**郑州大学信息工程学院**

****

**基于python的数字信号处理实验指导书**

专 业

班 级

姓 名

学 号

指导老师

目录

[一、python安装教程 1](#_Toc104038131)

[1.1 python解释器 1](#_Toc104038132)

[1.2 安装python第三方库 2](#_Toc104038133)

[1.3 Jupyter Notebook编辑器 2](#_Toc104038134)

[1.4 Google colab在线运行 4](#_Toc104038135)

[二、python语言入门教程 5](#_Toc104038136)

[2.1 python对象和变量 5](#_Toc104038137)

[2.2 DSP常用数值运算函数及功能 6](#_Toc104038138)

[2.2.1 基于 NumPy的数组运算 6](#_Toc104038139)

[2.2.2基于scipy的数值计算及应用 8](#_Toc104038140)

[2.2.3基于matplotlib的数据可视化 8](#_Toc104038141)

[三、数字信号处理实验 10](#_Toc104038142)

[实验一：系统响应以及系统稳定性 10](#_Toc104038143)

[实验二：时域采样 12](#_Toc104038144)

[实验三：频域采样 14](#_Toc104038145)

[实验四：零极点分析 16](#_Toc104038146)

[实验五：相关函数和线性卷积 18](#_Toc104038147)

[实验六：圆周卷积和线性卷积 20](#_Toc104038148)

[实验七：用FFT对信号做频谱分析 22](#_Toc104038149)

[实验八：IIR数字滤波器设计及软件实现 24](#_Toc104038150)

[实验九：FIR数字滤波器设计与软件实现 26](#_Toc104038151)

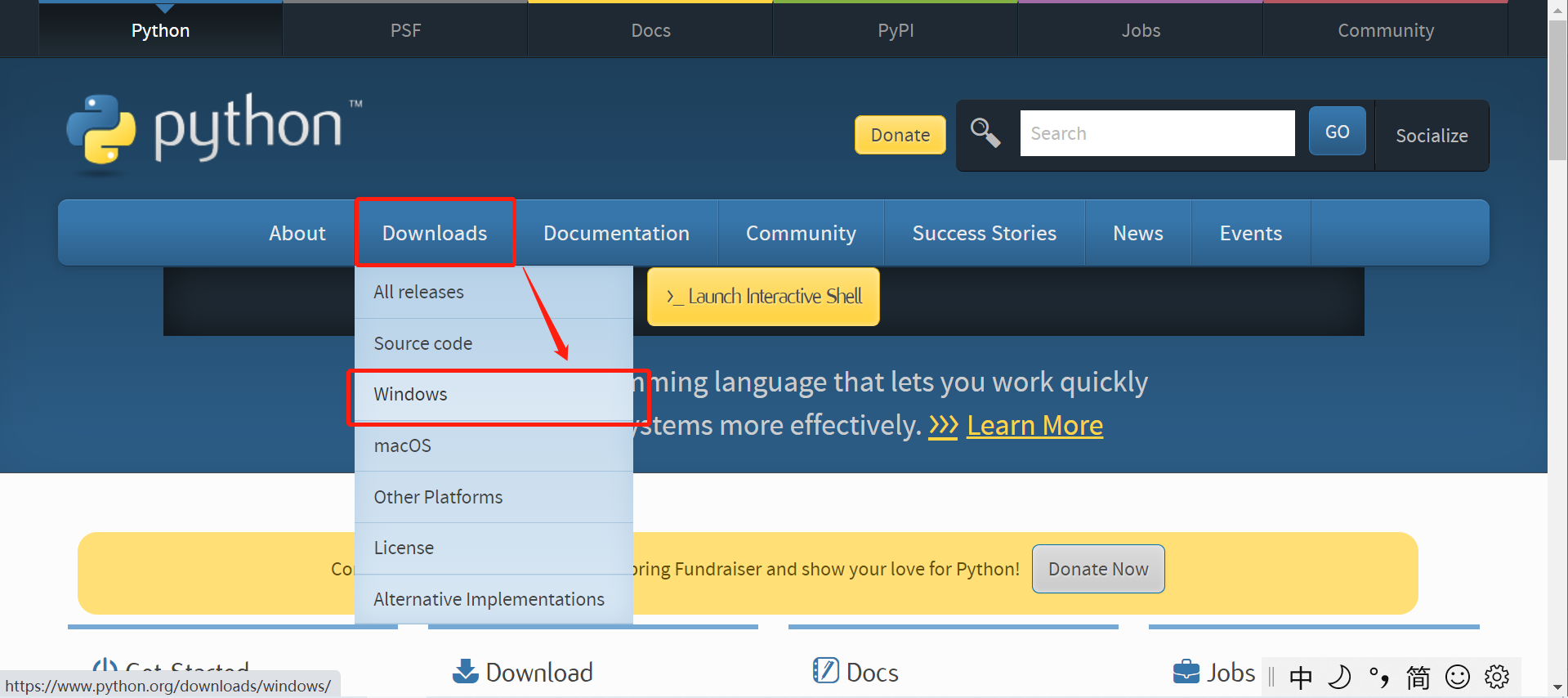
[实验十：综合实验——音频信号处理 27](#_Toc104038152)

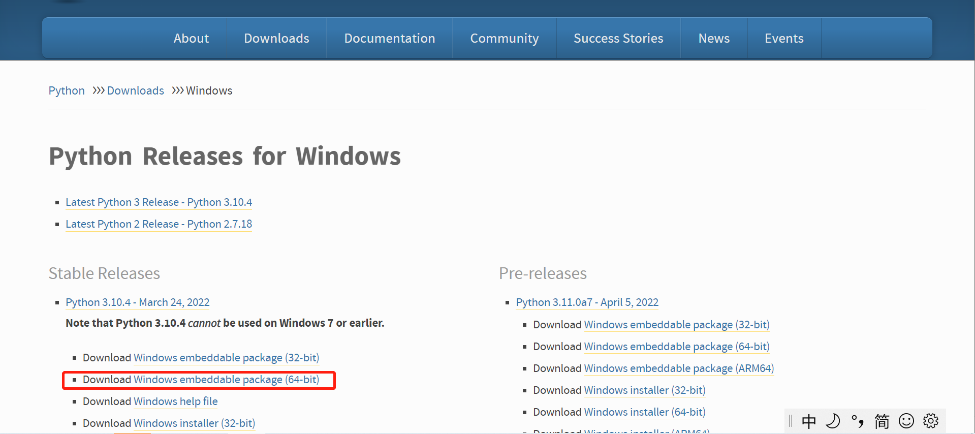
# 

# 一、python安装教程

## 1.1 python解释器

在Python官网（https://www.python.org）点击“Downloads”，在展开页面中选择需要下载的版本。以Windows系统的最新版本Python 3.10.4为例，如图1.1.1和1.1.2所示进行操作。



 图 1.1.1 python官网首页

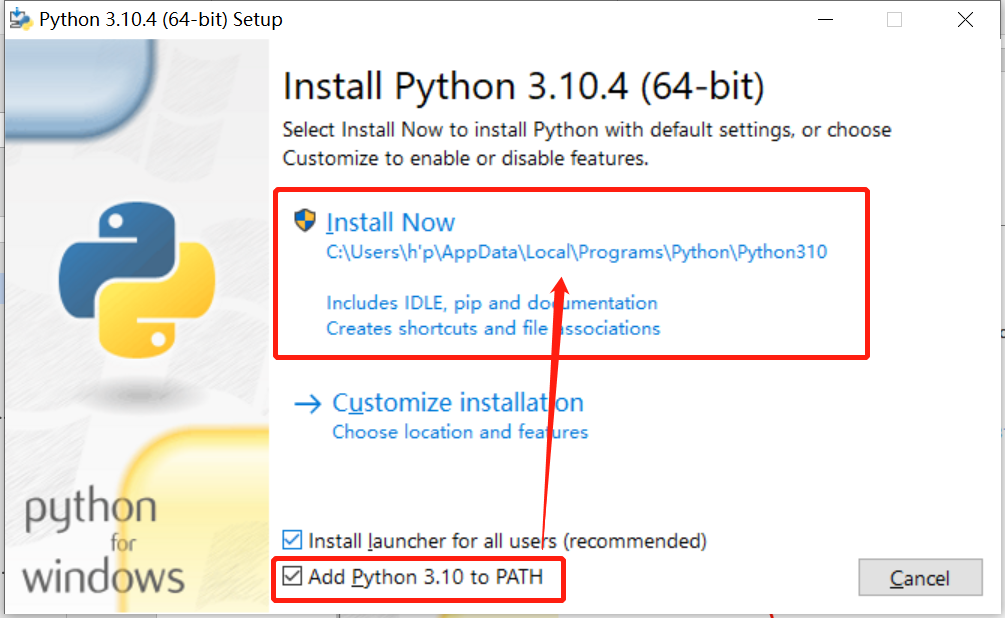
 图1.1.2 选择安装的python版本

图1.1.3 注意勾选“添加到路径”

完成后即安装成功

## 1.2 安装python第三方库

**（1）安装NumPy**

Python安装完成后，在命令行窗口输入以下命令，进入Python。

Python

输入以下命令进行安装。

from numpy import \*

**（2）安装matplotlib包**

在命令行窗口输入以下命令进行安装。

pip install matplotlib

出现如下图内容即安装成功。

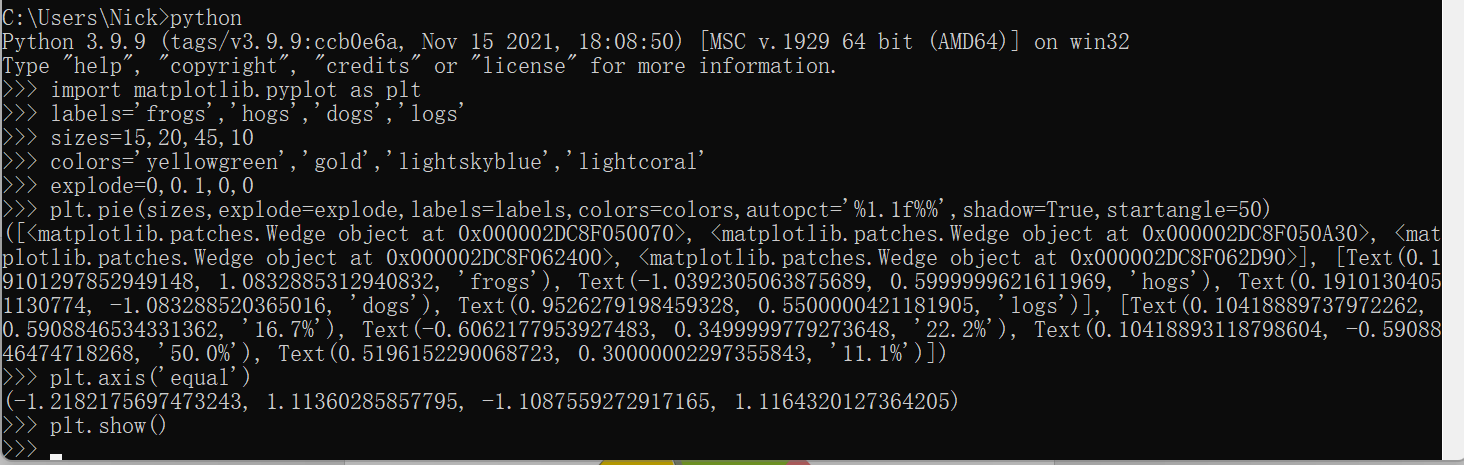


图1.2.1 matplotlib安装成功

**（3）安装scipy包**

在命令行窗口输入以下命令进行安装。

pip install scipy

## 1.3 Jupyter Notebook编辑器

**（1）介绍**

Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。

**（2）安装和使用方法**

pip install jupyter notebook #Python 2.x

pip3 install jupyter notebook #Python 3.x

当出现下图中内容时代表安装成功。

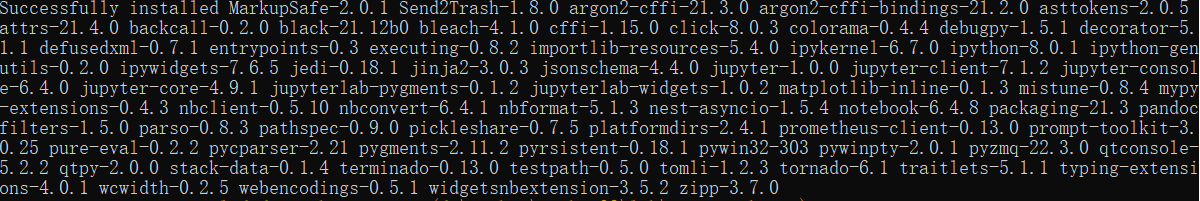


图1.3.1 Jupyter Notebook安装成功

另一种方法是通过安装Anaconda来安装Jupyter Notebook， Anaconda可以自动安装Jupter Notebook及其他工具，还有Python中超过180个科学包及其依赖项。

Anaconda在其官方网址https://repo.anaconda.com/archive/根据电脑系统和Python版本下载合适版本。

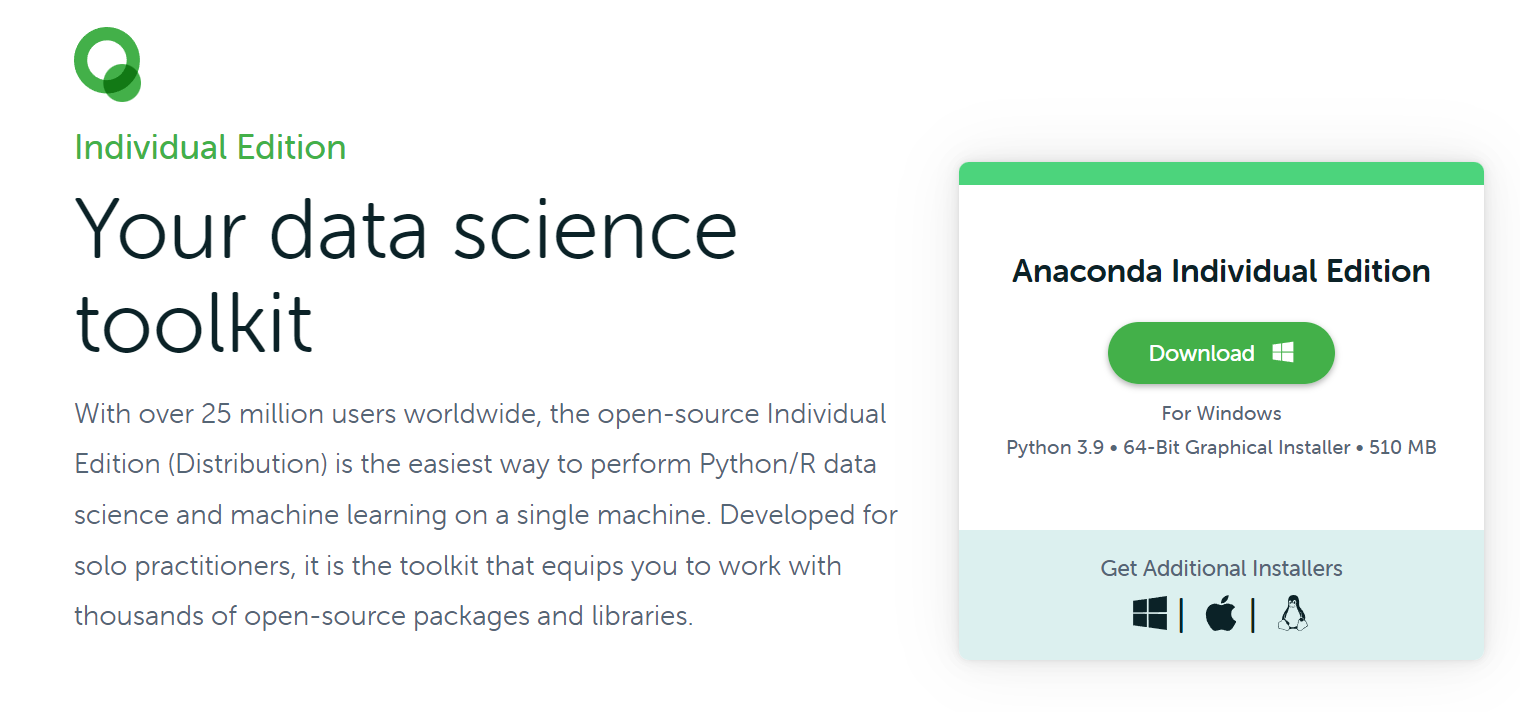


图1.3.2 安装Anaconda

下载完成后启动安装程序。安装过程均选用默认选项即可。

**（3）运行Jupyter Notebook**

在应用程序中找到Jupyter Notebook，单击打开，自动跳转网页，浏览器地址栏中默认地将会显示：http://localhost:8888。其中，“localhost”指的是本机，“8888”则是默认端口号。

第一次使用时，会直接出现如下主页面：



图1.3.3 Jupyter

“Upload”上传本机文件，“New”新建文件。

建立Python文件出现如下页面：

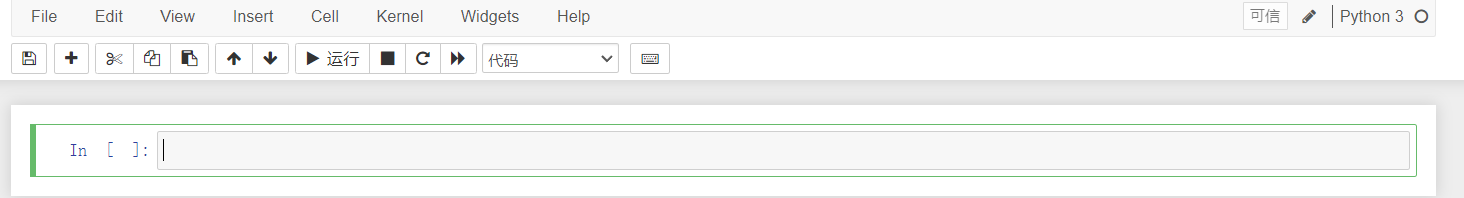
图1.3.4

图1.3.4就是NoteBook的交互界面,在这里我们可以对文档进行编辑、运行等操作。

更多详细内容参见Jupyter官方文档：https://docs.jupyter.org/en/latest/

## 1.4 Google colab在线运行

**(1)介绍**

Colaboratory 简称“Colab” ，是一种托管式 Jupyter 笔记本服务。用户无需进行设置，就可以直接使用，相当于Jupyter Notebook的在线版本，支持Google Dirve作为云端数据，提供了远端GPU作为运行的主机。

**(2)使用方法**

本课程的网络资源github仓库已经实现了程序的colab在线运行，可以在readme文件中点击超链接跳转，在线运行示例代码。

Github仓库链接：https://github.com/Chenying2000/py-SPT



图1.4.1 colab代码在线运行示例

# 二、python语言入门教程

## 2.1 python对象和变量

**（1）Python 标识符**

1、在 Python 里，标识符由字母、数字、下划线组成，但不能以数字开头。

2、Python 中的标识符是区分大小写的。

3、Python中的保留字不能用作常数或变数，或任何其他标识符名称。

**（2）行和缩进**

学习 Python 与其他语言最大的区别就是，Python 的代码块不使用大括号 来控制类，函数以及其他逻辑判断。Python 最具特色的就是用缩进来写模块。缩进的空白数量是可变的，但是所有代码块语句必须包含相同的缩进空白数量。

**（3）python变量类型**

**1 Python 数字**

数字数据类型用于存储数值，是不可改变的数据类型。

**2 Python字符串**

字符串或串(String)是由数字、字母、下划线组成的一串字符。Python的字串列表有2种取值顺序:

从左到右索引默认0开始的，最大范围是字符串长度少1。

从右到左索引默认-1开始的，最大范围是字符串开头。

**3 Python列表**

List（列表） 是 Python 中使用最频繁的数据类型，用 [ ] 标识。列表可以完成大多数集合类的数据结构实现。

**4 Python 元组**

元组（tuple）是另一个数据类型，用 () 标识，内部逗号隔开。元组不允许更新，相当于只读列表。

**5 Python 字典**

字典(dictionary)是除列表以外Python之中最灵活的内置数据结构类型。字典是无序的对象集合，字典是通过键来存取的，而不是通过偏移存取。字典用"{ }"标识。字典由索引(key)和它对应的值value组成。

**6 Python数据类型转换**

|  |  |
| --- | --- |
| [int(x [,base])](https://www.runoob.com/python/python-func-int.html) | 将x转换为一个整数 |
| [long(x [,base] )](https://www.runoob.com/python/python-func-long.html) | 将x转换为一个长整数 |
| [float(x)](https://www.runoob.com/python/python-func-float.html) | 将x转换到一个浮点数 |
| [complex(real [,imag])](https://www.runoob.com/python/python-func-complex.html) | 创建一个复数 |
| [str(x)](https://www.runoob.com/python/python-func-str.html) | 将对象 x 转换为字符串 |
| [tuple(s)](https://www.runoob.com/python/att-tuple-tuple.html) | 将序列 s 转换为一个元组 |
| [list(s)](https://www.runoob.com/python/att-list-list.html) | 将序列 s 转换为一个列表 |
| [set(s)](https://www.runoob.com/python/python-func-set.html) | 转换为可变集合 |
| [dict(d)](https://www.runoob.com/python/python-func-dict.html) | 创建一个字典。d 必须是一个序列 (key,value)元组。 |

表2.1-1 Python数据类型转换

**（4）python模块**

**1 模块介绍**

模块，可以理解为是对代码更高级的封装，这样可以方便其它程序或脚本导入并使用。模块能定义函数、类和变量，模块里也能包含可执行的代码。

**2 引入模块**

使用import语句为当前程序导入模块。格式形如：

import 模块名1[,模块名2[,... 模块名N]]

结合from 语句从模块中导入一个指定的部分到当前命名空间。语法如下：

from 模块名 import 成员名1[, 成员名2[, ... 成员名N]]

把一个模块的所有内容全都导入到当前的命名空间，使用如下声明：

from 模块名 import \*

## 2.2 DSP常用数值运算函数及功能

### 2.2.1 基于 NumPy的数组运算

**（1）创建数组**

**链接：**<https://numpy.org/doc/stable/reference/arrays.ndarray.html>

Numpy中的数组是一个值网格，所有类型都相同，并由非负整数元组索引。 维数是数组的排名; 数组的形状是一个整数元组，给出了每个维度的数组大小。

**案例：**常用的数组创建函数

>>> import numpy as np #导入numpy模块

>>> a = np.zeros((2,2)) # 创建全0数组

>>> b = np.ones((1,2)) # 创建全1数组

>>> c = np.random.random((2,2)) # 创建随机数组

>>> d = np.arrange(0,10,1) # 创建按从0到9，间隔为1的等差数列

>>> e = np.linspace (0,9,10) # 创建按从0到9，含有10个元素的等差数列

**（2） 数组访问**

**1基本索引（访问单元素）**

**链接：**<https://numpy.org/doc/stable/user/basics.indexing.html#basics-indexing>

数组的单元素索引的工作方式从0开始（matlab从1开始），并接受从数组末尾开始索引的负索引。

**案例：**通过索引访问数组中的指定元素

>>> x = np.arange(10) # x = [0 1 2 3 4 5 6 7 8 9]

>>> x[2]

输出：2

>>> x[-2]

输出：8

**2切片索引（访问多元素）**

**链接：**<https://numpy.org/doc/stable/user/basics.indexing.html#slicing-and-striding>

可以使用冒号（:）对numpy的数组进行切片。由于数组可能是多维的，因此必须为数组的每个维指定一个切片，基本的切片语法是i:j:k，其中i是开始索引，j是停止索引，k是步长

**案例：**访问多维数组中某一维度所有元素

>>> x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

>>> x[1:7:2]

输出：array([1, 3, 5])

**（3）算术运算：**

Numpy数组支持算术运算，包括加减乘除四则运算和简单的函数映射。

|  |  |
| --- | --- |
| \* | 乘法 |
| \*\* | 指数运算 |

### 2.2.2基于scipy的数值计算及应用

Scipy是开源的Python算法库和工具包。由线性代数、积分、插值、快速傅里叶变换、信号处理等子工具包组成。对于数字信号处理实验，我们只需要用到signal和fft两个子工具包，下表列出了本实验课程所需要用到的函数，更多详细信息参考官网信息。

**（1）傅里叶变换包（scipy.fft）**

**链接：**<https://scipy.github.io/devdocs/reference/fft.html>

|  |  |
| --- | --- |
| fft(x[, n, axis, norm, overwrite\_x, ...]) | 计算一维离散傅里叶变换 |
| ifft(x[, n, axis, norm, overwrite\_x, ...]) | 计算一维离散傅里叶逆变换 |

**（2）信号处理包（scipy.signal）**

**链接：**<https://scipy.github.io/devdocs/reference/signal.html>

|  |  |
| --- | --- |
| convolve(in1, in2) | 卷积两个n维数组 |
| correlate(in1, in2) | 互相关两个一维数组 |
| lfilter(b, a, x[, axis, zi]) | 使用 IIR 或 FIR 滤波器沿一维滤波 |
| bilinear(b, a[, fs]) | 使用双线性变换从模拟滤波器返回数字 IIR 滤波器 |
| firwin(numtaps, cutoff) | 使用窗函数法设计FIR滤波器 |
| freqs(b, a) | 计算模拟滤波器的频率响应 |
| freqz(b, a) | 计算数字滤波器的频率响应 |

### 2.2.3基于matplotlib的数据可视化

Matplotlib的全称是mathematic plot library，其功能是根据数据创建各种各样的可视化内容，这些内容可以是静态的、动态的，可交互的。

**链接：**https://matplotlib.org/。

本实验课程主要用到Matplotlib的子库pyplot，其代码风格贴近Matlab。

**案例：**画出离散图形（stem）和连续图形（plot）代码模板

>>> import matplotlib.pyplot as plt

>>>fig,ax = plt.subplots()#创建画布，ax是画布名称，可自定义

>>>ax.stem(x, y)#

>>>fig.savefig('图片名称.png',dpi=500)#dpi是图片分辨率

>>>fig,ax1 = plt.subplots()

>>>ax1.plot(x, y)

>>>fig.savefig('保存图片名称.png',dpi=500)

# 三、数字信号处理实验

## 实验一：系统响应以及系统稳定性

**一、实验目的**

(1) 握求系统响应的方法

(2) 握时域离散系统的时域特性

(3) 分析、观察及检验系统的稳定性。

**二、实验原理与方法**

在时域中，已知输入信号，可以由差分方程、单位脉冲响应或系统函数求出系统对于该输入信号的响应，本实验仅在时域求解。

在计算机上适合用递推法求差分方程的解，最简单的方法是采用 python第三方库scipy中的函数 lfilter 函数。也可以用python第三方库scipy中的函数 convolve 函数计算输入信号和系统的单位脉冲响应的线性卷积，求出系统的响应。

系统的时域特性指的是系统的线性时不变性质、因果性和稳定性。

注意：在以下实验中均假设系统的初始状态为零。

**三、相关python函数及用法介绍**

(1)Signal.lfilter（b，a）：使用 IIR 或 FIR 滤波器沿一维过滤数据，b是分子系数向量，a是分母系数向量

(2)Signal.freqz（b，a）：给定数字滤波器的 M 阶分子b和 N 阶分母a，计算其频率响应。

**四、实验内容及步骤**

(1) 编制程序，包括产生输入信号、单位脉冲响应序列的子程序，以及用 signal.lfilter 函数或 signal.convolve函数求解系统输出响应的主程序。程序中要有绘制信号波形的功能。

(2) 给定一个低通滤波器的差分方程为



输入信号

和

1. 别求出和的系统响应，并画出其波形。
2. 求出系统的单位脉冲响应，画出其波形。

(3)给定系统的单位脉冲响应为





用线性卷积法求分别对系统和 的输出响应，并画出波形。

(4) 给定一谐振器的差分方程为



令, 谐振器的谐振频率为。

① 用实验方法检查系统是否稳定。输入信号为 u(n) 时，画出系统输出波形

② 给定 输入信号为



求出系统的输出响应，并画出其波形

**四、 思考题**

**(1)** 如果输入信号为无限长序列，系统的单位脉冲响应是有限长序列，可否用线性卷 积法求系统的响应？如何求？

**(2)** 如果信号经过低通滤波器，把信号的高频分量滤掉，时域信号会有何变化？用实验内容 (2) 的结果进行分析说明

**五、实验报告要求**

(1) 简述在时域求系统响应的方法

(2) 简述通过实验判断系统稳定性的方法

(3) 对各实验所得结果进行简单分析和解释

(4) 简要回答思考题

(5) 打印程序清单和要求的各信号波形

## 实验二：时域采样

**一、实验目的**

时域采样理论是数字信号处理中的重要理论。要求掌握模拟信号时域采样前后频谱的变化，以及如何选择采样频率才能使采样后的信号不丢失信息。

**二、实验原理与方法**

时域采样定理的要点是： 对模拟信号  进行时域等间隔理想采样，形成的采样信号的频谱原模拟信号频谱以采样角频率为周期进行周期延拓。公式为



采样频率必须大千等于模拟信号最高频率的两倍以上，才能使采样信号的频谱 不产生频谱混叠。

通过上面的分析可以看到：时域采样频谱周期延拓。

**三、相关python函数及用法介绍**

（1）np.arange（start,end,1/f）：在给定的时间间隔（start，end）内返回均匀间隔（1/f）的值。

（2）fft.fftfreq(N0,1/f)：返回离散傅里叶变换采样频率。

**四、实验内容及步骤**

时域采样理论的验证。给定模拟信号



式中,,, 它的幅频特性曲线如图()所示

现用 FFT求该模拟信号的幅频特性，以验证时域采样理论。

按照的幅频特性曲线，选取三种采样频率， 观测时间选

为使用 FFT, 首先用下面的公式产生时域离散信号，对三种采样频、采样序列按顺序用、、表示。



因为采样频率不同，得到的 、、的长度不同，长度（点数）用公式计算 FFT 的变换点数为序列长度不够64的尾部加零。



式中，k代表的频率为



要求： 编写实验程序，计算、、的幅度特性，并绘图显示。观察分析频谱混叠失真。

**五、思考题**

如果序列的长度为 M, 希望得到其频谱在上的N点等间隔采样，当时，如何用一次最少点数的 DFT 得到该频谱采样？

**六、实验报告及要求**

(1) 运行程序，打印要求显示的图形

(2) 分析比较实验结果，简述由实验得到的主要结论

(3) 简要回答思考题

(4) 附上程序清单和有关曲线

## 实验三：频域采样

**一、实验目的**：

频域采样理论是数字信号处理中的重要理论。要求掌握频域采样会引起时域周期延拓的概念，以及频域采样定理及其对频域采样点数选择的指导作用。

**二、实验原理及方法：**

① 对信号的频谱函数在上等间隔采样N点，得：

则N点IDFT得到的序列就是原序列 以N为周期进行周期延拓后的主值区 序列，公式为：



②由上式可知，频域采样点数N必须大于等于时域离散信号的长度M（）, 才能使时域不产生混叠，则 N点IDFT得到的序列就是原序列。如果,比原序列尾部多了个零点 ；此外，如果长度，则 发生了时域混叠失真，而且的长度也比的长度短，因 此，和不相同。

通过上面的分析可以看到：频域采样时域信号周期延拓，结合实验二我们发现，时域采样和频域采样具有一定的对偶性，因此，将两个实验放在一起，便于大家理解。

**三、相关python函数及用法介绍：**

（1）np.arange（start,end,1/f）：在给定的时间间隔（start，end）内返回均匀间隔（1/f）的值。

（2）fftshift（）：函数将零频率分量移动到频谱的中心

**四、实验内容及步骤**

频域采样理论的验证，给定信号如下：



编写程序分别对频谱函数在区间上等间隔采样 32 点和 16 点， 得到和





再分别对和进行 32 点和 16 IFFT, 得到和





分别画出，和的幅度谱，并绘图显示 ，和的波形，进行对比和分析，验证总结频域采样理论。

**五、思考题：**

结合实验二，试简述时域采样与频域采样的关系

**六、实验报告及要求**

(1) 运行程序，打印要求显示的图形

(2) 分析比较实验结果，简述由实验得到的主要结论

(3) 简要回答思考题

(4) 附上程序清单和有关曲线

## 实验四：零极点分析

**一、实验目的**

（1）理解用系统函数的极点分布分析系统的因果性和稳定性

（2）学会利用系统的极零点分布分析系统的频率响应特性

（3）掌握由系统差分方程绘制其零极点分布图的python方法

**二、实验原理与方法**

（1）利用系统函数的极点分布分析系统的因果性和稳定性

系统稳定要求: ，这正是存在的条件，对照z变换与傅里叶变换的关系可知，系统稳定的条件是的收敛域包含单位圆。如果系统因果 且稳定，收敛域包含点和单位圆，那么收敛域可表示为



这样 H(z) 的极点集中在单位圆的内部。则具体系统的因果性和稳定性可由系统函数 H(z) 极点分布和收敛域确定。

（2）学会利用系统的极零点分布分析系统的频率响应特性

系统稳定时，将代入频率响应函数，得到：



在Z平面上，用极坐标标记零点向量和极点向量如图（3.1）

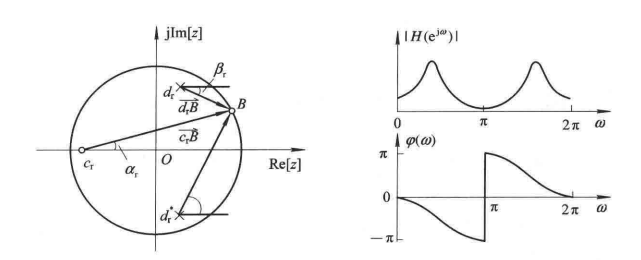


图 3.1 频率响应的几何表示

根据零极点的分布后，可以确定零极点位置对系统特性的影响。当B点转到极点附近时，极点向量长度最短，幅度特性可能出现峰值，且极点愈靠近单位圆，峰值愈高愈尖锐。如果极点在单位圆上，则幅度特性为,系统不稳定。对于零点，情况相反，且当零点处在单位圆上时，谷值为零。

总结以上结论 极点位置主要影响频响的峰值位置及尖锐程度，零点位置主要影响频响的谷点位置及形状。

**三、相关python函数及用法介绍**

（1）画单位圆：

circle = Circle(xy=(0.0,0.0),radius=1,alpha=0.5,facecolor='white',ec='cornflowerblue',linestyle='--',)

ax.add\_patch(circle)

（2）零极点表示法：

p1 = 0.9\*np.exp(-2\*np.pi\*0.3j)

**四、实验内容及步骤**

（1）设一阶系统的差分方程为



试画出系统的幅频特性并用几何法分析其幅度特性。

（2）已知，试编写python代码画出系统的幅频特性。

**五、思考题**

总结分析系统函数的极点分布是如何影响系统的因果性和稳定性的？

**六、实验报告及要求**

(1) 运行程序，打印要求显示的图形

(2) 简要回答思考题

## 实验五：相关函数和线性卷积

**一、 实验目的**

(1) 学习如何通过python计算序列的自相关和互相关；

(2) 学习如何通过不同的方法求相关函数

(3) 掌握各方法的原理，分析各方法所得的结果，了解相关的应用

**二、实验原理**

相关函数是描述随机信号在任意两个不同时刻和的取值之间的相关程度。

互相关函数是描述随机信号 x(n) y(n) 在任意两个不同时刻和的取值之间的相关程度，任意 x(n) y(n) 相关定义为



利用线性卷积来求相关函数 ，设原函数是，则 自相关函数也可以由下式计算 ：



信号和的互相关函数可以表示为（其中“\*”表示卷积运算）



互相关函数给出了在频域内两个信号是否相关的一个判断指标，把两测点之间信号的互谱与各自的自谱联系了起来。

**三、相关python函数及用法介绍**

（1）signal.correlate（a，b）：互相关两个一维数组

（2）signal.convolve（a，b）：卷积两个一维数组

**四、 实验内容及步骤**

给定两个信号：，其中 ，取整数

（1）利用定义编写求两个信号互（自）相关的程序。计算 ：，，，；分析两者之间的关系。

（2）比较编写求线性卷积和的程序，比较与，以及与

（3）利用傅里叶变换求：， ，，

（4）信号，，是一个白噪声，分别计算时的，，，

**五、 思考题**

（1）通过实验分析相关是否具有交换律？和有什么关系？

（2）分析总结相关和线性卷积的关系，并编写利用线性卷积求相关的程序

（3）分析实验结果有什么特点，思考总结相关的应用

**六、 实验报告要求**

（1）对各实验所得结果进行简单分析和解

（2）打印程序清单和实验所得结果的各信号波形

## 实验六：圆周卷积和线性卷积

**一、实验目的**

（1）利用不同原理（方法）编写实现圆周卷积运算的python程序。

（2）利用和求两序列的线性卷积。

（3）掌握圆周卷积与线性卷积之间的关系，编写程序，并通过实验验证两者关系。

**二、实验原理**

**（a）线性卷积**

线性时不变系统输入、 输出的关系为：当系统输入序列为，系统的单位脉冲响应，输出序列为这两个序列的线性卷积：



**（b）圆周卷积**

若序列长度为 的长度为L, 的长度为P,为这两个序列的N点圆周卷积， 则为：



圆周卷积的长度。具体步骤为:

（1）补零使两 列补零后长度均为N点。

（2）其中一个序列以N为周期进行周期延拓

（3）周期延拓后翻褶

（4）对应点相乘求和，然后取主值序列，得到

（5）移位1点。对应点相乘求和，然后取主值序列，得到。以此类推可求

**（c）圆周卷积与线性卷积关系**

长度为L的序列与长度为P的序列的N点圆周卷积等于这两序列的线性卷积以N周期进行周期延拓的主周期；当时,两者相等。

**（d）利用圆周卷积性质，基于和求圆周卷积和线性卷积**





**三、相关python函数及用法介绍**

（1）signal.fftconvolve：以fft算法卷积两个一维数组

（2）可以通过设置数组和引用线性卷积函数实现圆周卷积

**四、实验内容及步骤**

（1）已知两个有限长序列：





编写程序用定义法分别计算这两个序列的线性卷积和5点，6点，8点，10点圆周卷积，分析观察实验所得结果。

（2）编写和扩展函数程序，利用和扩展函数求（1）中的5,6 8,10 点圆周卷积

（3）求序列 和 求两序列的线性卷积；用变换实现 64点，128 点的圆周卷积。

（4）利用 signal.convolve函数及线性卷积和圆周卷积的关系 ，求练习（1） 练习（2）中的圆周卷积。

（5) 编写一个计算圆周卷积的函数，并利用调用函数的方法求解两序列不同点数的圆周卷积。

**五、思考题**

（1）分析线性卷积与圆周卷积之间的关系

（2）并分析它的应用 圆周卷积性质的应用

**六、实验报告要求**

（1）观察分析实验所得结果

（2) 印程序清单和实验所运行的结果图。

## 实验七：用FFT对信号做频谱分析

**一、实验目的**

学习用 FFT 对连续信号和时域离散信号进行谱分析的方法，了解可能出现的分析误差及其原因，以便正确应用 FFT。

**二、实验原理**

FFT 对信号作频谱分析是学习数字信号处理的重要内容。经常需要进行谱分析的信号是模拟信号和时域离散信号。对信号进行谱分析的重要问题是频谱分辨率D和分析误差。频谱分辨率直接和 FFT 的变换区间N有关，因为 FFT 能够实现的频率分辨率是，因此要求。可以根据此式选 FFT的变换区间N，误差主要来自于用 FFT 作频谱分析时，得到的是离散谱，而信号（周期信号除外）是连续谱，只有当N较大时，离散谱的包络才能逼近于连续谱 ，因此N要适当选择大一些。

周期信号的频谱是离散谱，只有用整数倍周期的长度作 FFT, 得到的离散谱才能代表周期信号的频谱。如果不知道信号周期，可以尽量选择信号的观察时间长一些。

**三、相关python函数及用法介绍**

(1)fft.fft()：计算一维离散傅立叶变换

**四、实验步骤及内容**

(1) 对以下序列进行谱分析：







选择 FFT 的变换区间N为8和16两种情况进行频谱分析。分别打印其幅频特性曲线，并进行对比、分析和讨论。

(2) 对以下周期序列进行谱分析





选择 FFT 的变换区间N为8和16两种情况分别对以上序列进行频谱分析。分别打印 其幅频特性曲线，并进行对比、分析和讨论

**四、思考题**

(1)对于周期序列，如果周期不知道，如何用 FFT 进行谱分析？

(2) 如何选择 FFT 的变换区间？ （包括非周期信号和周期信号）

(3) N=8 时，和的幅频特性会相同吗？为什么？呢？

**五、实验报告要求**

(1) 完成各个实验任务和要求，附上程序清单和有关曲线

(2) 简要回答思考题

## 实验八：IIR数字滤波器设计及软件实现

**一、实验目的**

(1) 熟悉用双线性变换法设计 IIR 数字滤波器的原理与方法；

(2) 学会调用python信号处理工具箱scipy库中滤波器设计函数设计各种 IIR 数字滤波器，学会根据滤波需求确定滤波器指标参数。

(3) 掌握 IIR 数字滤波器的python实现方法。

(4) 通过滤波器输入、输出信号的时域波形及其频谱，建立数字滤波的概念

**二、实验原理**

设计IIR 滤波器一般采用间接法（脉冲响应不变法和双线性变换法），应用最广泛的是双线性变换法。

基本设计过程：

1. 将给定的数字滤波器的指标转换成过渡模拟滤波 器的指标；
2. 设计过渡模拟滤波器；
3. 将过渡模拟滤波器系统函数转换成数字滤波器的系统函数。

**三、相关python函数及用法介绍**

（1）signal.residue(b, a)：该函数是scipy库中提供的，输入模拟滤波器系数向量，可以返回极点余数、极点和系数k

（2）signal.invresz（r，p，k）：该函数可以将r，p，k参数表示的数字滤波器转换为部分分式形式的系数向量。

（3）signal.bilinear(b,a,fs=1.0)：该函数是scipy库提供的模数滤波器转换函数，原理正是双线性变换法，是bilinear\_IIR()函数的核心。

**四、实验内容与步骤**

（1）利用脉冲响应不变变换法，把下列模拟域的模拟滤波器转换为数字滤波器，采样周期

|  |
| --- |
|  |

（2）利用脉冲响应不变变换法，该模拟域的模拟滤波器转换为数字滤波器，采样周期

|  |
| --- |
|  |

**四、思考题**

比较分析两种设计方法的结果有何不同，分析两种设计方法的优劣

**五、 实验报告要求**

(1) 简述实验目的及原理

(2) 画出实验主程序框图，打印程序清单

(3) 简要回答思考题

## 实验九：FIR数字滤波器设计与软件实现

**一、实验目的**

(1) 掌握用窗函数法设计 FIR 数字滤波器的原理和方法

(2) 掌握用等波纹最佳逼近法设计 IR 数字滤波器的原理和方法。

(3) 掌握 FIR 滤波器的快速卷积实现原理

(4) 学会调用python函数设计与实现 FIR 滤波器。

**二、相关python函数及用法介绍**

（1）signal.firwin(numtaps, cutoff, width=None, window='hamming')使用窗函数法设计 FIR 滤波器。该函数计算有限脉冲响应滤波器的系数。滤波器将具有线性相位；如果numtaps为奇数则为TypeI，如果*numtaps为偶数则为*Type II。

**三、实验内容及步骤**

（1）用窗函数法设计一个偶对称的线性相位FIR低通滤波器，给定通带截止频率，阻带截止频率为，阻带衰减为。

（2）用窗函数法设计一个偶对称的线性相位FIR高通滤波器，给定通带截止频率，阻带截止频率为，阻带衰减为。

（3）用窗函数法设计一个偶对称的线性相位FIR带通滤波器，给定下阻带截止频率，上阻带截止频率，通带下截止频率为，通带上截止频率为，阻带最小衰减为，抽样频率为。

**四、思考题**

简述窗函数法设计FIR滤波器的原理与步骤

**五、实验报告要求**

(1) 简述实验目的及原理

(2) 画出实验主程序框图，打印程序清单

(3) 简要回答思考题

## 实验十：综合实验——音频信号处理

**一、实验目的**

复习巩固本实验课程所学习的信号产生、信号采样与恢复、滤波器设计与信号滤波等基本内容，具备可以熟练运用python解决基本数字信号处理问题的能力。

**二、实验内容及步骤**

（1）原始信号输入：读取.wav音频文件，或仿真产生原始信号。

（2）信号采样：根据Nyquist采样定理，当采样率时，采样信号的频谱未发生混叠，因此选取三个采样率：=4410Hz，=3675Hz，=3150Hz。

（3）频谱分析：对采样产生的三个信号进行快速傅里叶变换，得到的频谱在范围内应该与原始信号频谱相同。

（4）数字滤波：采用Butterworth、Chebyshev1、Chebyshev2和Bessel四种数字滤波器，分别对原始信号和采样信号进行低通、高通、带通滤波，得到滤波后的信号及其频谱。

（5）内插恢复：将分别经过低通、高通和带通的三个采样信号进行内插恢复，与经过滤波的原始信号进行对比；并将内插恢复的信号生成.wav文件，通过音频输出设备播放。

**三、思考题**

（1）结合奈奎斯特定律，分析步骤（2）中不同采样率条件下的实验结果。

（2）比较步骤（4）数字滤波中不同滤波器的滤波性能好坏。

（3）比较原始信号与内插恢复信号，客观评价本次实验课程的学习效果。

**四、实验报告要求**

（1）画出实验主要程序框图，打印程序清单。

（2）简要回答思考题。

（3）总结回顾本次数字信号处理实验的收获，并反思自己的不足。