课 程 设 计 报 告

课程名称： 信息论

课程编号： 0108106006

任课教师： 张花国

实验项目： 信道容量 MATLAB 仿真

学　　号： 202021011119

姓　　名： 黄河

成　　绩：

目录

实验 信道容量 MATLAB 仿真

一、 课程设计目的

掌握信道容量计算原理，并进行软件编程仿真验证。

二、 课程设计要求

完成编程仿真，写成函数形式（MATLAB）。
输入：信道转移矩阵（考虑 3 种情况对称信道、准对称信道、非奇异方阵）；
输出：信道容量，最佳输入分布。

三、 软件设计方案

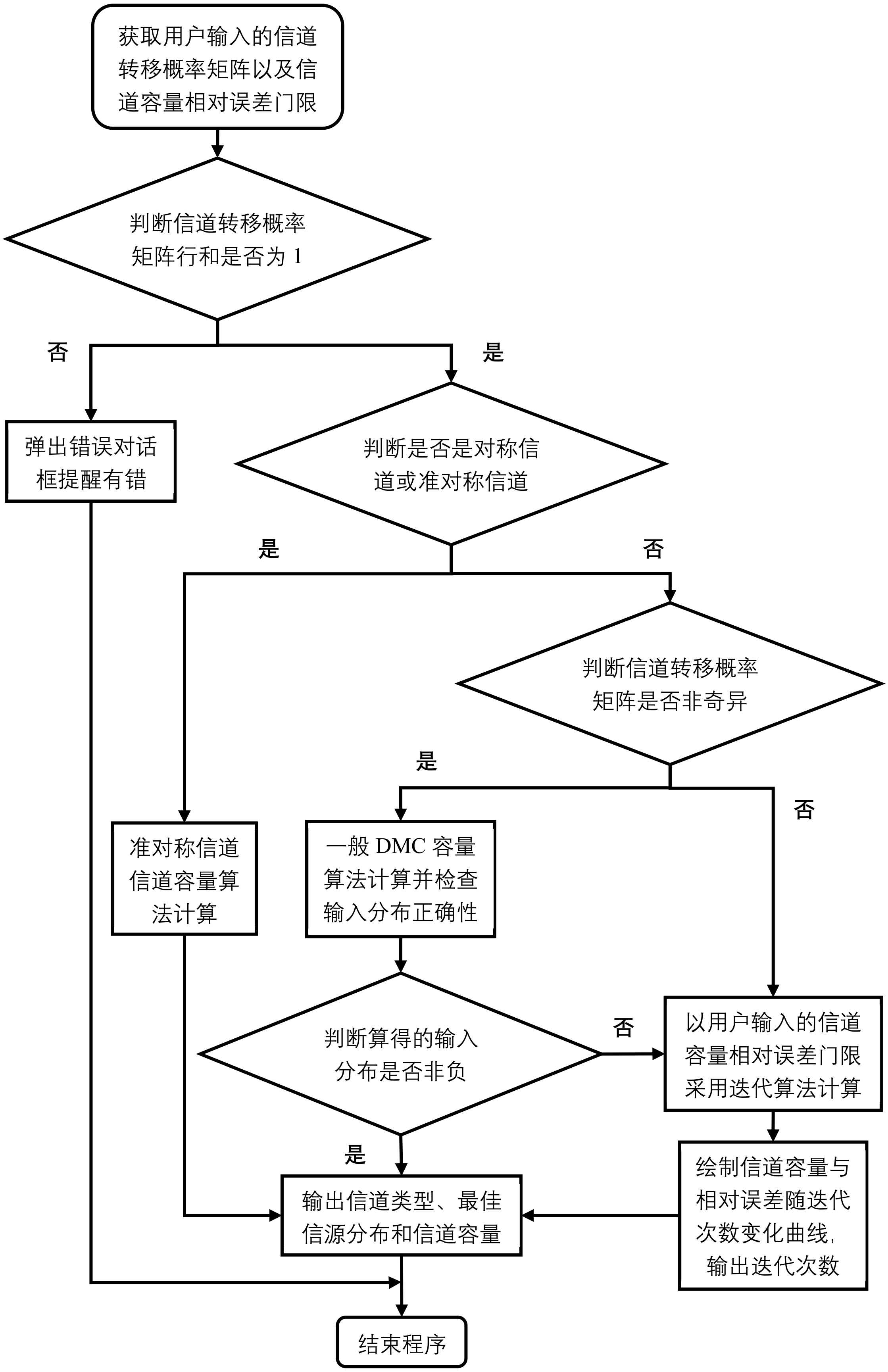
为使程序输入具有灵活性，输入的信道转移概率矩阵(Pij)由用户决定，只要满足矩阵的各行元素之和均为 1 即可（在主程序中以代码注释的方式提供各种情况下的 Pij 的例子供用户选择，若输入的矩阵不满足行和为 1 的条件，运行过程中会弹出错误对话框提醒用户检查后再次运行程序，并结束该次程序的运行）。

而为使输出结果简洁明了，将一些程序模块以函数的形式进行了封装，并在函数文件中以注释的形式编写了帮助文档（包括输入输出变量解释，特别地，为使输入输出变量简洁易懂，尽量采用结构体形式定义变量）。具体哪个程序是主程序，哪些程序是函数文件见代码文件夹中的“Readme.txt”文件，下面也会简要介绍各函数文件的函数名以及它们的作用，并将整体流程图绘制如下：

（1）Channel_capacity_parameter_settings.m 函数
获取用户的输入信道转移概率矩阵(Pij)以及信道容量相对误差门限(δ)。
（2）Symtc_Chanl_mtx.m 函数
判断信道转移概率矩阵 Pij 是否对称或准对称。
（3）Channel_capacity_symtc_algrtm.m 函数。
计算对称或准对称信道转移概率矩阵的信道容量(C)和最佳信源分布(Pi)。
（4）Channel_capacity_nonSngr_algrtm.m 函数
计算非奇异非对称信道转移概率矩阵的信道容量(C)和最佳信源分布(Pi)。先尝试用一般 DMC 容量算法计算并检查输入分布是否均非负；若非负，说明最佳信源分布在边界内取值，算得结果正确，跳出函数；若有负数，说明最佳信源分布在边界上取值（再用一般 DMC 容量算法计算就相当困难了），调用信道容量迭代算法计算，并返回算得结果。
（5）Channel_capacity_itrtn_algrtm.m 函数
信道容量迭代算法，计算奇异非对称信道转移概率矩阵的信道容量(C)和最佳信源分布(Pi)（实际上，对于所有满足信道转移概率矩阵定义的 DMC 矩阵，都可以用该函数计算信道容量(C)和最优信源分布(Pi)）。

（6）Channel_capacity_output.m 函数

在 Command Window 或 Figure 中可视化输出相应结果。



实验过程

一、 MATLAB程序

1. 主程序

```
% CHANNEL_CAPACITY_DMC    Channel capacity for DMC.
%
%
%
% 输入解释:
%           第一界面:
%                       A. 第二界面:
%                                       选择随机产生满足DMC的指定行数与列数以及迭代精
%                                       度(δ)的信道转移概率矩阵Pij;
%                       B. 第二界面:
%                                       选择自定义输入一个满足DMC的信道转移概率矩阵Pij
%                                       以及迭代精度(δ)。
%                                       下面提供各种情况下的Pij的例子供B项第二界面输入
%
% Examples of Pij:
%           symmetric:
%                       [1/3 1/3 1/6 1/6; 1/6 1/6 1/3 1/3]
%                                                       (result: C = 0.081704)
%                       or
%                       [1/2 1/3 1/6; 1/6 1/2 1/3; 1/3 1/6 1/2]
%                                                       (result: C = 0.125815)
%                       or
%                       [0.7 0.1 0.1 0.1; 0.1 0.7 0.1 0.1;
%                        0.1 0.1 0.7 0.1; 0.1 0.1 0.1 0.7]
%                                                       (result: C = 0.643220)
%           para-symmetric:
%                       [0.8 0.1 0.1; 0.1 0.1 0.8]          (result: C = 0.447067)
%                       or
%                       [1/2 1/4 1/8 1/8; 1/4 1/2 1/8 1/8]
%                                                       (result: C = 0.061278)
%                       or
%                       [1/3 1/3 0 1/3; 0 1/3 1/3 1/3; 1/3 0 1/3 1/3]
%                                                       (result: C = 0.389975)
%                       or
%                       [0.1 0.1 0.2 0.2 0.4; 0.1 0.2 0.4 0.1 0.2;
%                        0.1 0.1 0.4 0.2 0.2; 0.1 0.2 0.2 0.1 0.4]
%                                                       (result: C = 0.073534)
%           asymmetric nonsingular:
%                       default matrix in choice 'B'
%                       or
%                       [3/4 1/4 0; 1/3 1/3 1/3; 0 1/4 3/4]
%                                                   (result: C    = 0.75
%                                                          Pi = [1/2 0 1/2])
```

```
%                           or
%                              [0.5 0.25 0 0.25; 0 1 0 0; 0 0 1 0; 0.25 0 0.25 0.5]
%                                                  (result: C    = 1.321928
%                                                               Pi = [4/30 11/30 11/30 4/30])
%             asymmetric singular:
%                              [1 0; 1 0; 0.5 0.5; 0 1; 0 1]
%                                                       (result: C    = 1
%                                                                    Pi = [1/4 1/4 0 1/4 1/4])
%                              [0.35 0.45 0.20; 0.30 0.50 0.20;
%                               0.10 0.55 0.35; 0.35 0.45 0.20]
%                                         (result: C    = 0.071891
%                                                     Pi = [0.234114 0 0.531772 0.234114])
%
%
%
% $ Date: 2021-06-15 23:19:44
% Author: Huang He
% E-mail: 2327012749@qq.com

%%
clear; close all; clc

%% 获取用户的输入数据。
options = Channel_capacity_parameter_settings;

%% 若信道转移概率矩阵无误，分类讨论对称性并计算最佳信源分布Pi与信道容量C。
if options.cortn.Pij == 1
    %% 判断是否是对称信道或准对称信道的信道转移概率矩阵。
    symtc = Symtc_Chanl_mtx(options.Pij);

    %% 分类计算最佳信源分布Pi与信道容量C。
    if symtc ~= 0
        %% Pij是对称信道或准对称信道。
        chcp_result = Channel_capacity_symtc_algrtm(options.Pij);

    elseif (symtc == 0) && (options.cortn.nonSngr == 1)
        %%
        % Pij是非奇异非对称矩阵，算法中优先尝试采用一般算法求解，看是否在边界
        % 内取值。
        chcp_result = ...
            Channel_capacity_nonSngr_algrtm(options.Pij, options.delta);

    else
        %% Pij是奇异且非对称矩阵，采用迭代算法求解。
        chcp_result = ...
            Channel_capacity_itrtn_algrtm(options.Pij, options.delta);

    end

    %% 在Command Window或Figure中可视化输出。
    Channel_capacity_output(chcp_result, symtc);

end

```

2. 函数程序

(1) Channel_capacity_parameter_settings.m函数

```
function options = Channel_capacity_parameter_settings
% CHANNEL_CAPACITY_PARAMETER_SETTINGS    Get the user's input data.
%
%
%
% Notation explanation
%
%      Input:
%            None.
%
%      Output:
%            options
%                  structure variable
%                  Pij    : channel transition probability matrix;
%                  delta: channel capacity relative error threshold δ;
%                  slctn: (char variable)selection of choice 'A' or 'B';
%                  cortn
%                        structure variable
%                        Pij        : check the correctness of channel transition
%                                     probability matrix Pij:
%                                     ① if it's correct    , options.cortn.Pij = 1;
%                                     ② if it's incorrect, options.cortn.Pij = 0;
%                        delta    : check the correctness of channel capacity relative
%                                     error threshold δ:
%                                     ① if δ >    0, options.cortn.delta = 1;
%                                     ② if δ <= 0, options.cortn.delta = 0;
%                        nonSngr: judge whether the matrix is nonsingular:
%                                     ① if it's nonsingular, options.cortn.nonSngr = 1;
%                                     ② if it's singular or non-square matrix,
%                                          options.cortn.nonSngr = 0.
%
%
%
% $ Date: 2021-06-14 14:53:52
% Author: Huang He
% E-mail: 2327012749@qq.com

%% 用户选择随机生成一个r行s列的信道转移概率矩阵还是自己输入一个。
options.slctn = questdlg([
      'Do you want to:                                                  '...
      '              A. Randomly generate a channel transition probability mat'...
      'rix with known rows and columns                                 '...
      '                                 B. Manually input your own channel tran'...
      'sition probability matrix'], ...
      'Type selection', 'A', 'B', 'A');

dlg_title = 'Initial setting of CHANNEL CAPACITY';

%%
switch options.slctn

      case 'A'
```

```
        %% 随机生成一个已知行和列的信道转移概率矩阵。
        prompt = {[
            'Please input NUMBER OF    ROWS      of transition probability m'...
            'atrix Pij(or the number of inputs):'
            ], ...
            [
            'Please input NUMBER OF COLUMNS of transition probability m'...
            'atrix Pij(or the number of outputs):'
            ], ...
            [
            'Please enter the channel capacity relative error threshold'...
            ' δ(Please make sure that δ > 0, just can be used in ITER'...
            'ATIVE Algorithm for SINGULAR and ASYMMETRIC matrices):'
            ]};
        num_lines = [1 65];

        def = {'10', '10', '1e-12'};        % 输入的相对误差门限值δ默认为1e-12

        answer = inputdlg(prompt, dlg_title, num_lines, def);

        r = str2double(answer{1, 1});
        s = str2double(answer{2, 1});
        options.Pij = rand(r, s);
        options.Pij = options.Pij ./ sum(options.Pij, 2);
        options.cortn.Pij = 1;

    case 'B'
        %% 手动输入自己的信道转移概率矩阵。
        prompt = {[
            'Please input channel transition probability matrix Pij (Ple'...
            'ase make sure that the SUM of elements in each ROW of the '...
            'matrix is 1):'
            ], ...
            [
            'Please enter the channel capacity relative error threshold'...
            ' δ(Please make sure that δ > 0, just can be used in ITER'...
            'ATIVE Algorithm for SINGULAR and ASYMMETRIC matrices):'
            ]};
        num_lines = [5 65];

        def = {[
            '0.175 0.225 0.150 0.150 0.100 0.200; '...
            '0.050 0.300 0.055 0.200 0.150 0.245; '...
            '0.105 0.300 0.200 0.150 0.195 0.050; '...
            '0.150 0.225 0.050 0.275 0.050 0.250; '...
            '0.050 0.295 0.105 0.200 0.150 0.200; '...
            '0.225 0.100 0.200 0.150 0.075 0.250'
            ], ...
            '1e-12'};                            % 输入的相对误差门限值δ默认为1e-12

        answer = inputdlg(prompt, dlg_title, num_lines, def);

        % 读取输入的信道转移概率矩阵Pij。
        options.Pij     = str2num(answer{1, 1});

        [r, s] = size(options.Pij);          % 计算信道转移概率矩阵Pij的行数与列数
```

```
        
        %% 检查信道转移概率矩阵的错误，若出错，弹出错误警示框。
        % 由于matlab本身精度限制，使得计算行和时结果不一定为1，如：
        %           0.1 + 0.1 + 0.1 + 0.7 = 1 - 1.1102e-16 = 1 - eps / 2;
        %           1/3 + 1/3 + 1/6 + 1/6 = 1 - 1.1102e-16 = 1 - eps / 2;
        % 所以不能通过严格判断每一行的和是否为1检查Pij是否正确，而是容许eps/2
        % 的误差。
        if (  abs(sum(sum(options.Pij, 2) - ones(r,1))) > r*eps/2   ) || ...
                ( sum(sum(sign(sign(options.Pij)+0.5))) ~= r*s )
            [~] = errordlg([
                'Input data error, please check the channel transition '...
                'probability matrix and rerun the program!'
                ], 'Error');
            options.cortn.Pij = 0;
        else
            options.cortn.Pij = 1;
        end

end

options.delta = str2double(answer{end, 1}); % 读取δ值

%% 若输入的δ<=0，不会提醒用户输入有误，但会将其改为一个默认值1e-12。
if options.delta <= 0
    options.delta = 1e-12;
    options.cortn.delta = 0;
else
    options.cortn.delta = 1;
end

%% 判断矩阵是否非奇异，若不是方阵，默认奇异。
if r ~= s
    options.cortn.nonSngr = 0;
elseif det(options.Pij) ~= 0
    options.cortn.nonSngr = 1;
else
    options.cortn.nonSngr = 0;
end

end

```

(2) Symtc_Chanl_mtx.m函数

```
function symtc = Symtc_Chanl_mtx(Pij)
% SYMTC_CHANL_MTX   judge whether the channel transition probability matrix
%                   Pij is symmetric or para-symmetric.
%
%
%
% Notation explanation
%
%    Input:
%        Pij: channel transition probability matrix;
```

```
%
%     Output:
%             symtc: judge whether the matrix Pij is symmetric or para-symmetric:
%                         ①  if Pij is asymmetric       , symtc = 0;
%                         ②  if Pij is symmetric         , symtc = 1;
%                         ③  if Pij is para-symmetric, symtc = 2.
%
%
%
% $ Date: 2021-06-15 01:08:04
% Author: Huang He
% E-mail: 2327012749@qq.com

%%
[r, s] = size(Pij);
symtc = -1;                                             % symtc初始化为-1

if (r == 1) && (s == 1)
    %% Pij = 1的单输入单输出(SISO, single-input single-output)确定信道。
    symtc = 1;

elseif r == 1
    %%  单输入多输出(SIMO, single-input multiple-output)信道。
    if isequal( Pij, 1/s * ones(r, s) )
        symtc = 1;
    else
        symtc = 2;
    end

elseif s == 1
    %%
    % Pij = [1; ...; 1]的多输入单输出(MISO, multiple-input single-output)确
    %  定信道。
    symtc = 1;

elseif symtc_Row(Pij) ~= 1
    %%  多输入多输出(MIMO, multiple-input multiple-output)的非对称信道。
    symtc = 0;

elseif (symtc_Row(Pij) == 1) && (symtc_Row(Pij') == 1)
    %%  多输入多输出(MIMO, multiple-input multiple-output)的对称信道。
    symtc = 1;

else
    %%
    %  多输入多输出(MIMO, multiple-input multiple-output)的行对称信道；需要判
    %  断是否是准对称信道，若不是，则是一个非对称信道。



    %  找出含有相同元素的列并转置后排在一起。
    tag = 1:s;
    Pij_col = sort(Pij, 1);

    for i = 1:s
```

```
            for j = i:s
                if isequal( Pij_col(:,i), Pij_col(:,j) )
                    tag(j) = tag(i);
                end
            end
        end
        Pij_sort = sortrows([tag; Pij]');



        % 判断按列划分的几个含有相同元素的列组成的子矩阵是否是对称矩阵。
        for i = 1:s
            Pij_check = Pij_sort(    Pij_sort(:,1) == i, 2:end    );
            [nonzero,~]=size(Pij_check);
            if nonzero ~= 0
                if Symtc_Chanl_mtx(Pij_check') ~= 1            % 递归调用本身
                    symtc = 0;
                    break
                end
            end
        end

        if symtc == -1
            symtc = 2;
        end

end

end



%% Sub-function: symtc_Row.
function symtc_row = symtc_Row(mtx)
% SYMTC_ROW judge whether the matrix is row-symmetric.
%
%
%
% Notation explanation
%     Input:
%         mtx: r×s matrix;
%               NOTICE: r & s must be positive integers greater than 1 !
%
%     Outputs:
%         symtc_row: judge whether the matrix mtx is row-symmetric;
%                     ① if mtx is       row-symmetric, symtc_row = 1;
%                     ② if mtx isn't row-symmetric, symtc_row = 0;

%%
[r, ~] = size(mtx);
symtc_row = -1;                                             % symtc_row初始化为-1
mtx_row = sort(mtx, 2);

%% 判断是否关于输入对称/行对称。
for i = 1 : floor(log2(r))
    if ~isequal(    mtx_row( 1:2^(i-1), : ), ...
```

```
                mtx_row( (2^(i-1)+1):2^i, : )    )
            symtc_row = 0;                                          % 不关于输入对称
            break
        end
end

if (symtc_row == -1) && isequal(mtx_row, flipud(mtx_row))
        symtc_row = 1;                                              % 关于输入对称
elseif symtc_row == -1
        symtc_row = 0;                                              % 不关于输入对称
end

end

```

(3) Channel_capacity_symtc_algrtm.m函数

```
function chcp_result = Channel_capacity_symtc_algrtm(Pij)
% CHANNEL_CAPACITY_SYMTC_ALGRTM    Algorithm of channel capacity for DMC
%                                  matrix that satisfy the definition of
%                                  symmetric or para-symmetric channel
%                                  transition probability matrix, the channel
%                                  capacity(C) and optimal source
%                                  distribution(Pi) can be calculated by this
%                                  function.
%
%
%
% Notation explanation
%
%     Input:
%         Pij: channel transition probability matrix;
%
%     Output:
%         chcp_result
%             structure variable
%             Pi        : optimal source distribution(equal probability
%                         distribution);
%             C         : channel capacity, unit: bit/symbol;
%             k         : default to 1;
%             variation: default to 0.
%
%
%
% $ Date: 2021-06-15 20:56:09
% Author: Huang He
% E-mail: 2327012749@qq.com

%%
[r, ~] = size(Pij);
chcp_result.Pi = 1/r * ones(r, 1);

% rmmissing:    Removes missing entries (eg. NaN caused by 0*log2(0))
%                 from an array or table.
chcp_result.C = ...
        sum(rmmissing( Pij(1,:) .* log2(Pij(1,:) .* r ./ sum(Pij, 1)) ));
```

```

chcp_result.k = 1;
chcp_result.variation = 0;

end

```

(4) Channel_capacity_nonSngr_algrtm.m函数

```
function chcp_result = Channel_capacity_nonSngr_algrtm(Pij, delta)
% CHANNEL_CAPACITY_NONSNGR_ALGRTM    Algorithm of channel capacity for DMC
%                                     matrix that satisfy the definition of
%                                     nonsingular asymmetric channel
%                                     transition probability matrix, the
%                                     channel capacity(C) and optimal source
%                                     distribution(Pi) can be calculated by
%                                     this function.
%
%
%
% Notation explanation
%
%     Inputs:
%         Pij    : channel transition probability matrix;
%         delta: channel capacity relative error threshold δ.
%
%     Output:
%         chcp_result
%             structure variable
%             Pi          : optimal source distribution;
%             C           : channel capacity scalar or array, unit: bit/symbol;
%             k           : temporarily default to 1, if the optimal source
%                           distribution(Pi) takes values on the boundary, set
%                           to -1;
%             variation: default to 0, if the optimal source distribution(Pi)
%                           takes values on the boundary, it's a array.
%
%
%
% $ Date: 2021-06-15 21:55:52
% Author: Huang He
% E-mail: 2327012749@qq.com

%%
chcp_result.k = 1;
[r, ~] = size(Pij);
h_Yx = zeros(r, 1);
for i = 1:r
    h_Yx(i) = sum(rmmissing(Pij(i, :) .* log2(Pij(i, :))));
end

beta = Pij \ h_Yx;
chcp_result.C = log2(sum(2 .^ beta));
Pj = 2 .^ (beta - chcp_result.C);
chcp_result.Pi = Pij' \ Pj;
chcp_result.variation = 0;
```

```

if sum(chcp_result.Pi<0) > 0
    %% 输入分布在边界上取值时，求解十分麻烦，调用迭代算法函数求解。
    % 调用Channel_capacity_itrtn_algrtm.m函数。
    chcp_result = Channel_capacity_itrtn_algrtm(Pij, delta);
    chcp_result.k = -1;
end

end

```

(5) Channel_capacity_itrtn_algrtm.m函数

```
function chcp_result = Channel_capacity_itrtn_algrtm(Pij, delta)
% CHANNEL_CAPACITY_ITRTN_ALGRTM    Iterative algorithm of channel capacity;
%                                  ACTUALLY, for ALL DMC matrix that
%                                  satisfy the definition of channel
%                                  transition probability matrix, the channel
%                                  capacity(C) and optimal source
%                                  distribution(Pi) can be calculated by this
%                                  function.
%
%
%
% Notation explanation
%
%     Inputs:
%         Pij   : channel transition probability matrix;
%         delta: channel capacity relative error threshold δ.
%
%     Output:
%         chcp_result
%             structure variable
%             Pi          : optimal source distribution;
%             C            : channel capacity sequence generated in iteration
%                           process, unit: bit/symbol;
%             k            : iterations;
%             variation: relative error sequence generated in iteration
%                           process.
%
%
%
% $ Date: 2021-06-14 16:21:50
% Author: Huang He
% E-mail: 2327012749@qq.com

%% 初始化。
[r, ~] = size(Pij);
k = 0;                                    % 置迭代计数器k=0
chcp_result.C = -inf;                     % 置迭代初始值C=-∞
chcp_result.Pi(1:r) = 1/r;                % 信源分布Pi初始化为均匀分布
num = zeros(1,r);

%% 迭代设置。
error(1) = delta+1;
```

```
while error(k+1) > delta                          % 若相对误差大于门限值,则进行迭代
    k = k+1;                                      % 迭代计数器加一

    %% 计算φ的迭代值。
    fai = Pij' .* chcp_result.Pi ./ (sum( Pij .* chcp_result.Pi', 1 ))';

    %% 计算信源分布Pi的迭代值。
    for i = 1:r
        % rmmissing:    Removes missing entries (eg. NaN caused by 0*log2(0))
        %                   from an array or table.
        num(i) = 2^(sum(    rmmissing(Pij(i,:)' .* log2(fai(:,i)))    ));
    end
    chcp_result.Pi = num / sum(num);

    %% 计算信道容量C的迭代值。
    % 直接利用上一段子程序中已经计算出来的求和值。
    chcp_result.C(1,k+1) = log2(sum(num));

    %% 计算相对误差值。
    % 分母保护，在分母上加上一个无穷小量ε。
    error(k+1) = abs(chcp_result.C(1,k+1)-chcp_result.C(1,k)) / ...
        (chcp_result.C(1,k+1)+realmin);

end

chcp_result.k = k;
chcp_result.variation = error(2 : end);
chcp_result.C = chcp_result.C(2 : end);

end

```

(6) Channel_capacity_output.m函数

```
function Channel_capacity_output(parameters, symtc)
% CHANNEL_CAPACITY_OUTPUT    In Command Window or Figure, the optimal source
%                                    distribution 'Pi', channel capacity 'C', number
%                                    of iterations 'k' and the relative change of
%                                    channel capacity 'variation' in the process of
%                                    iteration are visualized..
%
%
%
% Notation explanation
%
% Inputs:
%           chcp_result
%                structure variable
%                Pi          : optimal source distribution;
%                C            : channel capacity sequence generated in iteration
%                                process, unit: bit/symbol;
%                k            :  ①  iterations in channel capacity iterative
%                                    algorithm 'Channel_capacity_itrtn_algrtm.m';
%                                ②  if Pij is a nonsingular asymmetric matrix and the
%                                    optimal source distribution(Pi) takes values on
```

```
%                                     the boundary,      'k' is -1;
%                                  ③ in other cases, 'k' is    1;
%                   variation: relative error sequence generated in iteration
%                                  process.
%               symtc: whether the matrix Pij is symmetric or para-symmetric:
%                          ① if Pij is asymmetric        , symtc = 0;
%                          ② if Pij is symmetric          , symtc = 1;
%                          ③ if Pij is para-symmetric, symtc = 2.
%
% Output:
%     None.
%
%
%
% $ Date: 2021-06-15 11:30:15
% Author: Huang He
% E-mail: 2327012749@qq.com

%% 可视化输出。

if (parameters.k ~=1) && (parameters.k~= -1)
    %%
    % 若采用迭代算法求解，绘制信道容量随迭代次数的变化曲线以及迭代过程中信道容
    % 量相对变化量随迭代次数的变化曲线以验证信道容量的收敛性。
    fig = figure;
    set(fig, 'units', 'normalized', 'position', [1/2 0.02 1/3 0.85])

    subplot(211)
    plot(1 : parameters.k, parameters.C, 'r-*')
    xlabel('Interations k'); ylabel('Channel capacity   /   (bit/symbol)')
    grid minor
    title(['The tendency of Value C(Channel capacity) with the ', ...
        'Interations k'])

    subplot(212)
    semilogy(   1 : parameters.k, parameters.variation, 'k-o'   )
    xlabel('Interations k'); ylabel('Relative variation'); grid minor
    title('The tendency of Relative variation with the Interations k')

    fprintf('这是一个奇异且非对称矩阵，采用迭代算法求解。\n\n')
    fprintf('迭代次数：\t\tk = ')
    fprintf('%d\r\n', parameters.k)                          % 输出迭代次数k

elseif parameters.k == -1
    %%
    parameters.k = length(parameters.C);

    fig = figure;
    set(fig, 'units', 'normalized', 'position', [1/2 0.02 1/3 0.85])

    subplot(211)
    plot(1 : parameters.k, parameters.C, 'r-*')
    xlabel('Interations k'); ylabel('Channel capacity   /   (bit/symbol)')
    grid minor
```

```
    title(['The tendency of Value C(Channel capacity) with the ', ...
        'Interations k'])

    subplot(212)
    semilogy(   1 : parameters.k, parameters.variation, 'k-o'   )
    xlabel('Interations k'); ylabel('Relative variation'); grid minor
    title('The tendency of Relative variation with the Interations k')

    fprintf('这是一个非奇异非对称矩阵，但要采用迭代算法求解,\n')
    fprintf('因为最佳信源分布在边界上。\n\n')
    fprintf('迭代次数：\t\tk = ')
    fprintf('%d\r\n', parameters.k)                     % 输出迭代次数k

elseif symtc == 0
    fprintf('这是一个非奇异非对称矩阵，采用一般算法求解,\n')
    fprintf('且求得的最佳信源分布不在边界上。\n\n')

elseif symtc == 1
    fprintf('这是一个对称DMC矩阵，最佳信源分布为等概分布。\n\n')

elseif symtc == 2
    fprintf('这是一个准对称DMC矩阵，最佳信源分布为等概分布。\n\n')

end

fprintf('最佳信源分布为：  Pi =\r\n\t\t\t\t')
fprintf('%f\r\n\t\t\t\t\t', parameters.Pi)              % 输出最佳信源分布Pi
fprintf('\n信道容量为：\t\tC = ')
fprintf('%f\r\n', parameters.C(parameters.k))           % 输出最终的C值

end

```

二、 仿真结果验证

1. 几种简单信道的仿真

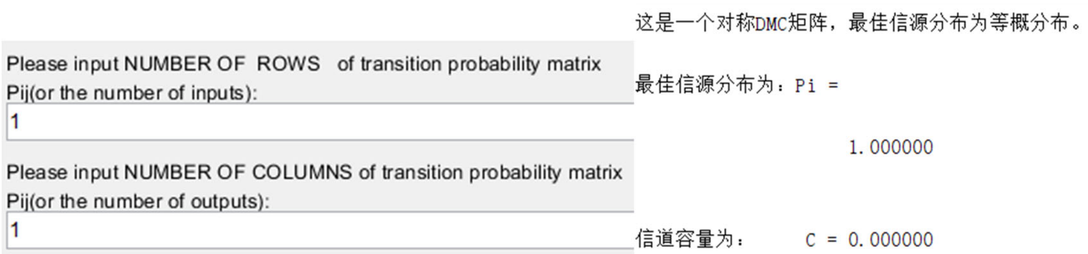


图 1. 单输入单输出(SISO)的无噪无损确定信道

Please input NUMBER OF ROWS of transition probability matrix Pij(or the number of inputs):

4

Please input NUMBER OF COLUMNS of transition probability matrix Pij(or the number of outputs):

1

这是一个对称DMC矩阵，最佳信源分布为等概分布。

最佳信源分布为：Pi =

0.250000

0.250000

0.250000

0.250000

信道容量为： C = 0.000000

图 2. 多输入单输出(MISO)的无噪有损信道

Please input NUMBER OF ROWS of transition probability matrix Pij(or the number of inputs):

1

Please input NUMBER OF COLUMNS of transition probability matrix Pij(or the number of outputs):

4

这是一个准对称DMC矩阵，最佳信源分布为等概分布。

最佳信源分布为：Pi =

1.000000

信道容量为： C = 0.000000

图 3. 单输入多输出(SIMO)的有噪无损信道

Please input channel transition probability matrix Pij: (Please make sure that the SUM of elements in each ROW of the matrix is 1)

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

这是一个对称DMC矩阵，最佳信源分布为等概分布。

最佳信源分布为：Pi =

0.250000

0.250000

0.250000

0.250000

信道容量为： C = 2.000000

图 4. 具有一一对应关系的无噪无损信道

Please input channel transition probability matrix Pij: (Please make sure that the SUM of elements in each ROW of the matrix is 1)

```
0.5 0.5
0.5 0.5
```

这是一个对称DMC矩阵，最佳信源分布为等概分布。

最佳信源分布为：Pi =

0.500000

0.500000

信道容量为： C = 0.000000

图 5. 二元对称信道

Please input channel transition probability matrix Pij: (Please make sure that the SUM of elements in each ROW of the matrix is 1)

1 0
0.5 0.5

这是一个非奇异非对称矩阵，采用一般算法求解，
且求得的最佳信源分布不在边界上。

最佳信源分布为：Pi =

0.600000

0.400000

信道容量为： C = 0.321928

图 6. Z 信道

Please input channel transition probability matrix Pij: (Please make sure that the SUM of elements in each ROW of the matrix is 1)

0.9 0.1 0
0 0.1 0.9

这是一个准对称DMC矩阵，最佳信源分布为等概分布。

最佳信源分布为：Pi =

0.500000

0.500000

信道容量为： C = 0.900000

图 7. 二元纯删除信道

Please input channel transition probability matrix Pij: (Please make sure that the SUM of elements in each ROW of the matrix is 1)

0.3 0.7 0 0
0.7 0.3 0 0
0 0 0.4 0.6
0 0 0.6 0.4

这是一个非奇异非对称矩阵，采用一般算法求解，
且求得的最佳信源分布不在边界上。

最佳信源分布为：Pi =

0.257766

0.257766

0.242234

0.242234

信道容量为： C = 1.074576

图 8. 和信道

2. 离散对称/准对称信道的仿真

Please input channel transition probability matrix Pij: (Please make sure that the SUM of elements in each ROW of the matrix is 1)

1/2 1/3 1/6
1/6 1/2 1/3
1/3 1/6 1/2

这是一个对称DMC矩阵，最佳信源分布为等概分布。

最佳信源分布为：Pi =

0.333333

0.333333

0.333333

信道容量为： C = 0.125815

图 9. 离散对称信道

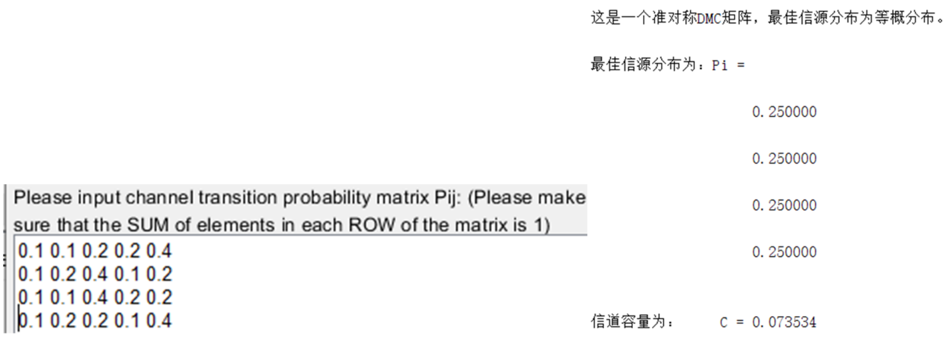


图 10. 离散准对称信道

3. 非奇异方阵信道的仿真

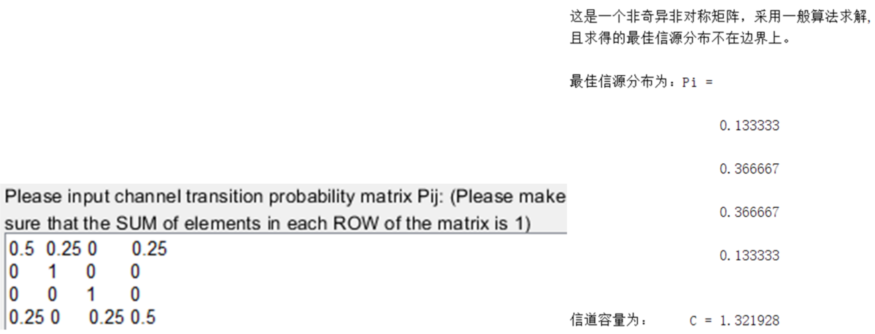


图 11. 最佳信源分布在边界内取值情况

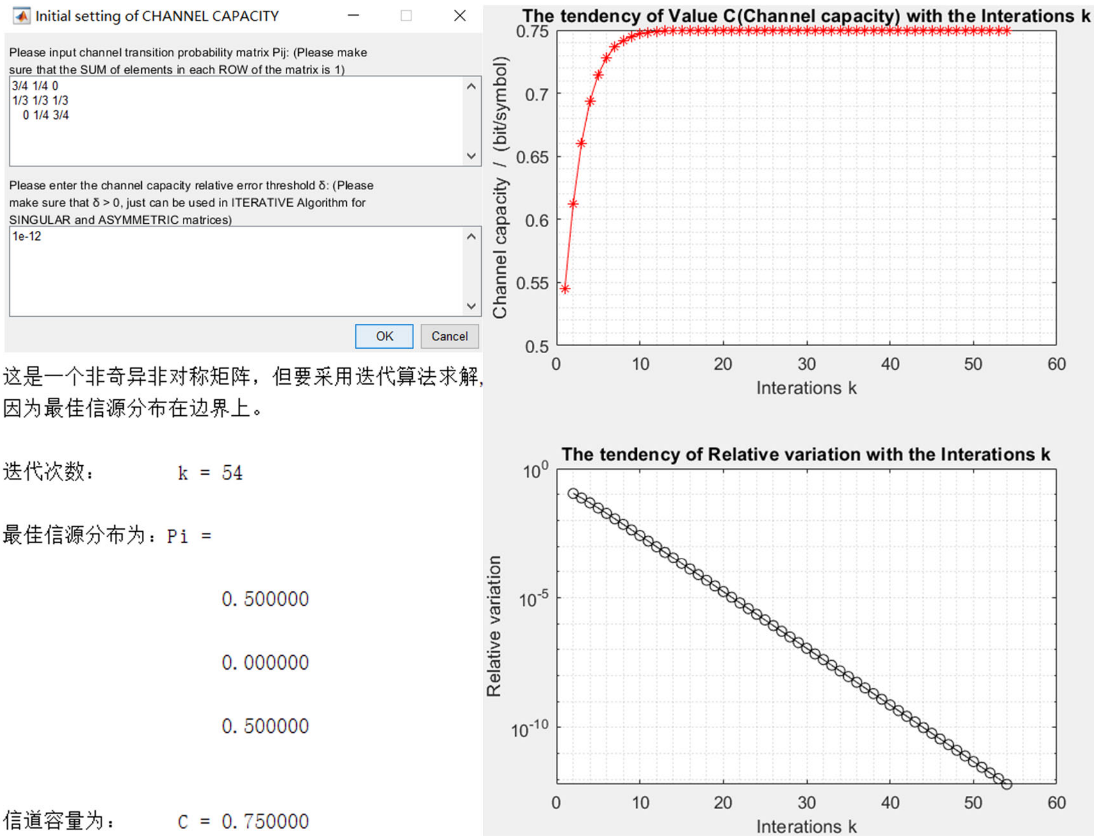


图 12. 最佳信源分布在边界上取值情况

4. 奇异方阵/非方阵信道的仿真

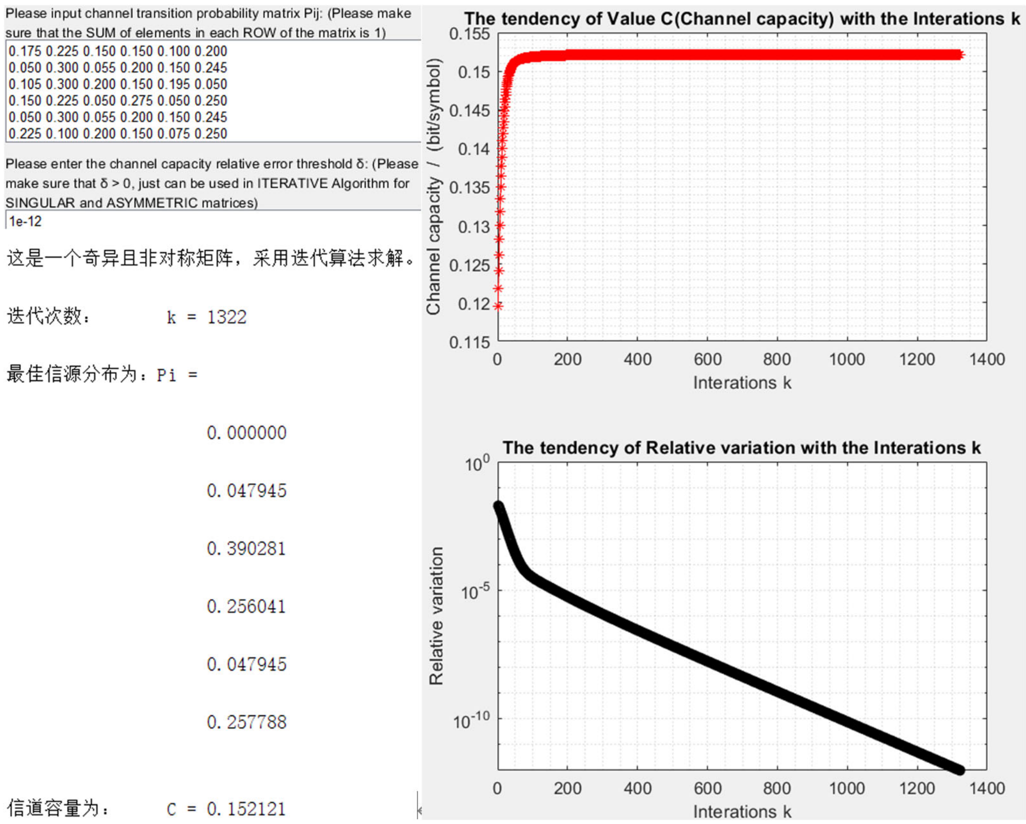


图 13. 奇异方阵

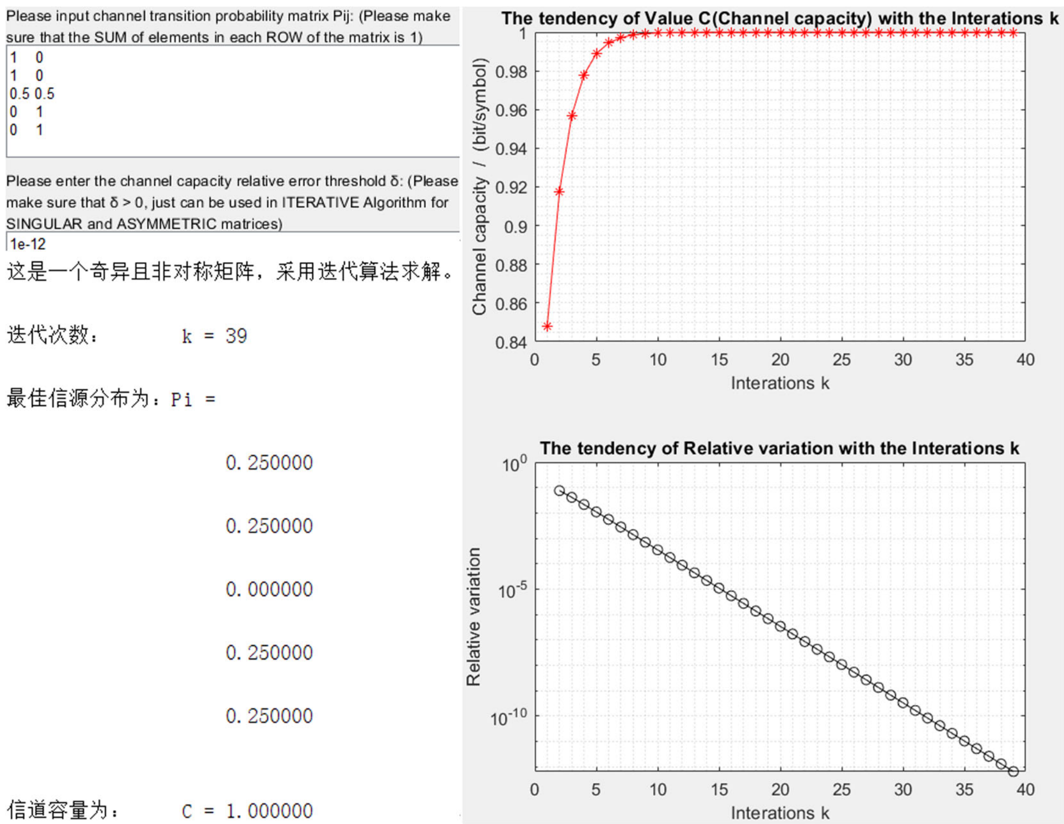


图 14. 非方阵