0.0.1 Rapor 8 : Trimodal-Ising → Enes Yıldırım

[3]:
```
import sympy as sy
import numpy as np
import matplotlib.pyplot as plt
from cmath import sqrt
```

Bu rapor kapsamında M_1 ve M_2 değerlerine erişeceğiz. Öncelikle bir önceki rapordan elde edilen verileri geri çağıralım. Bu veriler initialdata.txt isimli dosyada depolanmıştır.

[ ]:
```
idatas = np.genfromtxt("initialdatas.txt", delimiter="\t");
idatas2 = np.genfromtxt("initialdatas2.txt", delimiter="\t"); idatas
```

Burada colon sıralası : $p \to T_- \to T_+ \to h_0 \to V_0 \to z_1 \to z_2$

Dataları ayrıştıralım.

[5]:
```
p_repo = idatas[:,0]; p_repo
```

[5]:
```
array([0.37, 0.38, 0.39, 0.4 , 0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47,
       0.48, 0.49, 0.5 , 0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58,
       0.59, 0.6 , 0.61, 0.62, 0.63])
```

[6]:
```
t1_repo = idatas[:,1];
```

[7]:
```
t2_repo = idatas2[:,2];
```

[42]:
```
h_repo = idatas[:,2];
h_repo2 = idatas2[:,2];
```

[43]:
```
v_repo = idatas[:,3];
v_repo2 = idatas2[:,4];
```

[44]:
```
z1_repo = idatas[:,4]
```

[11]:
```
z2_repo = idatas2[:,6]
```

Devam edecek olursak

$$M^2 = F_2\gamma M^2 + \frac{F_3}{3}\gamma^2 M^3 + \frac{F_4}{12}\gamma^3 M^4 + \frac{F_6}{360}\gamma^5 M^6$$

eşitliğine aşinayız. Burada paramanyetik faz için yani $M = 0$ için ifadenin

$$\frac{\gamma^2 F_3 M^2}{2!} + \frac{\gamma^5 F_6 M^5}{5!} = 0$$

eşitliği elde edilir. Bu eşitlikte $\omega = \gamma M$ değişken düzenlemesi yapılacak olursa eşitliğimiz,

$$F_6\omega^5 + 60F_3\omega^2 = 0$$

eşitliği elde edilir. Bu durumda omeganın iki versiyonu (kökü) olduğu bulunur.

$$\omega_1 = 0$$

ve

$$\omega_2 = \left(\frac{-60F_3}{F_6}\right)^{1/3}$$

[12]:
```
p, beta, v0, h0, tm, m1, m2, gam, lam, tanp, tanm, omg1, omg2 = sy.
↪symbols("p,\\beta,V_0,h_0,T_-, M_1, M_2, \\gamma, \\lambda, mode_+, mode_-,␣
↪\\omega_1, \\omega_2") ;
# Değişen sembolleri tanımla
A,B,C,D,E,F,G = sy.symbols("A,B,C,D,E,F,G");
```

[13]:
```
tanp = sy.Lambda((beta, v0, h0),(sy.tanh(beta*(v0 + h0)))); tanp
```

[13]: $((\beta,\ V_0,\ h_0) \mapsto \tanh\left(\beta\left(V_0 + h_0\right)\right))$

[14]:
```
tanm = sy.Lambda((beta, v0, h0, lam),(sy.tanh(beta*(v0 - lam*h0)))); tanm
```

[14]: $((\beta,\ V_0,\ h_0,\ \lambda) \mapsto \tanh\left(\beta\left(V_0 - \lambda h_0\right)\right))$

[15]:
```
F_3 = sy.Lambda((p,A,B),2*(-p*(1-A**2)*A - (1-p)*(1-B**2)*B));F_3
```

[15]: $((p,\ A,\ B) \mapsto -2Ap\left(1 - A^2\right) - 2B\left(1 - B^2\right)\left(1 - p\right))$

[16]:
```
F_6 = sy.Lambda((p,A,B),(8*p*(1-A**2)*(15*A**4 - 15*A**2 + 2) +␣
↪8*(1-p)*(1-B**2)*(15*B**4 - 15*B**2 + 2)));
F_6
```

[16]: $((p,\ A,\ B) \mapsto 8p\left(1 - A^2\right)\left(15A^4 - 15A^2 + 2\right) + \left(1 - B^2\right)\left(8 - 8p\right)\left(15B^4 - 15B^2 + 2\right))$

[17]:
```
omg1 = 0;
omg2 = sy.Lambda((A,B),((-60*A)/(B))**(1/3)); omg2
```

[17]: $\left((A,\ B) \mapsto 3.91486764116886\left(-\frac{A}{B}\right)^{0.333333333333333}\right)$

[18]:
```
m1 = sy.Lambda((A,B),(A*B)); m2 = sy.Lambda((A,B),(A*B)) ; m2
```

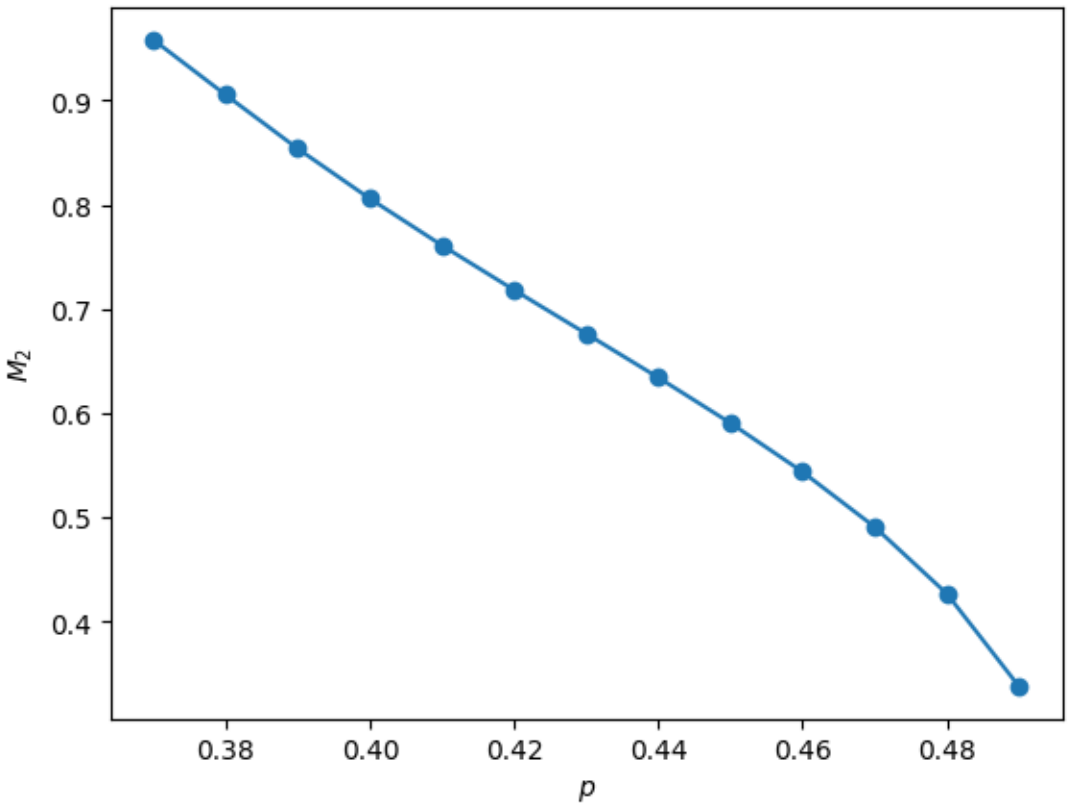
[18]: $((A,\ B) \mapsto AB)$

[ ]:
```
m1_repo = []
m2_repo = []
for i in range(len(p_repo)):
    m1_repo.append(m1(omg1,t1_repo[i]))
    m2_repo.append(m2(omg2(
        F_3(p_repo[i],tanp(1/t1_repo[i],v_repo[i],h_repo[i]),tanm((1/
↪t1_repo[i]),v_repo[i],h_repo[i],1)),
```

```
        F_6(p_repo[i],tanp(1/t1_repo[i],v_repo[i],h_repo[i]),tanm((1/
↪t1_repo[i]),v_repo[i],h_repo[i],1))
    ),t1_repo[i]));
m2_repo
```

[20]:
```
plt.scatter(p_repo[0:13],m2_repo[0:13])
plt.plot(p_repo[0:13],m2_repo[0:13])
plt.xlabel("$p$")
plt.ylabel("$M_2$")
```

[20]:
```
Text(0, 0.5, '$M_2$')
```



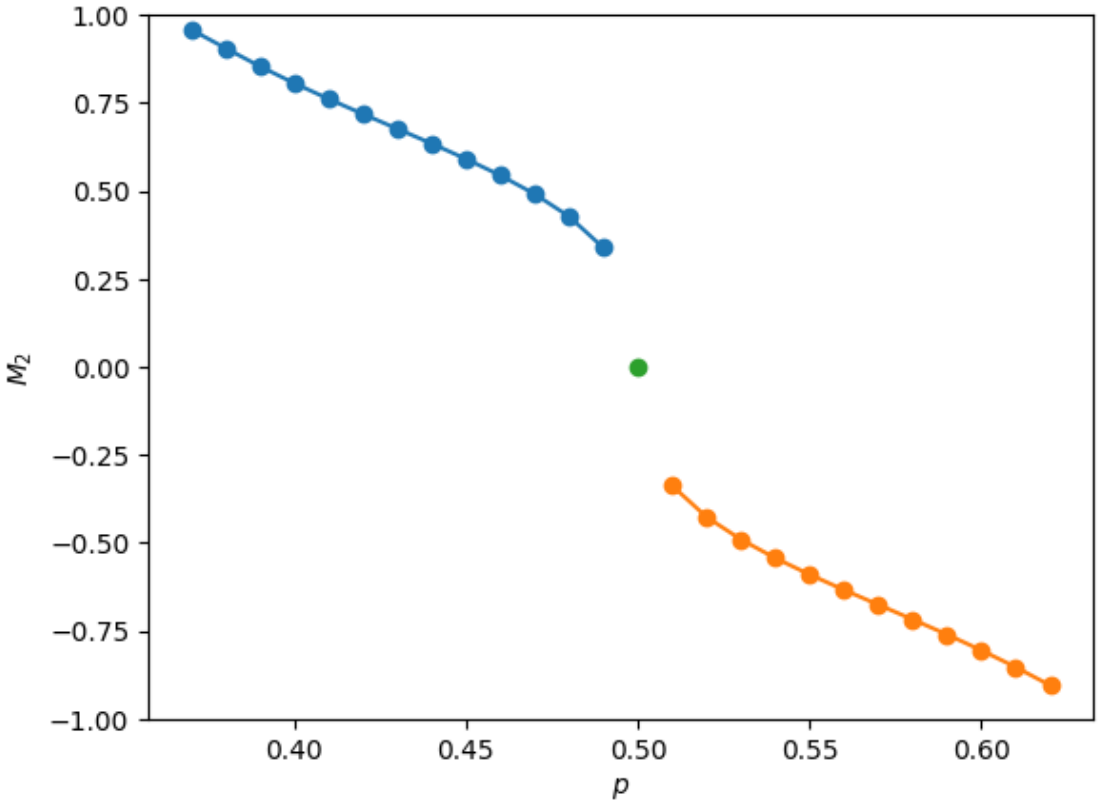
Dikkat ederseniz $p = 0.50$ seviyesine kadar çizdirdik. (Bu seviyede M = 0'dır) Burada devam eden p değerleri için grafiğin ters şekilde konumlanacağını biliyoruz. O halde;

[21]:
```
m_repo_2 = []
for i in range(14):
    m_repo_2.append(-1*m2_repo[13-i])
for i in range(len(m_repo_2)):
    m2_repo[13+i] = m_repo_2[i]
m_repo_2
```

[21]:
```
[nan,
 -0.337929677620462,
 -0.426881656800280,
 -0.490822489303615,
 -0.543637193747151,
 -0.590504757354852,
 -0.634124473500557,
 -0.676226147300673,
 -0.718117258047708,
 -0.760935622996380,
 -0.805778515137953,
 -0.853712132902232,
 -0.905381230639107,
 -0.958069730065018]
```

[22]:
```
plt.scatter(p_repo[0:13],m2_repo[0:13]);
plt.plot(p_repo[0:13],m2_repo[0:13]);
plt.scatter(p_repo[13:26],m_repo_2[0:13]);
plt.plot(p_repo[13:26],m_repo_2[0:13]);
plt.scatter(0.50,0);
plt.ylim((-1.0,1.0));
plt.xlabel("$p$");
plt.ylabel("$M_2$");
plt.savefig("M_2values.png")
```

```
C:\Users\enesh\AppData\Local\Programs\Python\Python311\Lib\site-
packages\matplotlib\collections.py:192: RuntimeWarning: invalid value
encountered in cast
  offsets = np.asanyarray(offsets, float)
C:\Users\enesh\AppData\Local\Programs\Python\Python311\Lib\site-
packages\matplotlib\cbook\__init__.py:1340: RuntimeWarning: invalid value
encountered in cast
  return np.asarray(x, float)
```



Bu değerleri hesapladıktan sonra bu değerlerin free enerji üzerine etkitilmesi işlemine geçelim. M_1 değerleri 0 olsa da M_2 değerleri Free enerji değerini daha düşük kılar bu sebeple M_2 değerleri daha fazla fiziksel anlam teşkil ettiğinden bu değerleri kullanacağız. Şimdi bu değerler üzeriden tıpkı ilk kısımda yaptığımız gibi bir tablo oluşturmak üzere Free enerji değerlerini yerlerine yazalım bunun için öncelikle bize Free enerji fonksiyonu gerekiyor. Bunu da raporlarımız kapsamında zaten tanımladık. O halde free enerji fonlsiyonumuzu tanımlayalım.