数字信号处理

Digital signal processing

第二章离散时间信号与系统

-2.3.3常系数差分方程

华中科技大学 人工智能与自动化学院

- School of AI & Automation.
- Huazhong University of Science and Technology

蔡超

caichao@hust.edu.cn

回忆离散时间线性时不变系统

定义　**同时满足线性性和时不变性的离散时间系统。**

即：　$x(n) \longrightarrow T[\ ] \longrightarrow y(n)$

同时满足：　$y(n) = T[x(n)]$

1. **线性**

$$y_1(n) = T[x_1(n)], y_2(n) = T[x_2(n)],$$

$$\begin{aligned} y(n) &= T[a_1x_1(n) + a_2x_2(n)] \\ &= a_1T[x_1(n)] + a_2T[x_2(n)] \\ &= a_1y_1(n) + a_2y_2(n) \end{aligned}$$

2.时不变性　$T[x(n-n_0)] = y(n-n_0)$

回忆离散时间线性时不变系统

- 线性时不变离散系统任意激励下的响应 $y(n)$ 与单位脉冲响应 $h(n)$ 之间的关系

$$
\begin{aligned}
y(n) &= T[x(n)] = T[\sum_{m=-\infty}^{\infty} x(m)\delta(n-m)] \\
&= \sum_{m=-\infty}^{\infty} x(m)T[\delta(n-m)] = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \\
&\triangleq x(n) * h(n)
\end{aligned}
$$

2.3.3 线性常系数差分方程

- 线性非移变系统可以用线性常系数差分方程来描述。差分方程是由函数序列的差分来表示的。一个函数序列的一阶后向差分表示为:

$$\nabla y(n) = y(n) - y(n-1)$$

二阶后向差分表示为

$$\nabla^2 y(n) = \nabla[y(n) - y(n-1)] = y(n) - 2y(n-1) + y(n-2)$$

引人单位延迟算子D，即 $Dy(n)=y(n-1)$。于是

$$\nabla y(n) = y(n) - y(n-1) = y(n) - Dy(n) = (1-D)y(n)$$

因此有 $\nabla = 1 - D$

二阶后向差分可表示为

$$\nabla^2 y(n) = (1-D)^2 y(n) = (1-2D+D^2)y(n)$$
$$= y(n) - 2y(n-1) + y(n-2)$$

类似地k阶后向差分表示为

$$\nabla^k y(n) = (1-D)^k y(n)$$

因此，按二项式定理将(1-D)^{k}展开后，便可得到k阶差分的表示式。

差分方程是描述函数序列差分之间关系的方程。例如，对于一个二阶差分方程：

$$\nabla^2 y(n)+\nabla y(n)+2=0$$

将 $\nabla=1-D$ 代入上式，得到

$$(1-D)^2 y(n)+(1-D)y(n)+2=0$$

展开后得

$$y(n)-\frac{3}{2}y(n-1)+\frac{1}{2}y(n-2)+1=0$$

这就是一个二阶线性常系数差分方程。

线性常系数差分方程

线性时不变系统

有限维 $\mathrm{LTI}(Linear\ \ Time\ Invariant)$ 离散时间系统可用线性常系数差分方程（*linear constant coefficient different equation*）表示

$$\sum_{k=0}^{N} d_k y(n-k) = \sum_{k=0}^{M} p_k x(n-k)$$

或

$$y(n) = -\sum_{k=1}^{N} \frac{d_k}{d_0} y(n-k) + \sum_{k=0}^{M} \frac{p_k}{d_0} x(n-k)$$

$\max\left(M,N\right)$ 称为差分方程的阶数

由实际问题直接得到差分方程

例如：

$y(n)$表示一个国家在第n年的人口数

a(常数)：出生率

b(常数)：死亡率

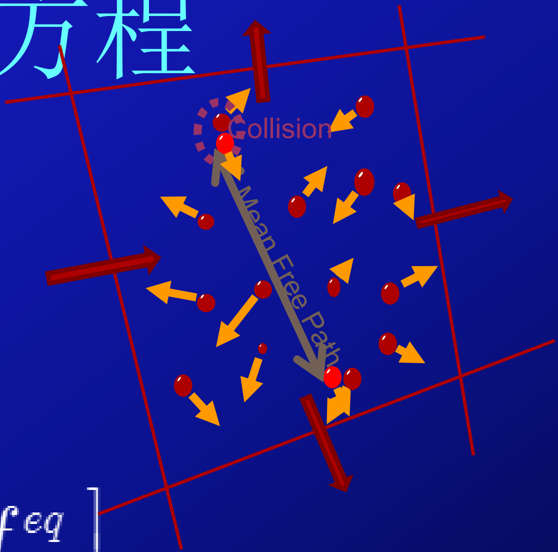
设$x(n)$是国外移民的净增数

则该国在第$n+1$年的人口总数为：

$$y(n+1)=y(n)+ay(n)-by(n)+x(n)$$

$$=(a-b+1)y(n)+x(n)$$

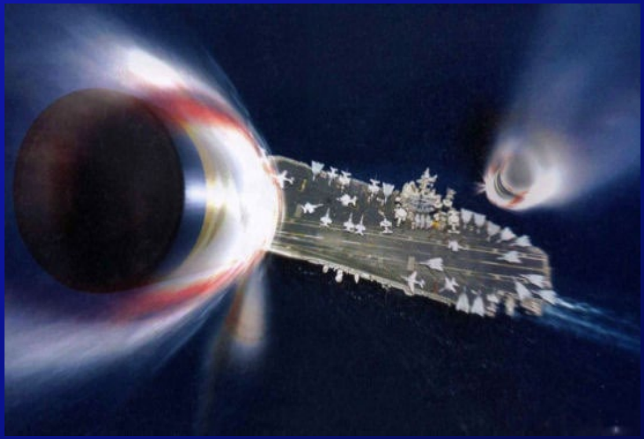
由微分方程导出差分方程

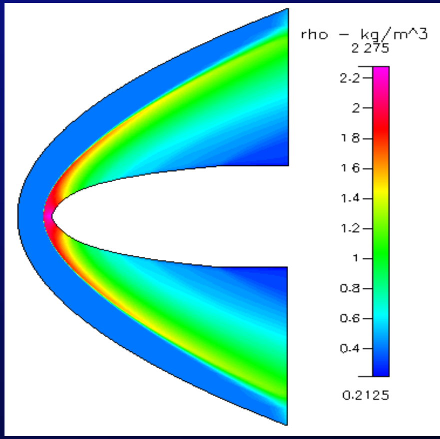


当前仍有大量的方程求解问题：

比如气体动理学模型（Boltzmann方程）

$$\partial_t f + \xi \cdot \nabla f = \Omega \equiv -\frac{1}{\tau}\left[f - f^{eq} \right]$$





线性常系数差分方程

定理2.3.1 设

$$y_1(n), y_2(n), \cdots, y_k(n)$$

是N阶常系数齐次线性差分方程

$$d_0 y(n) + d_1 y(n-1) + \ldots + d_N y(n-N) = 0$$

特解顾名思义就是一个特殊的解，微分方程可能还有别的解

的k个**特解**，则线性组合

$$y(n) = C_1 y_1(n) + C_2 y_2(n) + \cdots + C_k y_k(n)$$

也是该差分方程的解，其中 $C_1, C_2, \cdots, C_k$ 为任意常数.

定理2.3.2 *N*阶常系数齐次线性差分方程一定存在*N*个线性无关的特解．若

$$y_1(n), y_2(n), \cdots, y_N(n)$$

是方程

$$d_0 y(n) + d_1 y(n-1) + \ldots + d_N y(n-N) = 0$$

的*N*个线性无关的解，则方程 的通解为

我们往往要根据系统在初始时刻所处的状态对差分方程附加一定的条件，这种附加条件称为初始条件,满足初始条件的解称为特解.

通解是指满足这种形式的函数都是微分方程的解,有很多个。

$$Y = C_1 y_1(n) + C_2 y_2(n) + \cdots + C_N y_N(n),$$

其中 $C_1, C_2, \cdots, C_N$ 为任意常数．

通解是没有初始条件下的解,特解则是有初始条件限制.例如y'=0的通解就是y=C，C是常数，如y=0就是该微分方程的特解.

定理2.3.3 *N*阶非齐次线性差分方程

$$d_0 y(n) + d_1 y(n-1) + \ldots + d_N y(n-N) = f(x(n))$$

对应的齐次方程

$$d_0 y(n) + d_1 y(n-1) + \ldots + d_N y(n-N) = 0$$

的通解与它自己本身的一个特解之和构成该方程的全解，即全解等于

$$Y = C_1 y_1(n) + C_2 y_2(n) + \cdots + C_N y_N(n) + y^*(n),$$

其中 $y^*(t)$ 是它自己本身的一个特解.

解法

1.迭代法

逐次代入求解， 概念清楚， 比较简便， 适用于计算机，缺点是不易得出通式解答。

2.时域经典法：齐次解 + 特解；

求解过程比较麻烦

自由响应 强迫响应

3. z变换法→反变换→$y(n)$

相对简单，通用，需要用到留数定理

1. 迭代法

解差分方程的基础方法

差分方程本身是一种递推关系。

缺点：得不到$y(n)$输出序列的解析式

例1：已知常系数线性差分方程

$$y(n)-ay(n-1)=x(n)$$

若边界条件

$$y(-1)=0$$

求其单位抽样响应。

解：令输入$x(n)=\delta(n)$，则输出$y(n)=h(n)$，

又已知$y(-1)=0$

由$y(n)=ay(n-1)+x(n)$，得

$$y(0)=ay(-1)+x(0)=1$$

$$y(1)=ay(0)+x(1)=a$$

$$y(2)=ay(1)+x(2)=a^2$$

$$y(3)=ay(2)+x(3)=a^3$$

$$\vdots$$

$$y(n)=a^n,\quad n\geq 0$$

由$y(n-1)=\frac{1}{a}[y(n)-x(n)]$，得

$$y(-2)=\frac{1}{a}[y(-1)-x(-1)]=0$$

$$y(-3)=\frac{1}{a}[y(-2)-x(-2)]=0$$

$$\vdots$$

$$y(n)=0,\quad n\leq -1$$

$$\therefore h(n)=y(n)=a^n u(n)$$

2时域经典法：齐次解+特解；

全解（total solution）的计算

自由响应：也称固有响应，由系统本身特性决定，与外加激励形式无关。对应于齐次解。
强迫响应：形式取决于外加激励。对应于特解。这种分解法是从数学上理解的，没有实际的物理意义。

解的形式：$y(n)=y_c(n)+y_p(n)$

$y(n)$：全解（*total solution*）

$y_c(n)$：方程$\sum_{k=0}^{N} d_k y(n-k)=0$的解，称为齐次解（通解）（*complementary solution*）

$y_p(n)$：方程$\sum_{k=0}^{N} d_k y(n-k)=\sum_{k=0}^{M} p_k x(n-k)$，$x(n)\neq 0$的一个解，

称为特解（*particular solution*）

零输入响应：没有外加激励信号的作用，只由起始状态（起始时刻系统储能）所产生的响应（注意它不等同于自由响应，自由响应只和系统有关，而零输入响应和系统及初始储能有关），这种响应随时间按指数规律衰减。
零状态响应：不考虑原始时刻系统储能的作用（起始状态等于零），由系统的外加激励信号产生的响应。

（2.1）对应齐次差分方程通解的解法：

齐次差分方程：

$$\sum_{k=0}^{N} d_k y(n-k) = \sum_{k=0}^{N} d_k D^k y(n) = 0;$$

则有特征方程：

$$\sum_{k=0}^{N} d_k \lambda^{N-k} = d_0\lambda^N + d_1\lambda^{N-1} + \cdots + d_{N-1}\lambda + d_N = 0$$

得到了齐次方程的N个特解
$y(n) = \lambda_k^n, k = 1,2,\ldots,N$

$\sum_{k=0}^{N} d_k \lambda^{N-k}$称为离散时间系统的特征多项式（*characteristic polynomial*）

为什么呢？

设上式的根为λ_1，$\lambda_2,\cdots\lambda_N$，则齐次解的一般形式为

$$y_c(n) = \alpha_1\lambda_1^n + \alpha_2\lambda_2^n + \cdots + \alpha_N\lambda_N^n$$

如果有L个重根，则齐次解为

$$y_c(n) = \alpha_1\lambda_1^n + \alpha_2 n\lambda_1^n + \alpha_3 n^2\lambda_1^n + \cdots + \alpha_L n^{L-1}\lambda_1^n + \alpha_{L+1}\lambda_2^n + \cdots + \alpha_N\lambda_{N-L}^n$$

设$y(n)=\lambda_1^n$为齐次方程

$$\sum_{k=0}^{N} d_k y(n-k)=0$$的一个解,

当然也就是

$$\sum_{k=0}^{N} d_k y(N-k)=0$$的一个解。

代入方程得：

$$\sum_{k=0}^{N} d_k y(N-k)=\sum_{k=0}^{N} d_k \lambda_1^{N-k}=0$$

即：

$$\sum_{k=0}^{N} d_k \lambda_1^{N-k}=$$

$$d_0\lambda_1^N+d_1\lambda_1^{N-1}+\cdots+d_{N-1}\lambda_1+d_N=0$$

所以：λ_1是该多项式的一个根。

反之：设λ_1是多项式

$$d_0\lambda^N+d_1\lambda^{N-1}+\cdots+d_{N-1}\lambda+d_N=0$$

的一个根，则

$$d_0\lambda_1^N+d_1\lambda_1^{N-1}+\cdots+d_{N-1}\lambda_1+d_N$$

$$=\sum_{k=0}^{N} d_k \lambda_1^{N-k}=0,$$

两边乘以λ_1^{n-N}得

$$\sum_{k=0}^{N} d_k \lambda_1^{n-k}=0,$$

令$y(n)=\lambda_1^n$，有

$$\sum_{k=0}^{N} d_k y(n-k)=0$$

例1：求以下齐次差分方程的通解

$$y(n)-5\,y(n-1)+6\,y(n-2)=0$$

解：特征方程为

$$\alpha^2-5\alpha+6=0$$

特征根为

$$\alpha_1=2,\quad \alpha_2=3$$

于是齐次差分方程的通解为

$$y(n)=c_1 2^n+c_2 3^n$$

（2.2）原方程特解

解法1-卷积法：

求方程的特解的方法有卷积法, 比较系数法和递推法等。这里主要介绍卷积法。卷积法的思路是: 由于在**零状态下**，线性非移变系统对输入$x(n)$的响应$y(n)$可用单位脉冲响应表示的线性卷积来计算。

$$y(\mathrm{n}) = h(n) * x(n)$$

零状态响应：不考虑原始时刻系统储能的作用（起始状态等于零），由系统的外加激励信号产生的响应。

为此，将方程写成D算子的形式，即

$$\sum_{k=0}^{N} d_k y(n-k) = \sum_{k=0}^{M} p_k x(n-k) \longrightarrow \sum_{k=0}^{N} d_k D^k y(n) = \sum_{k=0}^{M} p_k D^k x(n)$$

由此求得

$$y(n) = \frac{\sum_{k=0}^{M} p_k D^k}{\sum_{k=0}^{N} d_k D^k} x(n) = H(D) x(n)$$

令： $x(n)=\delta(n)$，则有： $h(n)=H(D)\delta(n)$

$$H(D)=\frac{\sum_{k=0}^{M} p_k D^k}{\sum_{k=0}^{N} d_k D^k}=\frac{p_0+p_1D+p_2D^2+\ldots+p_MD^M}{d_0+d_1D+d_2D^2+\ldots+d_ND^N}$$

$$=\frac{A_1}{1-a_1D}+\frac{A_2}{1-a_2D}+\frac{A_3}{1-a_3D}\ldots+\frac{A_N}{1-a_ND} \quad (M<N)$$

$$h(n)=H(D)\delta(n)$$

$$=\frac{A_1}{1-a_1D}\delta(n)+\frac{A_2}{1-a_2D}\delta(n)+\frac{A_3}{1-a_3D}\delta(n)\ldots+\frac{A_N}{1-a_ND}\delta(n)$$

$$=\sum_{k=1}^{N} g_k(n)$$

注意D是延迟算子，是对序列的延迟。式中：

$g_i(n)=A_i(\delta(n)+a_i\delta(n\text{-}1)+a_i^2\delta(n\text{-}2)+\ldots)=A_ia_i^nu(n)$

这里的$\delta(n)$，$\delta(n\text{-}1)$，$\delta(n\text{-}2)$...都是序列

式中：$g_i(n)=\dfrac{A_i}{1-a_iD}\delta(n)=A_i(1+a_iD+a_i^2D^2+\ldots)\delta(n)=A_ia_i^nu(n)$

所以：$h(n)=\sum_{i=1}^{N} A_ia_i^nu(n)$

9/24/2021 3:41:52 PM

方程的特解为

$$y_p = h(n) * x(n) = \left[\sum_{i=1}^{N} A_i a_i^n u(n) \right] * x(n)$$

例2： 设一个因果线性非移变系统由下列差分方程描述：

$$y(n)-3y(n-1)+2y(n-2)=x(n)$$

求系统的单位阶跃响应。

解：求单位阶跃响应就是求系统在零状态下，对单位阶跃输入$u(n)$的响应，即求特解

求特解。根据卷积法，首先令$x(n) = \delta(n)$，则得

$$h(n) - 3h(n-1) + 2h(n-2) = \delta(n)$$

或：

$$h(n) - 3Dh(n) + 2D^2h(n) = \delta(n)$$

由此得出：

$$h(n) = \frac{1}{1 - 3D + 2D^2}\delta(n) = \left(\frac{2}{1-2D} - \frac{1}{1-D}\right)\delta(n)$$

因此：

$$h(n) = (2 \times 2^n - 1^n)u(n)$$

故系统的单位阶跃响应为：

$$y_2(n) = h(n) * u(n) = [(2 \times 2^n - 1^n)u(n)] * u(n\ )$$

$$= (2^{n+2} - n - 3)u(n)$$

（2.2）原方程特解

解法2-比较法：

线性时不变系统输入与输出有相同的形式

输入	输出
$x(n)=e^{an}$	$y(n)=Ae^{an}$
$x(n)=e^{j\omega n}$	$y(n)=Ae^{j\omega n}$
$x(n)=\cos\omega n$	$y(n)=A\cos(\omega n+\theta)$
$x(n)=\sin\omega n$	$y(n)=A\sin(\omega n+\theta)$
$x(n)=n^k$	$y(n)=A_k n^k+A_{k-1}n^{k-1}+\cdots+A_1 n+A_0$
$x(n)=A$	$y(n)=C$
$x(n)=(r)^n$	$y(n)=C(r)^n$
$x(n)=(r)^n$（r与特征根重）	$y(n)=C_1 n(r)^n+C_2(r)^n$

（2.3）原方程的全解:

解的形式：$y(n)=y_c(n)+y_p(n)$

$y(n)$：全解（*total solution*）

$y_c(n)$：方程$\sum_{k=0}^{N} d_k y(n-k)=0$的解，称为齐次解(通解)（*complementary solution*）

$y_p(n)$：方程$\sum_{k=0}^{N} d_k y(n-k)=\sum_{k=0}^{M} p_k x(n-k)$，$x(n)\neq 0$的一个解，

称为特解（*particular solution*）

例3 $y(n)+y(n-1)-6y(n-2)=x(n)$

输入序列为$x(n)=8u(n)$，初始条件为$y(-1)=1$，$y(-2)=-1$

解：（1）齐次解

$$\lambda^n+\lambda^{n-1}-6\lambda^{n-2}=0 \Rightarrow \lambda_1=-3, \lambda_2=2$$

$$y_c(n)=a_1(-3)^n+a_2(2)^n$$

（2）特解

设$y_p=\beta \Rightarrow \beta=-2$，$n\geq 0$

（3）全解

$$y(n)=a_1(-3)^n+a_2(2)^n-2，\quad n\geq 0$$

代入初始条件可得$a_1=-1.8$，$a_2=4.8$

$$y(n)=-1.8(-3)^n+4.8(2)^n-2，\quad n\geq 0$$

序列的能量与功率

- 有界信号

若存在有界常数B，使序列$x(n)$满足

$$|x(n)| \le B < \infty$$

则称序列为有界信号。

- 序列的总能量

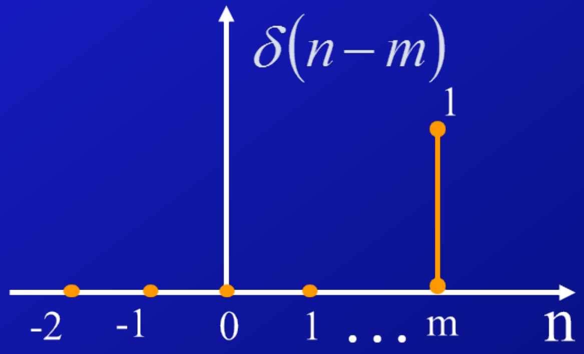
有界信号的总能量定义为序列各样点值的平方和，即：

$$E = \sum_{n=-\infty}^{\infty} |x(n)|^2$$

当$E < \infty$时，称信号为能量有限信号。

若序列的长度为有限长时，只要信号$x(n)$为有限值，则信号的能量就是有限的。但当信号的长度为无限长时，即使信号有界，其能量也不一定是有限的。

序列的能量与功率



- 序列的平均功率

1、对非周期序列 $x(n)$，若序列为无限长，其平均功率定义为：

$$P=\lim_{k\to\infty}\frac{1}{2K+1}\sum_{n=-K}^{K}|x(n)|^2=\lim_{k\to\infty}\frac{1}{2K+1}E$$

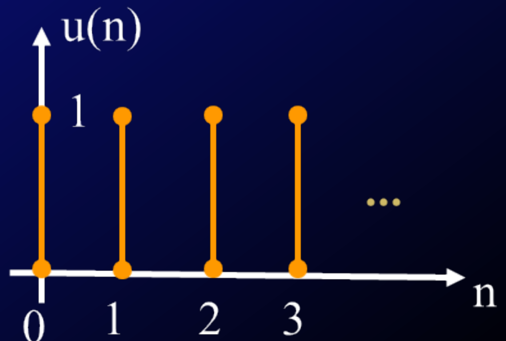
能量为有限值，平均功率等于0的信号称为**能量信号**。

能量为无限值，平均功率为有限值的信号称为**功率信号**。

2、对周期为 N 的周期序列 $\tilde{x}(n)$，其平均功率定义为：

$$P=\frac{1}{N}\sum_{n=0}^{N-1}|\tilde{x}(n)|^2$$

显然，周期序列通常为功率信号



例：设离散信号 $x(n)$ 的表达式为 $x(n)=6(-1)^{n}u(n)$

试判断该信号是能量信号还是功率信号。

解：∵该信号为有界信号，其总能量为：

$$E=\sum_{n=-\infty}^{\infty}|x(n)|^{2}=\sum_{n=0}^{\infty}36=\infty$$

可见信号的能量是无限的，但其功率为：

$$P=\lim_{k\to\infty}\frac{1}{2K+1}(36\sum_{n=0}^{K}1)=\lim_{k\to\infty}\frac{36(K+1)}{2K+1}=18$$

∴该信号是功率信号。

有限脉冲响应与无限脉冲响应

- **LTI离散时间系统的分类**
  - 脉冲响应的长度
    - 有限脉冲响应（finite impulse response，FIR）系统

$$h(n)=0 \quad n<N_1 \ or \ n>N_2$$

$$y(n)=\sum_{k=N_2}^{N_1} h(k)x(n-k)$$

    - 无限脉冲响应（infinite impulse response，IIR）系统

$h(n)$为无限长

$$y(n)=\sum_{k=0}^{\infty} h(k)x(n-k)$$

离散时间系统的因果性与稳定性

- 系统的因果性

  系统在 n 时刻的输出只取决于 n 时刻和 n 时刻以前的输入，而与 n 时刻以后的输入无关。

  系统的**因果性**表明了系统的物理可实现性。

  如果系统的输出与将来的输入有关，该系统为非因果系统，是物理不可实现的。

- 线性时不变系统具有因果性的充要条件

$$h(n) = h(n)u(n)$$

即要求描述系统特性的h(n)为一因果序列

离散时间系统的因果性与稳定性

- 系统的稳定性

  系统对于任何有界输入，输出也是有界的。

  称这种稳定性为有界输入—有界输出(BIBO)稳定性。

- **系统的稳定条件**

$$\sum_{n=-\infty}^{\infty}|h(n)|<\infty$$

- **典型例题**

若描述某离散系统特性的单位脉冲响应为：

$$h(n)=-a^{n}u(-n-1)$$

试讨论系统的因果性与稳定性。

离散时间系统的因果性与稳定性

解： **因果性** 因在n<0时，h(n)≠0，

故系统为非因果系统

稳定性

$$\sum_{n=-\infty}^{\infty}|h(n)|=\sum_{n=-\infty}^{-1}|a^{n}|=\sum_{n=1}^{\infty}|a|^{-n}=\begin{cases}\dfrac{1}{|a|-1} & |a|>1 \quad \text{稳定} \\ \infty & |a|\le 1 \quad \text{不稳定}\end{cases}$$

作业：

- P.86 复习思考题2.4
- P.87 习题：2.3，2.5，2.8（1，3，5），2.10，2.11，2.12，2.13。

扩展： 差分方程图像处理

$$\frac{\partial I}{\partial t} = \nabla^2 I$$

Where I is the continuous input signal, ∇^2 is the Laplacian and t is the scale parameter. Solutions to this equation are equivalent to convolution with a linear invariant Gaussian kernel.

$$\frac{\partial}{\partial t}I(x,y,t) = \frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}I(x,y,t\right)\right] + \frac{\partial}{\partial y}\left[\frac{\partial}{\partial y}I(x,y,t\right)\right]$$

$$\approx \frac{1}{(\Delta x)^2}\left[\big(I(x+\Delta x,y,t) - I(x,y,t)\big) - \big(I(x,y,t) - I(x-\Delta x,y,t)\big)\right] +$$

$$\frac{1}{(\Delta y)^2}\left[\big(I(x,y+\Delta y,t) - I(x,y,t)\big) - \big(I(x,y,t) - I(x,y-\Delta y,t)\big)\right]$$

$$I(x,y,t+\Delta t) \approx I(x,y,t)$$

$$+\Delta t\left\{\frac{1}{(\Delta x)^2}\left[\big(I(x+\Delta x,y,t) - I(x,y,t)\big) - \big(I(x,y,t) - I(x-\Delta x,y,t)\big)\right]\right.$$

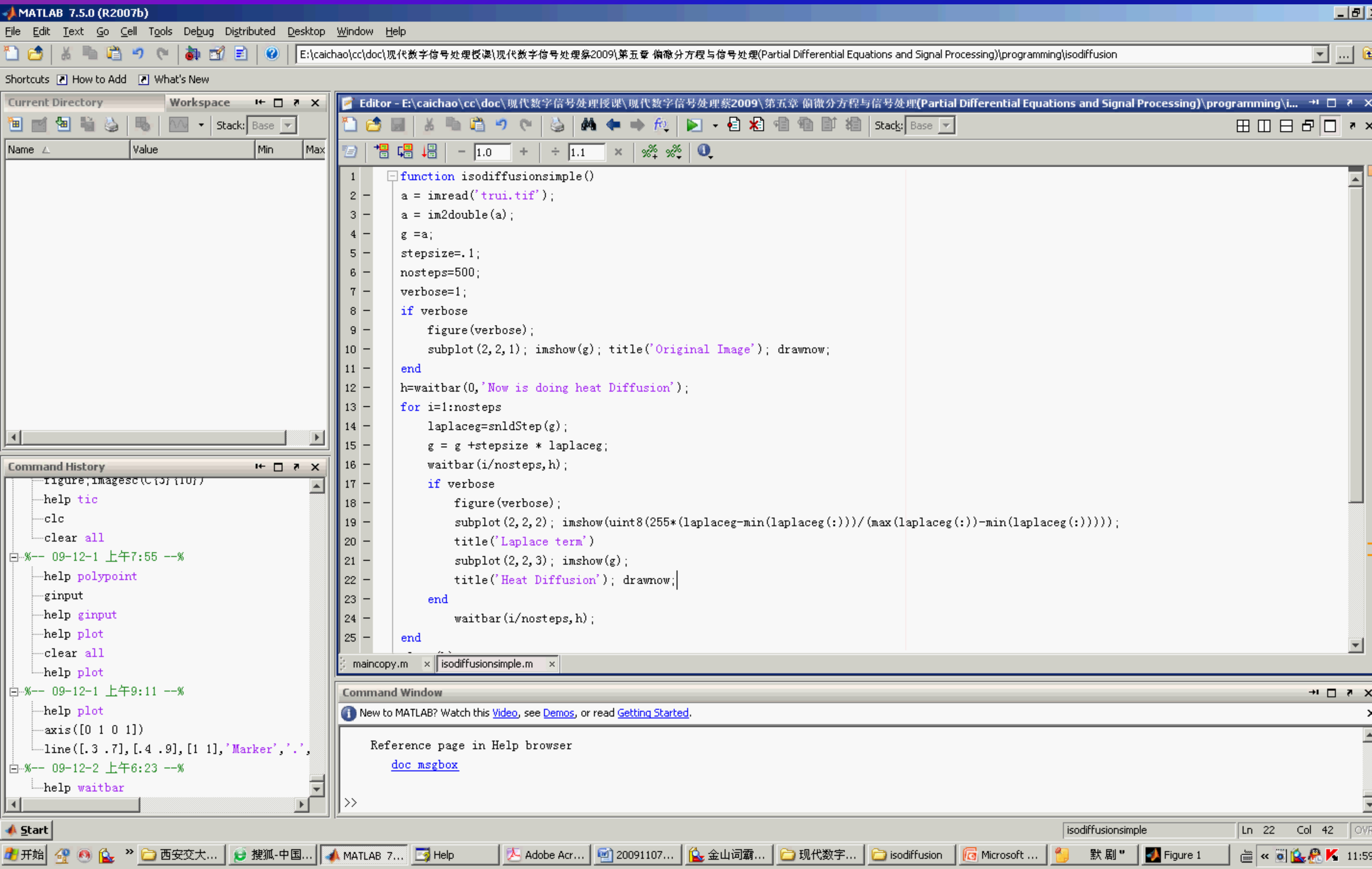
$$\left.+\frac{1}{(\Delta y)^2}\left[\big(I(x,y+\Delta y,t) - I(x,y,t)\big) - \big(I(x,y,t) - I(x,y-\Delta y,t)\big)\right]\right\}$$

$$= I(x,y,t) + \Delta t\left[\Phi_e + \Phi_w + \Phi_n + \Phi_s\right]$$

How to implement

```
function isodiffusionsimple()
a = imread('trui.tif');
a = im2double(a);
g =a;
stepsize=.24;
nosteps=100;
verbose=1;
h=waitbar(0,'Now is doing heat Diffusion');
for i=1:nosteps
    g = g +stepsize * snldStep(g);
end
```

```
function r = snldStep( L )
% Discrete numerical scheme
of dL/dt for scalar diffusion
N = size(L, 1);
M = size(L, 2);
Lpc = translateImage( L, 1, 0 );
Lmc = translateImage( L, -1,
0 );
Lcp = translateImage( L, 0, 1 );
Lcm = translateImage( L, 0, -
1 );
r = ( (Lpc-2*L+Lmc)+ ...
    (Lcp-2*L+Lcm) );
```



9/24/2021 3:41:52 PM

2.4 答案

解 $\omega(n)=x(n)*h_1(n)$

$$=\sum_{k=-\infty}^{\infty} u(k)[\delta(n-k)-\delta(n-k-4)]$$

$$=u(n)-u(n-4)$$

$$y(n)=\omega(n)*h_2(n)$$

$$=\sum_{k=-\infty}^{\infty} a^k u(k)[u(n-k)-u(n-k-4)]$$

$$=\sum_{k=n-3}^{\infty} a^k, n\geqslant 3$$