

一、函数说明及调用关系

1.1 UD.h

本头文件用于存放均匀随机数相关函数

函数名	函数类型	输入	输出	说明
get_UDS()	float*	/	随机数数组的首地址	产生 (0, 1) 之间的随机数
USD2txt	void	/	/	产生 (0, 1) 之间的随机数并存入 txt 文件.

调用关系



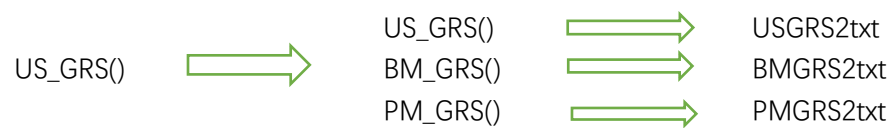
1.2 GD.h

本头文件用于存放与高斯随机数相关的函数

函数名	函数类型	输入	输出	说明
US_GRS()	float*	N	存放高斯随机数数组的首地址	使用均匀变量求和法产生 N 个服从标准正态分布的高斯随机数
USGRS2txt	void	/	/	使用均匀变量求和法产生 N 个服从标准正态分布的高斯随机数并将其存入 txt 文件
BM_GRS()	float*	N	存放高斯随机数数组的首地址(其中偶数地址存放 X,奇数地址存放 Y)	使用 Box-Muller 算法分别产生 N 个服从标准正态分布的高斯随机数 X 和 Y
BMGRS2txt	void	/	/	使用 Box-Muller 算法分别产生 N 个服从标准正态分布的高斯随机数并分别存入 txt 文件.

PM_GRS()	float*	N	存放高斯随机数数组的首地址(其中首地址存放生成的高斯随机数个数,其他奇地址存放 X,偶数地址存放 Y)	使用极坐标法进行 N 次尝试,分别生成不确定目个服从标准正态分布的高斯随机数 X 和 Y
PMGRS2txt	void	/	/	使用极坐标法进行 N 次尝试,分别生成不确定目个服从标准正态分布的高斯随机数 X 和 Y, 并并存入 txt 文件.
Rayleigh_RS	float*	N	存放 N 个服从瑞利分布的随机数(sigma = 1)数组的首地址。	生成 N 个服从瑞利分布的随机数(sigma = 1)
Rayleigh_RS2txt	void	N	/	生成 N 个服从瑞利分布的随机数(sigma = 1)并存入 txt

函数调用关系

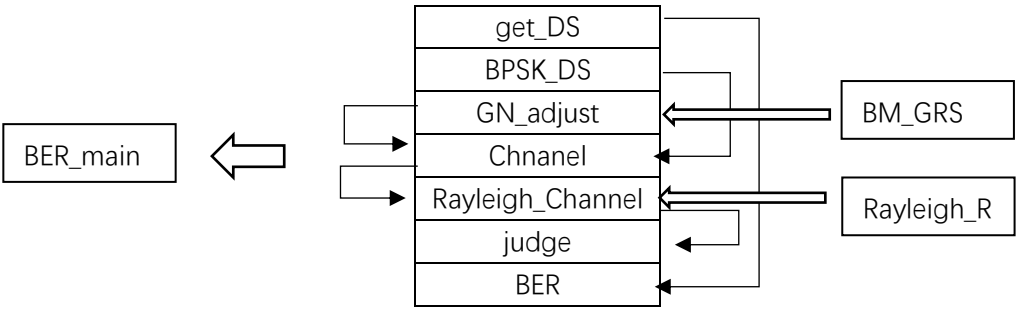


1.3 BPSK.h

函数名	函数类型	输入	输出	说明
get_DS	Int*	信源长度 N	原始信信息数组的首地址	获取随机信源
BPSK_DS	Int*	信源 DS[], 信源长度 N	调制后信源数组的首地址	DS 为输入信源, +1、-1 表示相位
Chnanel	float*	待处理信号 coding[], 信道噪声 N[], 信息长度 length 信道种类 style	调制后数组的首地址	信道函数
Rayleigh_Channel	float*	待处理信号 coding[], 信道噪声 r[], 信息长度 length	调制后数组的首地址	重写衰落信道

GN_adjust	float*	标准正态分布数组 GN[], 信号长度 N, 信噪比 SNR_dB	满足信噪比条件的高斯随机数数组的首地址	高斯噪声修正函数
judge	int*	receive_coding[], 信号长度 N	判决结果首地址	判决函数
BER	float	原始信号 S[] 判决信号 R[] 信号长度 N	误码率	误码率计算函数
BER_main	float	信号长度 N 信噪比 SNR_dB	误码率	仿真主程序

函数调用关系图：

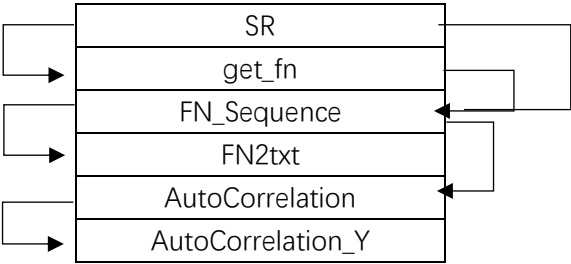


1.4 FN.h

本头文件用于存放与伪随机噪声相关的函数

函数名	函数类型	输入	输出	说明
SR	void	初始数组, 寄存器长度, 反馈值 fn	/	N 位随机寄存器(直接在地址上操作)
get_fn	int	上一次寄存器中的数据	反馈值 fn	利用 10 位寄存器和上一次寄存器中的数据生成反馈值 fn
FN_Sequence	Int*	长度 N	伪噪声序列数组的首地址	生成长度为 N 的伪噪声序列
FN2txt	void	/	/	生成长度为 N 的伪噪声序列并存入 txt 中
AutoCorrelation	Int	随机序列 FN[] m, 长度 N	自相关函数值	计算第 m 点对应的自相关函数值
AutoCorrelation_Y	void	长度 N	/	得到全部点对应的自相关函数值并存入 txt 文件

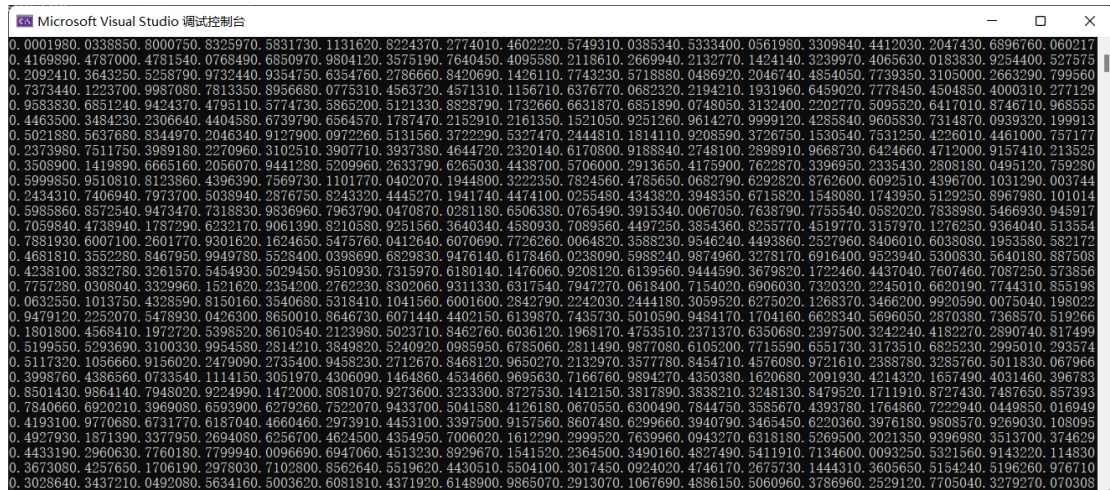
函数调用关系图:



二、仿真结果及解释

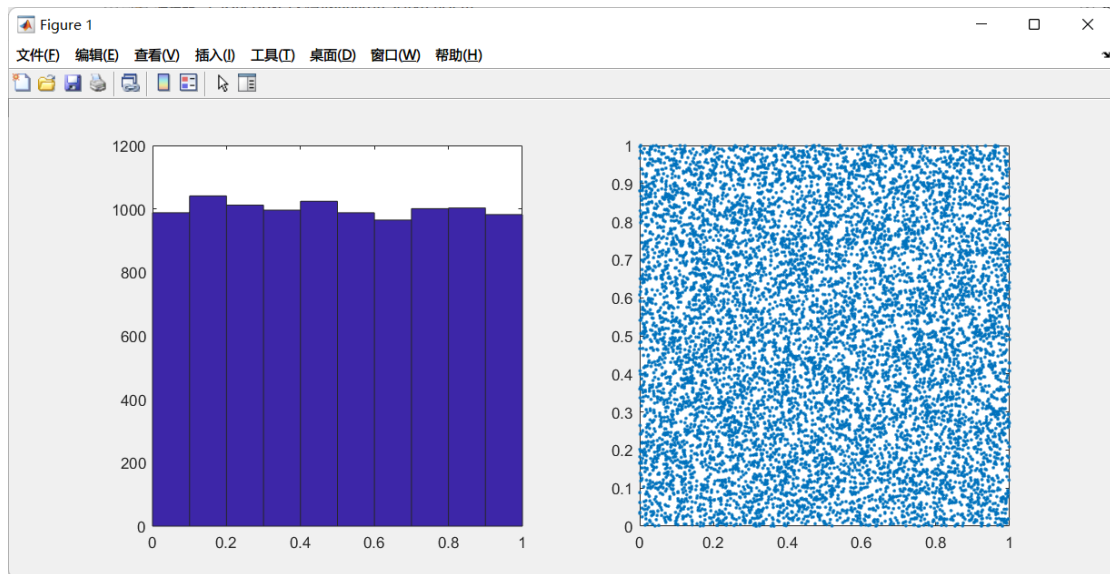
2.1 随机数仿真结果

使用 WH 算法生成 $(0, 1)$ 之间的随机数，并存入 txt 文件



```
Microsoft Visual Studio 调试控制台
0.0001980, 0.038850, 0.000750, 0.8325970, 0.5831730, 1.131620, 0.8224370, 0.2774010, 0.4602220, 0.5749310, 0.0385340, 0.5333400, 0.0561980, 0.3309840, 0.4412030, 0.2047430, 0.6896760, 0.060217
0.4169890, 0.4787000, 0.4781540, 0.0768490, 0.6850970, 0.9804120, 0.3575190, 0.7640450, 0.4095580, 0.2118610, 0.2669940, 0.2132770, 0.1424140, 0.3239970, 0.4065630, 0.0183830, 0.9254400, 0.527575
0.2092410, 0.3643250, 0.5258790, 0.9732440, 0.9354750, 0.6354760, 0.2786660, 0.8420690, 0.1426110, 0.7743230, 0.5718880, 0.0486920, 0.2046740, 0.4854050, 0.7739350, 0.3105000, 0.2663290, 0.799560
0.7373440, 0.1223700, 0.9987080, 0.7813350, 0.8956680, 0.0775310, 0.4563720, 0.4571310, 0.1156710, 0.6376770, 0.0682320, 0.2194210, 0.1931960, 0.6459020, 0.7778450, 0.4504850, 0.4000310, 0.277129
0.9583830, 0.6851240, 0.9424370, 0.4795110, 0.5774730, 0.5865200, 0.5121330, 0.8828790, 0.1732660, 0.6631870, 0.6851890, 0.0748050, 0.3132400, 0.2202770, 0.5095520, 0.6417010, 0.8746710, 0.968555
0.4463500, 0.3484230, 0.2306640, 0.4404580, 0.6739790, 0.6564570, 0.1787470, 0.2152910, 0.2161350, 0.1521050, 0.9251260, 0.9614270, 0.9999120, 0.4285840, 0.9605830, 0.7314870, 0.0939320, 0.199913
0.5021880, 0.5637680, 0.8344970, 0.2046340, 0.9127900, 0.0972260, 0.5131560, 0.5722290, 0.5327470, 0.2444810, 0.1814110, 0.9208590, 0.5726750, 0.1530540, 0.7531250, 0.4226010, 0.4461000, 0.757177
0.2373990, 0.7511750, 0.999180, 0.2070960, 0.3102510, 0.3907710, 0.3937280, 0.4644720, 0.2320140, 0.6170800, 0.9188340, 0.2748100, 0.2989910, 0.9668730, 0.6424660, 0.4712000, 0.8157410, 0.213525
0.3508900, 0.1419890, 0.6665160, 0.2056070, 0.9441280, 0.5209960, 0.2633790, 0.6265030, 0.4438700, 0.5706000, 0.2913650, 0.4175900, 0.7622870, 0.3986950, 0.2335430, 0.2808180, 0.0495120, 0.759280
0.5998850, 0.9510810, 0.9123860, 0.4986390, 0.7569730, 0.1101770, 0.0402070, 0.1944800, 0.3222350, 0.7824560, 0.4785650, 0.0682790, 0.6292820, 0.8762600, 0.6092510, 0.4396700, 0.1031290, 0.003744
0.2434310, 0.7406940, 0.7973700, 0.5038940, 0.2876750, 0.8243320, 0.4445270, 0.1941740, 0.4474100, 0.0255480, 0.4343820, 0.3948350, 0.6715820, 0.1548080, 0.1743950, 0.5129250, 0.8967980, 0.101014
0.5985860, 0.8572540, 0.9473470, 0.7318830, 0.9836960, 0.7963790, 0.0470870, 0.0281180, 0.6506380, 0.0765490, 0.3915340, 0.0067050, 0.7638790, 0.7755540, 0.0582020, 0.7838980, 0.5466930, 0.943917
0.7059840, 0.4738940, 0.1787290, 0.632170, 0.9061390, 0.8210580, 0.9251560, 0.3640340, 0.4580930, 0.7089560, 0.4497250, 0.3854360, 0.8256770, 0.4519770, 0.3157970, 0.1276250, 0.9364040, 0.513554
0.7881930, 0.6007100, 0.2601770, 0.9301620, 0.1624650, 0.5475760, 0.0412640, 0.6070690, 0.7726260, 0.0064820, 0.3588230, 0.9546240, 0.4493860, 0.2527960, 0.8406010, 0.6038080, 0.1953580, 0.582172
0.4681810, 0.3552280, 0.8467950, 0.9949780, 0.5528400, 0.0398690, 0.6829830, 0.9476140, 0.6178460, 0.0238090, 0.5988240, 0.9874960, 0.3278170, 0.6916400, 0.9523940, 0.5300830, 0.5640180, 0.887508
0.4238100, 0.3832780, 0.3261570, 0.5454930, 0.5029450, 0.9510930, 0.7315970, 0.6180140, 0.1476060, 0.9208120, 0.6139560, 0.9444590, 0.3679820, 0.1722460, 0.4437040, 0.7607460, 0.7087250, 0.573856
0.7757280, 0.0308040, 0.3329960, 0.1521620, 0.2354200, 0.2762230, 0.8302060, 0.9311330, 0.6317540, 0.7947270, 0.0618400, 0.7154020, 0.6906030, 0.7320320, 0.2245010, 0.6620190, 0.7744310, 0.855198
0.0632550, 0.1013750, 0.4328590, 0.8150160, 0.3540680, 0.5318410, 0.1041560, 0.6001600, 0.2842790, 0.2242030, 0.2444180, 0.3059520, 0.6275020, 0.1268370, 0.3466200, 0.9920590, 0.075040, 0.198022
0.9479120, 0.2252070, 0.5478930, 0.0426300, 0.8650010, 0.8646730, 0.6071440, 0.4402150, 0.6139870, 0.7435730, 0.5010590, 0.9484170, 0.1704160, 0.6628340, 0.5696050, 0.2870380, 0.7368570, 0.519266
0.1801800, 0.4568410, 0.1972720, 0.5398520, 0.8610540, 0.2123980, 0.5023710, 0.8462760, 0.6036120, 0.1968170, 0.4753510, 0.2371370, 0.6350680, 0.2397500, 0.3242240, 0.4182270, 0.2890740, 0.817499
0.5199550, 0.5293690, 0.3100330, 0.9954580, 0.2814210, 0.3849820, 0.5240920, 0.0985950, 0.6785060, 0.2811490, 0.9877080, 0.6105200, 0.7715590, 0.6551730, 0.3173510, 0.6825230, 0.2995010, 0.293574
0.5117320, 0.1056660, 0.9156020, 0.2479090, 0.2735400, 0.9458230, 0.2712670, 0.8468120, 0.9650270, 0.2132970, 0.3577780, 0.8454710, 0.4576080, 0.9721610, 0.2388780, 0.3285760, 0.5011830, 0.067966
0.3998760, 0.4386560, 0.0733540, 0.114150, 0.3051970, 0.4306090, 0.1464860, 0.4534660, 0.9695630, 0.7166760, 0.9894270, 0.4350380, 0.1620680, 0.2091930, 0.4214320, 0.1657490, 0.4031460, 0.396783
0.8501430, 0.9864140, 0.7948020, 0.9224990, 0.1472000, 0.8081070, 0.9273600, 0.3233300, 0.8727530, 0.1412150, 0.3817890, 0.3838210, 0.3248130, 0.8479520, 0.1711910, 0.8727430, 0.7487650, 0.857393
0.7840660, 0.6920210, 0.3969080, 0.6593900, 0.6279260, 0.7522070, 0.9433700, 0.5041580, 0.4126180, 0.0670550, 0.6300490, 0.7844750, 0.585670, 0.4393780, 0.1764860, 0.7222940, 0.0449550, 0.016949
0.4193100, 0.9770680, 0.6731770, 0.6197040, 0.4660460, 0.2973910, 0.4453100, 0.3397500, 0.9157560, 0.807480, 0.6299660, 0.3940790, 0.3465450, 0.6220360, 0.3976180, 0.9808570, 0.2689030, 0.108955
0.4927930, 0.1871390, 0.3377850, 0.2694080, 0.6256700, 0.4624500, 0.4354950, 0.7066020, 0.1612290, 0.2999520, 0.7639960, 0.0943270, 0.6318180, 0.5269500, 0.2021350, 0.9396980, 0.3513700, 0.746229
0.4433190, 0.2960630, 0.7760180, 0.7799940, 0.0996690, 0.6947060, 0.4513230, 0.8929670, 0.1541520, 0.2364500, 0.3490160, 0.4827490, 0.5411910, 0.7134600, 0.093250, 0.5321560, 0.9143220, 0.114830
0.3673080, 0.4257650, 0.1706190, 0.2978030, 0.7102800, 0.8562640, 0.5519620, 0.4430510, 0.5504100, 0.3017430, 0.0924020, 0.4746170, 0.2675730, 0.1444310, 0.3605650, 0.5154240, 0.5196260, 0.976710
0.3028640, 0.3437210, 0.0492080, 0.5634160, 0.5003620, 0.6081810, 0.4371920, 0.6148900, 0.9865070, 0.2913070, 0.1067690, 0.4886150, 0.5060960, 0.3786960, 0.2529120, 0.7705040, 0.3279270, 0.070308
```

读入 matlab 文件，绘制直方图和散点图



可以看到不同区间的点数量基本相同， (x_n, x_{n+1}) 的可能情况也基本都有出现，证明这些数是在 $(0, 1)$ 之间均匀分布的并且不相关。

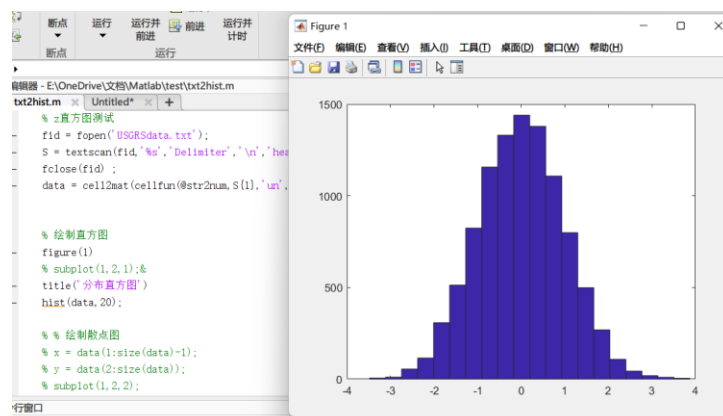
2.2 高斯随机数和瑞利随机数的生成

2.2.1 均匀求和法

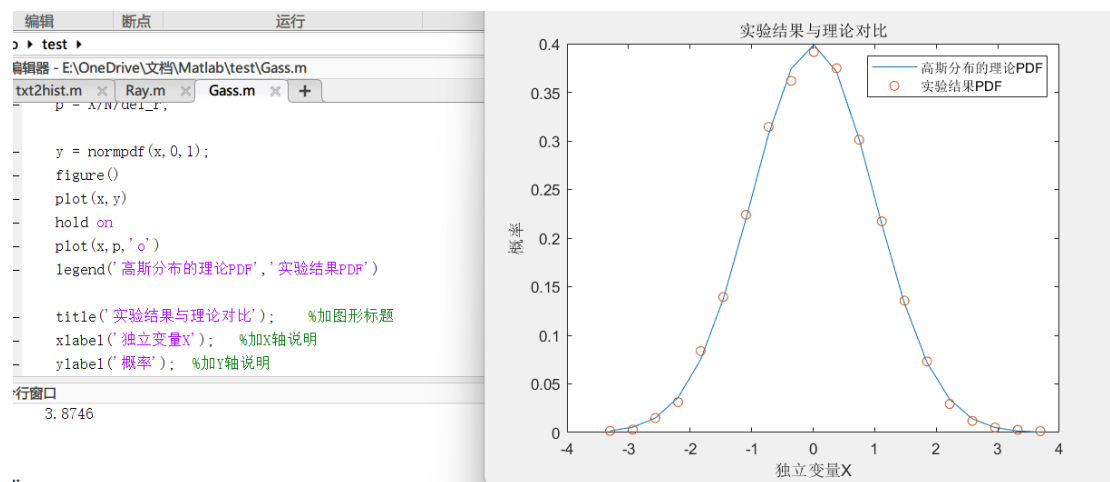
生成随机数并写入 txt 文件

```
Microsoft Visual Studio 调试控制台
0.930045 -1.100730 0.301890 0.453338 -1.300901 1.220908 0.324326 0.063241 -0.120013 -0.779659 0.400694 -0.556566 0.095375 -1.452443
0.661728 1.426666 -0.925226 1.000422 0.460217 -0.334991 1.116465 1.086234 1.003703 -0.231605 0.237198 -1.938289 1.347710 -0.606387 -0.414593 0.2
0.01788 0.019953 0.060824 -0.530236 0.267536 0.853126 1.254242 -0.789644 0.967662 -1.169144 0.026368 -0.608039 -0.757772 -0.847921 -0.768277 -0.05
0.997 -0.384288 0.178487 0.641753 0.479319 -0.033660 1.257837 -1.185901 0.036164 -1.443154 0.099901 1.082076 0.629460 0.737060 -1.634989 0.486868
-0.429881 0.663111 0.066395 0.982946 -0.644082 0.172359 0.617118 -1.370977 -0.373913 0.607858 1.189708 0.011663 0.378945 0.312095 0.537905 -0.4
11251 -0.736143 -1.182677 -0.319827 0.222118 -0.265715 0.849114 1.377044 1.450855 -0.423653 1.110344 0.320188 1.117716 0.353974 0.171164 1.91637
-1.489316 -0.723339 -2.382157 0.722772 -0.302193 0.212509 0.921447 0.928207 -1.169462 -0.344754 -0.741167 0.034419 -1.217216 0.831633 -0.113774
7 -1.307008 -0.155096 -0.667984 0.705650 0.791964 -0.166035 -0.704847 0.360793 -1.002177 0.972217 -0.196399 1.579449 0.332792 -0.516909 1.735482
0.176594 -0.780600 -0.874031 0.367072 1.129378 -1.866669 1.341544 0.812460 0.248691 -0.227927 -1.208927 -1.927345 -0.094704 -1.477454 1.001817
0.184731 1.476339 0.150007 0.078039 -0.125669 -0.272949 -0.033335 -1.772257 -0.031409 0.572337 -0.493782 -0.525232 -1.616748 -0.063745 0.388309
-0.333068 0.232362 0.307433 0.329455 -0.680885 -0.543944 -1.594646 -1.398755 0.154078 -0.250641 0.668756 -0.302497 1.743131 0.491759 -0.810383
0.261653 -0.515305 0.755714 -0.480112 -0.919073 -1.173449 0.675781 0.493541 -0.558642 1.335435 0.992176 -1.112049 -1.018909 -1.157723 0.323235 -
0.583034 1.035772 -1.497953 0.335782 0.789769 1.486974 -0.710685 0.527022 -0.487236 -1.009781 0.568336 -2.112353 -0.012700 0.056446 -1.084694 -
1.634265 0.339002 -0.807865 -0.199870 0.539503 1.179008 -0.517810 0.132909 -0.554547 0.905689 -0.678372 0.125692 -1.331707 -0.068612 -1.291893 -
2.044383 0.432602 -0.136019 0.591466 -1.136071 0.246292 -0.349710 0.710169 -0.356039 0.510002 1.063332 -0.594283 -1.208264 0.750323 0.911585 -0.
27189 -0.217559 0.042958 0.386251 -0.664664 -1.265533 -0.345051 -0.358122 0.446817 1.746016 1.087236 -1.064914 1.388122 -1.122048 -0.179865 0.7
46426 1.317701 0.232388 -1.188065 -0.517887 -0.933046 1.151658 1.74347 -1.299103 -1.508669 -2.131184 1.092604 0.738156 -0.498089 -0.469207 1.38
4978 -0.587865 -0.019631 -0.500373 -0.434732 -0.140553 -0.443357 0.050569 -0.402104 1.187636 0.935745 -0.476976 -0.903289 0.157017 0.815073 -0.6
18333 0.451719 0.628443 -0.337379 1.248817 0.146017 1.329629 0.191440 -0.097255 0.258840 -1.378978 -0.483314 0.949422 -1.467410 0.961723 0.51861
4 -0.359241 1.039370 1.513882 -1.427672 1.907303 0.924029 -0.355990 1.637119 -0.058191 -0.605708 1.232982 0.121573 1.69143 -1.327681 0.547487 0.
175981 1.006421 1.726950 2.035194 -1.093565 0.447663 1.598453 -0.629584 -0.868346 0.807932 -1.169584 -1.132448 -0.397621 -2.319853 -0.027217 -1.
972833 0.637174 -0.362512 -0.618309 1.645552 0.273355 -1.633706 0.273944 -1.799795 -0.172365 1.534504 -0.237708 -0.190875 -0.596044 0.880208 1.2
43623 0.356019 -0.269854 -1.477507 0.209456 -0.49103 0.029984 -0.535956 1.209001 -1.453963 -0.628430 1.416069 0.120748 0.523363 -0.077155 0.690
961 -1.053317 0.161133 0.358907 -0.485158 0.392320 0.498471 -0.558056 0.343282 -2.079303 -0.547701 1.551509 -0.033044 -0.161729 -0.574754 -0.452
008 -1.684275 -0.900583 1.608811 -0.301385 0.398529 -1.572047 -0.553525 0.638578 0.093616 0.011256 -0.842502 1.239172 -0.072976 0.263960 1.35821
4 -0.463367 -0.957968 -0.343301 -1.683977 0.618309 -0.033346 -0.829021 0.305972 1.535010 -0.177491 -0.302907 0.286405 -1.572444 -0.196722 -0.084
240 -0.030387 -0.904212 0.381508 0.204443 -0.373408 -0.790079 -1.415856 1.897828 -0.846931 0.327732 0.157431 -2.508450 -0.879332 -0.869848 -0.46
5795 0.397872 -0.160210 0.438566 0.932662 2.714464 0.986888 0.446375 0.640770 -0.605015 -2.097848 -0.473443 -1.258293 0.601502 -1.033310 1.02028
6 1.636467 -0.845341 0.213491 -0.711631 -0.080719 1.637793 -0.314826 -0.309417 0.294279 -1.031842 0.951881 0.098617 -0.125337 0.076506 1.15702
5 0.335854 0.544729 -0.664066 0.002678 0.468227 0.985222 0.996211 0.874695 3.163799 -0.074641 0.361038 -0.347483 0.358706 0.077376 1.431901 -0.607
```

把生成的数据 txt 读入 matlab 并绘制直方图



与标准数据进行对比

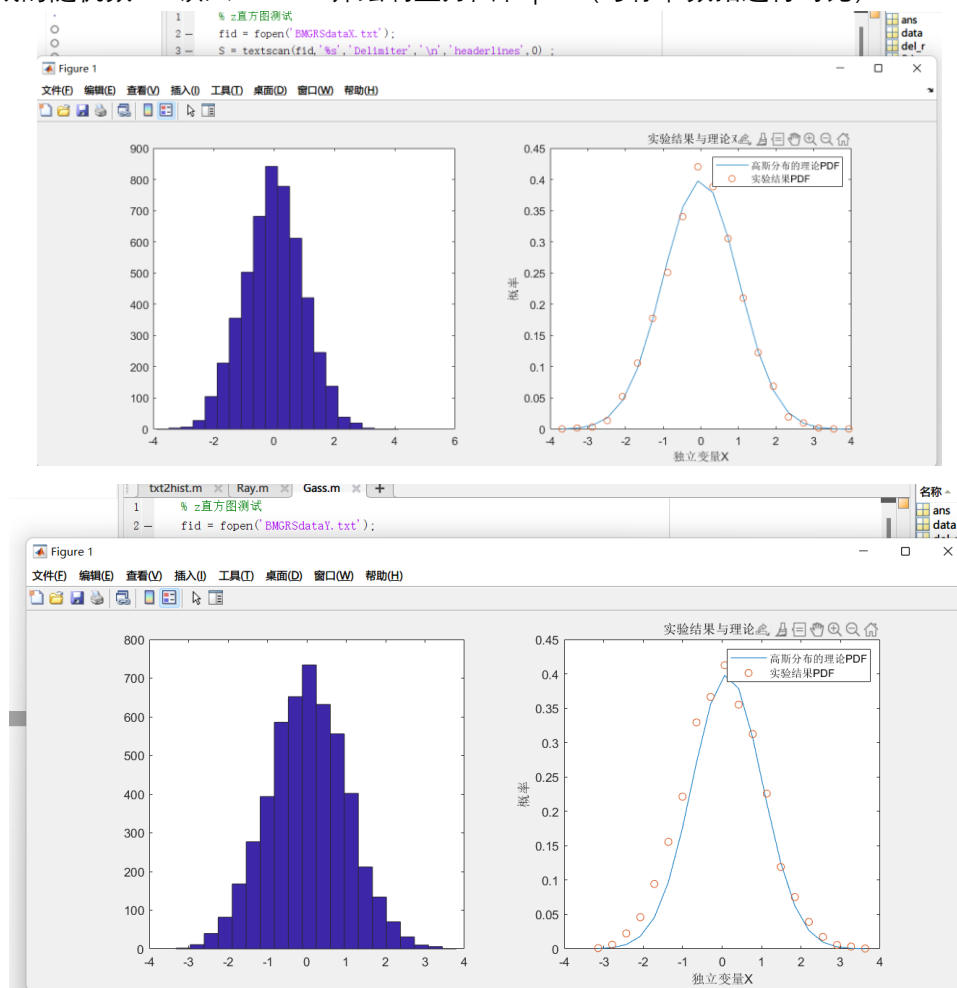


2.2.2 B-W 算法

每一次生成的一对高斯随机数如下，并分别存入 txt 文件中

```
Microsoft Visual Studio 调试控制台
4.129733      0.004853
-0.665251    -1.142642
-0.396608    -0.761419
-0.231980    -1.373745
1.235512     0.689682
2.112944     0.106977
-1.700303    0.335467
-0.372323    0.562653
-0.640974    -0.913656
-0.024371    0.380499
-0.910759    0.189076
-0.345273    -1.395394
-1.424296    0.943835
0.889634     1.069574
0.017655     1.659765
1.911822     0.223353
-0.953265    0.537766
-0.217990    -0.414168
1.496635     0.994623
-1.036030    -0.172567
0.067450     -0.278344
-1.500554     1.253740
-1.958679     0.893516
-0.147681    -0.348027
0.682651     0.618284
0.557553     -0.488316
-0.153408     0.731486
0.033844     -1.676525
0.046126     -0.003074
-0.175272    -0.931804
```

将生成的随机数 txt 读入 matlab 并绘制直方图和 pdf（与标准数据进行对比）



2.2.3 PM 算法

生成服从标准正态分布的随机数组并存入 txt

```
AS为1.307088
随机数是-0.772222

-0.442242
-0.948407
S为1.046448
跳过

0.135749
-0.182289
S为0.227282
AS为3.610709
随机数是0.490150

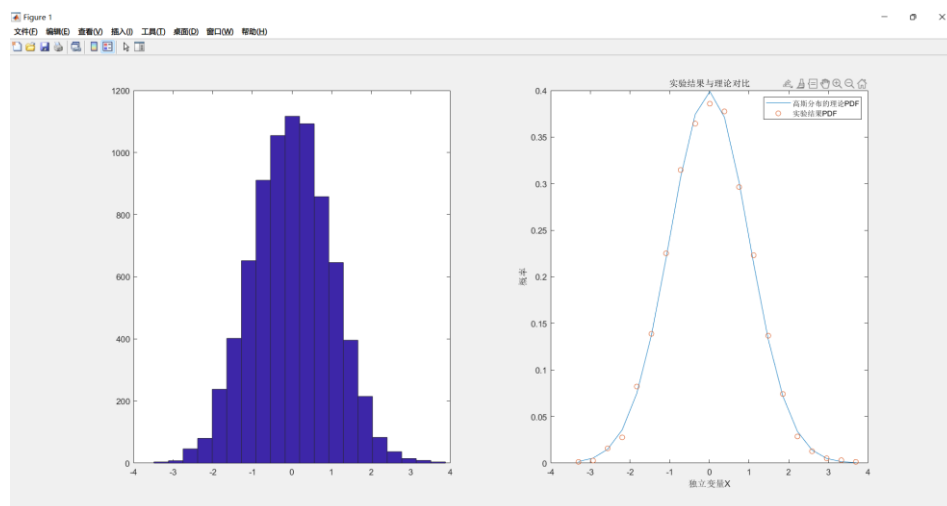
总共进行了10000次尝试，有7869次符合条件
E:\OneDrive\文档\Vs2019 Projects\Simulation Homework1
0) 已退出，代码为 0。
而在调试停止时自动关闭控制台，请启用“工具”\“选项”
```

S 理论上被拒绝的概率为 0.2146，实验中 S 被拒绝的概率为：

$$\frac{10000 - 7869}{10000} = 0.2131$$

误差 $\frac{(0.2146 - 0.2131)}{0.2146} = 0.6\%$ ，基本符合。

将生成的随机数读入 matlab 并绘制直方图和 pdf（与标准数据进行对比）

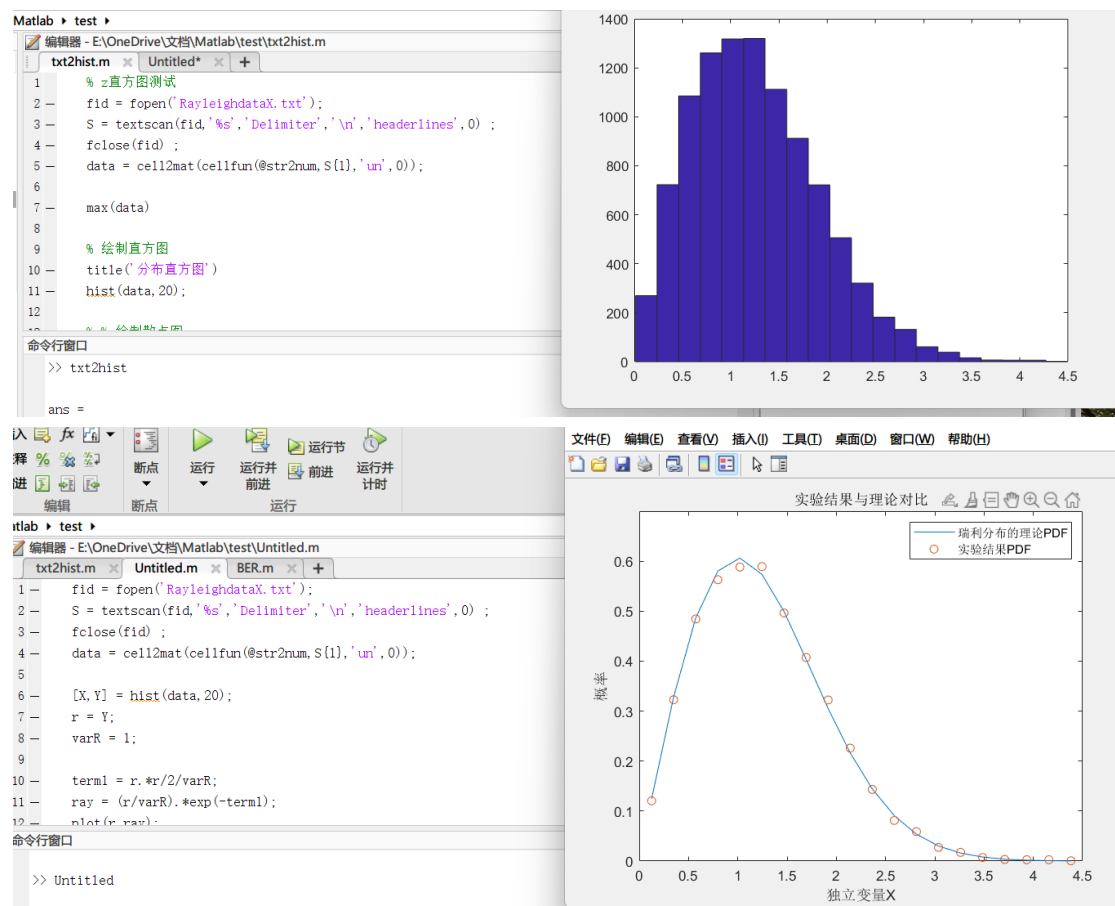


2.2.4 瑞利分布的生成

生成服从 $\sigma=1$ 的瑞利分布并存入 txt

```
Microsoft Visual Studio 调试控制台
1. 4187920. 5799201. 0795901. 5361102. 0331751. 6897810. 9640532. 2365261. 3878850. 1393130. 8494021. 1822730. 3891961. 5307531. 298449
1. 7193161. 6014901. 9709371. 2869410. 3077621. 3244431. 3867261. 5051060. 4848402. 1152940. 8944470. 6570791. 9458461. 0223471. 290161
1. 2817290. 2536761. 8584170. 7439740. 9871971. 4530072. 3105590. 8727240. 0553071. 0175850. 8772010. 5118300. 7262551. 3965721. 456801
0. 3547531. 1427490. 5623220. 3130310. 9684300. 9658371. 3677641. 7189261. 0289750. 6318751. 3966021. 7857970. 6104532. 8327640. 974948
1. 3055961. 3896671. 5170220. 9772710. 6786102. 8686730. 8961090. 6372281. 9766041. 3373290. 7244721. 3929132. 5781381. 9589590. 737909
0. 8018490. 6574122. 4378331. 4031820. 8203110. 6152050. 8973552. 1291951. 6750451. 4206290. 6888801. 2399421. 2978810. 6095851. 266284
0. 5630401. 8392921. 0219360. 5387121. 5941972. 0022651. 9611760. 8453721. 2395840. 9289230. 5450271. 4546671. 6635401. 8106220. 626462
1. 1519551. 5791941. 6390683. 2082231. 7446372. 3342961. 1852931. 3435871. 5095351. 1673081. 3786102. 5839801. 6871801. 3703281. 619155
0. 4378411. 6429300. 5880081. 7449360. 8858951. 8859171. 2643792. 1133600. 3469190. 8564532. 2606921. 1454540. 7784321. 2554502. 099874
1. 8853791. 5496391. 0024220. 4561760. 9896260. 4325371. 3390630. 8528360. 4846101. 5983141. 3389191. 2496952. 3660911. 0980891. 348230
1. 0419760. 9245471. 8196420. 0934810. 8757541. 5394780. 2789231. 5793361. 9835010. 3980090. 9840011. 8096060. 6292910. 4014051. 678844
1. 3340431. 2604470. 9081750. 9205071. 2400830. 6302302. 0836152. 1044271. 2514911. 8025101. 8074151. 6061870. 3540520. 2674571. 065466
1. 1045600. 4618700. 7575771. 1243611. 8023381. 5035620. 8942531. 0869901. 1337211. 9388130. 5551341. 2412790. 7351421. 0387301. 868972
0. 8566350. 1532241. 2512310. 3365821. 1799511. 5912901. 8810271. 3929840. 8580801. 0443571. 0256111. 5218052. 6048940. 9763131. 029112
0. 9422443. 4035392. 0630320. 9189781. 6110701. 1976891. 0709650. 7614341. 8459952. 1900120. 7589000. 4048791. 1931821. 7094141. 222138
0. 8903520. 7523411. 6991260. 5988231. 7053570. 7795600. 6177161. 5267172. 2401982. 2981491. 6851522. 0691130. 8507671. 7217190. 171479
1. 4738441. 3859471. 7149581. 4929130. 3434772. 0161280. 5287500. 3280291. 2627750. 8672511. 6100821. 4122341. 5338551. 7743481. 525659
0. 3990881. 2953800. 8865371. 7561940. 8414451. 9449130. 5686542. 6796392. 0960081. 0304951. 6580670. 9620840. 6324850. 8257641. 959052
0. 4014651. 9119651. 0565920. 3290612. 4598690. 3824120. 7174811. 8019601. 3868300. 5479511. 4045250. 8766190. 4913531. 9045730. 618303
0. 8298170. 5194681. 4446011. 2035450. 7458870. 9975930. 9713320. 3387582. 6550212. 1493240. 8097131. 0819860. 7337791. 0544970. 581316
2. 3369581. 4495891. 4654780. 4186650. 8033821. 7638951. 5734370. 8584920. 7262061. 3436081. 5923871. 0257002. 1998160. 5098531. 180600
0. 3370660. 3587841. 5796860. 6773602. 0454821. 9314370. 8612251. 2418860. 9678121. 7284891. 1564322. 0286941. 3318021. 0096210. 651798
0. 8041291. 0152891. 5723941. 3583402. 4698821. 2380551. 1155841. 4308740. 1179400. 4079031. 3411940. 9070400. 6669072. 5483680. 467642
1. 9617000. 2638891. 0757041. 4666142. 3694391. 4486830. 3176801. 4675451. 1620010. 4601960. 7431351. 3806182. 1679520. 4557700. 894173
0. 7794041. 0418751. 0842931. 5294591. 8804560. 5971470. 5816121. 7813971. 5981101. 063819
E:\OneDrive\文档\2019 Projects\Simulation Homework1 Random sequence\Random sequence\Debug\RandomSequence.exe (进程 3372)
已退出。 代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . . .
```

将生成的随机数 txt 读入 matlab，并绘制直方图和 pdf（与标准数据进行对比）



2.3 BPSK 误码率

仿真主程序结果，误码率和对应的噪声功率如下，这里令 E_b 为 1，将 ENR 的单位从分贝转化后可以得到对应的噪声功率。计算信噪比从-3dB 到 13dB。

```
Microsoft Visual Studio 调试控制台
噪声功率为1.995262
误码率为0.212700
噪声功率为1.584893
误码率为0.187600
噪声功率为1.258925
误码率为0.166000
噪声功率为1.000000
误码率为0.144700
噪声功率为0.794328
误码率为0.126100
噪声功率为0.630957
误码率为0.101300
噪声功率为0.501187
误码率为0.090300
噪声功率为0.398107
误码率为0.074900
噪声功率为0.316228
误码率为0.060500
噪声功率为0.251189
误码率为0.050100
```

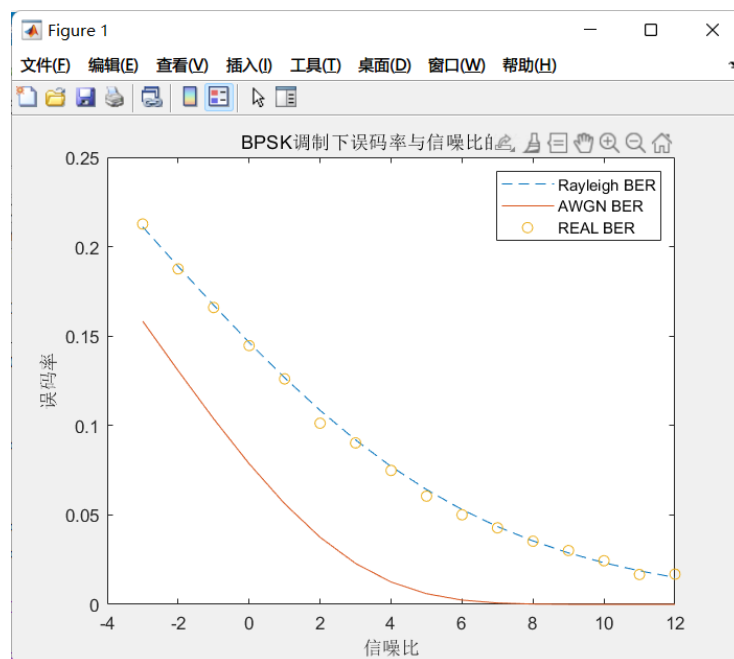
BPSK 在高斯信道下的理论误码率为:

$$P_e = Q(\sqrt{SNR}) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

考虑衰落信道后理论码率变为:

$$P_e = \frac{1}{2} \left(1 - \sqrt{\frac{E_b/N_0}{1 + E_b/N_0}}\right)$$

把生成的 txt 读入 matlab 并与计算理论值对比



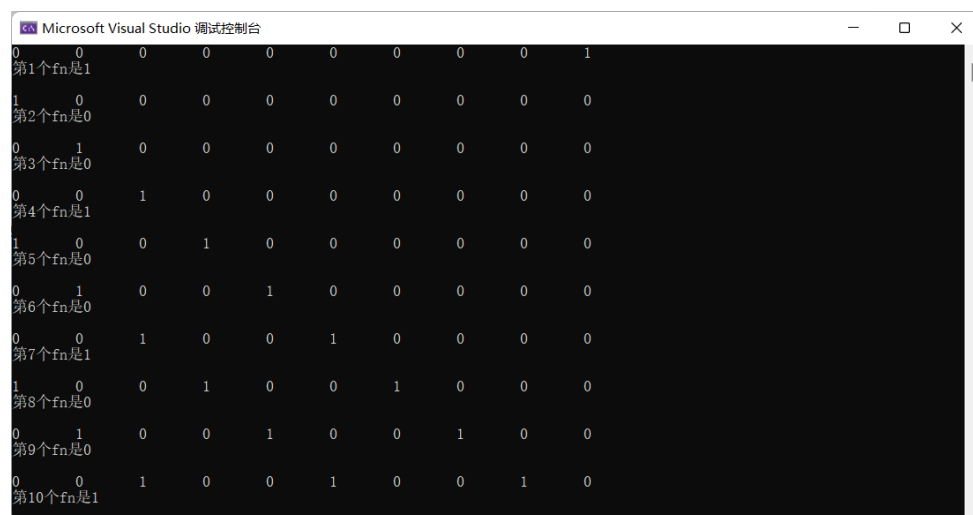
基本吻合，在衰落之后的误码率大于 AWGN 信道下的误码率

2.4 Fn 序列

采用 10 位寄存器，周期为 $2^{10} - 1 = 1023$ ，由表 7-1，连接向量为：

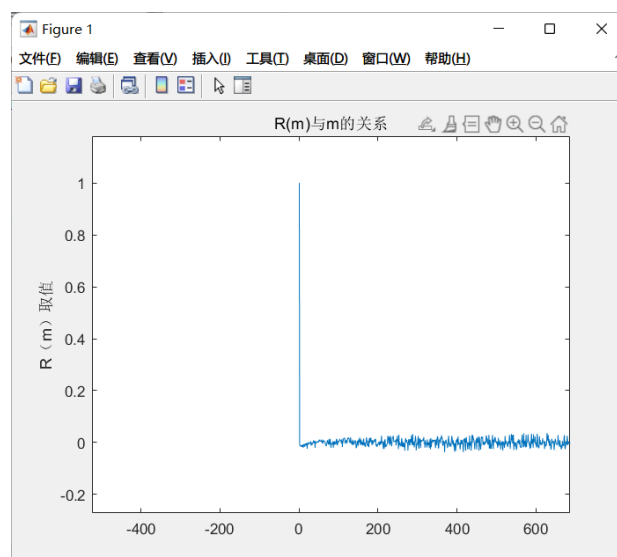
$$G = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1]$$

主程序运行结果：



可以实时看到移位寄存器的移位情况与生成的 fn。

计算自相关函数，读取生成的函数在 matlab 中绘制，可以看到类似于冲击函数



这里使用的函数为，

$$R(m) = \sum_{n=m}^{N-1} x(n-m)x(n)$$

没有考虑 FN 序列的循环性。