

İki Boyutlu (2D) Sistemde Lennard-Jones Gazının Monte Carlo Yöntemi İle Simülasyonu

FİZ521 - Fizikte Bilgisayar Uygulamaları
182163004 - Ömer TAYFUROĞLU

İçerik

1. Giriş	2
2. Metod	2
3. Uygulama	3
3.1. 50 Atomlu Metropolis Monte Carlo Uygulaması.....	5
4. Data Analizi ve Sonuçlar.....	5
4.1. Enerji Minimizasyonu.....	5
5. Sonuç.....	9
6. Referanslar.....	9

1. Giriş

Bu projede , Monte Carlo tekniğini kullanarak Lennard-Jones (LJ) gazının iki boyutlu (2D) bir sistem düşünülerek simülasyonu gerçekleştirilmiştir. Python bilgisayar dilinde hazırlanan kod kararlı bir şekilde çalıştırılmış, Lennard-Jones gazının, enerjisi ve ısı kapasitesi ölçümlerinin hesapsal çalışması gerçekleştirilmiştir.

Python bilgisayar dilinde yazılan bu kodu çalıştırabilmek için aşağıdaki programlama dili ve harici kütüphanelerin bilgisaya önceden yüklenmesi gerekir;

- ✓ Python 2.7
- ✓ Numpy 1.14
- ✓ Matplotlib 2.2
- ✓ Pandas 0.23

Python dilinde yazılan ve yukarıda bahsedilen kütüphaneler kullanılarak hazırlanan bu kod, Ubuntu 16.04 LTS, 64 bit işletim sisteminde test edilmiştir. Söz konusu bilgisayar programlarının hepsi Ubuntu yazılım merkezi (Ubuntu software center) aracılığıyla yüklenebilmektedir.

2. Metod

Bu çalışmada, iki boyutlu (2D) sistemde Lennard-Jones gazı simülasyonu elde edilirken Monte Carlo yöntemi kullanılmıştır. Monte Carlo yönteminde, değişim ve ilerleme Moleküler dinamik yönteminde kullanılan Newton hareket denkleminde olduğu gibi zamana göre tanımlanmış deterministik kurallara bağlı olarak değil, bunun yerine rastgele üretilen sayı dizileri ile gerçekleştirilmektedir. Bu çalışmada, farklı iki teknik olan Metropolis Monte Carlo (MMC) ve Markov Chain Monte Carlo (MCMC) methodları ile enerji minimizasyon simülasyonu elde edilmiş ve karşılaştırılmıştır. MMC tekniği rastgele elde edilen konfigürasyonlarda hesaplanan toplam enerjilerin Boltzman Faktörüne göre kabul edilip ve ya reddedilmesi şeklinde istatistiksel olarak ilerleme söz konusudur [1]. MCMC tekniğinde ise konfigürasyonlar markov chain tekniği (rastgele yürüyüş - random walk) ile değiştirilerek atomların uzayda uygun konumda yerleşmesi ve bu şekilde düşük enerjili yapının elde edilmesi şeklinde ilerleme söz konusudur. Her iki yöntem mevcut sistemin enerjisini minimize edecek şekilde çalışmaktadır.

Simülasyon iki ana bölüm üzerine tasarlanmıştır. İlk etapta, iki boyutlu taneciklerin koordinatlarını rasgele değiştiren `displace.py` modülü, mevcut sistemin enerjisini hesaplayan `calc_energy.py` modülü, oluşan konfigürasyon ve diğer dataları dinamik olarak görselleştiren `dynamic_vis.py` modülü ve elde edilen dataların analizini gerçekleştiren `data_analysis.py` modülü oluşturulmuştur. İkinci etapta, Monte Carlo ana işlevlerini gerçekleştiren `markov_chain_MC.py` ve `metropolis_MC.py` modülleri oluşturulmuştur. Bu modüller içerisinde gerekli değişkenler eklenmiş ve simülasyonu gerçekleştiren `monte_carlo_sim` fonksiyonu tanımlanmıştır. Bu fonksiyonun içerisinde gerektiğinde birinci etapta oluşturulan modüllerin içerisindeki fonksiyonları da çağrılarak simülasyon için gerekli işlemlerin gerçekleştirilmesini sağlanmıştır. Ayrıca Lenard – Jones (LJ) potansiyelini tanımlarken aşağıdaki gibi indirgenmiş birimler kullanıldı [2].

- Uzunluk birimi; σ
- Enerjinin birimi; ϵ
- Kütle birimi; m

Örneğin, buradan hareketle sıcaklığın birimi; ϵ/k_B olarak elde edilmiştir.

İndirgenmiş birimlerde Lenard – Jones potansiyeli şu şekilde elde edilmiştir;

$$U^{LJ} = 4 \left[\left(\frac{1}{r} \right)^{12} - \left(\frac{1}{r} \right)^6 \right]$$

Bu yöntem, potansiyel enerjinin hesaplanması sırasında bir enerji düzeltmesi yapılmasını gerektirir [2]. Bu düzeltmeyi şu şekilde yapabiliriz;

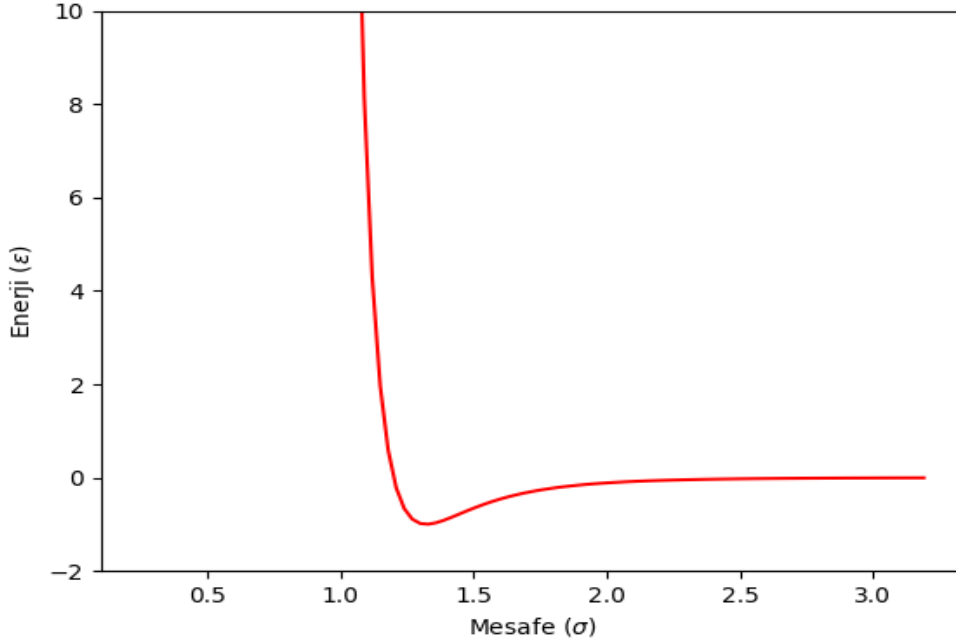
$$r_{kes} = 4 \left[\left(\frac{1}{r_k} \right)^{12} - \left(\frac{1}{r_k} \right)^6 \right]$$

Burada, r_k simülasyon kutusunun uzunluğunun yarı çapının kesilim değerini belirler.

3. Uygulama

Başlangıçta, `calc_energy.py` modülü kullanılarak iki boyutlu (2D) bir kutunun içerisinde iki boyutlu taneciklerin indirgenmiş LJ potansiyeli elde edilir. Burada, kodun hatalı çalışıp çalışmadığı iki atom arasındaki LJ potansiyel grafiği ile denetlenir. Şeki 1’ de ki enerji değişim grafiğine bakıldığında iki tanecik arasında ki LJ enerji değerinin doğru

hesaplandığı görülmektedir. Bu değerin doğru hesaplanması yapacağımız simülasyonların güvenilirliği açısından oldukça önemlidir.



Şekil 1. İki boyutlu (2D) bir sistemde iki atom arasındaki mesafe ile Lennard – Jones potansiyelinin değişim grafiği.

Taneciklerin rasgele hareketlerini elde etmek için `displace.py` modülü içerisinde fonksiyonlar tanımlandı. MMC ve MCMC için farklı fonksiyonlar tanımlanmıştır. MCMC için tanımlanan algoritma enerji minimizasyonunda önemli bir rol oynamaktadır. Bu algoritma taneciklerin kutunun duvar sınırları arasında kalması ve kendi ile çakışmaması ile en ideal durumda taneciklerin konumlanması şartını sağlamakta ve bu şekilde enerji minimizasyonu yapılmaktadır. Metropolis MC simülasyonunda ise enerji minimizasyonu `markov_chain_MC.py` modülünde tanımlandığı gibi, rasgele elde edilen konfigürasyonlardan bir sonraki konfigürasyonun bir öncekinden daha düşük enerjili olanın kabul edilmesi (accepted) ile enerji minimizasyonu gerçekleştirilir. Ayrıca enerjinin büyük çıkması durumunda (algoritmada uygulandığı şekilde) 0 ile 1 arasında rast gele çekilen bir reel sayı, Boltzman Faktörü ile karşılaştırılarak konfigürasyon kabul edilir ve ya reddedilir. Bu teknikler dögüsel (iteratif) olarak tekrar edilerek doğru sonucu ulaşılmaktadır. Modüller ve içerdikleri fonksiyonlar çalıştırılarak ve taneciklerin hareketi ve enerji grafikleri dinamik olarak gözlenerek doğrulandı.

3.1. 50 Atomlu Monte Carlo Simülasyonu

Yukarıda belirttiğimiz gibi koddaki hataların ayıklanarak doğru çalışması (ki bu şekilde yapacağımız simülasyonun sonuçları deneysel sonuçlarla daha iyi örtüşecektir.) sağlandıktan sonra Tablo 1’ de belirtilen parametreler uygulanarak Metropolis MC ve Markov Chain MC simülasyonu gerçekleştirildi. Simülasyon boyunca her 100 adımda bir örnekleme yapılarak (sampling), alınan örneğin koordinatları kaydedildi. Bu koordinatlardaki enerji değerli konfigürasyonlarla birlikte daha sonra analiz edilmek üzere kaydedildi. Bunun yanısıra kabul edilen konfigürasyonların (accepted) tüm konfigürasyonlara (accepted + rejected) oranı ($100 \times \text{accepted} / (\text{accepted} + \text{rejected})$) yüzde olarak belirlendi ve simülasyon boyunca monitör edildi. Denemeler sonucu bu değer, tavsiye edilen %30 ve % 50 değerleri arasına ayarlanarak kodun daha verimli çalışması sağlandı.

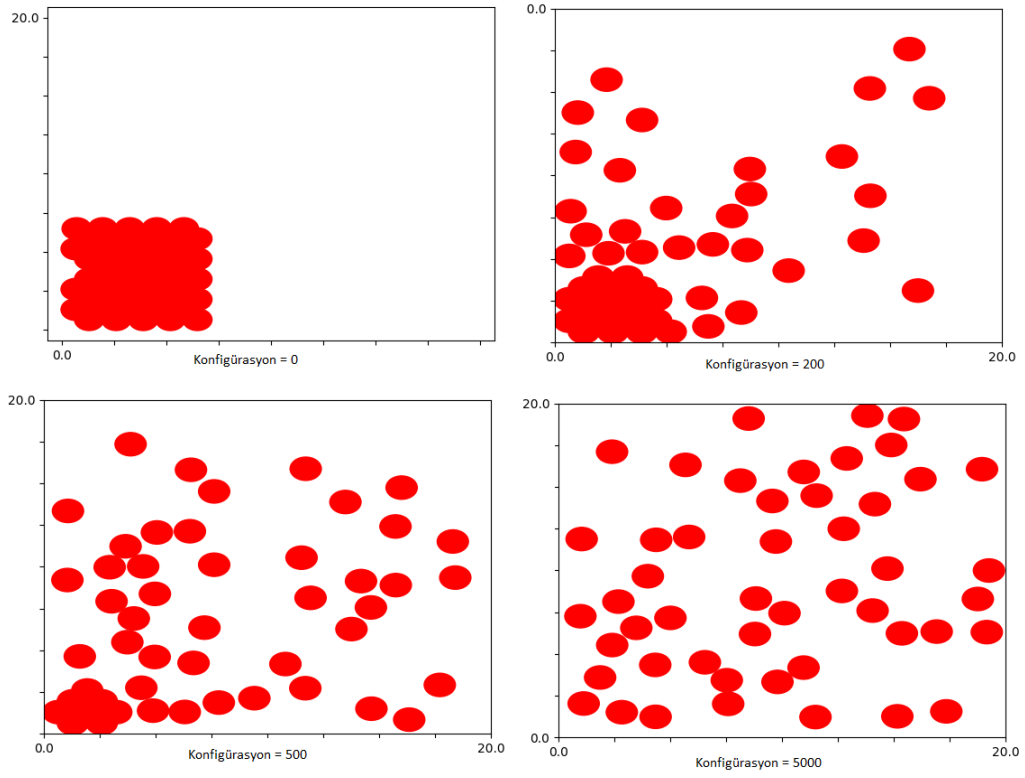
Tablo 1. 50 Atomlu Monte Carlo Simülasyonu Parametreleri

Parametre	Değer	Birim
Sıcaklık	1.4275	k_B
Yer değiştirme Adımı	$0.2 \times (\text{random}(-1, 1))$	σ
Atom Sayısı	50	
Kesilim Değeri (cutoff)	$\frac{1}{2} \times (L_{\text{box}})$	σ
Döngü Sayısı	15×10^3	

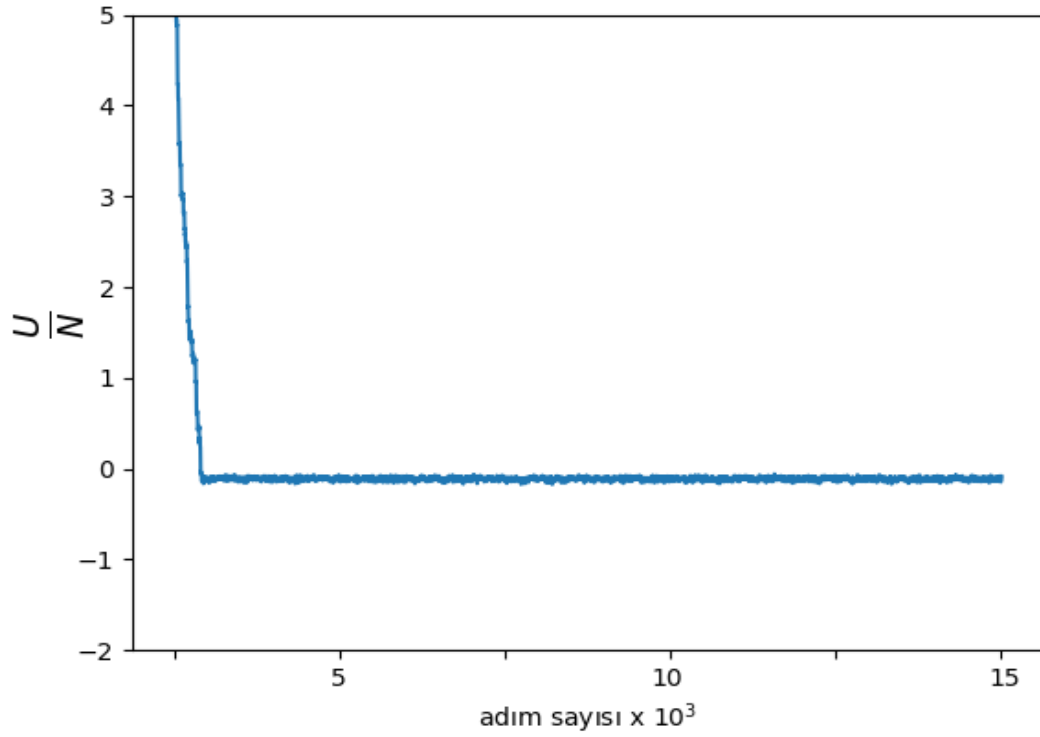
4. Data Analizi ve Sonuçlar

4.1. Enerji Minimizasyonu

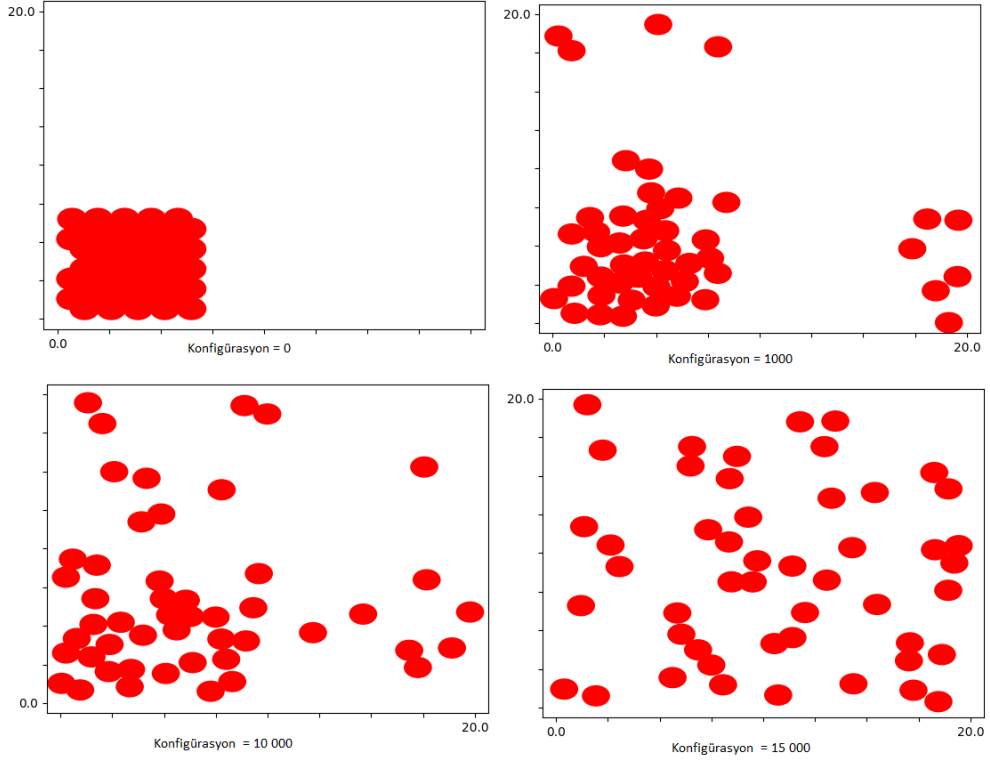
Metropolis MC ve Markov Chain MC simülasyonları 15000 adım (step) ile gerçekleştirildi. MMC de kabul edilen konfigürasyonların kabul edilme oranının (accepted) yaklaşık olarak % 30 – % 50 arasında olmasına dikkat edildi. Her iki yöntem ile yapılan hesaplamalardan elde edilen sonuçlar aşağıda gösterilmiş ve enerji grafikleri karşılaştırılmıştır. Konfigürasyon figürlerinden (Şekil 2, Şekil4) anlaşıldığı gibi başlanğıçta kutunun belli bir yerinde toplanmış olan tanecikler simülasyon ilerledikçe kutunun her tarafına homejen bir şekilde dağılmış olup her iki yöntemde de yaklaşık aynı dağılım yapısı gözlemlenmiştir. Bunun yanısıra enerji grafiklerine bakıldığında (Şekil 3, Şekil 5, Şekil 6)MCMC yöntemiyle daha hızlı minumun enerjili yapıya ulaşılmış aynı zamanda daha az enerji dalgalanması gözlemlenmiştir. Ayrıca Tablo 2 de görüldüğü gibi her iki yöntemle elde edilen ortalama toplam enerji değerleri bir birilerine oldukça yakındır.



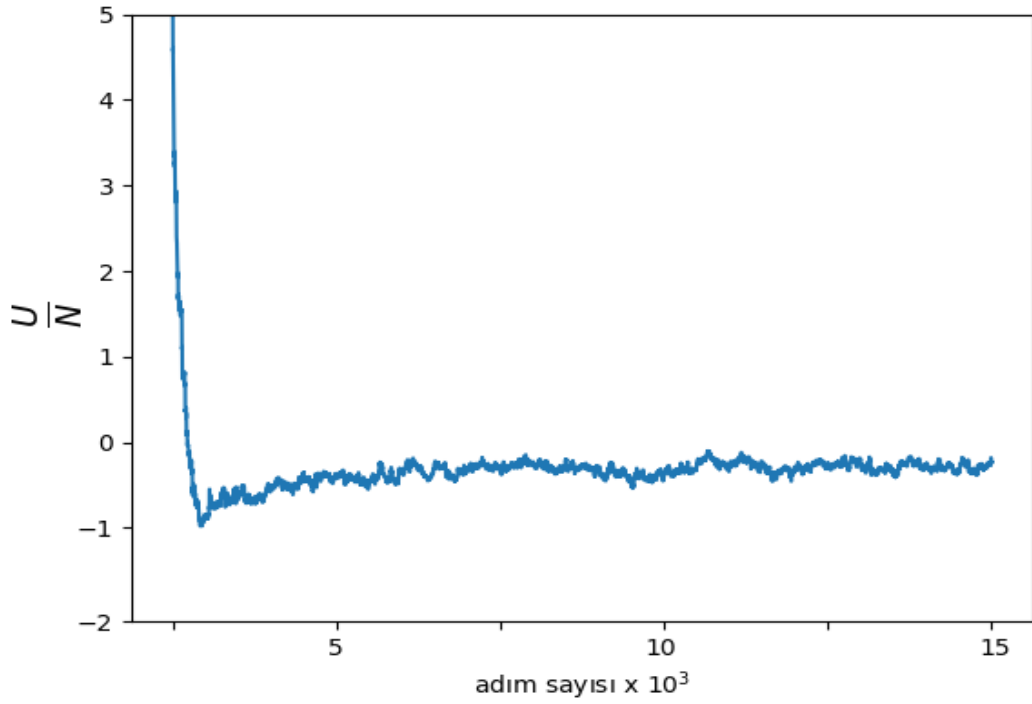
Şekil 2. Markov Chain MC yöntemiyle yapılan simülasyondan belirli aralıklarda alınan konfigürasyon örnekleri.



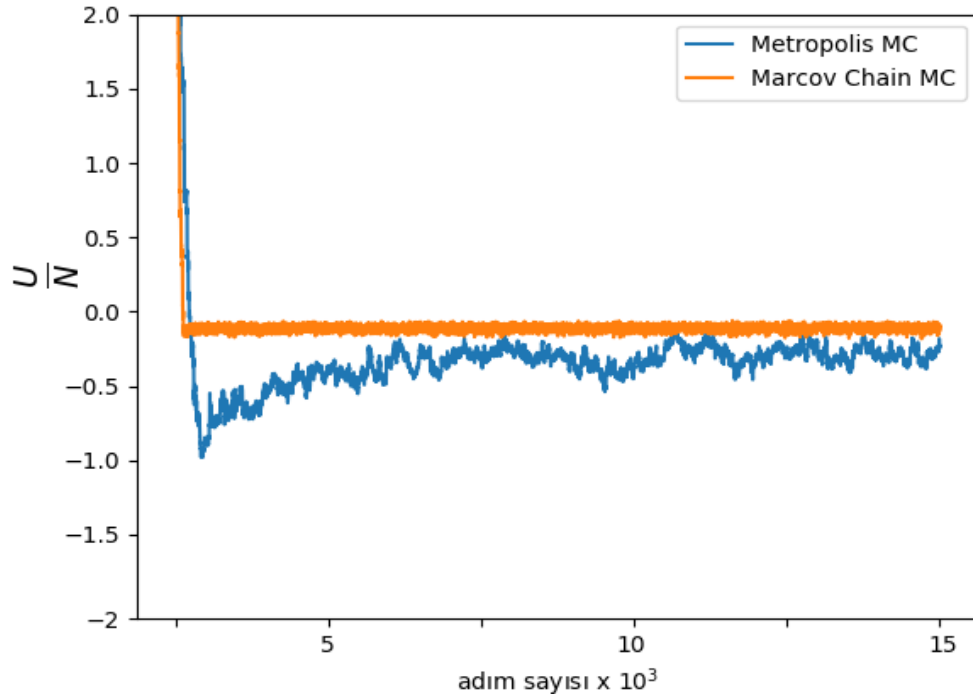
Şekil 3. Markov Chain MC yöntemiyle yapılan simülasyonun adım sayısına göre enerji grafiği.



Şekil 4. Metropolis MC yöntemiyle yapılan simülasyondan belirli aralıklarda alınan konfigürasyon örnekleri.



Şekil 5. Metropolis MC yöntemiyle yapılan simülasyonun adım sayısına göre enerji grafiği.



Şekil 6. Metropolis MC ve Markov Chain MC yöntemleriyle yapılan simülasyonun sayısına göre enerji grafiklerinin karşılaştırılması

Sabit tanecik sayısı (N), sabit sıcaklık (T) ve sabit hacimli sistemlerde (NVT ensemble) sistemin toplam potansiyel enerjisinin simülasyondaki varyasyon hesabından aşağıdaki formül ile hesaplayabiliriz [4].

$$C_V^{NVT} = \frac{\text{var}(U)}{T^2}$$

Hazırladığımız data_analysis.py modülündeki calc_heat_cap_MC fonksiyonunda yukarıdaki eşitliği kullanarak ısı kapasiteleri her iki yöntemle için (MMC ve MCMC) hesaplanmıştır. Ayrıca toplan enerji değerleri ve diğer parametreler aşağıdaki tabloda özetlenmiştir.

Tablo 2. 50 tanecikli iki boyutlu (2D) bir sistemde MMC ve MCMC simülasyon sonuçları

Kullanılan Yöntem	E_{pot} (ortalama) (ϵ)	E_{Fark} (ϵ)	C_v (kB)	$C_v \text{ Fark}$ (kB)
Metropolis MC	-4.756	-0.316	1.783	0.249
Markov Chain MC	-4.439		1.534	
DeneySEL			1.567 [4]	

5. Sonuç

Bir Lennard-Jones gazının kararlı Monte Carlo ve simülasyonları kodlanmış ve enerjileri ve ısı kapasiteleri gibi ilgi makroskopik fiziksel özellikleri istatistik mekanik verilerinden elde edilmiştir. Bu verilerin aynı şartlarda deneysel veriler ile, belli hata oranları da dikkate alındığında, tutarlı olduğu görülmektedir. Ayrıca, Python programlama dili bu tür simülasyonları elde etmek için, gerek kodlama tekniği, gerek ise zengin kütüphanelerinin varlığı nedeniyle çok yönlü bir araç olduğunu görülmüştür.

6. Referanslar

[1] A Guide to Monte Carlo Simulations in Statistical Physics, Cambridge University Press New York, NY, USA, 2005

[2]] Understanding Molecular Simulation - From Algorithms to Applications. New York, Academic Press, 2002

[3] Statistical Mechanics: Algorithms and Computations, Oxford Press New York, NY, USA, 2006

[4] Engineering ToolBox, (2003). Specific Heat and Individual Gas Constant of Gases. [online] Available at: https://www.engineeringtoolbox.com/specific-heat-capacity-gases-d_159.html