Exercise 2

In this section, we consider the Wigner spike model for spiked matrix estimation. The model assumes we are given data $Y \in \mathrm{R}^{n \times n}$, a symmetric matrix defined as:

$$Y = \sqrt{\frac{\lambda}{N}}\, x^* x^* + \xi,$$

where:

- $x^* \in \mathrm{R}^N$ is a random vector, representing the true signal, with *i.i.d*. entries drawn from a prior distribution $P_X(x)$.
- ξ is symmetric *i.i.d*. noise with $\xi_{i\,j} = \xi_{j\,i} \sim N(0, 1)$ for $i \leq j$.

The posterior probability $P_{X|Y}(x \mid y)$ can be written in the form of a **Boltzmann Distribution**, in particular , the dependence on **Y**, which in turns depends on the realization of the random variables: **ξ, x^***, identify a **disordered system**, where the key assumption would be that, in the Thermodynamic Limit ($n \to \infty$), the quantities of are all **self-averaging**, namely they do not depend on the particular realization of the disorder random variables, but rather concentrate around the average over the disorder.

As in all Statistical Physics problems, the fundamental task is to compute the **Free entropy density:** $\Phi = \lim_{n\to\infty} E_{x^*,z}\left[\frac{1}{n} \log Z_n\right]$. This can be done introducing N independent copies of the system and relying in the **Replica Method**, with the so called replica trick:

$$E_{x^*,z}\left[\log Z\right] = \lim_{N\to 0} E_{x^*,z}\left[\frac{Z^N - 1}{n}\right] = \lim_{N\to 0} \frac{E_{x^*,z}\left[Z^N\right] - 1}{N}.$$

In the problem considered this allows to find an expression for the Free Entropy density, depending on the overlap with the truth: $m = \frac{1}{n} E_{x^*,z}\left[\hat{x}\, x^*\right]$ and the signal to noise ratio (SNR = λ.)

$$\Phi(m, \lambda) = -\frac{\lambda}{4} m^2 + \mathrm{E}_{x^*,z}\left[\log \int d\,x\, P_X(x)\, e^{-\frac{\lambda m}{2} x^2 + \left(\lambda\, m\, x^* - \sqrt{\lambda\, m}\, z\right) x}\right], \qquad \text{where } z \sim N(0, 1)$$

The equilibrium overlap m^* maximizes the previous expression. Imposing this condition one can obtain the following self-consistent equation:

$$m = \mathrm{E}_{x^*,z}\left[\frac{\int d\,x\, P_X(x)\, x\, x^*\, e^{-\frac{\lambda m}{2} x^2 + \left(\lambda\, m\, x^* + \sqrt{\lambda\, m}\, z\right) x}}{\int d\,x\, P_X(x)\, e^{-\frac{\lambda m}{2} x^2 + \left(\lambda\, m\, x^* + \sqrt{\lambda\, m}\, z\right) x}}\right]$$

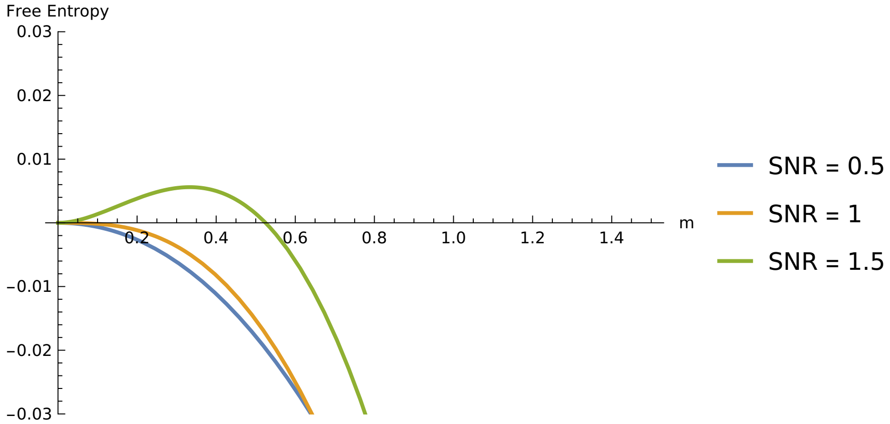
The importance of m^* in this context is its relation with the minimum square error, indeed can be shown that: $MMSE(\lambda) =$ E_{x^*}[(x^*)^2] – m^*(\lambda) .

The exercise ask to solve the self-consistent equation for a **Gauss, Rademacher and Rademacher-Bernoulli** random variables, and observe possible phase transition in the overlap varying SNR, where a transition from a zero to a non zero m, means a theoretical prediction about a transition between a maximum MMSE to a decreasing one, going from a region in which is impossible to recover the signal to one in which is possible with a certain optimal MMSE.

In the PDF file the self-consistent equation is simplified consider the information provided by the prior under examination. The resulting expression are the one implemented in the cose.

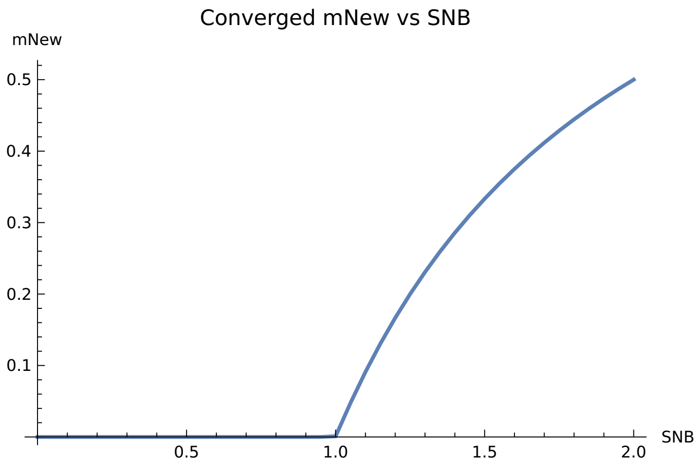
```
(*Free Energy plot*)
Frs[m_, SNR_] := -SNR / 4 * m ^ 2 - 1 / 2 * Log[1 + SNR * m] + SNR * m / 2
Plot[{Frs[m, 0.5], Frs[m, 1], Frs[m, 1.5]}, {m, 0, 1.5},
 PlotRange → {-0.03, 0.03},
 PlotLegends → {"SNR = 0.5", "SNR = 1", "SNR = 1.5"},
 AxesLabel → {"m", "Free Entropy"}]
```

Out[122]=



```
(*Self consistent overlap for a Gaussian Prior*)
results = Table[Module[{mNew = 0.1, mOld = 0, tolerance = 10 ^ -6},
    While[Abs[mNew - mOld] > tolerance, mOld = mNew;
     mNew = mOld * SNR / (1 + SNR * mOld)];
    {SNR, mNew}],
   {SNR, 0, 2, 0.05}];
ListLinePlot[results, AxesLabel → {"SNR", "mNew"}, PlotLabel → "Converged mNew vs SNR"]
```

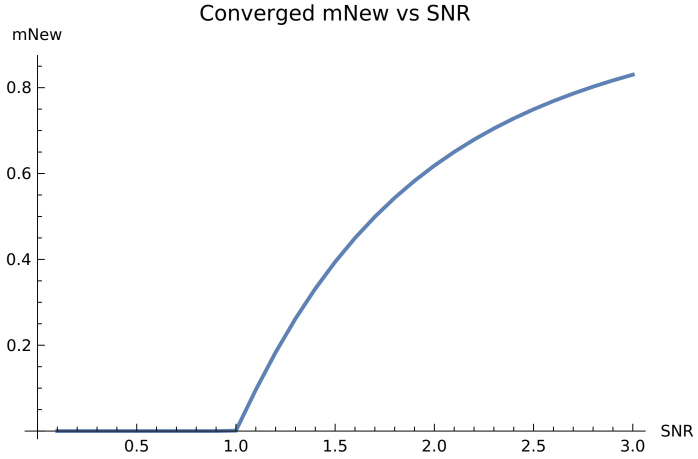
Out[150]=



```
(*Self consistent overlap for a Rademacher Prior*)
SNR = 1; (*Example value of SNR*)
mUpdate[m_, SNR_] :=
  NIntegrate[Exp[-z ^ 2 / 2] / Sqrt[2 * Pi] * Tanh[SNR * m + Sqrt[SNR * m] * z], {z, -5, 5}];

results1 = Table[Module[{mNew = 0.1, mOld = 0, tolerance = 10 ^ -6},
    While[Abs[mNew - mOld] > tolerance, mOld = mNew;
     mNew = mUpdate[mOld, SNR]];
    {SNR, mNew}],
   {SNR, 0.1, 3, 0.1}];
ListLinePlot[results1, AxesLabel → {"SNR", "mNew"},
 PlotLabel → "Converged mNew vs SNR"]
```

Out[30]=



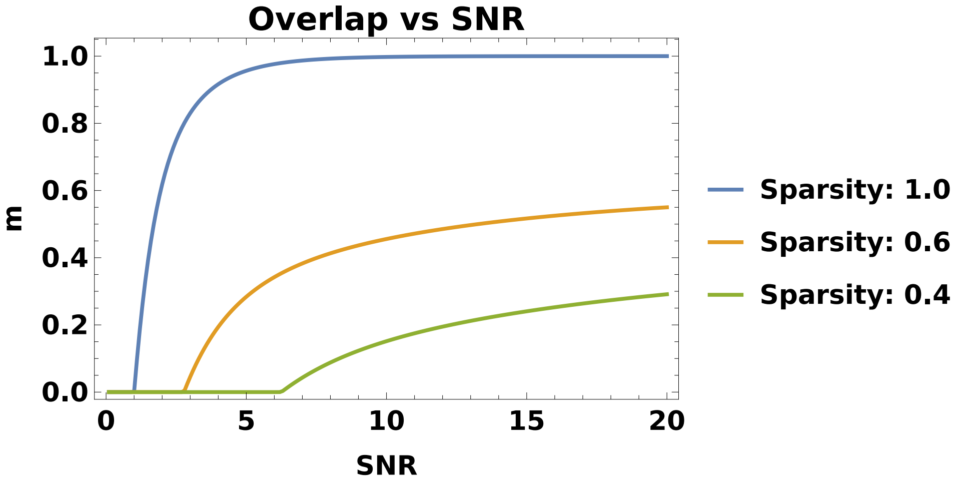
In the third case the prior gives with a probability $1 - \rho$ result 0 or a Rademacher random variable otherwise. The parameter $1 - \rho$ is also referred as **sparsity**, since when it increases the amount of *null* information increases. Starting from $\rho = 1$, one recover coherently the previous case, when sparsity increases, the **critical signal to noise ratio** increases too, meaning that we need a greater amount of signal for being able to recover it, in the meanwhile, when we are in the **recovery phase**, for the same *SNR* the maximum theoretically achievable overlap with the truth decrease with sparsity.
For really high sparsity, like $\rho = 0.05$, one can see that there is an almost vertical transition, rather resembling a First Order Phase Transition, in this case the maximum overlap is almost negligible.

In[17]:=

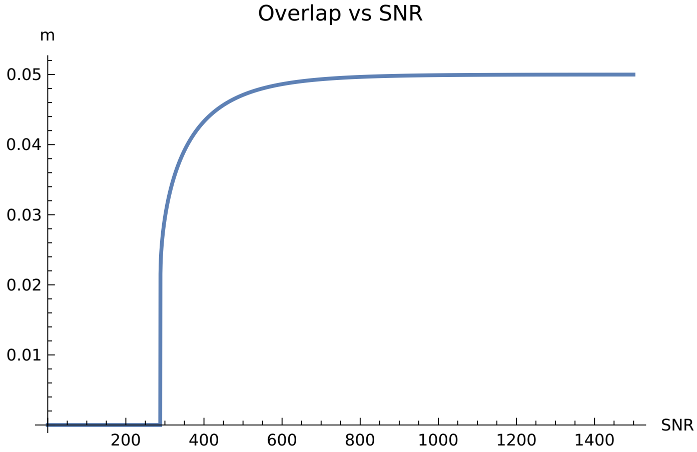
```
(*Self consistent overlap for a Rademacher Prior with sparsity*)
rho = 0.7; (*Example value of rho*)
mUpdate2[m_, SNR_, rho_] :=
 NIntegrate[Exp[-z ^ 2 / 2] / Sqrt[2 * Pi] * rho ^ 2  * Tanh[SNR * m + Sqrt[SNR * m] * z] /
    ((1 - rho) * Exp[SNR * m / 2] / Cosh[SNR * m + Sqrt[SNR * m] * z] + rho ), {z, -10, 10}]
LowSpars = Table[Module[{mNew = 0.1, mOld = 0, tolerance = 10 ^ -6},
    While[Abs[mNew - mOld] > tolerance, mOld = mNew;
     mNew = mUpdate2[mOld, SNR, 1]];
    {SNR, mNew}],
   {SNR, 0.1, 20, 0.1}];
midSpars = Table[Module[{mNew = 0.1, mOld = 0, tolerance = 10 ^ -6},
    While[Abs[mNew - mOld] > tolerance, mOld = mNew;
     mNew = mUpdate2[mOld, SNR, 0.6]];
    {SNR, mNew}],
   {SNR, 0.1, 20, 0.1}];
highSpars = Table[Module[{mNew = 0.1, mOld = 0, tolerance = 10 ^ -6},
    While[Abs[mNew - mOld] > tolerance, mOld = mNew;
     mNew = mUpdate2[mOld, SNR, 0.4]];
    {SNR, mNew}],
   {SNR, 0.1, 20, 0.1}];
ListLinePlot[{LowSpars, midSpars, highSpars},
 PlotLegends → {"Sparsity: 1.0", "Sparsity: 0.6", "Sparsity: 0.4"},
 Frame → True, FrameLabel → {"SNR", "m"}, PlotRange → All,
 LabelStyle → Directive[Bold, 14], PlotLabel → "Overlap vs SNR"]

highhighSpars = Table[Module[{mNew = 0.1, mOld = 0, tolerance = 10 ^ -6},
    While[Abs[mNew - mOld] > tolerance, mOld = mNew;
     mNew = mUpdate2[mOld, SNR, 0.05]];
    {SNR, mNew}],
   {SNR, 0.1, 1500, 0.5}];
ListLinePlot[highhighSpars, AxesLabel → {"SNR", "m"},
 PlotLabel → "Overlap vs SNR", PlotRange → Full]
```

Out[22]=


Overlap vs SNR
m
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
SNR
Sparsity: 1.0
Sparsity: 0.6
Sparsity: 0.4

Out[24]=


Overlap vs SNR
m
0.05
0.04
0.03
0.02
0.01
200
400
600
800
1000
1200
1400
SNR