

Giriş

Dalga Boyu bazında ölçümü yapılmış verilerin ortalama ve standart sapma değerleri alınarak, bu değerlere belli bir iterasyon sayısı ile Monte Carlo Simülasyonu uygulanmaktadır. Monte Carlo Simülasyonu sonucunda ise dalga boyu bazında veriler arasında Korelasyon Katsayı matrisi hesaplanır ve sonucu ısı haritası olarak gösterilir.

Bu hesaplama ve yöntemler Python Programlama dili kullanılarak geliştirilmiştir.

Dosya Formatı ve Yükleme

Program içerisinde Monte Carlo Simülasyonu yapmak için ölçümü yapılmış verileri içeren bir input excel dosyasına gereksinim duyar. Bu input excel dosyasının formatı sırasıyla şu şekilde ayarlanmalıdır:

1. Gerekli “Sheet” lerin oluşturulması

Data | Distributions

Oluşturulmak istenen input excel dosyasının içerisinde Data ve Distributions bilgilerini içeren 2 farklı sheet oluşturulmalıdır. Eğer bu sheet isimlerini farklı yaparsanız, Python Programı içerisinde bu kısmı düzenlemeniz gerekir.

```
data_SheetName = "Data"
distributions_SheetName = "Distributions"
```

2. “Data” sheet’nin içerisinde verilerin ayarlanması

	A	B	C	D	E	F	G
1	WL	AVERAGE 1	STD.DEV.1	AVERAGE 2	STD.DEV. 2	AVERAGE 3	STD.DEV. 3
2	350	1,10E-03	3,75E-10	0	-0,000114878	3,43E-06	1,15E-07
3	351	1,13E-03	4,30E-10	0	-4,0173E-05	3,56E-06	1,64E-07
4	352	1,16E-03	4,42E-10	0	3,127E-05	3,59E-06	7,67E-08
5	353	1,19E-03	4,54E-10	0	2,8109E-05	3,65E-06	7,60E-08
6	354	1,23E-03	4,66E-10	0	-8,3735E-05	3,73E-06	1,37E-07
7	355	1,26E-03	4,77E-10	0	-0,00010526	3,73E-06	2,13E-07

“Data” sheet’nin içerisinde her dalga boyu ve her data için ortalama ve standart sapma değerleri tutulmalıdır. Örneğin elimizde x1, x2 ve x3 olarak 3 farklı data olsun. Her data

için bir ortalama ve bir standart sapma değerleri olmalıdır. Yani Excel tablosunda 6 sütun sadece bu alan için ayrılmalıdır. Eğer input excel içerisinde ortalama değeri var fakat standart sapma değeri yoksa Program doğru çalışmayacaktır ve hata verecektir.

Başka önemli bir detay ise ortalama ve standart sapma değerleri yan yana olmalıdır. Ortalama, ortalama , standart sapma gibi bir sırayla olmamalıdır.

Sütunların ilk satırı başlık olarak ayrıldığı için bu başlıkların isimleri değiştirilebilir. Program çalışırken başlık isimlerinin herhangi bir etkisi yoktur, sadece doğru sıra ve yerde verilerin yerleştirilmesine gereksinim duyar. Ayrıca Dalga boyu değerleri için de bir sütun ayrılır. Toplam da 6 + 1'den 7 sütun(3 data için bir örneği temsilen) input excel dosyasından doldurulur.

Hangi datanın hangi sütunda olduğu önemlidir. Çünkü formül de hesap ona göre yapılmaktadır.(İlerleyen yazılarda detaylı bir şekilde anlatılmıştır.)

3. “Distributions” sheet’nin içerisinde verilerin ayarlanması

	A	B	C
1	AVERAGE 1	AVERAGE 2	AVERAGE 3
2	normal	normal	normal

Data sheet’nin de kaç data için ortalama ve standart sapma değeri girildiyse, Distributions sheet’de de o data sayısında dağılımları belirtilmelidir. Örneğin data sheet’in de 4 data için değerler girilmiş olsun, Distributions sheet’in de 4 dağılım belirtilmelidir.

İlk satır data isimleri(herhangi bir isim girilebilir sonucu etkilememektedir.), 2. satır ise dağılım isimleri olmalıdır. Dağılım olarak 4 farklı dağılım çeşidi vardır.

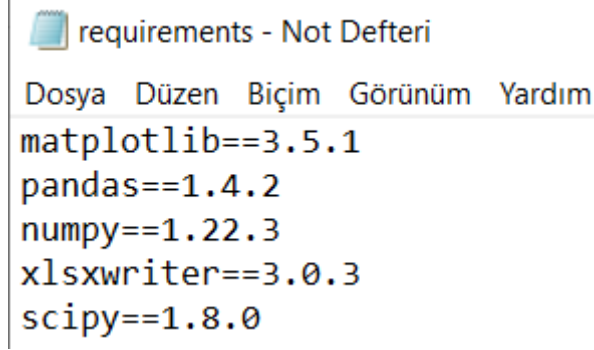
- normal
- T
- uniform
- triangle

Bu seçenekler dışında herhangi bir seçenek girilmemelidir.

4. Excel Dosyasının İsmi belirlenmesi

Belirlemiş olduğunuz dosya ismine göre Program içerisinde ona göre değiştirilerek Programdan gerekli input excel dosyasının okunması sağlanır.

Gerekli Kütüphanelerin Yüklenmesi



```
requirements - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
matplotlib==3.5.1
pandas==1.4.2
numpy==1.22.3
xlsxwriter==3.0.3
scipy==1.8.0
```

Program Python Programlama dilinin 3.8.5 versiyonu kullanılarak, Anaconda Environment da geliştirilmiştir.

Python kütüphaneleri için Proje dosyasının içerisinde bulunan requirements.txt dosyasında belirtilen sürüm ve kütüphaneler indirilmelidir. Her kütüphane tek tek de indirilebilirken, toplu bir şekilde proje dosya yolu içerisinde terminalde şu komut satırı çalıştırılarak da indirilebilir.

pip install -r requirements.txt

Başlangıç Parametrelerinin Yüklenmesi

```
5 drawNumber = 1000 # Iteration Number
6
7 #input_FileName = "makaleDegerleri input.xlsx"
8 #input_FileName = "Tayfsal Responsivity input.xlsx" # It has to be excel file.
9 input_FileName = "flux input.xlsx"
10 #input_FileName = "flux2m input.xlsx"
11 #input_FileName = "Philip input.xlsx"
12 modelFunctionName = "flux"
13 data_SheetName = "Data"
14 distributions_SheetName = "Distributions"
15
16 """
17 There are three options now.
18 modelFunctionName = "flux" or
19 modelFunctionName = "tayfsal" or
20 modelFunctionName = "article" or
21 modelFunctionName = "flux2m"
22 """
```

Yukarıdaki şekilde görüldüğü gibi main.py içeriğinde başlangıçta tanımlanması gereken değişkenler bulunmaktadır. Bunlar:

drawNumber→ Bizim iterasyon sayımıza denk gelmektedir.

input_FileName→ Hangi dosyadan okuma yapmak istiyorsak uzantısıyla beraber yazılmalıdır.

modelFunctionName→ Burada girilen model fonksiyona göre hangi formül kullanılarak Monte Carlo Simülasyonun sonucunun hesaplanacağı durumdur. Şu an da program içerisinde flux, tayfsal, article ve flux2m dışında herhangi bir model fonksiyonu bulunmamaktadır. Eğer yeni bir model fonksiyon tanımlanacak ise MCOOutput1.py içerisinde formula metot içerisinde gerekli elseif bloğunun içeriği ayarlanmalıdır.(Daha sonradan detaylı bir şekilde anlatılmıştır.)

Data_SheetName→ Oluşturmuş olduğumuz input excel dosyası içerisinde 2 sheet'in olması gerektiğinden bahsetmiştik. Ortalama ve standart sapma değerlerini içeren sheet isimi buraya yazarız.

Distributions_SheetName→ Yine bunun hakkında da önceki sayfalarda detaylı bir şekilde bahsetmiştik. Input excel dosyası içerisinde hangi sheet de data'ların dağılımlarının bilgisini vermişsek o sheet ismi buraya yazılır.

Yeni Formül Ekleme

```
for i in range(0,len(draw_matrix)):

    # if your data number is not equal to 13, it will calculate 0 for the output1 result. You should define a formula with
    if len(draw_matrix[0]) == 13 and modelFunctionName == "flux": # data number --> 13

        formula = (draw_matrix[i][0]+draw_matrix[i][1])*(draw_matrix[i][2]/draw_matrix[i][3])*(draw_matrix[i][4]/draw_matrix[i][5])

    elif len(draw_matrix[0]) == 13 and modelFunctionName == "tayfsal":

        first_Transaction = (draw_matrix[i][0]+draw_matrix[i][5]+draw_matrix[i][6]+draw_matrix[i][7]+draw_matrix[i][8]+draw_matrix[i][9]+draw_matrix[i][10]+draw_matrix[i][11]+draw_matrix[i][12])
        second_Transaction = (draw_matrix[i][1]+draw_matrix[i][5]+draw_matrix[i][6]+draw_matrix[i][7]+draw_matrix[i][8]+draw_matrix[i][9]+draw_matrix[i][10]+draw_matrix[i][11]+draw_matrix[i][12])
        formula = (first_Transaction/second_Transaction)*draw_matrix[i][4]

    elif len(draw_matrix[0]) == 12 and modelFunctionName == "article":

        formula = (draw_matrix[i][0]/draw_matrix[i][1])*draw_matrix[i][2]*draw_matrix[i][3]*draw_matrix[i][4]*draw_matrix[i][5]*draw_matrix[i][6]*draw_matrix[i][7]*draw_matrix[i][8]*draw_matrix[i][9]*draw_matrix[i][10]*draw_matrix[i][11]

    elif len(draw_matrix[0]) == 11 and modelFunctionName == "philip":

        formula = draw_matrix[i][2]*(draw_matrix[i][0]/draw_matrix[i][1])*(1-draw_matrix[i][3]-draw_matrix[i][4]-draw_matrix[i][5]-draw_matrix[i][6]-draw_matrix[i][7]-draw_matrix[i][8]-draw_matrix[i][9]-draw_matrix[i][10])

    elif len(draw_matrix[0]) == 17 and modelFunctionName == "flux2m":

        formula = (draw_matrix[i][0] - draw_matrix[i][1])*(draw_matrix[i][2]/draw_matrix[i][3])*(draw_matrix[i][4]/draw_matrix[i][5])

    output_list.append(formula)
```

MCOOutput1.py dosyası içerisinde output1 çıktısını üretirken ve Korelasyon hesabı için mc_result matrisini oluşturmada önemli noktalardan biri de formüllerin ayarlanması.

len(draw_matrix[0])→ Data sayısına denk gelmektedir.

modelFunctionName→ Bizim başlangıç parametrelerinde belirlemiş olduğumuz model fonksiyon ismidir.

Burası hazırlamış olduğunuz input dosyasının hangi model fonksiyon ile output1 çıktısının üretileceğinin belirlendiği yerdir. Örneğin input excel dosyası içerisinde flux model fonksiyonu ile çıktı üretmek istiyorsak, Toplam 13 data'ya(x1,x2,x3...x13) ihtiyacımız vardır.

Örnek bir uygulama ile anlatalım. Elimizde şöyle bir input excel dosyası olsun:

	A	B	C	D	E	F	G
1	WL	AVG1	STD1	AVG2	STD2	AVG3	STD3
2	256	2027,82	21,29211	0	0,001	907,627	0,192703
3	257	2084,42	21,88641	0	0,001	907,627	0,192703

Data Sheet

	A	B	C
1	x1	x2	x3
2	normal	uniform	normal

Distributions Sheet

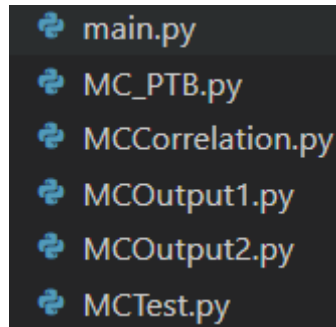
Data Sheet içerisinde bulunan Ortalama ve standart sapma değerlerini kullanarak formülden sonuç hesaplamak isteyelim. Formülümüz ise şu şekilde olsun:

Formül = $(x_1+x_3)/x_2$ modelFunctionName = “deneme”

```
elif len(draw_matrix[0]) == 3 and modelFunctionName == "deneme":  
    formula = (draw_matrix[i][0]+draw_matrix[i][2])/draw_matrix[i][1]
```

Şekline programa yeni koşul altında ekleme işlemi gerçekleştirilmelidir. Buradaki x1 avg1 ve std1’e, x2 avg2 ve std2, x3 ise avg3 ve std3 değerlerine karşılık gelmektedir.

Monte Carlo Programı İçeriği



- main.py
- MC_PTB.py
- MCCorrelation.py
- MCOutput1.py
- MCOutput2.py
- MCTest.py

main.py → Programın başlangıç değerlerinin belirlendiği ve MCTest içerisindeki mainMC metodunun çağırıldığı modüldür.

MC_PTB.py → PTB nin bizimle paylaştığı random üretme metotunu ve ortalama/standart sapma değerlerinin hesaplandığı metotları içerir.

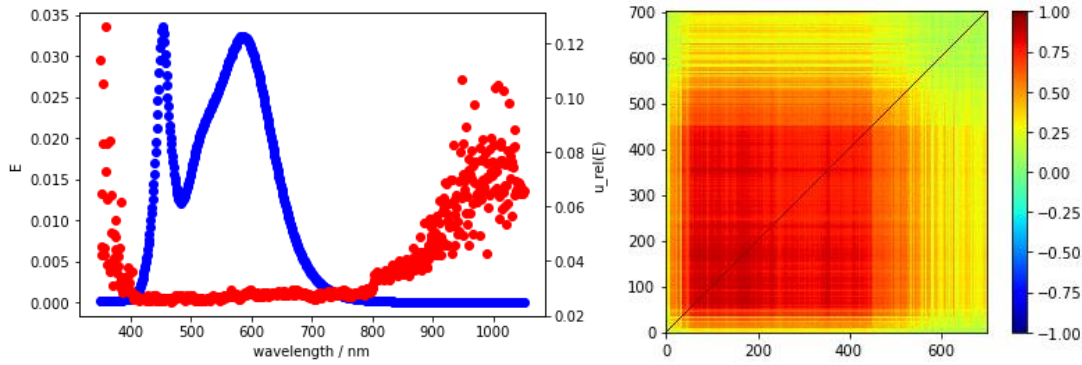
MCCorrelation.py → Korelasyon Katsayı Matrisinin hesaplanması ve onların görselleştirilmesi için metotları içerir.

MCOutput1.py → Monte Carlo Sonucunda oluşan Output excel dosyasının output1 çıktısını üretir.

MCOutput2.py → Monte Carlo Sonucunda oluşan Output excel dosyasının output2 çıktısını üretir.

MCTest.py → input excel dosyasının okunması ve mainMC'yi barındıran modüldür. Programın temel akışı burada gerçekleşmektedir.

Korelasyon Sonuçları



Monte Carlo Simülasyonu sonucunda Korelasyon Hesabı için 2 tane grafik çıkartmaktayız. Soldaki grafik MCCorrelation.py içerişindeki resultPlot metotunun içerisinde çıkartılır.

Function to plot the spectral output values and the relative standard deviation as a function of wavelength

Sağ taraftaki grafik ise Monte Carlo Simülasyonu sonucunda oluşturulan mc_result matrisinin Korelasyon Katsayı Matrisinin hesaplanması ile gösterilen ısı haritasıdır. Dalga boyu bazında, dataların arasında ki Korelasyonu görmüş oluyoruz böylece. Bu sonucu formülden hesaplanan değerler etki edip, korelasyonu oluşturmaktadır.

Genel Sonuçlar Dosyası

output1 | output2 | intervals

Output excel dosyasında 3 farklı sheet oluşturulmaktadır.

	A	B	C	
1	Iteration N	1000		
2	Wave Leng	Mean	Standart Dev	
3	350	0,000153	1,74E-05	
4	351	0,000153	1,01E-05	
5	352	0,000156	7,13E-06	
6	353	0,000158	6,73E-06	
7	354	0,00016	1,35E-05	
8	355	0,000156	1,55E-05	
9	356	0,000157	6,97E-06	
10	357	0,000164	7,29E-06	
11	358	0,000162	1,15E-05	
12	359	0,000165	2E-05	
13	360	0,000158	1E-05	
14	361	0,000164	6,39E-06	
15	362	0,00016	1,34E-05	
16	363	0,000166	6,99E-06	
17	364	0,00017	6,79E-06	

Output1 sonucunda her dalga boyu için belli iterasyon sonucunda hesaplanan formüller çıktılarının ortalama ve standart sapmalarının alınması ile hesaplanır.

	A	B	C	D	E	F	G	H	I
1	Wave Leng	mean1	std1	mean2	std2	mean3	std3	mean4	std4
2	350	0,001102	3,73E-10	5,44E-06	0,000117	3,43E-06	1,14E-07	2,72E-05	2,78E-07
3	351	0,001133	4,2E-10	-2,9E-06	3,94E-05	3,56E-06	1,61E-07	2,87E-05	3,3E-07
4	352	0,001164	4,33E-10	8,61E-07	3,11E-05	3,59E-06	7,79E-08	2,95E-05	3,64E-07
5	353	0,001194	4,64E-10	-8,7E-07	2,77E-05	3,65E-06	7,37E-08	3,03E-05	3,11E-07
6	354	0,001225	4,71E-10	-1E-06	8,53E-05	3,72E-06	1,45E-07	3,13E-05	2,67E-07
7	355	0,001256	4,53E-10	1,82E-06	0,000101	3,75E-06	2,04E-07	3,29E-05	3,55E-07
8	356	0,001289	5,58E-10	-8,7E-07	3,76E-05	3,75E-06	6,67E-08	3,36E-05	3,43E-07

Output2 sonucunda ise her dalga boyu için data'ların ayrı ayrı belli iterasyon sonucunda ortalama ve standart sapmalarının sonuçlarının gösterildiği kısımdır.

	A	B	C	D	E	F	G	H	I
1	Wave Leng	Low1	High1	Low2	High2	Low3	High3	Low4	High4
2	350	0,001102	0,001102	-0,00022	0,00025	3,21E-06	3,65E-06	2,67E-05	2,78E-05
3	351	0,001133	0,001133	-7,6E-05	7,71E-05	3,24E-06	3,88E-06	2,8E-05	2,93E-05
4	352	0,001164	0,001164	-6,1E-05	6,16E-05	3,43E-06	3,73E-06	2,87E-05	3,02E-05
5	353	0,001194	0,001194	-5,4E-05	5,06E-05	3,51E-06	3,8E-06	2,97E-05	3,09E-05
6	354	0,001225	0,001225	-0,00018	0,000161	3,45E-06	4,01E-06	3,07E-05	3,18E-05
7	355	0,001256	0,001256	-0,00019	0,000213	3,33E-06	4,12E-06	3,22E-05	3,36E-05
8	356	0,001289	0,001289	-7,3E-05	7,28E-05	3,62E-06	3,88E-06	3,29E-05	3,42E-05

Monte Carlo Simülasyonu yapılırken sumMC metotundaki interval hesaplamının sonucunu görmek için oluşturulmuş çıktıdır.