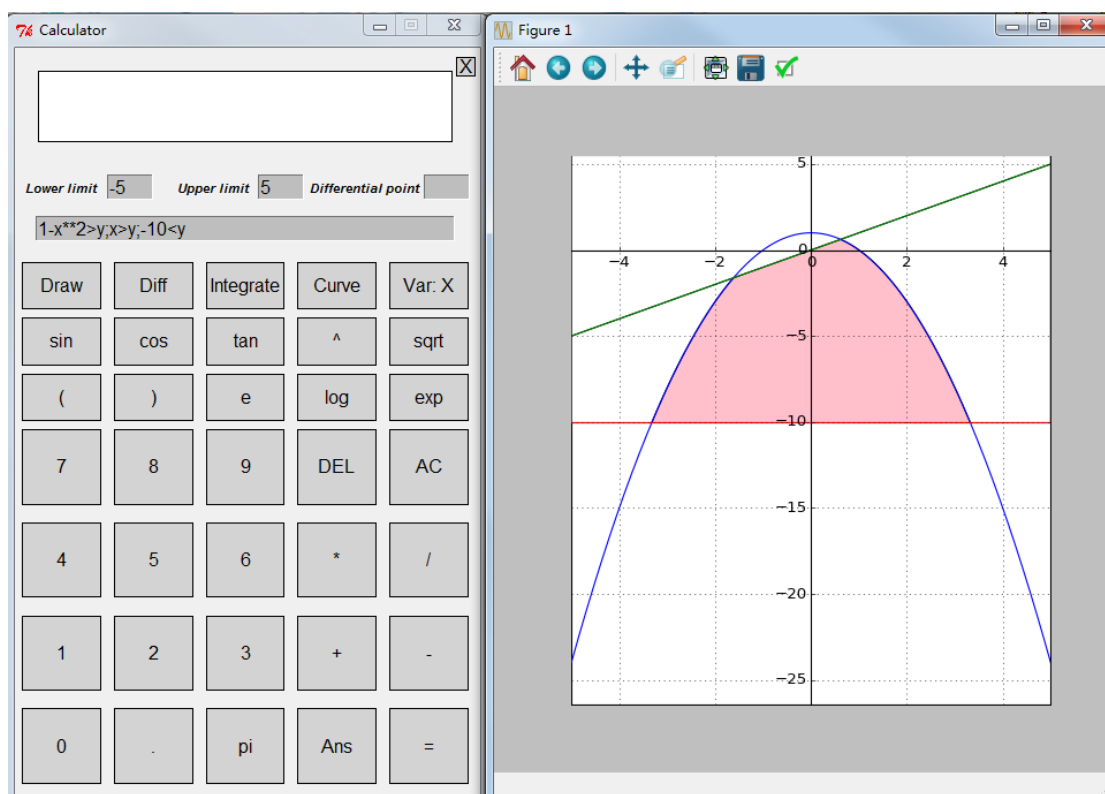
例如我们想求解以下非线性规划：

$$x^2 + y \leq 1 \quad x < y \quad y + 10 \geq 0$$

先进行变形，将 y 都放到不等式右边，并将小于等于改成小于号，大于等于改成大于号（作图时开闭没有影响）：

$$1 - x^2 > y \quad x < y \quad 10 < y$$

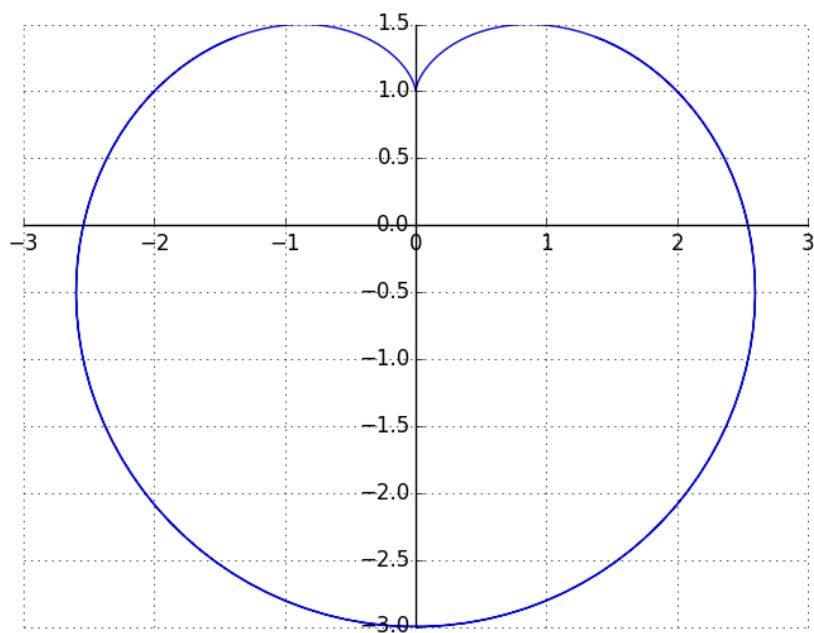
然后在长条输入框中键入 $1-x^2>y;x>y;-10<y$ ，图中粉色空间即为解的空间。



➤ 通过参数方程作图:

虽然前面可以作函数图像,但是有一些曲线并不是函数图像,比如圆锥曲线。要作这样的图,只需在长条输入框中输入用分号分隔的参数方程即可(当然函数图像也可以用过参数方程作出)。参数为 t 。同样可以输入 t 的取值范围,默认为 0 到 2π 。

著名的心形线的参数方程为 $x = 2 \sin(t) - \sin(2t)$, $y = 2 \cos(t) - \cos(2t)$ 。在灰色长条输入框中输入 $2*\sin(t)-\sin(2*t);2*\cos(t)-\cos(2*t)$, 按下 **curve** 得到的图像如下:



4、实现细节

共分为 graphics.py, main.py, mydraw.py 和 mymath.py 四个文件。通过设计类和函数最大限度地实现了代码重用，通过合理的模块分类使得代码结构清晰易读。

➤ **graphics.py**

著名的 python GUI 库。

➤ **main.py**

一个主函数 FuncGraph，主要是利用 mymath 中定义的 button 类和 drawlabel 函数布置界面。在一个 while 循环中检测鼠标点击并作出相应动作。这里用的是 try except 语句大大提高了函数的健壮性，在出现非法输入时将显示”Math Error”。有部分功能既可通过按键输入，也可以在输入框中输入，则优先考虑输入框。

➤ **mydraw.py**

主要是实现了三个不同的作图功能：

my_plot 对应普通函数作图，包括了画坐标轴，画函数图像，染色，返回极值等操作，基本是基于 matplotlib 的函数。值得一提的地方是 $C = \text{eval}(\text{expr} + '+x-x')$ 这一句。由于自变量 x 实际上是一个 numpy 库中的一个 ndarray，或者说是 n 维数组，而且 numpy 重载了 $+/-/*$ 等常规运算符，因此你直接写 $C=x+2$ 得到的 C 即是对 x 这个数组中所有元素加 2 得到的数组。但是问题在于， expr 传进来是一个字符串。这里困扰了我很久，一度准备自己手写一个 parser，最后通过反复查阅资料， eval 即可将环境考虑进来，将字符串转为表达式。之所以加 $'+x-x'$ ，是因为有时候 expr 输进来是一个常数，这时 eval 会将 expr 转化为 int/float，因此需要加上 $'+x-x'$ 强制其检测 x 的类型进行正确的类型转换。后面的代码中也有类似的 trick，不再赘述。

my_plot2 对应线性规划/非线性规划解的空间作图。这里的 trick 在于如何作出解的空间。思考一下，几个不等式的组合其实是对各个解的空间作与操作，也是可以看作对非解的空间作或操作后取反。由此，我的做法是先将整个平面染为粉色，然后将不满足各个不等式的空间染白，剩下的即为几个不等式解的空间的交集。

my_plot3 对应参数方程作图。这里没有什么特别的地方，matplotlib 可以处理参数方程的情况。

➤ **mymath.py**

这里实现了一个按钮类，drawlabel 函数和用到的一些数学运算。

button 类主要是对 Text 类作了重新封装，通过定义按钮中心和长宽即可方便地定义一个按钮。其一个子函数 clicked 用于检测鼠标是否点在了按钮上，即是否在矩形范围内。button 类的存在充分实现了代码重用，可读性大大增强。

buttonAction 主要是实现了按钮被点击后计算器屏幕的更新。

drawlabel 也是对 text 封装了一下，便于方便地在指定位置显示文字。

diff 是微分函数，是 $f'(x) \approx \frac{f(x+\varepsilon) - f(x-\varepsilon)}{2\varepsilon}$ 的简单实现。

inte 是积分函数，是矩阵法近似求数值积分的简单实现。