**设计报告题目（自拟）**

（大作业题目编号）

姓名，学号

班级，个人email

**摘要**

简要概况这个设计的目的、或出发点，及设计过程/原理，设计结果及影响，字数200-300字左右。【注：设计报告整体按照本模板来组织，字号、格式要求、段落间距等都要与本模板一致！】【注意，正文小四号字，各节标题四号字，具体以本模板样式为准】

**关键字：**列出3-6个本报告有关的关键字

**1 引言**

介绍一下本报告设计内容的背景知识，现状等，一般需要标注一些必要的参考文献作为支撑材料，参考文献以索引数字加中括号的方式引用，就像这样[1]。

然后，结合背景以及研究背景中有可能面临的问题，介绍本设计报告的必要性、意义、价值等。再略详细介绍本设计原理、过程，创新点（如果有），结果总结等。

**2 正文内容（题目自拟，如下根据具体情况自行设置各级小节）**

**2.1 小节子标题**

在此小节应主要介绍所设计内容的基本原理、现有方法或做法的过程。如有图示表示应尽量提供。

**2.2 小节子标题**

详细介绍本设计的原理、方法、过程等。侧重本设计的内容（而非现有方法的内容）。

**（1）二级小节子标题**

如果小节下面还有二级小节，可按如上方式用括号列子标题。

本设计中如果有一些图或表，最好也要列上，格式如下。

表1 表的标题

|  |  |  |
| --- | --- | --- |
| 表格内容。注意， | 表格的标题 | 列于表格的上方 |
| 另外，表和图 | 字号都是**五号**， | 比正文小。 |

下面是一个图的示例。

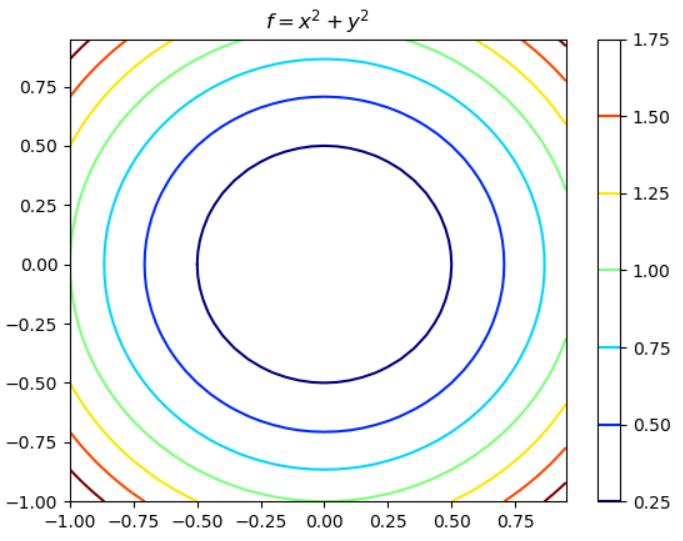


图1 图标题，如果本图里面包含几个子图（用(a),(b)来标识），可以写完图的标题后再继续写：(a)子图标题，(b)子图标题。

还可能有公式，按照如下格式来写：

|  |  |
| --- | --- |
|  | (1) |

引用公式时要指明公式号，如，公式（1）表明了什么。

要注重内容的前后逻辑性，内容要清晰顺畅。能从所写的内容上清楚明白做的是什么东西，为了解决什么问题，具体怎么做的，有无贡献点等。

**3 仿真验证部分**

**3.1 小节子标题**

这部分主要是用Python对本设计内容做的代码编写、及结果验证。本部分最好将要做的具体内容、或要验证的具体内容按小节的方式有条理的叙述出来。并配合必要的Python仿真结果等。

注意，用Python做仿真验证的一些必要前提条件、参数应该列出。

这部分一般会有大量的图（用Python仿真的）来作为仿真验证结果。还要有对图表的细致分析来说明本设计的内容、方法、或创新点。

**4 参考文献**

[1] 作者(如果有)，文章名或专著名，年份，出版杂志或出版社（注：参考文献正文用5号字）

[2] 作者(如果有)，文章名或专著名，年份，网页链接则直接写出网址(如[www.xidian.edu.cn](http://www.xidian.edu.cn))

**5 源代码**

将自己编写的本设计报告有关的源代码都列于此。如果有多个模块或子程序，要先指明模块名及功能（加粗显示），再依次罗列代码。示例：

**模块t9\_1.py**

**（功能：测试1，注，代码正文用5号字）**

import numpy as np

import scipy.integrate as inte

def fun1(x):

#待测试积分的函数 （x^3-x^2+1）

return x\*\*3-x\*\*2+1

if 1:#测试主代码

range1,range2=0,5#积分区间

#调用quad()函数求积分

out1=inte.quad(fun1,range1,range2)[0]#

#直接按离散叠加方式求积分（精度低）

interval=0.1#离散序列间隔

xt=np.arange(range1,range2,interval)#离散序列

out2=fun1(xt).sum()\*interval

#理论分析结果（用于对比）

xr=np.array([range1,range2])

tmp=xr\*\*4/4.-xr\*\*3/3.+xr#待积分表达式理论计算结果

out\_theory=tmp[1]-tmp[0]#积分理论结果

#结果打印

print u'积分理论分析结果：',out\_theory

print u'离散叠加方式的积分结果：',out2,u',误差：',out\_theory-out2

print u'用quad函数求解的积分结果：',out1,u',误差：',out\_theory-out1

**模块t9\_1\_2.py**

**（功能：测试2）**

# -\*- coding: utf-8 -\*-

import numpy as np

import scipy.integrate as inte

def f2(x,y):

#待测试积分的函数 （x\*y）

return x\*y

def range\_y():

#y的范围

return [0,0.5]

def range\_x(y):

#x的范围

return [0,1-2.\*y]

if 1:#测试主代码

#调用nquad()函数求积分

out1=inte.nquad(f2,[range\_x,range\_y])#

#结果打印

print u'用nquad函数求解的积分结果：',out1[0],u',相对误差：',out1[1]