深 圳 大 学 实 验 报 告

课程名称：智能网络与计算

实验项目名称：实验三 物理层信道容量分析实验

学院：计算机与软件学院

专业：计算机科学与技术

指导教师：车越岭

报告人：吴嘉楷 学号：2022150168 班级：国际班

实验时间：2024 年 10 月 23 日

实验报告提交时间：2024 年 10 月 25 日

教务处制

实验三 物理层信道容量分析实验

实验目的与要求：

1. 了解什么是凸优化问题；
2. 学会使用 Matlab CVX 工具箱解决最优功率分配问题，使得信道容量最大化；
3. 了解注水算法；

方法、步骤：

电脑， Matlab， CVX 工具箱

实验过程及内容：

1. 安装 matlab 软件

由于 matlab 正版软件需要付费激活，但好在深圳大学内部网存在 matlab 正版软件资源，因此直接在深圳大学内部网 => 正版软件 => matlab 下载即可。

下载链接：http://ms.szu.edu.cn/soft/detail/30

2. 安装 CVX 工具箱

（1） 访问 CVX 官网，下载压缩包

官网链接：https://cvxr.com/cvx/download/

（2） 解压文件夹并在命令行工具中 cd 进入解压目录

（3） 输入 cvx_setup 配置 CVX 路径

图 1 配置 CVX

3. 最优功率分配问题描述

考虑T = 10 个时隙。在每个时隙 i,发射机的发射功率为 Pi（W），发射机到接收机的信道状态与接收机的背景噪声的比值为 ai。假设单位带宽，则收发机之间 T 个时隙的总信道容量可表示为$maximize \Sigma_{i=1}^{T} \log_2(1+P_i a_i)$。

考虑发射机的发射功率之和不能超过 Pmax=1（W）。发射机的最优功率分配问题可以表示成如下的凸优化问题：

$$maximize \Sigma_{i=1}^{T} \log_2(1+P_i a_i) \tag{1}$$

$$\text{subject to:} \ \mathrm{P_i} \geq 0, \ \Sigma_{\mathrm{i=1}}^{\mathrm{T}} \mathrm{P_i} = 1 \tag{2}$$

4. 打开 matlab 软件，新建方法一的脚本：使用 CVX 求解

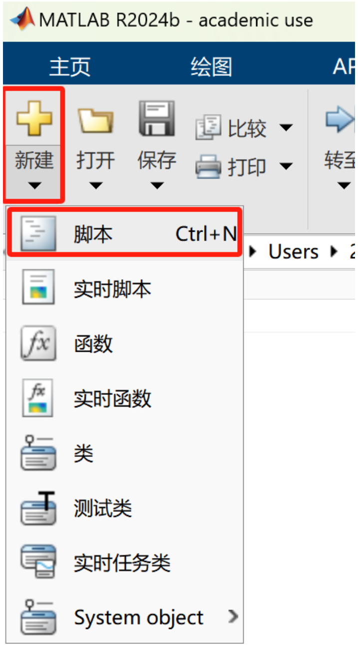
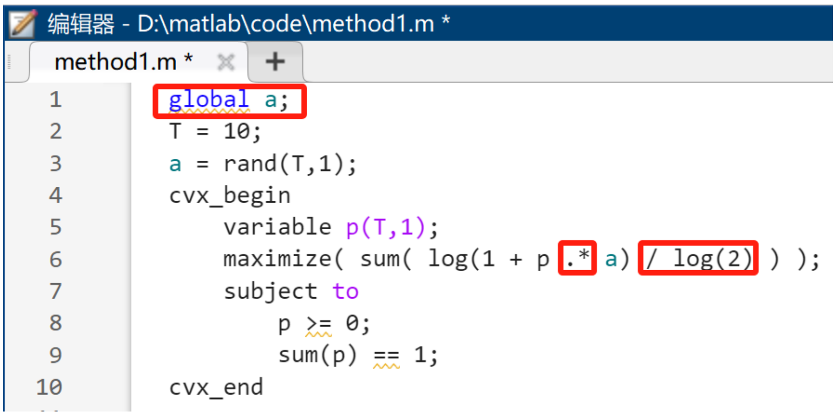


图 2 新建 method1.m

5. 编写方法一的脚本代码

根据最优功率分配问题描述，得到了方程（1）以及约束条件（2），因此可以定义优化问题中的目标函数和约束条件。设定 T = 10，生成范围为 [0, 1) 的随机信道状态比值，并显示最终结果。

脚本代码如图 3 所示：



```
global a;
T = 10;
a = rand(T,1);
cvx_begin
    variable p(T,1);
    maximize( sum( log(1 + p .* a) / log(2) ) );
    subject to
        p >= 0;
        sum(p) == 1;
cvx_end
```

图 3 方法一的代码

其中，使用 global 声明 a 是为了控制变量，使得方法一和方法二可以使用同样的数据集，从而更好的进行对比。

使用“.*”是因为 p_i 和 a_i 是逐个相乘的，而不是矩阵相乘，故不可以直接使用“*”。

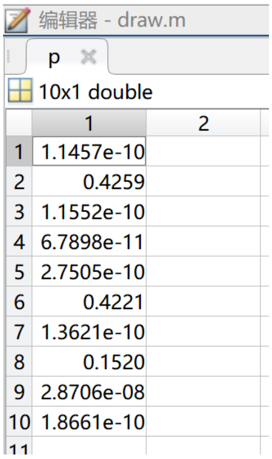
还要除以“log(2)”是因为 log()方法是以 e 为底的，利用换底公式，可以把对数的底数换成 2。

6. 运行脚本一得到 10 个最优值 p_i

（1） 点击菜单栏中的运行按钮

图 4 运行脚本一

（2） 得到 10 个最优值



编辑器 - draw.m

p

10x1 double

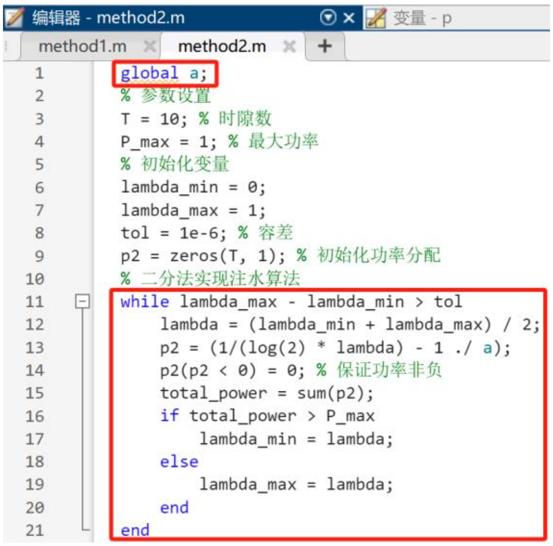
	1	2
1	1.1457e-10	
2	0.4259	
3	1.1552e-10	
4	6.7898e-11	
5	2.7505e-10	
6	0.4221	
7	1.3621e-10	
8	0.1520	
9	2.8706e-08	
10	1.8661e-10	

图 5 最优功率分配

7. 编写方法二的脚本代码

程序通过二分法实现了注水算法，以求解最优功率分配问题。首先，global a; 表示信道增益 a 是由方法一的脚本生成的全局变量。然后定义了时隙数 T 和最大总功率 P_max。程序利用 lambda_min 和 lambda_max 初始化注水线的上下界，通过二分法调整注水线 lambda 使总功率接近 P_max。在每次迭代中，计算各时隙的功率分配 P，并根据总功率大小调整 lambda 的取值，直到误差小于设定的容差 tol。

脚本代码如图 6 所示：



编辑器 - method2.m　　变量 - p

method1.m　method2.m

```
global a;
% 参数设置
T = 10; % 时隙数
P_max = 1; % 最大功率
% 初始化变量
lambda_min = 0;
lambda_max = 1;
tol = 1e-6; % 容差
p2 = zeros(T, 1); % 初始化功率分配
% 二分法实现注水算法
while lambda_max - lambda_min > tol
    lambda = (lambda_min + lambda_max) / 2;
    p2 = (1/(log(2) * lambda) - 1 ./ a);
    p2(p2 < 0) = 0; % 保证功率非负
    total_power = sum(p2);
    if total_power > P_max
        lambda_min = lambda;
    else
        lambda_max = lambda;
    end
end
```

图 6 方法二的代码

其中，使用了 global 声明的全局变量 a，目的是为了控制变量，使得方法一和方法二可以使用同样的数据集，从而更好的进行对比。

红色框圈出的循环部分为注水算法迭代调整 lambda 值的核心，即二分法。起初，注水线的上下界分别设置为 lambda_min = 0 和 lambda_max = 1。在每次迭代中，通过取二者的中值 lambda = (lambda_min + lambda_max) / 2 来更新注水线。根据这个 lambda，利用公式计算每个时隙的功率 p_i。

接着，程序将所有时隙的功率分配 p_i 求和，得到总功率 total_power。如果总功率大于 P_max，说明注水线太低，需要增加注水线值，因此将 lambda_min 更新为当前的 lambda；反之，则需要减少注水线，因此更新 lambda_max。这个过程反复进行，直到上下界的差值小于设定的容差 tol，意味着已经找到了使总功率等于 P_max 的最优注水线 lambda。

8. 运行脚本二得到 10 个最优值 p2

变量 - p2

p2

10x1 double

	1	2
1	0	
2	0.4259	
3	0	
4	0	
5	0	
6	0.4221	
7	0	
8	0.1520	
9	0	
10	0	
11		

图 7 最优功率分配（二）

9. 对比两个方法得到的最优值 p_i

编辑器 - draw.m

p

10x1 double

	1	2	3
1	1.1457e-10		
2	0.4259		
3	1.1552e-10		
4	6.7898e-11		
5	2.7505e-10		
6	0.4221		
7	1.3621e-10		
8	0.1520		
9	2.8706e-08		
10	1.8661e-10		
11			
12			
13			
14			
15			
16			
17			

变量 - p2

p2

10x1 double

	1	2	3
1	0		
2	0.4259		
3	0		
4	0		
5	0		
6	0.4221		
7	0		
8	0.1520		
9	0		
10	0		
11			
12			
13			
14			
15			
16			
17			

图 8 方法一、二的运行结果对比

由上图可见，方法一（CVX）和方法二（注水算法）二者得到的最优功率分配情况几乎一至，只是精度上有所差异而已。由此，也印证了脚本二代码设计的成功。

10. 编写画图代码

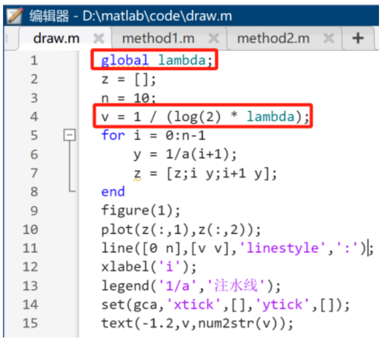


图 9 绘图代码

程序首先定义了全局变量 lambda，然后初始化了一个空数组 z，它将用于存储生成的坐标数据。变量 n 表示循环的次数，而 v 是一个与 lambda 相关的常数，计算方式为 **v = 1 / (log(2) * lambda)**，这是**注水线的高度**。

在接下来的 for 循环中，代码遍历从 0 到 n-1，并在每次迭代中，计算出函数 y = 1/a(i+1) 的值，随后将当前迭代的索引 i、计算出的 y 值、以及它们对应的坐标对 [i y] 和 [i+1 y] 添加到数组 z 中。这些数据点表示某种递减函数 1/a(i+1) 在各个离散点的取值。

接着，程序使用 plot 函数绘制这些生成的数据点，并用 line 函数在图中画出一条与 x 轴平行的虚线，代表注水线 v，这条线表示一个常数值，用于和函数值进行对比。

xlabel('i') 给 x 轴标上标签 "i"，legend('1/a','注水线') 设置图例，分别标注函数曲线和注水线。set(gca,'xtick',[],'ytick',[]) 则隐藏了 x 轴和 y 轴的刻度。最后，text(-1.2,v,num2str(v)) 在图形的某个位置 (-1.2, v) 标注注水线的数值。

11. 绘图结果

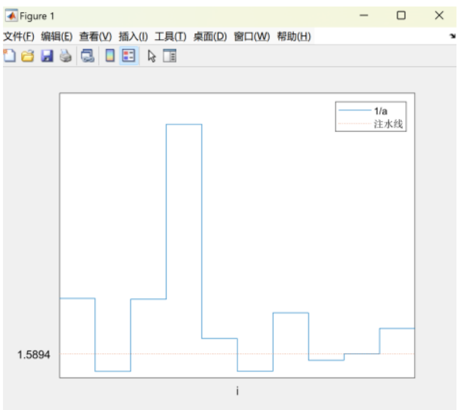


图 10 结果图

思考题：

1. 思考为什么称作注水算法。注水线，信道状态 *a*，与功率分配的关系如何？

注水算法之所以被称为“注水”，是因为该算法类似于向容器中注水。信道增益 a_i 较大的时隙可以被认为是“深”的容器，在这些时隙中需要分配更多的功率，而 a_i 较小的时隙则分配较少功率或不分配功率。

注水线反映了功率分配的标准：如果信道增益 a_i 足够高，功率分配 P_i 会超过注水线；如果信道增益低于注水线，功率分配会被设为零。这种方式可以保证总功率分配的最优。

2. 找出注水线的具体值

注水线：$\frac{1}{\ln 2 \times \lambda}$

由上述公式可知，我们只需要求出 λ 的值即可求出注水线的值。在脚本二中，我们使用了二分法去不断地缩小 λ 的上下界，最终迭代得到一个满足约束条件的 λ 值。

二分算法如图 11：

```
while lambda_max - lambda_min > tol
    lambda = (lambda_min + lambda_max) / 2;
    p2 = (1/(log(2) * lambda) - 1 ./ a);
    p2(p2 < 0) = 0; % 保证功率非负
    total_power = sum(p2);
    if total_power > P_max
        lambda_min = lambda;
    else
        lambda_max = lambda;
    end
end
```

图 11 二分法求 λ

然后，我们利用注水线的公式即可计算得出具体值：1.5894（在本数据集中）。

3. 利用 Matlab 画出类似于下图的结果图

（1） 编写画图代码

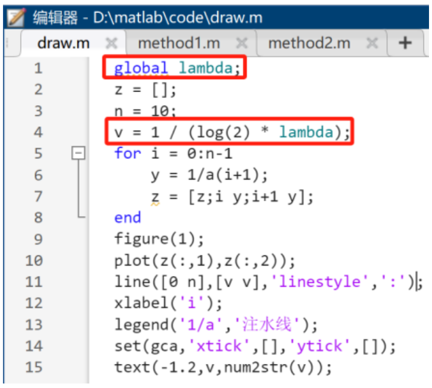


图 12 画图代码

红色框圈出的部分为修改点，其他部分基本与示例代码一致。

（2）运行代码得出结果图

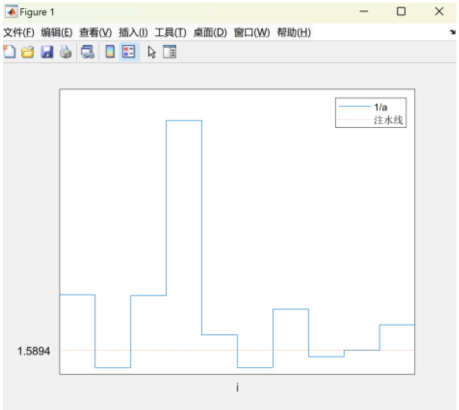


图 13 注水算法结果图

实验结论：

1. 成功应用 Matlab CVX 工具箱求解最优功率分配

利用 Matlab 中的 CVX 工具箱，我们有效地将复杂的最优功率分配问题建模为凸优化问题，并成功求解，得出最优功率分配。

2. 验证了 CVX 求解结果的正确性

通过经典的注水算法，我们对 CVX 工具箱的求解结果进行了验证。实验结果显示，两种方法得出的最优功率分配值在大多数情况下高度一致，这不仅验证了 CVX 工具箱求解的准确性，也体现了凸优化方法在通信系统设计中的可靠性。

3. CVX 工具箱的有效性

CVX 工具箱的使用显著简化了最优功率分配问题的编程过程。通过高级的数学建模和求解功能，我们避免了手动编写复杂的优化算法，从而提高了开发效率和代码的可维护性。

4. 深入理解功率分配与信道状态的关系

通过实现注水算法，我们进一步加深了对功率分配与信道状态之间关系的理解。这一算法的实现不仅验证了理论模型，还为后续的系统设计和优化提供了重要的参考。

5. 学会利用二分法求解注水线

在限定总功率的条件下，我们采用二分法成功求解了注水线，实现了功率分配的最优化。这一过程展示了计算机科学中数值求解方法在实际问题中的应用，以及如何通过算法设计来满足特定的系统约束。

心得体会：

通过这次实验，我深入理解了凸优化问题的基本概念以及 Matlab 在处理这些问题时的强大功能。特别是在解决最优功率分配问题时，利用 CVX 工具箱能够迅速地求解出最优值，而注水算法则进一步加深了我对算法求解过程的理解。在编写注水算法时，二分法的应用不仅有效解决了最优功率分配问题，还让我对数值迭代和容差控制有了更深刻的体会。

同时，我也领悟到，Matlab 作为一款强大的工具，极大地简化了数学建模和仿真过程，让我对未来更多的数学问题和算法实现充满期待。这次实验为我后续的学习打下了坚实的基础。

然而，实验中也遇到了一些难点：

1. **Matlab CVX 工具箱的安装与配置**：在安装 CVX 时，出现 cvx_setup 命令不被识别的错误，重新下载 CVX 后方才解决问题。
2. **对 Matlab 语法的不熟悉：**在编写方法一代码时，出现过 p * a 报错，才发现此处应该使用 .* 而不是矩阵乘法，不然需要检验矩阵的维度。
3. **编写注水算法**：如何在二分法中合理设定 lambda 上下界以及准确控制循环迭代次数是一个挑战。通过逐步调整上下界参数和反复运行代码，最终达到了预期的效果。

指导教师批阅意见：

成绩评定： 分

指导教师签字：

年　　月　日
备注：