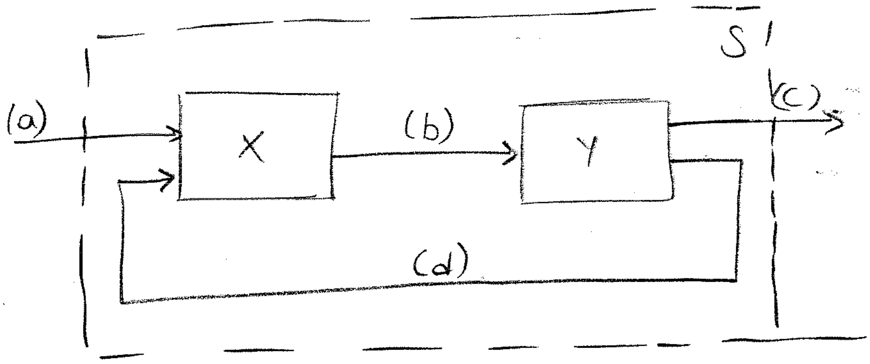
SLAYT 10- Harici Sinyaller ve Olaylar

* İç ve Dış Sinyaller

→ Genelde sistemler iç veya dış sinyaller ile bağlanan modüllerden oluşur.
→ Dış sinyaller port olarak isimlendirilen giriş / çıkıştan gelen veya giden sinyallerdir.
→ İç sinyaller ise sadece sistem modüllerini bağlar ve tamamen sistem içinden beslenirler.



* a ve c giriş ve çıkışlar olduğu için dış sinyallerdir.
* d ve b iç sinyallerdir

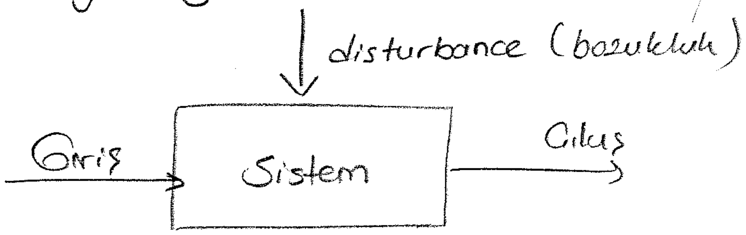
X ve Y modülleri ile S sistemi

İç ve Dış Sinyaller

* İç sinyaller ileri beslemeli ve geri beslemeli olabilir.
* İleri beslemeli sistemler girişten çıkışa ilerler. Böylece birden çok ziyaret edilen modül yoktur.
* Diğer taraftan geri beslemeli sinyaller döngüler oluşturur ve birden çok ziyaret edilir.
* Yukarıda (b) sinyali ileri beslemeli, (d) sinyali ise geri beslemelidir.
* İç girişler bazen açık sinyaller ve bozukluklara bölünebilir.
* Açık girişler arzu edilen sinyallerken, bozukluklar ilgilenilmesi gereken basit sıkıntılardır.

Bozuk Sinyaller

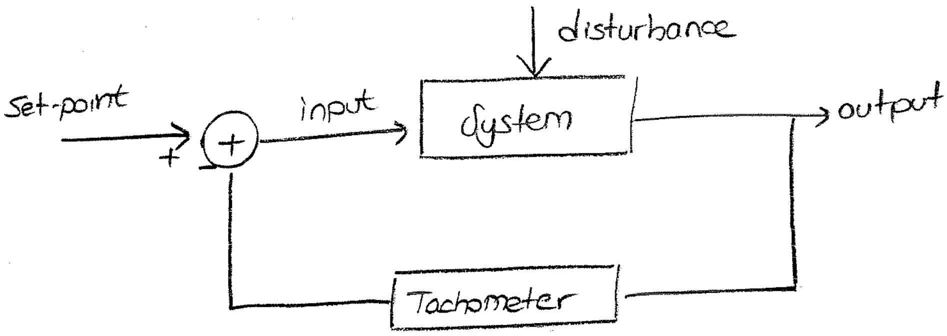
* Olasılıklı modeller bir otonom sistem rastgele bir formda bozukluk olarak adlandırılan bir sinyal aldığında oluşur.



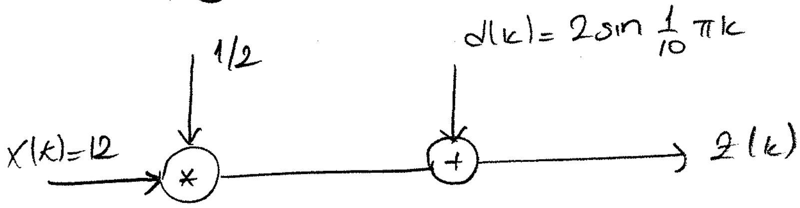
Örnek: Sabit yük ile bir motor sürücü sistemi düşünün.

→ Sabit bir giriş gerilimi sabit bir hızda motoru döndürmek için uygulanır. Fakat yük birden değişirse sistem aniden yavaşlar. Çünkü giriş sinyali dengeyi sağlayamaz. Bu ekstra yük, bozukluk olarak ifade edilir. Mühendisler bu problemi çözmek için geri beslemeli bir model önerirler.

→ Çıkış hızı bir takometre ile ölçülür ve gerçek hıza karşılık gelen gerilim geri besleme olarak verilir. Sistem gelen bozukluğa göre kendini ayarlar.



Örnek: Giriş sinyali $x(k)=12$, transfer fonksiyonu $1/2$ olarak verilmiştir. Sistemin deterministik bozukluk sinyali $d(k)=2\sin(1/10\,\pi k)$. Sistemi analiz edin ve bozukluğun etkisini azaltmak için geri besleme uygulayın.



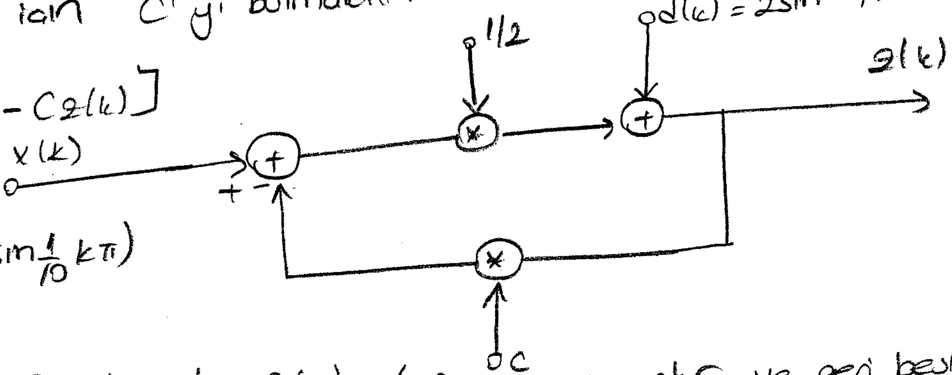
* Eğer sistemde bozukluk olmasaydı $z(k)=12\cdot\frac{1}{2}=6$ olacaktı

$$z(k)=6+2\sin(1/10\,\pi k)$$

* Bozukluk ± 2'lik değişim uygular.

Örnek: Burada C kazancı ile bir geri besleme verilmelidir. Temel fikir $d(k)$'yı azaltmak için C'yi bulmaktır.

$$z(k)=d(k)+\frac{1}{2}\left[12-Cz(k)\right]$$



$$z(k)=\frac{4}{C+2}\left(3+\sin\frac{1}{10}k\pi\right)$$

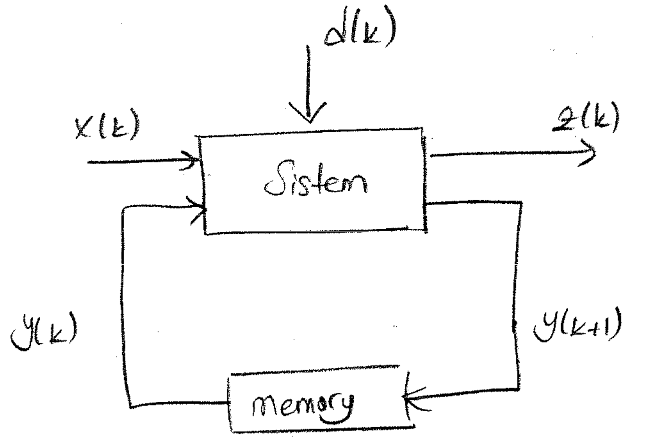
* Denklemde $C=0$ olduğunda $z(k)=6+2\sin 1/10\,\pi k$ olur ve geri besleme olmaz. C arttırıldığında ise $d(k)$'nın etkisi azaltılır. Fakat $z(k)=6$'nın etkisi de azaltılır ve ortalama çıkış sinyali $z(k)=12/(2+C)$ olur.

DURUM MAKİNELERİ

* Açık giriş = sonlu bir X   Durumların sayısı = Y
  Bozuk sinyal = sonlu bir D

* Hem dış giriş hem de durum sayısı sonlu ise Z çıkışı da sonludur.

* Eğer $X = \{x_i\}$, $D = \{d_i\}$ ve $Y = \{y_i\}$ ise $Z = \{x_i\} = \{(x_i, d_i, z_i)\}$
  X, Y ve D'nin bir kümesidir.



$$y(k+1) = f(x(k), d(k), y(k))$$
$$z(k) = g(x(k), d(k), y(k))$$

Senkron Sonlu Durum Makinesi

Örnek: Düzenli bir saatin vuruşlarında A, B, C, D harflerinden ardışık olarak çalışan bir sayısal ardışıl sistem düşünelim. Bu devre dış d(k) giriş sinyaline göre aşağı veya yukarı birbirini izler.

Başlangıç durumu $y(0) = A$

- Eğer $d(k) = 1$ ise, ardışıl sistem bulunduğu durumda kalır.
- Eğer $d(k) = 2$ ise, durum artan sırada listelenir. A, B, C, D, A, B, C, D ...
- Eğer $d(k) = 3$ ise, durum azalan sırada listelenir. A, D, C, B, A, D, C, B

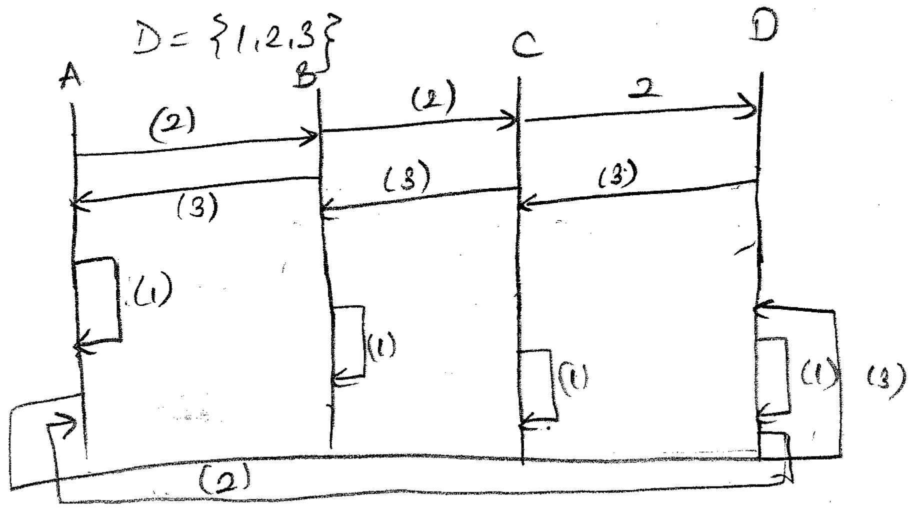
Bu sistemi tasarlayınız.

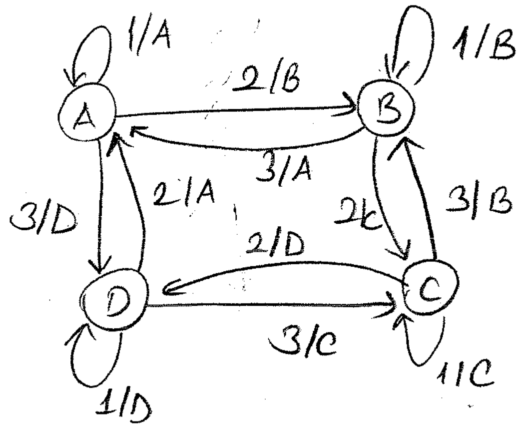
Çözüm: Bu örnekte durum sadece saat vurusu ile değiştiğinden senkron bir sistemdir. Aynı zamanda durum ve girişlerin sonlu bir kümesi vardır.

$X = \emptyset$

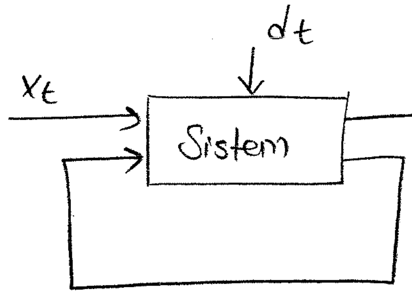
$Y = \{A, B, C, D\}$

$D = \{1, 2, 3\}$



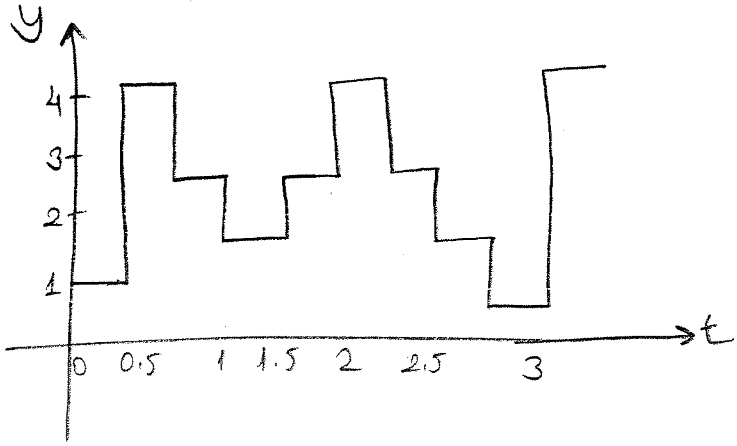


→ Eğer olaylar düzensiz zamanda düşüyorsa yani olaylar arası zaman değişken veya rastgele ise sistem asenkrondur. Önceki durum, senkron sistemlerde tam bilinmesine rağmen asenkronda bilinmez.



Senkron Makine Kodu

```
k=0
t=to
y=yo
Print t,y
for k=1 to n
    t=1+δ
    d = INT (1+3*RND)
    Call = Φ (d,y,z)
    print t,y,z
next k
```




```
Subroutine Φ (d,y,z)
  case y
  if y=A
      if d=1 then y1=A, z1=A
      if d=2 then y1=B, z1=B
      if d=3 then y1=D, z1=D
  if y=B
      if d=1 then y2=B, z2=B
      if d=2 then y2=C, z2=C
      if d=3 then, y2=A, z2=A
  if y=C
      if d=1 then y3=C, z3=C
      if d=2 then y3=D, z3=D
      if d=3 then y3=B, z3=B
  if y=D
      if d=1 then y4=D, z4=D
      if d=2 then y4=A, z4=A
      if d=3 then y4=C, z4=C
end case
y=y1
z=z1
return
```

* Rastgele durum üretiminde olasılıklar % 25 alınmıştır. Her defasında rastgele üretilen sayı ile gidilecek durum belirtilir.
- Durumlar olasılığa bağlı olduğundan farklı olasılık değerleri de verilebilir. Örneğin $Pr[D=1] = 0.1$, $Pr[D=2] = 0.3$, $Pr[D=3] = 0.6$ değerleri için giriş aşağıdaki gibi üretilir.

Kümülatif Olasılık

$0.1 * 10 \rightarrow 1$

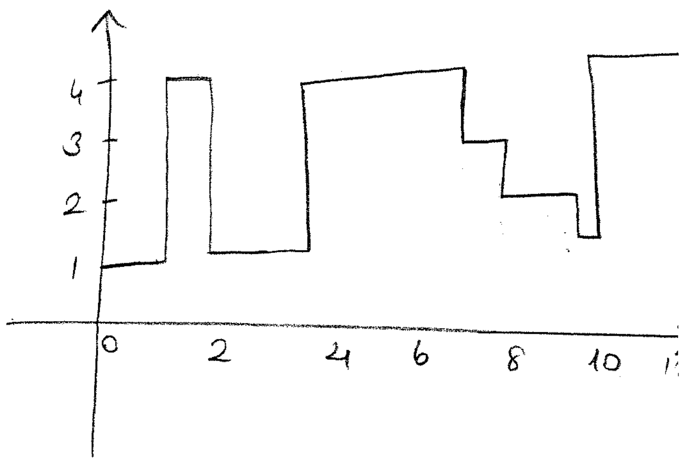
$0.4 * 10 \rightarrow 4$

$1.0 * 10 \rightarrow 10$

```
r = 10*RND
if r < 1 then D = 1
if 1 ≤ r < 4 then D=2
if 4 ≤ r then D=3
```

Asenkron Makine Kodu

```
    t=to
    y=yo
    print t,y
[1] t=t-μ*ln(RND)
    d= INT (1+3*RND)
    call Ø (d,y,z)
    print t,y,z
    if t < tmax then goto [1]
```

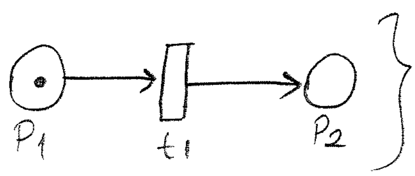


PETRİ AĞLARI

* N tane hücreye sahip bilgisayarlarda 2^N tane farklı hafıza değeri tanımlanabilmektedir.
* Çoğu bilgisayar ardışıl işlem yapmaktadır. 1 zamanda 1 işlem gerçekleştirmektedir. (Sınırlı hafıza ve ardışıl işlem → Von Neuman Mimarisi)
* Çok işlemcili makineler aynı anda birden fazla işin paralel olarak işlenmesini sağlar. Burada eş zamanlı modellemeye ihtiyaç duyulur. Bu yapılar Petri ağları olarak adlandırılır.

Temel Petri Ağ Elemanları

Yerler → Daireler
Geçişler → Çizgiler
Arklar → Oklar
İşaretler → Noktalar

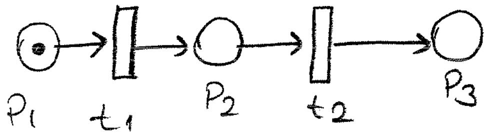


2 yer
1 geçiş

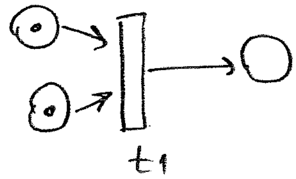
* Giriş yerlerinin herbirinde en az bir işaret varsa, geçiş düğümü iletime hazırdır. Durum geçişi $(1,0) \Rightarrow (0,1)$ olur.

Petri Ağlarının Özellikleri

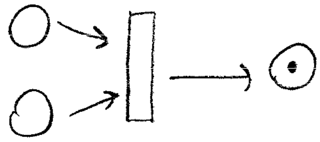
1. **Sıralı Geçiş:** t_1 iletiminden sonra t_2 iletimi gerçekleşir.



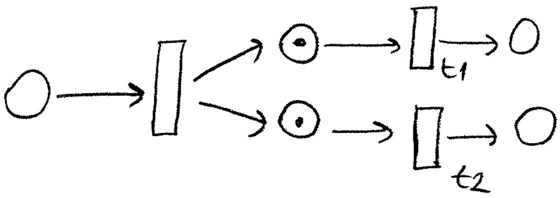
2. **Senkronize Geçiş:** Giriş yerlerinin herbirinde en az bir işaret olduğunda t_1 etkin olacaktır.



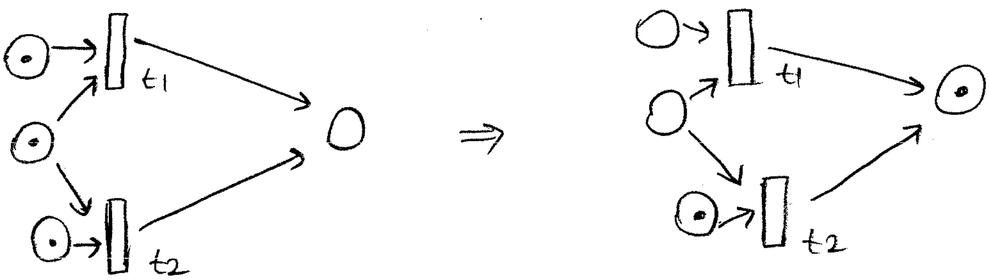
3. **Birleştirme:** İşaretler birçok yerden aynı geçiş hizmetine ulaştığında meydana gelir.



4. **Eş Zamanlılık:**



5. **Çakışma:**



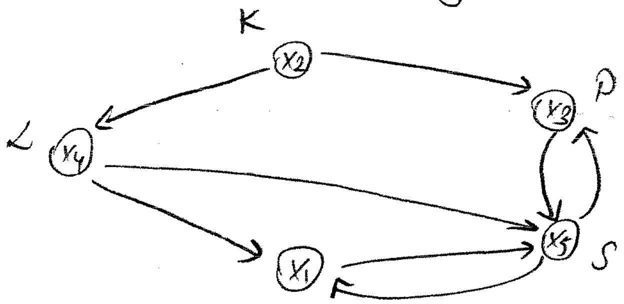
Örnek: Tablo mesaj merkezi sistem durumları

Şehir Merkezi	Düğüm	Yerel Durum
C	1	x_1
K	2	x_2
P	3	x_3
L	4	x_4
S	5	x_5

X_i: Burada belirli durumu temsil ettiğinden yerel durum olarak adlandırılır.

Sistem durumları yerel durumların vektörü şeklinde gösterilir.

X vektörü [2,0,1,3,0] olsun. Mesaj iletmek için; C → 2, P → 1, L → 3 mesaja sahiptir. K ve S'nin hiç mesajı bulunmamaktadır.



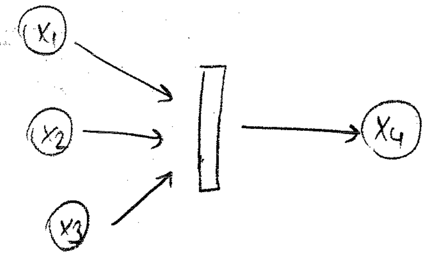
Başlangıç durumu verildiğinde sonraki durum olasılıkları tahmin edilebilir. Örneğin başlangıç durumu [2,0,1,3,0] olsun. O halde;

Geçiş	Bir Sonraki X
C → S	[1,0,1,3,1]
K → P	Mümkün değil. k=0
K → L	Mümkün değil. k=0
P → S	[2,0,0,3,1]
L → S	[2,0,1,2,1]
L → C	[3,0,1,2,0]
S → C	Mümkün değil. S=0
S → P	" S=0
S → K	" S=0

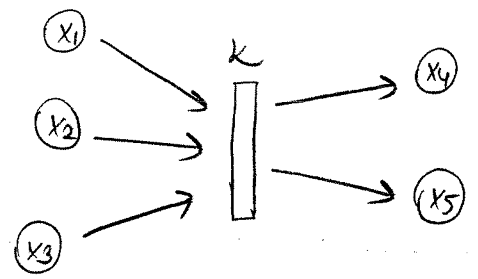
* Petri ağlarında gidilecek yerler ve mesaj sayıları gibi değerler tanımlanır. Gidiş süresi ile ilgili bilgi verilmez.
* Olası geçişlerin kümesine eylem kümesi denir ve hareket için rastgele bir seçim yapılır.

* Eylem kümesinde (δ) tüm X_i girişleri tanımlıysa geçiş pozitiftir.

if $x_1>0$, $x_2>0$, $x_3>0$ then $S_{yeni} = S_{eski} \cup \{k\}$



* Eylem kümesinde tüm kurallara karşılık gelen bir değer bulunmalıdır. δ kümesinden rastgele bir eleman seçilir. Bu geçiş k olarak kabul edilir. Her bir durum geçişinde k artırılarak yeni durumlar tespit edilir. Bu güncellemeyle yeni bir durum oluşur ve bu olay simülasyon bitene kadar devam eder.



k, δ'nın bir elemanı ise;

$x_1 = x_1 - 1$

$x_2 = x_2 - 1$

$x_3 = x_3 - 1$

$x_4 = x_4 + 1$

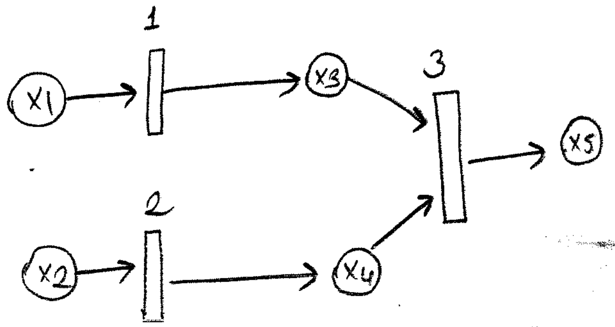
$x_5 = x_5 + 1$

* Petri ağları döngü içerisinde 3 aşamalı olarak gerçekleştirilir. Başlangıç şartları girildikten sonra döngü sürekli tekrarlanır. Döngü içerisinde mümkün geçişler, eylem kümesine eklenir. Eylem kümesinden rastgele bir hareket seçilir ve seçilen harekete göre geçişler güncellenir.

1. Başlangıç değerlerini gir, dağılımı göster
2. k=0
3. print k, durum
4. for k=1 to n
5. Tüm geçişleri belirle, işaretle
6. Eylem kümesinden rastgele bir geçiş seç.
7. Sadece bu geçişe dahil olan node'ları güncelle.
8. Print k, durum
9. next k

<u>Örnek</u>: Şekilde gösterilen bir petri ağı düşünelim. 1. yer için 8, 2. yer için 5 mesaj tanımlanmış, diğer yerlerin mesajı 0 olarak verilmiştir.

Petri ağ modeli kullanarak, geçici ve uzun vadeli bir davranış simule ediniz.



$X(0) = [8, 5, 0, 0, 0]$

Çözüm: Petri ağının 3 aşaması;

1. Tüm geçişleri tespit et ve işaretle
2. Bunlardan birini rastgele seç.
3. Yürüt.

* Verilen örnekte 1. ve 2. geçiş paraleldir. Hangisinin önce başlayacağını bilemeyiz, birini ilk başlaması için seçeriz. x_3 ve x_4 mesajları yerine ulaşıncaya kadar 3 geçişi aktif değildir. 3 geçişi mesajını x_5'e ulaştırır. Ama bunlar gerçekleşinceye kadar x_1 ve x_2 çoktan gerçekleşmiş olacaktır.

1. Tüm Yasal Geçişlerin Belirlenmesi ve işaretlenmesi

- Yasal geçişler bir S eylem kümesi oluşturularak yapılır.
- Her geçişte yasal olarak bulunan davranışlar bu kümeye eklenir.
- Boş bir eylem kümesi yasal hiçbir işlem yapılamayacağını gösterir.

$$S = \emptyset$$
if $x_1 > 0$ then $S = S \cup \{1\}$
if $x_2 > 0$ then $S = S \cup \{2\}$
if $x_3 > 0$ and $x_4 > 0$ then $S = S \cup \{3\}$

2. Yasal Geçişlerden Birinin Rastgele Seçilmesi

S boş küme değilse rastgele bir eleman seçilir, boş ise 0 döndürülür.

$$RAND(S) = \begin{cases} 0, & S = \emptyset \\ y \in S, & S \neq \emptyset \end{cases}$$

* S stringinden rastgele bir karakter döndüren algoritma;

SORU: Madeni para ile oynanan yazı tura oyunu şu şekildedir. Para atıldığında yazı gelirse bir sonraki aşamada tura veya yazı gelme şansı %50'dir. Bununla birlikte eğer tura gelirse bir sonraki aşamada tura gelme şansı %75'tir. Monte Carlo benzetimi ile tura gelme olasılığını belirleyen bir MATLAB kodu yazınız.

Yazı gelirse; Y → %50, T → %50

Tura gelirse; Y → %25, T → %75

```
function para = montecarlo(n)

tura_olasilik = 0.5
yazi=0; tura=0;
for i=1:n
   x=rand;
   if ( 0 < x < 0.5)
      yazi = yazi+1
      tura_olasilik = 0.5
   else
      tura = tura+1;
      tura_olasilik = 0.75
   end
end
para = (tura) / (yazi+tura);
```

kümülatif olasılık

0.5 0 - 0.5

Y

1 0.5 - 1

4

SORU:

Aralık	Tekrar Sayısı
0-10	120
11-20	100
21-30	90
31-40	80
41-50	140
51-60	70
61-70	150
71-80	60
81-90	90
91-100	100

Soru : int Ufuk_Gitsin() isimli [0,100] aralığında değer döndüren bir alt program yazılmış ve % 99 tek düze olduğu iddia edilmiştir, ancak 1.000 defa çağrılarak test edildiğinde yukarıdaki frekans tablosu gözlemlenmiştir. İddianın doğruluğunu analiz ediniz. ($x_c^2 = 21.66$)

Çözüm.

$\boxed{n = 1000}$ $\quad e_k = \frac{n}{m} = \frac{1000}{10} = 100$

$m = 10$

$f_1 = 120$ $\quad f_6 = 70$
$f_2 = 100$ $\quad f_7 = 150$
$f_3 = 90$ $\quad f_8 = 60$
$f_4 = 80$ $\quad f_9 = 90$
$f_5 = 140$ $\quad f_{10} = 100$

$$\boxed{X^2 = \sum_{k=1}^{m} \frac{(f_k - e_k)^2}{e_k}}$$

$$X^2 = \frac{(120-100)^2}{100} + \frac{(100-100)^2}{100} + \frac{(90-100)^2}{100} + \frac{(80-100)^2}{100} + \frac{(140-100)^2}{100} +$$
$$\frac{(70-100)^2}{100} + \frac{(150-100)^2}{100} + \frac{(60-100)^2}{100} + \frac{(90-100)^2}{100} + \frac{(100-100)^2}{100}$$

$$= 4 + 1 + 4 + 16 + 9 + 25 + 16 + 1 + 0 = 76$$

$X^2 = 76$ $\quad X_c^2 = 21.66$

$\boxed{X^2 > X_c^2 \text{ olduğu için uniform değildir.}}$

SORU: Aşağıdaki parçalı olasılık yoğunluk fonksiyonunu kullanarak ters dönüşüm tekniği ile rassal değişken üreten denklemi elde ederek ilgili algoritmayı yazınız.

$$f(x) = \begin{cases} 1/3 & , \quad 0 \leq x \leq 2 \\ \frac{1}{24} & , \quad 2 < x \leq 10 \\ 0 & , \quad \text{diğer durumlarda} \end{cases}$$

Çözüm:

(1) $U = F(x) = \int_0^x \frac{1}{3} dx = \frac{1}{3} x \qquad 0 \leq x \leq 2$

$U = \frac{1}{3} x \Rightarrow x = 3U$

$x = 0$ için $\quad u = 0$

$x = 2$ için $\quad u = 2/3$

(2) $U = F(x) = \int_0^2 \frac{1}{3} dx + \int_2^x \frac{1}{24} dx = \frac{2}{3} + \frac{1}{24} x - \frac{2}{24} = \frac{7}{12} + \frac{x}{24}$

$u = \frac{x}{24} + \frac{14}{24} \Rightarrow \left(u - \frac{14}{24}\right) \cdot 24 = x$

$\boxed{24u - 14 = x}$

$x = 2$ için $\quad u = \frac{2}{3}$

$x = 10$ için $\quad u = 1$

$$F(x) = \begin{cases} 3u & , \quad 0 \leq u \leq 2/3 \\ 24u - 14 & , \quad 2/3 < u \leq 1 \end{cases}$$

Algoritması:

1. $u \sim U(0,1)$ rastgele sayı üret.
2. if $u \leq 2/3$ $\quad x = 3u$
3. if $u > 2/3$ $\quad x = 24u - 14$
4. return

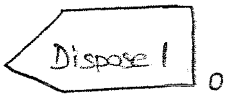
SLAYT 9- ARENA MODÜLLERİ

1.CREATE

Bir simülasyon modelinde varlıklar için başlangıç noktasını tasarlar.

2.DISPOSE

Bir simülasyon modelinde varlıklar için son noktayı tasarlar.



3.PROCESS

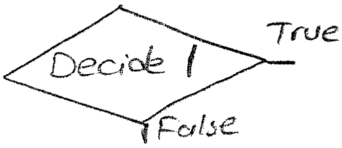
Bir simülasyon modelinde ana process metodunu tasarlar. Kullanılacak kaynakları ifade eder.

4.DECIDE

Bu modül sistemde karar verme processi için izin verir. Karar alınmasında bir veya daha fazla duruma (koşula) ya da bir veya daha fazla olasılığa dayanarak seçmeyi içerir.

Durumlar, özellik değerlerine, değişken değerlerine, varlık tiplerine ya da bir ifadeye dayanabilir.

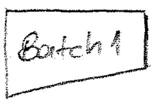
İkili ihtimal ya da ikili durumdan herhangi biri seçildiğinde, Decide modülünün iki çıkış noktası vardır. Doğru ve yanlışlar için birer çıkış noktası vardır.



2-way by Condition : 2'li durum
2-way by Change : 2'li seçim
N-way by Condition : Çoklu durum
N-way by Change : Çoklu seçim

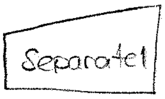
5.BATCH

Bu modül, simülasyon modülü içinde gruplama mekanizmasını tasarlar.



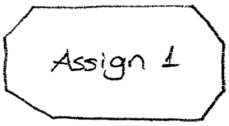
6. SEPARATE

Bu modül, coklu varlıkların icine gelen bir varlığı kopyalamakta ya da önceden oluşturulan bir varlık yığınını bölmekte kullanılabilir.



7. ASSIGN

Bu modül, değişkenlere varlık özelliklerine, varlık tiplerine, varlık resimlerine ya da sistem değişkenlerine yeni değer atanması için kullanılır. Tek bir assign modülle coklu atamalar yapılabilir.

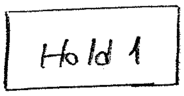


8. RECORD

Bu modül, simülasyon modelinde istatistikleri biriktirmekte kullanılır.

9. HOLD

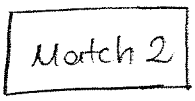
Bu modülde eğer varlık bir sinyal için tutuluyorsa, sinyal modülü varlığa sonraki modüle geçmek için izin vermede kullanılır.



10. MATCH

Match modülü farklı kuyruklarda bekleyen varlıkları belli sayılarda gruplar bir araya getirir.

Match komutunun işlev görebilmesi için belirtilen kuyruklarda en azından bir varlık olması gerekir.



11. ROUTE

Route (rota) modülü, belirtilen bir istasyona bir varlığı transfer eder veya istasyona ziyaret sırasında, sonraki istasyona gezen birimi tanımlamak için kullanılır.

Route 1

12. STATION

Route komutu kullanıldığında gezen birimin gideceği yerleri tanımlamak için kullanılır.

Station modülü hareketli kaynakları veya durağan olmayan kaynakların olduğu ilgili park alanına sahip olabilir.

Station 1

13. ACCESS

Access modülü, varlığın bir istasyondan diğerine hareketi için konveyörün bir ya da daha fazla hücresine yer tahsis eder.

Access 1

14. CONVEY

Convey modülü aracılığıyla bulunduğu istasyondan belirtilen varış istasyonuna taşır.

Convey 1

15. EXIT

Bu modül Access modülü ile Conveyöre alınan gezen birimi herhangi bir işlem için konveyörden almaya yarar.

Convey komutu ile taşınan bir gezen birim mutlaka ilgili istasyona geldiğinde processe girmeden önce conveyörden alınmalıdır. Aksi halde taşıyıcı sürekli dolu görünecek bu da yanıltıcı sonuçlar doğuracaktır.

Exit 1

16. REQUEST

İstek modülü, bir varlığa bir taşıyıcı ünitesini tayin eder ve varlığın yerine üniteye hareket eder.

Request 1

17. TRANSPORT

Bu modül yine geçen birimin taşınmasında kullanılır. Bu modülde taşıyıcı sınırlaması vardır. İstediğimiz kadar taşıyıcıyı biz tanımlarız.
Request komutu ile çağrılan taşıyıcı Transport modülü ile ilgili istasyona gittikten sonra Free modülü ile mutlaka boşaltılmalıdır.

Transport 1

18. FREE

Bu modül varlığın en son pay edilmiş taşıyıcısını salıvermek için kullanılır. Eğer sırada taşıyıcı istemek veya pay etmek için bekleyen bir varlık varsa, taşıyıcı o varlığa verilir.

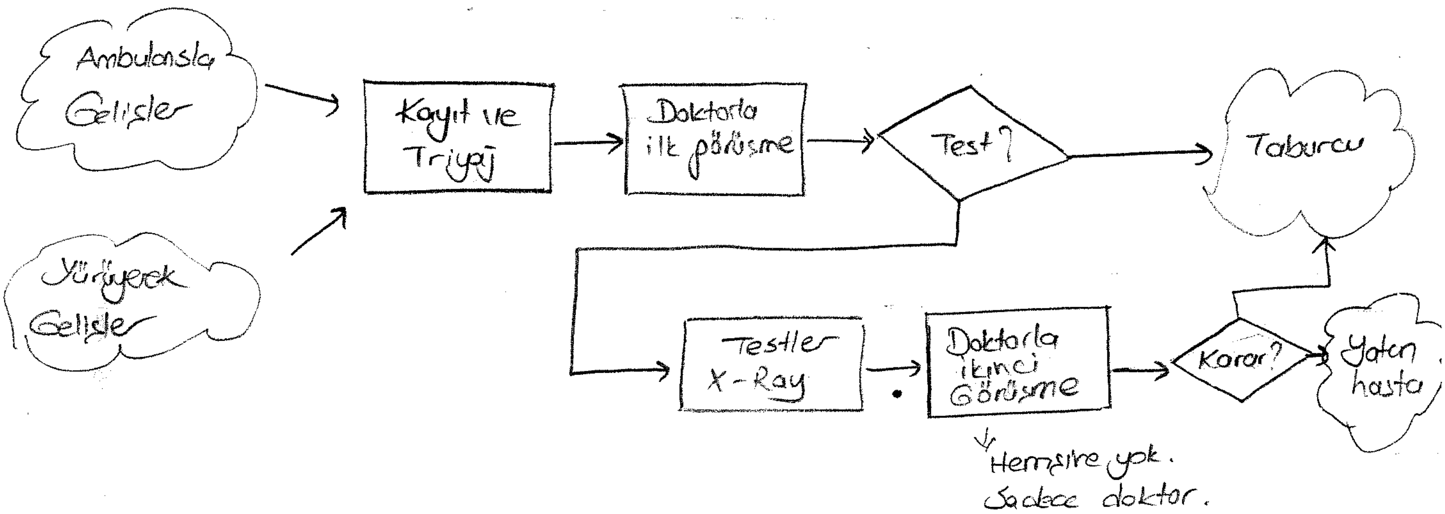
Free 1

ACİL SERVİS MODELİ

Bu bölümde bir hastanenin Acil Servis (AS)'nın benzetim modeli yaratılacaktır.

* İki şekilde hasta gelişi olmaktadır; yürüyerek ve ambulansla.
* Her iki hasta tipi içinde; gelişlerinden hemen sonra kayıt ve triyaj yapılmaktadır.
* Triyaj sonucunda hastalara kırmızı, sarı, yeşil renklerinden birisi verilerek önceliklendirilmektedir.
* Gerçek hayatta bu önceliğe göre hastalar doktor muayenesi için kuyruğa girdiklerine rağmen biz şimdilik bu modelde doktorla ilk görüşme için yine FIFO bir kuyruk olduğunu varsayalım. Ayrıca doktorla ilk görüşme sırasında bir hemşirenin de hazır bulunması gerekmektedir.

Acil Servis Sürecinin Şematik Gösterimi



Süreler	Değer
* Ambulans gelişleri arası süre	Üssel dağılım (ortalama 30 dk)
* Yürüyerek gelişler arası süre	Üssel dağılım (ortalama 5 dk)
* Kayıt ve triyaj süresi (1 hemşire)	Üçgensel dağılım (en az 2 dk, genelde 5 dk, en çok 10 dk)
* Doktorla ilk görüşme (1 hemşire 1 doktor)	Kırmızı hastalar: Lognormal dağılım (ort: 30 dk Std. sapma 15 dk) Sarı hastalar: Üssel dağılım (ort. 20 dk) Yeşil hastalar: Üçgensel dağılım (en az 5 dk, genelde 8 dk, en çok 12 dk)
* Testler ve X-Ray	Tüm hastalar için; üçgensel dağılım (en az 20 dk, genelde 40 dk, en çok 60 dk)
* Doktorla ikinci görüşme	Tüm hastalar için; üçgensel dağılım (en az 5 dk, genelde 10 dk, en çok 15 dk)

Oranlar

* Triyaj rengi oranları

Ambulansla gelenler için;

% 70 kırmızı

% 30 sarı

* ''

Yürüyerek gelenler için;

% 1 kırmızı

% 19 sarı

% 80 yeşil

* Test ihtiyacı oranı

Sarı ve kırmızı hastalar için

% 10 ihtiyaç var

% 90 ihtiyaç yok.

Yeşil hastalar için test istenmiyor.

* İkinci doktor görüşmesi sonrası doktorun hastayı taburcu etme kararı

Tüm hastalar için;

% 20 hasta yatsın

% 80 taburcu olsun.

Kaynaklar

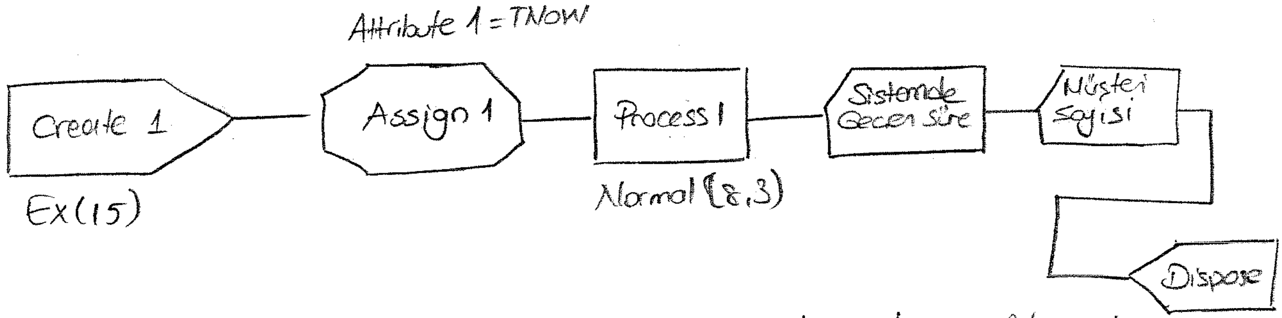
* Doktor sayısı ⟶ 3
* Hemşire sayısı ⟶ 6
* Test makinesi ⟶ 1

* Her modelleme projesinde olduğu gibi bu projede de bir amacımızın olması gerekmektedir. Bu örnekte;
  - İlk doktor muayenesi için ortalama bekleme zamanının 10 dk'yı geçmemesini
  - Doktorun kullanım oranının % 70'i geçmemesini istiyoruz.

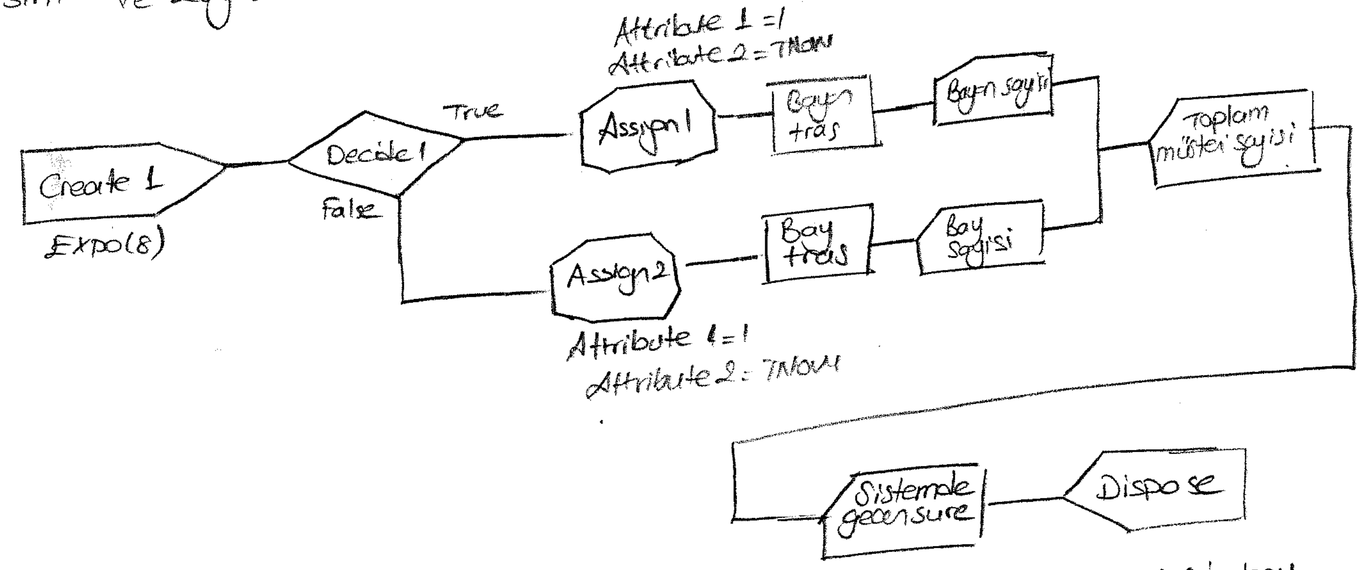
ARENA ÖRNEKLER

1) Gelişler arası süre Ex(15) olan ve tıraş süreleri normal (8,3) dağılımına uyan bir berberde 200 dk çalışılması durumunda oluşacak hizmet gören müşteri sayısı, hizmet görenlerin ortalama sistemde geçirdiği süre ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız.

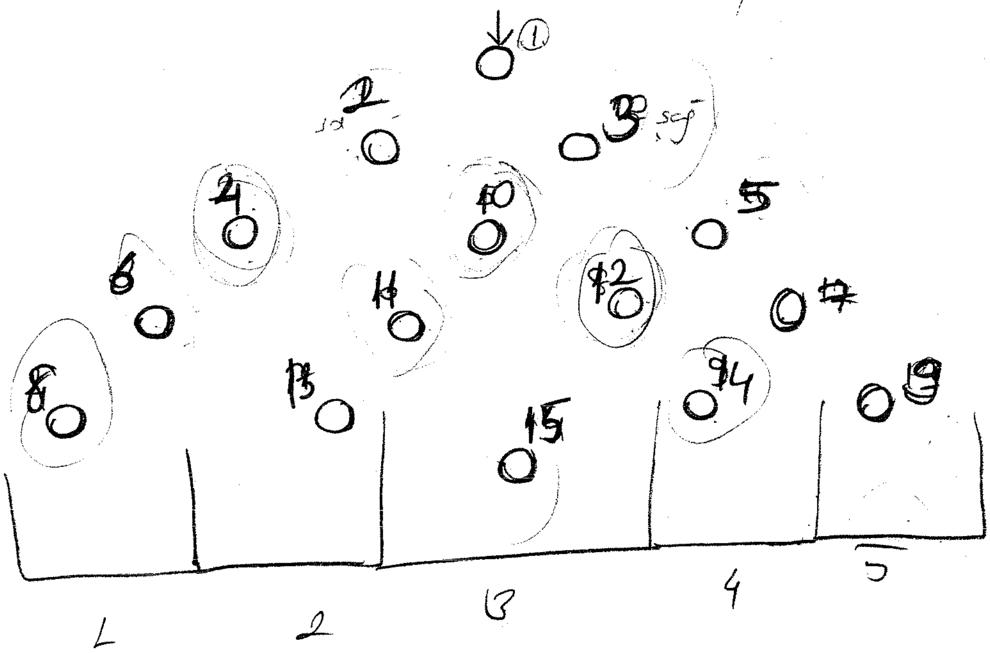
Çözüm:

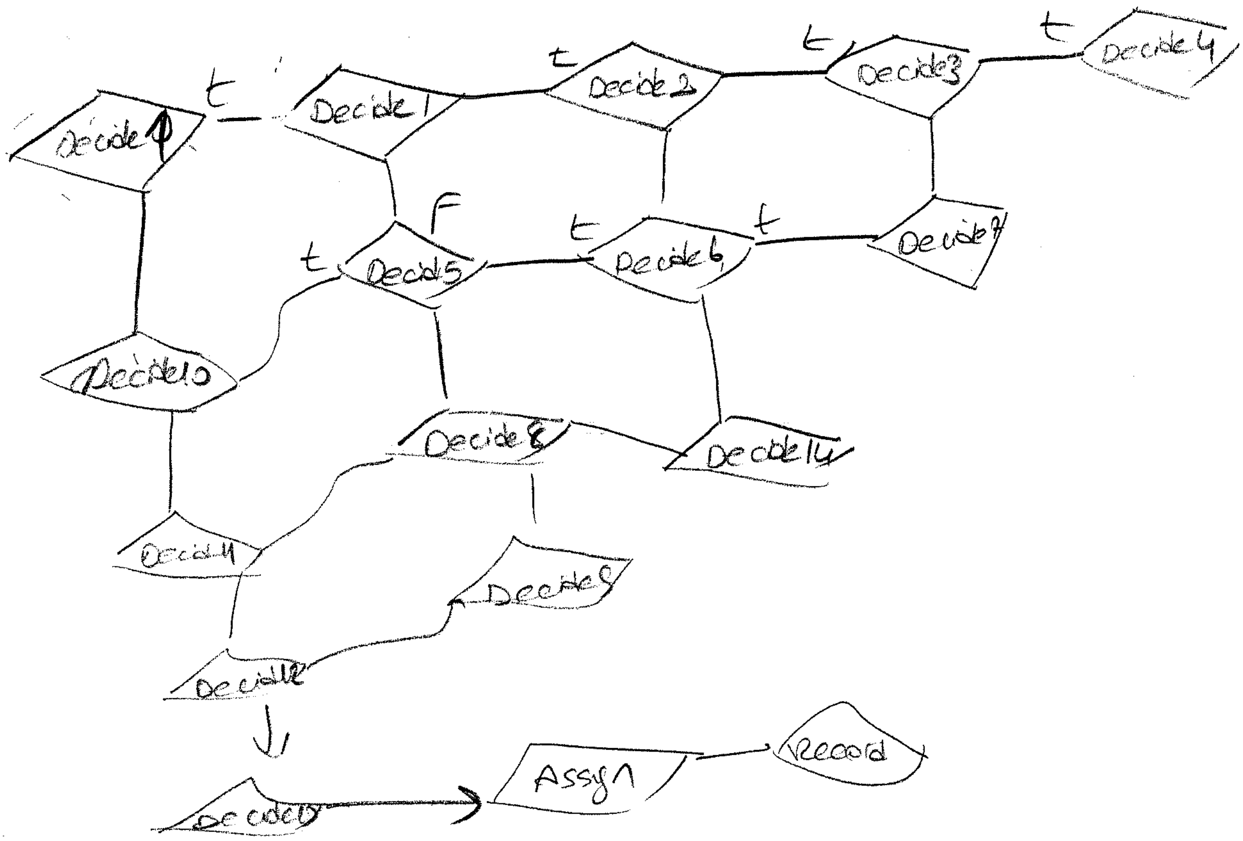


2) Gelişlerarası süre Ex(8), gelen müşterilerin %60'nın bayan %40'ının bay olduğu, bayan ve bay müşteriler için ayrı ayrı tıraş koltuklarının bulunduğu ve tıraş sürelerinin bayanlar için normal (10,4), baylar için normal (8,3) dağılımına uyan bir berberde 200 dk çalışılması durumunda hizmet gören müşteri sayısı, hizmet gören bay ve bayan sayıları, hizmet görenlerin ortalama sistemde geçirdiği süreleri (bay ve bayan için ayrı), kayıp müşteri sayısını ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız.



3) Gelişlerarası süre Ex(8) olan, gelen müşterilerin %60'ı bayan %40'ı bay olduğu, bayan ve bay müşteriler için ayrı ayrı tıraş koltuklarının bulunduğu, bayanlar için bekleme salonu kapasitesinin 5 kişi, baylar için 4 kişi olduğu ve tıraş sürelerinin bayanlar için normal (10,4), baylar için normal (8,3) dağılımına uyan bir berberde 200 dakika çalışılması durumunda oluşacak; hizmet gören müşteri sayısı, hizmet gören bay ve bayan sayılarını, hizmet görenlerin ortamda sistemde geçirdiği süreleri, kayıp müşteri sayısını ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız.

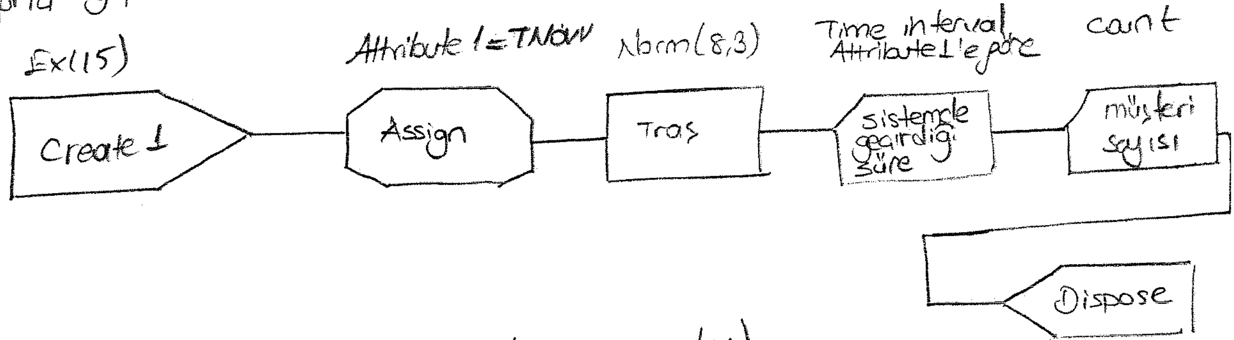



Decide 0
Decide 1
Decide 2
Decide 3
Decide 4
Decide 5
Decide 6
Decide 7
Decide 10
Decide 8
Decide 14
Assign
Record

ARENA ÖRNEKLERİ

Berber Örneği;

1) Gelişler arası süre Ex(15) olan ve tıraş süreleri normal (8,3) dağılımına uyan bir berberde 200 dk çalışılması durumunda oluşacak hizmet gören müşteri sayısı, hizmet görenlerin ortalama sistemde geçirdiği süre ve kuyruk durumlarını gösteren Arena simülasyonu yapınız.



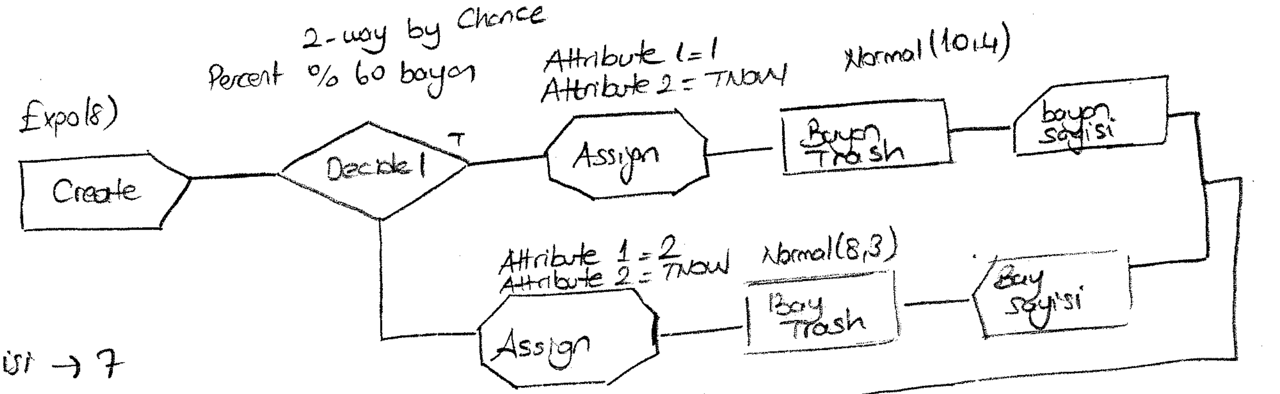
Run ⟶ Setup ⟶ Replication Length (200 minutes)

Sistemde hizmet gören müşteri sayısı ⟶ 12

Sistemde geçen süre ⟶ 0.1734 (ortalama)

Geliştirilmiş Berber Örneği

2) Gelişler arası süre Ex(8) olan, gelen müşterilerin %60 bayan, %40'ı bay olduğu, bayan ve bay müşteriler için ayrı ayrı tıraş koltuklarının bulunduğu ve tıraş sürelerinin bayanlar için normal (10,4), baylar için normal (8,3) dağılımına uyan bir berberde 200 dakika çalışılması durumunda oluşacak hizmet gören müşteri sayısı, hizmet gören bay ve bayan sayılarını, hizmet görenlerin ortalama geçirdiği süreleri (bay ve bayan için ayrı), kayıp müşteri sayısını ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız.

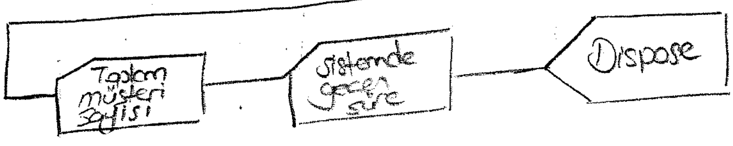


Bay sayısı ⟶ 7

Bayan sayısı ⟶ 10

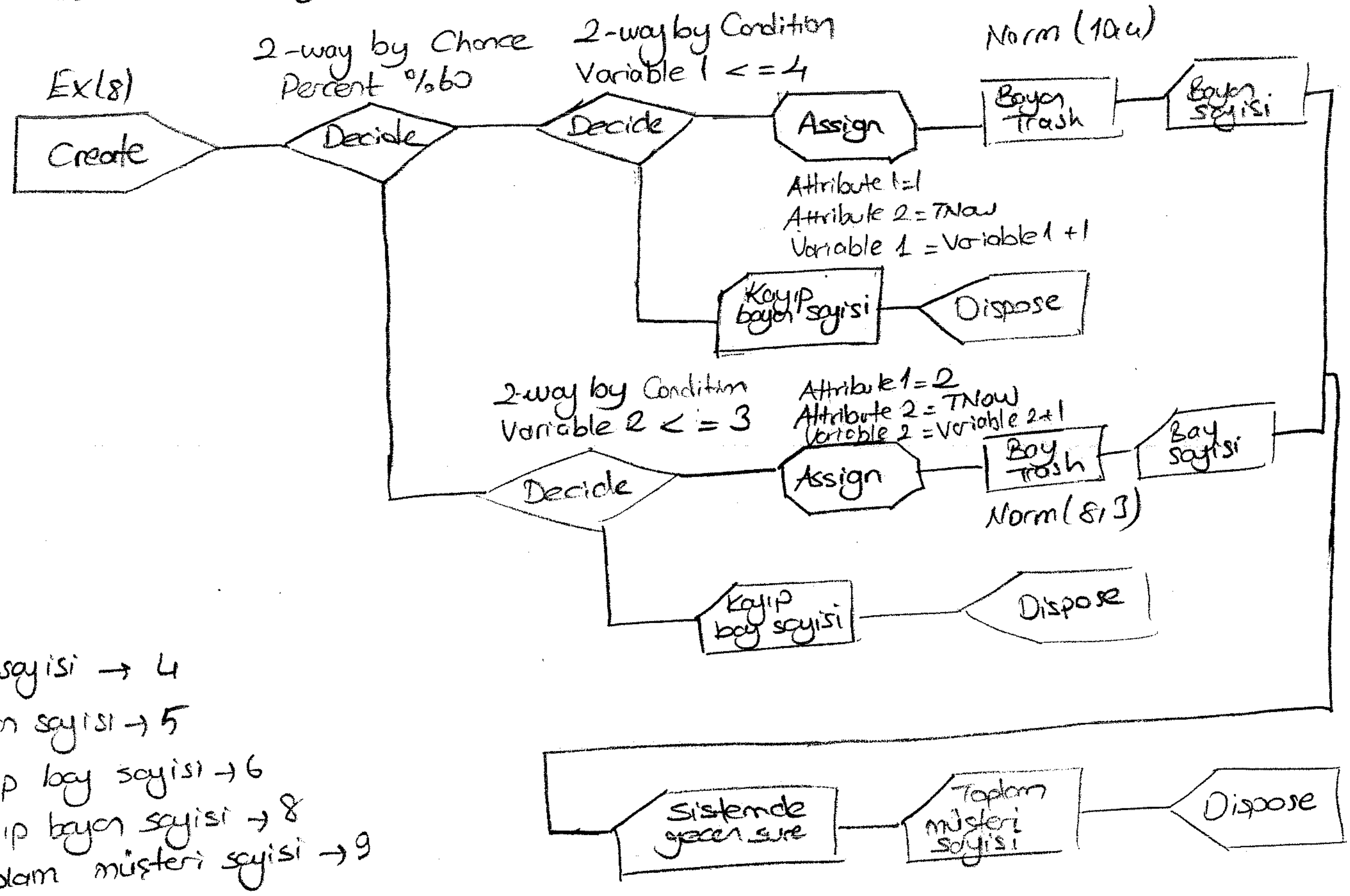
Toplam müşteri sayısı ⟶ 17

Sistemde geçen süre ⟶ 0.6772 (ort)



Bekleme Yeri Sınırlı Berber Örneği

3) Gelişler arası süre Ex(8) olan, gelen müşterilerin %60'ı bayan, %40'ının bay olduğu, bayan ve bay müşteriler için ayrı ayrı traş koltuklarının bulunduğu bayanlar için bekleme salonu kapasitesinin 5 kişi, baylar için 4 kişi olduğu ve traş sürelerinin bayanlar için normal (10,4) baylar için normal (8,3) dağılımına göre 200 dakika çalışılması durumunda düşecek hizmet gören müşteri sayısı, hizmet gören bay ve bayan sayılarını, hizmet görenlerin ortalama sistemde geçirdiği süreleri (bay ve bayan için ayrı ayrı), kayıp müşteri sayısını ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız

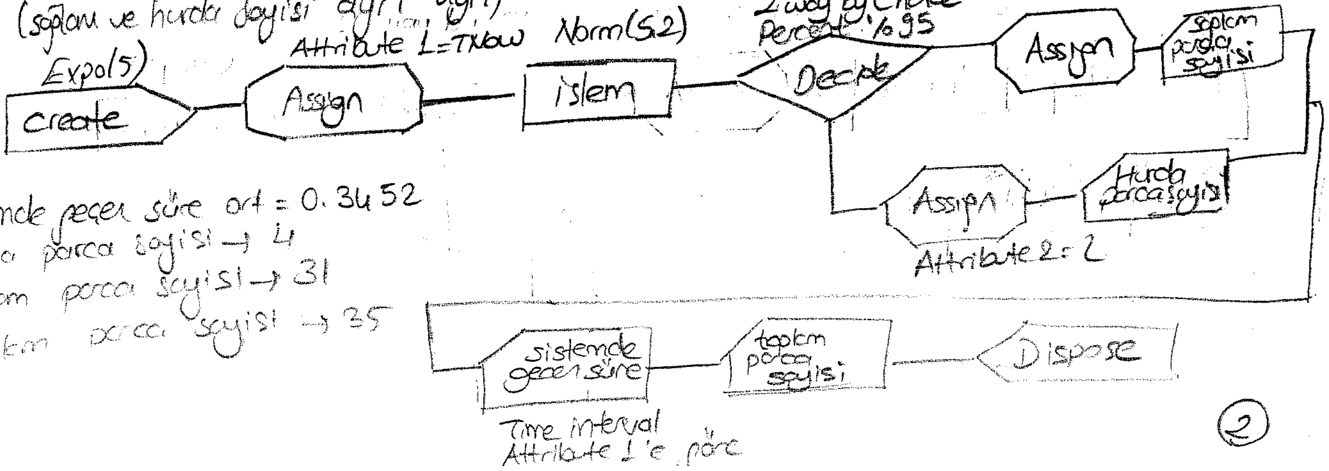


Bay sayısı → 4
Bayan sayısı → 5
Kayıp bay sayısı → 6
Kayıp bayan sayısı → 8
Toplam müşteri sayısı → 9

Sistemde geçen süre ort → 0.5001

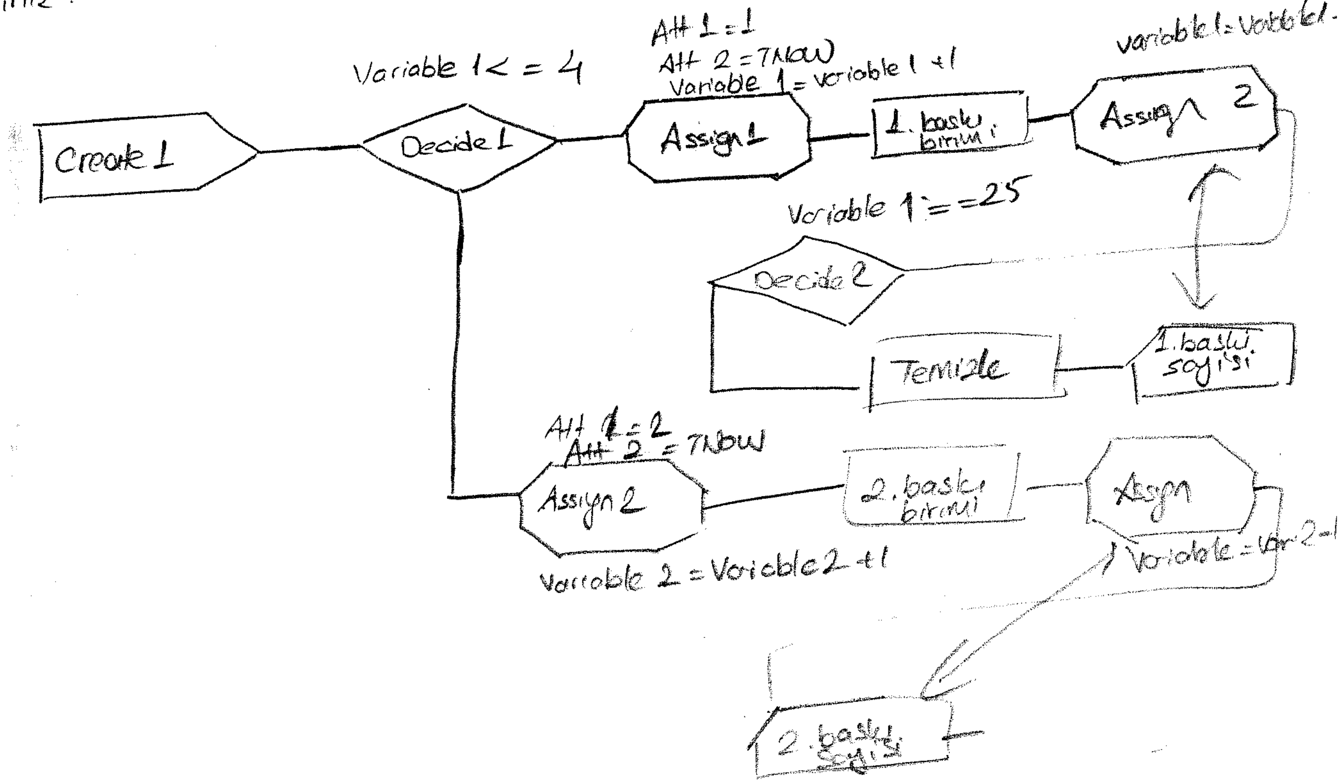
Atölye Örneği

4) Gelişler arası süre Ex(5) olan, gelen parçaların %5'inin hurda %95'inin sağlam olduğu, işlem süresinin normal (5,2) dağılımına uyan bir atölyede işlem gören parça sayıları, sistemde geçen süreler ve kuyruk durumunu veren Arena simülasyonunu yapınız. (sağlam ve hurda sayısı ayrı ayrı)



Sistemde geçen süre ort = 0.3452
Hurda parça sayısı → 4
Sağlam parça sayısı → 31
toplam parça sayısı → 35

ÖRNEK: Bir davetiye baskı sürecinde kağıt yığınları EXPO(10) varışlar arası zaman ile sisteme varmaktadır. Gelen kağıtlar için iki baskı birimi bulunmaktadır. Birincil ve ikincil. Bütün varışlar birincil baskı birimine yönlendirilmektedir. Eğer birincil baskı birimi önündeki kuyruk 5'ten az ise kağıt yığınları birincil baskı birimi önünde bir kuyruk oluşturur. Kağıt yığınlarının birincil baskı biriminde TRIA (9,12,15) süresince bir işlem görmektedir. Eğer birincil baskı biriminde 5 veya daha fazla kağıt var ise gelen işler ikincil baskı birimine yönlendirilecektir. İkincil baskı birimi sonsuz yığın kuyruğuna sahiptir. Birinci baskı birimi 25 davetiye bastıktan sonra temizleme için kapatılması gerekmektedir. Bu işlem EXPO(30) kadar süre almaktadır. Temizleme esnasında birincil baskı birimi önündeki kağıt yığını bu birim aktif olana kadar bekleyecektir. Bu simülasyonu Arena ile gerçekleştirerek her bir kaynağın kullanım oranlarını, bir davetiyenin sistemde geçirdiği zaman vb. istatistikleri hesaplayınız ve bunları grafiksel olarak gösteriniz. Simülasyonu 50 saat için çalıştırınız.



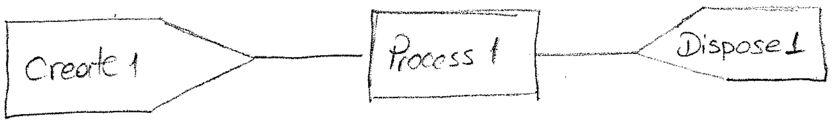
Örnek: Gelişler arası süre Ex(15) olan ve tıraş süreleri normal (8,3) dağılıma uyan bir berberde 200 dakika çalışılması durumunda oluşacak hizmet gören müşteri sayısı, hizmet görenlerin sistemde geçirdiği süre ve kuyruk durumlarını gösteren Arena programı=?

ÖRNEKLER

1) Herhangi bir tipteki varlıklar sistemimize gelmekte, işlem görmekte ve işi biten varlıklar sistemden ayrılmaktadır. Ancak işleme tabi tutulmaları için bir kaynağın (veya işlem yapan bir çalışanın veya makinenin) var olması gerektiğini ve eğer bu kaynaktan elimizde yoksa varlıkların kaynaklardan elimizde olana kadar bekleyeceklerini biliyoruz. Birden fazla varlık kaynak için bekliyorsa, önce gelenin önce işlem göreceğini kabul ederiz.

Kuyruk Teorisi Notasyonu: M/M/n/FIFO

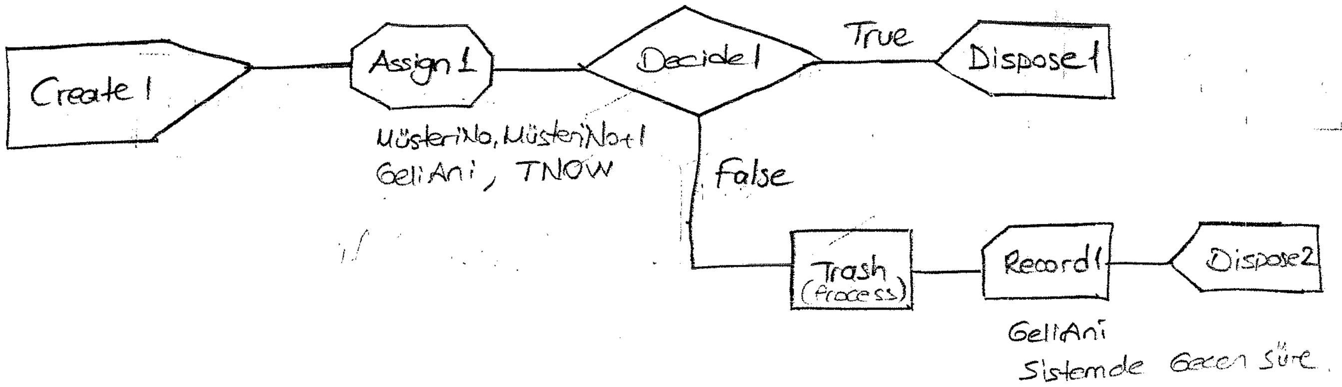
Sistemimizde gelişler arası sürenin ortalama 8 dk olduğunu, herbir varlığın ortalama 10 dakika işlem gördüğünü ve işlem yapan 2 sunucumun olduğunu kabul edelim.



* Varlıkların "Process" içerisinde ne kadar zaman geçireceğini belirtmek için öncelikle "Delay Type" seçilmelidir. Bu listede "Constant" (sabit) seçilirse, varlıkların süreç içinde sabit bir süre kalacaklarını belirtmiş oluruz. Bizim problemimizde varlıklar üssel dağılımdan ortalama 10 dk kaldıklarına göre "Delay Type" olarak "Expression" seçilir, ve alttaki listeden "EXPO (mean)" seçilir. Mean yazan yere 10 yazılır. "Units" bölümü "Minutes" olarak seçilmelidir. Böylece ortalaması 10 dk olan üssel dağılımdan değerler üretecektir.

2) Sistemimizde gelişler arası sürenin ortalama 12 dk olduğunu ve herbir varlığın ortalama 8 dk işlem gördüğünü ve işlem yapan sunucu 1 iken ve 2 iken sistemin arenada simülasyonunu yapınız. Grafiğini çiziniz.

3) Bir erkek kuaföründe tıraş kuyruğunun simülasyonu yapılmıştır. Kuaföre gelen müşteriler sıraya girer. Müşteri sırası FIFO mantığıyla çalışmaktadır. Bir müşteri kuaföre girdiğinde eğer tıraş kuyruğu 3 kişi ise kuaförden çıkmaktadır. Tıraş kuyruğu 3 kişiden az ise müşteri kuyruğa girerek tıraş olmaktadır.



STOKASTİK ÜRETEÇLER

RASSAL SAYI ÜRETEÇLERİNDEN İSTENEN ÖZELLİKLER

1. Rassallık
2. Büyük Periyod
3. Yeniden Üretilebilirlik
4. Hesaplama Etkinliği

TEK DÜZE DAĞILIMLI RASTGELE SAYILAR

* Dil derleyicileri $[0,1]$ aralığında tek düze dağılımlı rastgele sayılar için olanak sağlar. Böylece yordamlar $U(0,1)$ olarak bilinir.
* Örneğin; BASIC dilinde RND çağrısı $0 <= X <= 1$ aralığında bir x kesri döndürecektir.
* Bu, ayrık bir rastgele değişkendir; fakat pratikte sürekli olduğu varsayılır.
* 100 defa RND fonksiyonunu çağırırsak;

%10'u 0 ile 0.1 aralığında

%10'u 0.1 ile 0.2 aralığında

⋮

%10'u 0.9 ile 1.00 aralığında

dağılımlar oluşacaktır.

RASSAL SAYI ÜRETİMİ İÇİN TEKNİKLER

* Orta Kare Yöntemi
* LCG
* Hull-Dobell Yöntemi

1) ORTA KARE YÖNTEMİ:

1. m basamaklı genellikle tek olan bir sayı başlangıç değeri olarak alınır.
2. Bu sayının karesi alınır. Sonucun ortasında m basamaklı sayı alınır.
3. Bu sayı rassal sayı olarak kaydedilir.
4. Elde edilen sayının tekrar karesi alınır. Ortasında m basamaklı sayı tekrar rassal sayı olarak kaydedilir.
5. İstenilen sayıda rassal sayı üretene kadar bu işlemlere devam edilir.

Örnek: $X_0 = 5497$ ($m=4$ basamaklı)

$(X_0)^2 = 30\underbrace{2170}_{4 \text{ basamaklı}}09$ $\quad U_1 = \frac{2170}{10000} = 0,2170$ ($U(0,1)$ olduğundan)

↳ 1. rassal sayımız

$X_1 = 2170$ ($m=4$ basamaklı)

$(X_1)^2 = 4\underbrace{7089}_{4 \text{ basamaklı}}00$ $\quad U_2 = \frac{7089}{10000} = 0,7089$

↳ 2. rassal sayımız

$X_2 = 7089$ (m=4 basamaklı)

$(X_2)^2 = 50\underbrace{2539}_{\text{4 basamaklı}}21$

$U_3 = \frac{2539}{10000} = 0,2539$ ↳ 3. rassal sayımız

$X_3 = 2539$

1) DEZAVANTAJLARI

* İlk sayı ve tekrar uzunluğu (periyod) arasındaki ilişkiyi önceden bilmek mümkün değildir. Çoğu kez tekrar uzunluğu kısadır.
* Elde edilen sayılar rassal olmayabilir. Yani dizide dejenerasyon söz konusu olabilir.

2) LCG (Linear Congruential Generators = Lineer Eşleşiksel Üreteçler)

* Tek düze rastgele sayı üreteçlerinin çoğu LCG şeklindedir.
* LCG deterministik olup bir algoritmaya bağlıdır.
* Başlamak için bir ilk değer çekirdeğe (z_0) ihtiyaç duyar.
* Bu çekirdek ve z_k dizisinin ardışıl terimleri bir LCG formülüne uygulanır.
* Ardından z_k, $0 \leq U_k \leq 1$ aralığında bir U_k çıkışına normalize edilir.

$\boxed{z_0 : \text{çekirdek}}$ $\boxed{z_{k+1} = (a.z_k + c) \bmod m}$

a: çarpan
c: artım
m: genlik

$\boxed{U_k = \frac{z_k}{m}}$

Örnek: $a=5$; $c=3$, $m=16$, ve $z_0 = 7$ değerleri ile LCG kullanarak oluşturulan sayı dizisini belirleyelim.

$z_0 = 7$

$U_0 = \frac{7}{16} = 0,4375$

$z_{k+1} = (a.z_0 + c) \bmod m$

$z_1 = (5.7+3) \bmod 16$

$z_1 = \underline{\underline{6}}$

$U_1 = \frac{6}{16} = 0,375$

$z_2 = (5.6+3) \bmod 16$

$z_2 = 1$ //

$U_2 = \frac{1}{16} = 0,0625$

$z_3 = (5.1+3) \bmod 16$

$z_3 = 8$

$U_3 = \frac{8}{16} = 0.5$

$z_4 = (5.8+3) \bmod 16$

$z_4 = 11$ //

$U_4 = \frac{11}{16} = 0,6875$

k	z_k	u_k
0	7	0.4375
1	6	0.375
2	1	0.0625
3	8	0.500
4	11	0.688
5	10	0.625
6	5	0.313
7	12	0.750
8	15	0.938
9	14	0.875
10	9	0.563
11	0	0.000
12	3	0.188
13	2	0.125
14	13	0.813
15	4	0.250
16	7.	0.4375

* Dizinin ilk 17 elemanı yandaki gibidir.
* m adet tekrar için m farklı sayının oluştuğu durumda seçilen LCG'nin **tam periyoda** sahip olduğu söylenir.
* m tekrarlı durum için m tane farklı rastgele sayı oluştuğundan LCG seçimi tam periyoda sahiptir. ($m=16$)
* z_k'nın bir tekrarında tam bir döngü izler.
* Buradaki LCG tam periyoda sahiptir.

$u_{16} = \frac{z_{16}}{m} = \frac{7}{16} = 0.4375$ (u_0'ın aynısı)

Bu yüzden max m farklı değer üretir dedik.

* Tam periyoda sahip olup olmadığı uygun a, c ve m değerleri ile belirlenir.

Bunun şartları;

1. c ile m aralarında asal (ikisinin de en büyük ortak böleni 1), $a=1+4k$ ($k=0,1,2,\dots$), $m=2^b$ ve $c\neq 0$ ise;

$$\boxed{\text{Periyot} = m = 2^b \text{ 'dir}}$$

2. z_0 başlangıç değeri tek sayı, a değeri; $a=3+8k$ veya $a=5+8k$ ($k=0,1,2\dots$) $m=2^b$ ve $c=0$ ise;

$$\boxed{\text{Periyot} = m/4 = 2^{b-2} \text{ 'dir}}$$

(3) a çarpanına bağlı; en küçük k tamsayısı için ; $a^k - 1 \mod m = 0$, m asal sayı , $c=0$ ise;

$\boxed{\text{Periyot} = m-1 \text{ 'dir.}}$

(öyle bir k bulunacak ki olabilecek kadar küçük olacak ve m'e bölümü 0 olacak.)

Örnek: $a=13$ $m=2^6=64$ $z_0 = 1,2,3$ ve 4 için tam periyoda sahip olup olmadığını bulunuz. ($c=0$)

$m=2^6$, $z_0=1$ ve 3 için $a = 5+8k_{41} = 13$ $c=0$ - - -> kural 2

$\text{Periyot} = m/4 = 64/4 = \underline{\underline{16}}$

$z_0=1$ ve 3 için periyot $= \underline{\underline{16}}$

$z_0=2$ ve 4 için hiçbir kurala uymadığı için tek tek bakarız.

i	X_i	X_i	X_i	X_i
0	1	2	3	4
1	13	26	39	52
2	41	18	59	36
3	21	42	63	20
4	17	34	51	4
5	29	58	23	52
6	57	50	43	
7	37	10	47	
8	33	2	35	
9	45	26	7	
10	9		27	
11	53		31	
12	49		19	
13	61		55	
14	25		11	
15	5		15	
16	1		3	

* $z_0=1$ ve 3 için 2. kural sağlandığı için periyot = 16 olmuştur. 2. kuralı sağladığından tam periyodu 16 çıkar.
* $z_0=2$ ve 4 için hiçbir kuralı sağlamadığı için tek tek bakarız.
  $z_0=2$ için periyodu = 8
  $z_0=4$ için periyodu = 4

$z_0=1$ ve 3 için kuralı sağladığından periyodiktir deriz.

Örnek: $z_0 = 63$ $a = 19$ $c = 0$ $m = 10^2 = 100$

Çözüm: $c=0$ $z_0 = 63$ tek sayı $a = 3+8\underbrace{k}_{2} = 19$ $m = 2^b \neq 10^2$ olduğu için hiçbir kuralı sağlamaz. Tek tek bakarız.

$z_0 = 63$

$z_{k+1} = z_1 = (19.63+0) \mod 100 = 97$

$z_2 = (19.97+0) \mod 100 = 43$

$z_3 = (19.43+0) \mod 100 = 17$

$z_4 = (19.17+0) \mod 100 = 23$

$z_5 = (19.23+0) \mod 100 = 37$

$z_6 = (19.37+0) \mod 100 = 3$

$z_7 = (19.3+0) \mod 100 = 57$

$z_8 = (19.57+0) \mod 100 = 83$

$z_9 = (19.83+0) \mod 100 = 77$

$z_{10} = (19.77+0) \mod 100 = 63$ → **10. adımda kendini tekrar ediyor.**

Periyot = 10

3) HULL - DOBELL TEOREMİ

* Bu teorem tam periyodu elde etmek için gerekli ve yeterli şartları sağlar.
* LCG ancak ve ancak aşağıdaki üç şartı sağlıyorsa tam periyoda sahiptir.

I. a ve c asal olmalı.

II. m sayısının bölünebildiği bütün asal sayılara, a-1'de bölünebilmeli

III. Eğer m 4'e bölünebiliyorsa a-1'de 4'e bölünmeli.

Örneğin; $a=5$ $c=3$ $m=16$ ve $z_0 = 7$

* 3 ve 5 asal (I. şart)
* 16'nın bölünebildiği asal sayılar : 2 , $a-1=5-1=4$ a-1'de 2'ye bölünmeli. 4 , 2'ye bölünür. (II. şart)
* Eğer $m=16$, 4'e bölünebiliyorsa ; $a-1=5-1=4$ 'te 4'e bölünmeli. 4 , 4'e bölünür. (III. şart)

Bütün şartları sağladığı için **tam periyoda** sahiptir.

→ Bir bilgisayar uygulaması bu algoritmayı donanım açısından ele alır. Çünkü işlemler hesaplama ve hız odaklıdır.

→ İşlem makineye shift register kullanılarak yaptırılır.

→ m , 2'nin kuvveti şeklinde ele alınır.

* Yukarıdaki örnekte; $m=16=2^4$ 'tür. Dolayısıyla LCG 4-bit shift register ile tamsayıları gösterebilir.

$R = [\,r_1 \quad r_2 \quad r_3 \quad r_4\,]$

Register içeriği 4 bit olacaktır.

* $Z_6 = 5$ olduğundan $R = [0101]$ 'dir. Z_7'yi elde etmek için;

* $Z_7 = 5 \cdot Z_6 + 3$ $R = [1\ 1100] = 28$ Burada baştaki 1 shift register 4 bit olduğundan kaybedilir. $28 \pmod{16} = 12$ $R = [1100]$

* İkili nokta uygulandığında $(0.1100)_2 = 0.75$'dir.

→ Gerçek bilgisayarlarda farklı ölçüde üreteçler vardır. IBM'in RANDU üreteçleri, $a = 2^{16}+3$ $c=0$ ve $m = 2^{31}$ değerlerine sahiptir.

ÜRETEÇLERİN İSTATİSTİKSEL ÖZELLİKLERİ

* Donanım hesaplanabilirliği için seçilen mod işlemi ve periş bir periyoda sahip olmanın yanı sıra bir $U[0,1]$ üreteci istatistiksel anlamda iyi olmalıdır.

Şu iki özelliği sağlaması önemlidir;

1) Üreteç Tek Düze Olmalı (Uniform): Herhangi bir L uzunluk aralığında oluşan sayıların miktarı, diğer bir L uzunluk aralığında oluşan sayıların miktarına yakın olmalıdır.

2) Dizi Bağımsız Olmalı: Herhangi bir sayı, bir sonrakine etkisini göstermemeli. Aksi halde dizi boşluk veya gruplama eğilimi gösterir.

* Üreteçleri test etmek için teorik ve deneysel araçlar vardır.
* Birinci özelliği test etmek için Chi-Square (Ki Kare) testi uygulanır.

Ki - Kare Testi

* Beklenen frekans değerleri ile gözlenen frekans değerlerini karşılaştırıp aradaki uyuma bakılmasıdır.

Frekans Dağıtım Tablosu

Aralık Sayısı	Aralık	Deneysel frekans (f_k)	Beklenen frekans (e_k)
1	$[0, 1/m]$	f_1	e_1
2	$[1/m, 2/m]$	f_2	e_2
3	$[2/m, 3/m]$	f_3	e_3
⋮	⋮	⋮	⋮
m	$[\frac{m-1}{m}, 1]$	f_m	e_m

→ Bu test için FDT'den (Frequency Distribution Table) faydalanılır.

→ n tane rastgele sayı oluşturularak ve her birini bir m sınıfına atayarak $f_1, f_2, \dots, f_m$ frekansları çizelgeye geçirilir.

→ Her bir sınıf için beklenen frekansı ile karşılaştırılır. $e_k = \frac{n}{m}$

* $$\boxed{\chi^2 = \sum_{k=1}^{m} \frac{(f_k - e_k)^2}{e_k} = \frac{m}{n} \sum_{k=1}^{m} \left(f_k - \frac{n}{m}\right)^2}$$

* $V = m-1$ → bağımsızlık derecesi

Örnek: SNAFU olarak isimlendirilen $U[0,1]$ üreteci 100 sayı üretilerek test edilmiş ve frekansları sayılmıştır. Frekans değerleri aşağıda verilmiştir.

$0.00 \le x < 0.25$ Sonuçlar; $f_1 = 21$
$0.25 \le x < 0.50$ $f_2 = 31$
$0.50 \le x < 0.75$ $f_3 = 26$
$0.75 \le x < 1.00$ $f_4 = 22$ 'dir.

Üretecin uniform olup olmadığını bulunuz.

Çözüm: $n = 100$ (örnek sayısı), $m = 4$ (aralık sayısı) $\Big\}$ $\frac{n}{m}$ her sınıfta olması beklenen sayıların sayısı

$$e_k = \frac{n}{m} = \frac{100}{4} = \underline{\underline{25}}$$

Ki-Kare Testinin Formülü;

$$X^2 = \sum_{k=1}^{m} \frac{(f_k - e_k)^2}{e_k} = X^2 = \frac{(21-25)^2}{25} + \frac{(31-25)^2}{25} + \frac{(26-25)^2}{25} + \frac{(22-25)^2}{25}$$

$$= \underline{\underline{2.48}}$$

*Bağımsızlık derecesi $V = m - 1$

$V = 4 - 1 = 3 \rightarrow$ ki-kare tablosunun 3. satırı

X^2 değeri; $\alpha = \%95$ $X_c^2 = 7.81$ olduğu ki-kare tablosundan bulunabilir.

$X^2 < X_c^2$ olduğundan uniform olduğu söylenebilir.

TEK DÜZE OLMAYAN RASTGELE DEĞİŞKENLERİN ÜRETİMİ

* İsteğe bağlı sayıları oluşturabilmek için bilinen bazı algoritmalar;

1. Ters Dönüşüm Metodu
2. Red Metodu
3. Konvolüsyon Metodu

1) TERS DÖNÜŞÜM (INVERSE) METODU

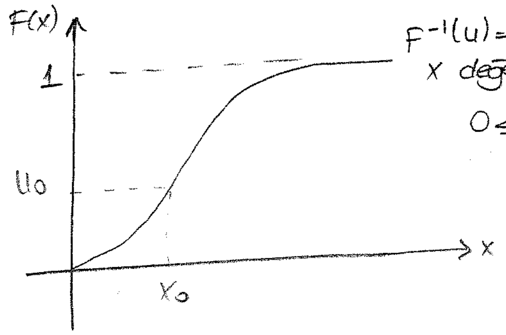
* $f(x)$ dağılık yoğunluk fonksiyonunun verildiğini kabul edelim. Amaç $f(x)$'ten bir rassal değişken üretmektir.

$$F(x) = \int_{-\infty}^{\infty} f(x)\, dx \qquad 0 \le F(x) \le 1$$

$u = F(x)$ için $\quad x = F^{-1}(u)$ ters fonksiyon

$u \sim U(0,1)$

Yani x'e karşılık gelen U'ların üretilmesi gerek.



$F^{-1}(u) = x$ ifadesi; u değerine karşılık gelen x değerlerini belirler.

$0 \le F(x) \le 1$ $F(x)$ artan bir fonksiyondur.

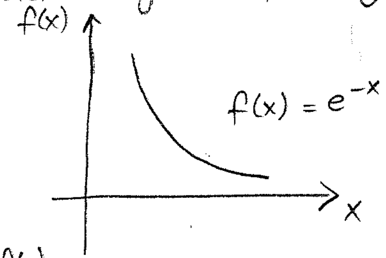
Algoritması;

1. $u \sim U(0,1)$ rastgele değişken üret.
2. $x = F^{-1}(u)$ 'dan x rastgele değerlerini hesapla.
3. RETURN

NOT: Hangi dağılımlar için rassal değişkenler üretilir?

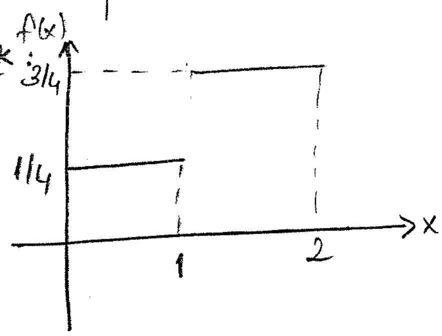
→ Üssel
→ Uniform
→ Üçgen dağılım
→ Sürekli ve ayrık deneysel dağılımlar

NOT: Üstel fonksiyonda $f(x)$ şöyledir;



$0 \leq f(x) \leq 1$ olmalı.

Örnek:



$$f(x) = \begin{cases} 1/4, & 0 \leq x < 1 \\ 3/4, & 1 \leq x < 2 \end{cases}$$

Yukarıda verilen $f(x)$ parçalı fonksiyonu kullanarak ters dönüşüm yöntemiyle rassal değişken üretiniz.

Çözüm: ① $0 \leq x \leq 1$ aralığı için;

$$U = F(x) = \int_0^x \frac{1}{4}\,dx = \frac{1}{4}\Big|_0^x = \frac{1}{4}x \qquad (0 \leq x \leq 1)$$

$$U = \frac{1}{4}x \Rightarrow \boxed{x = 4u} \Rightarrow \left.\begin{array}{lll} x=0 & \text{için} & u=0 \\ x=1 & \text{için} & u=\frac{1}{4} \end{array}\right\} \; 0 \leq u \leq \frac{1}{4}$$

② $1 \leq x < 2$ aralığı için;

$$U = F(x) = \int_0^1 \frac{1}{4}\,dx + \int_1^x \frac{3}{4}\,dx = \frac{1}{4} + \frac{3}{4}x - \frac{3}{4} = \frac{3}{4}x - \frac{1}{2}$$

$$U = \frac{3}{4}x - \frac{1}{2} \Rightarrow \boxed{x = \frac{4u}{3} + \frac{2}{3}} \Rightarrow \begin{array}{lll} x=1 & \text{için} & u=1/4 \\ x=2 & \text{için} & u=1 \end{array}$$

$$F^{-1}(u) = \begin{cases} 4u, & 0 \leq u \leq 1/4 \\ \frac{4u}{3} + \frac{2}{3}, & \frac{1}{4} \leq u < 1 \end{cases}$$

Örnek:

$$f(x) = \begin{cases} \lambda . e^{-\lambda x} & , \quad x \geq 0 \\ 0 & , \quad x < 0 \end{cases}$$

olasılık yoğunluk fonksiyonu verilmiştir. Buna göre $F^{-1}(u) = ?$

Çözüm: Verilen olasılık yoğunluk fonksiyonu; aynı zamanda kümülatif olasılık yoğunluk fonksiyonudur. (Eğri altındaki alan)

$$F(x) = \int_{-\infty}^{0} f(t)dt = \int_{-\infty}^{0} \lambda . e^{-\lambda x} dx = 1 - e^{-\lambda x} \Rightarrow \begin{cases} 1-e^{-\lambda x} & x \geq 0 \\ 0 & , \quad x < 0 \end{cases}$$

$$F(x) = U = 1 - e^{-\lambda x}$$

$U = 1 - e^{-\lambda x} \Rightarrow e^{-\lambda x} = 1 - u$ (Her iki tarafın ln' inini alırsak)

$$-\lambda x = \ln(1-u)$$

$$x = -\frac{\ln(u)}{\lambda}$$

1. integral al.
2. U'ya eşitle.
3. X'i yalnız bırak.

Son olarak rassal U değişkenleri üret. Buna karşılık gelen X'i hesapla. $\lambda = 1$ olsun.

Hesaplanan U'iler;

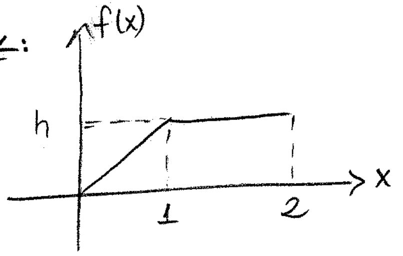
$U = 0.1306$ için $X_1 = -\ln(1-u) = 0.14$

$U = 0.0422$ için $X_2 = -\ln(1-u) = 0.0431$

$U = 0.6597$ için $X_3 = -\ln(1-u) = 1.078$

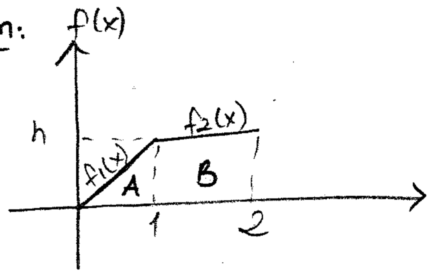
$U = 0.7915$ için $X_4 = -\ln(1-u) = 1.56$

Örnek:



f(x)'e uygun rassal değişken üreten algoritmayı ters dönüşüm ile çıkartınız.

Çözüm:



(0,1) aralığında çalıştığımızdan eğri altındaki alan maksimum 1 olabilir. A+B=1

$$1.h + \frac{1.h}{2} = 1$$

$$h = \frac{2}{3}$$

$f_1(x) = h.x$ → orjinden geçer

$f_2(x) = h$ → sabit

$$F(x) = \begin{cases} hx & , \quad 0 \leq x \leq 1 \\ h & , \quad 1 \leq x \leq 2 \end{cases}$$

$$F(x) = \begin{cases} \frac{2}{3}x & , \quad 0 \le x \le 1 \\ \frac{2}{3} & , \quad 1 \le x \le 2 \end{cases}$$

① $U = \int_0^x \frac{2}{3}x\,dx = \frac{1}{3}x^2 \qquad 0 \le x \le 1$

$U = \frac{1}{3}x^2 \Rightarrow x^2 = 3u \Rightarrow x = \sqrt{3u}$

$x=0$ için; $u=0$

$x=1$ için; $u=\frac{1}{3}$

② $U = \int_0^1 \frac{2x}{3}\,dx + \int_1^x \frac{2}{3}\,dx = \frac{1}{3} + \frac{2}{3}x - \frac{2}{3}$

$U = \frac{2x}{3} - \frac{1}{3} \Rightarrow x = \frac{3u+1}{2}$

$1 \le x \le 2$

$x=1$ için; $u=\frac{1}{3}$

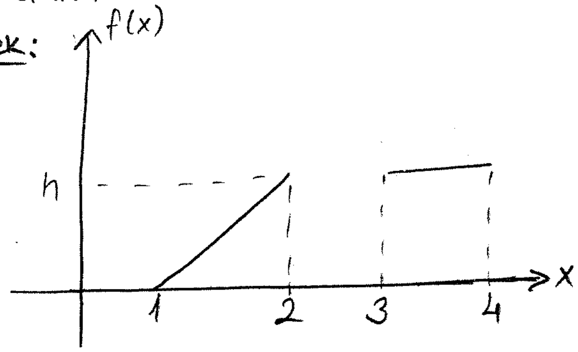
$x=2$ için; $u=1$

$$F^{-1}(u) = \begin{cases} \sqrt{3u} & , \quad 0 \le u \le \frac{1}{3} \\ \frac{3u+1}{2} & , \quad \frac{1}{3} \le u \le 1 \end{cases}$$

Algoritması:

1) $u \sim U(0,1)$
2) if $u < \frac{1}{3} \Rightarrow x = \sqrt{3u}$
3) if $u > \frac{1}{3} \Rightarrow x = \frac{3u+1}{2}$
4) RETURN

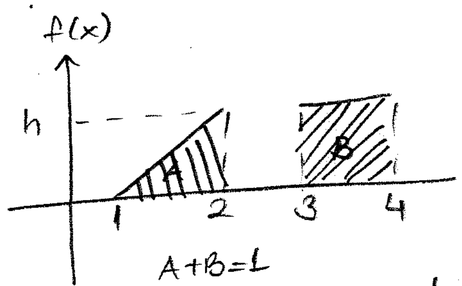
Örnek:



$f(x)$ fonksiyonundan ters dönüşüm tekniği ile rassal değişken üreten algoritmayı yazınız.

Çözüm:

$$f(x) = \begin{cases} f_1(x) & 1 \le x \le 2 \\ f_2(x) & 3 \le x \le 4 \end{cases}$$



$A+B=1$

$\frac{1.h}{2} + 1.h = 1$

$h = \frac{2}{3}$ //

* $f_1(x) \Rightarrow (1,0)$ noktasından geçmekte. Bir noktası verilen doğru denklemi;

$f_1(x) = m.(x - x_1)$ ile bulunur. $m = \tan\alpha = \frac{h}{(2-1)} = \frac{2/3}{1} = 2/3$

$f_1(x) = \frac{2}{3}.(x-1) \Rightarrow f_1(x) = \frac{2}{3}x - \frac{2}{3}$

$f_2(x) \Rightarrow$ sabit h noktasında olduğundan $f_2(x) = \frac{2}{3}$ 'tür.

$$F(x) = \begin{cases} \frac{2}{3}x - \frac{2}{3}, & 1 \leq x \leq 2 \quad (1) \\ \frac{2}{3}, & 3 \leq x \leq 4 \quad (2) \end{cases}$$

(1) $U = F(x) = \int_1^x \left(\frac{2}{3}x - \frac{2}{3}\right) dx = \frac{1}{3}x^2 - \frac{2}{3}x \Big|_1^x = \frac{1}{3}x^2 - \frac{2}{3}x + \frac{1}{3}$

$U = \frac{1}{3}x^2 - \frac{2}{3}x + \frac{1}{3}$

$3u = x^2 - 2x + 1$

$3u = (x-1)^2 \Rightarrow x = 1 + \sqrt{3u}, \quad 1 \leq x \leq 2$

$x = 1$ için; $u = 0$

$x = 2$ için; $u = 1/3$ $\qquad 0 \leq u \leq \frac{1}{3}$

(2) $U = F(x) = \int_1^2 \left(\frac{2}{3}x - \frac{2}{3}\right) dx + \int_3^x \frac{2}{3} dx = \frac{1}{3} + \frac{2x}{3} - 2 = \frac{2x}{3} - \frac{5}{3}$

$U = \frac{2x-5}{3}$,

$3U = 2x - 5 \Rightarrow x = \frac{3U+5}{2}, \quad 3 \leq x \leq 4$

$x = 3$ için; $U = \frac{1}{3}$

$x = 4$ için; $U = 1$ $\qquad \frac{1}{3} \leq u \leq 1$

Sonuç :

$$F^{-1}(u) = \begin{cases} 1 + \sqrt{3u}, & 0 \leq u \leq \frac{1}{3} \\ \frac{3u+5}{2}, & \frac{1}{3} \leq u \leq 1 \end{cases}$$

Örnek :

$$f(x) = \begin{cases} x, & 0 \leq x \leq 1 \\ 2-x, & 1 \leq x \leq 2 \\ 0, & \text{diğer durumlarda} \end{cases} \qquad F(x) = ?$$

Çözüm :

(1) $U = F(x) = \int_0^x x\,dx = \frac{x^2}{2} \Rightarrow u = \frac{x^2}{2} \Rightarrow x = \sqrt{2u}$ $\quad 0 \leq x \leq 1$

$x = 0$ için; $u = 0$

$x = 1$ için; $U = \frac{1}{2}$ $\quad 0 \leq u \leq \frac{1}{2}$

(2) $U = F(x) = \int_0^1 x\,dx + \int_1^x (2-x)\,dx = \frac{1}{2} + 2x - \frac{x^2}{2} - \frac{3}{2}$ $\quad 1 \leq x \leq 2$

$= -\frac{x^2}{2} + 2x - 1 = -\frac{x^2 + 4x - 2 + 6}{2}$

$x = 1$ için; $U = \frac{1}{2}$

$x = 2$ için; $U = 1$

$\frac{1}{2} \leq u \leq 1$

$U = 1 - \frac{(2-x)^2}{2}$

$1 - u = \frac{(2-x)^2}{2}$

$2 - 2u = (2-x)^2 \Rightarrow 2 - \sqrt{2-2u} = x$

$$F^{-1}(u) = \begin{cases} \sqrt{2u}, & 0 \leq u \leq \frac{1}{2} \\ 2 - \sqrt{2-2u}, & \frac{1}{2} \leq u \leq 1 \end{cases}$$

Örnek :

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{diğer durumlarda} \end{cases}$$

Çözüm :

$F(x) = U = \int_a^x \frac{1}{b-a}\,dx = \frac{x}{b-a} - \frac{a}{b-a}$ $\quad a \leq x \leq b$

$U = \frac{x}{b-a} - \frac{a}{b-a} \Rightarrow U + \frac{a}{b-a} = \frac{x}{b-a} \Rightarrow x = U.(b-a) + a$

$x = a$ için; $u = 0$

$x = b$ için; $u = 1$

$F^{-1}(u) = \{ u(b-a) + a, \quad 0 \leq u \leq 1$

Deneysel Sürekli Dağılımdan Rassal Değişken Üretimi

* Deneysel dağılım eğer elimizde veriyi modelleyecek bir fonksiyon yok ise sadece veriler var ise ; verilerin deneysel dağılımını kullanmak gerekir. Bu dağılım aslında parçalı lineer bir fonksiyon gibi düşünülerek yapılır.
* Eldeki deneysel veriler artan sırada sıralanır.
* Daha sonra her deneysel iki nokta arasındaki eğim bulunur.
* Kümülatif olasılıklar kullanılarak değişken üretimi sağlanır.

Örnek : 2.76 1.83 0.8 1.45 1.24 → Gözlemler

$C = \max(x_i : i = 1:15) = 2.76$

$x_1 < x_2 < x_3 < x_4 < x_5$ → Küçükten büyüğe sıralanır.

$0.8 < 1.24 < 1.45 < 1.83 < 2.76$ ($x_0 = 0$ kabul edilir.)

$n = 5$ (gözlem sayısı)

$\frac{1}{n} = \frac{1}{5} = 0.2$ (olasılık)

$x_{i-1} < x < x_i$ (x_{i-1} ile x_i arasında rastgele sayı üreteceğiz)

$$a_i = \frac{x_i - x_{i-1}}{1/n} \quad \text{(eğim)}$$

$$X = F^{-1}(u) = x_{i-1} + a_i\left(u - \frac{(i-1)}{n}\right)$$

$$\frac{i-1}{n} < u \leq \frac{i}{n}$$

$x_{i-1} < x < x_i$ Aralık	olasılık (1/n)	Kümülatif Olasılık (i/n)	Eğim (ai)
$0.0 < x \leq 0.8$	0.2	0.2	4
$0.8 < x \leq 1.24$	0.2	0.4	2.20
$1.24 < x \leq 1.45$	0.2	0.6	1.05
$1.45 < x \leq 1.83$	0.2	0.8	1.90
$1.83 < x \leq 2.76$	0.2	1.0	4.65

$0.0 < x \leq 0.8$ için eğim $= \frac{x_i - x_{i-1}}{1/n} = \frac{0.8 - 0}{0.2} = 4$

$U = 0.71$ için; $\frac{i-1}{n} < 0.71 \leq \frac{i}{n} \Rightarrow \frac{3}{5} < u \leq \frac{4}{5}$ (4. aralıktadır)

$$X = X_{i-1} + a_i \left(u - \frac{(i-1)}{n} \right) \qquad a_4 = \frac{X_4 - X_3}{1/n} = \frac{1.83 - 1.45}{0.2} = 1.9$$

$$= X_{(4-1)} + a_4 \left(u - \frac{(4-1)}{5} \right)$$

$$= X_3 + 1.9 \left(0.71 - \frac{3}{5} \right)$$

$$= 1.45 + 1.9(0.71 - 0.6)$$

$$= \underline{1.66} \quad (4. \text{ aralıkta})$$

* Eğer elimizde büyük bir veri varsa ve herbir aralıkta üretilen rassal sayıların frekansı var ise eğim ve rassal değişken üretimi değişmektedir.

$$a_i = \frac{X_i - X_{i-1}}{C_i - C_{i-1}} \qquad X = F^{-1}(u) = X_{i-1} + a_i (u - C_{i-1})$$

$$\hookrightarrow C_{i-1} < u \leq C_i$$

Örnek: 100 makinenin tamir zamanları veri olarak toplanmıştır. Bu veriler farklı aralıklarda gözlemlerin sayısı olarak kaydedilmiş ve şu şekilde verilmiştir. ($X_0 = 25$)

i	Aralık	Frekans	İlgili Olasılık	Kümülatif Olasılık	Eğim (a_i)
1	$0.25 \leq x < 0.5$	31	0.31	0.31	0.81
2	$0.5 \leq x < 1.0$	10	0.10	0.41	5.00
3	$1.0 \leq x < 1.5$	25	0.25	0.66	2.00
4	$1.5 \leq x < 2.0$	34	0.34	1.0	1.47

$\underline{u = 0.83}$ için X değeri = ?

$C_{i-1} < u \leq C_i \Rightarrow 0.66 < 0.83 \leq 1.0$ olduğundan 4. aralıktadır

$$X = F^{-1}(u) = X_{(4-1)} + a_4 (0.83 - 0.66)$$

$$= 1.0 + 1.47(0.83 - 0.66)$$

$$= \underline{\underline{1.24}}$$

REDDETME TEKNİĞİ

* Reddetme tekniği, sürekli ve sınırlı olan herhangi bir $f(x)$ olasılık yoğunluk fonksiyonundan rassal değişken üretmek için kullanılan genel bir metottur.

Sürekli bir x rassal değişkeni için;

$$0 \leq f(x) \leq f_{max} \quad a \leq x \leq b \text{ 'dır.}$$

* Reddetme tekniği direk teknikler başarısız veya etkin olmadığında kullanılır.

Reddetme Tekniğinin Adımları

* Bu teknikte öncelikle bir t fonksiyonunun tanımlanması gerekir.
* Her x_i için $t(x) \geq f(x)$ olmalıdır.

$$c = \int_{-\infty}^{\infty} t(x)\,dx \geq \int_{-\infty}^{\infty} f(x)\,dx = 1$$

$t(x)$ fonksiyonu bir olasılık yoğunluk fonksiyonu değildir. Çünkü $c > 1$

$$r(x) = \frac{t(x)}{c} \text{ bir olasılık yoğunluk fonksiyonudur.}$$

* $r(x)$ olasılık yoğunluk fonksiyonundan y rassal değişkeni aşağıdaki algoritma ile üretilebilir.

1. $r(x)$ yoğunluk fonksiyonundan y rassal değişkeni üret.
   $u_1 \sim u(0,1)$; $y = x$
2. $u_2 \sim u(0,1)$ üret (y'den bağımsız)
3. $u_2 \leq \dfrac{f(y)}{t(y)}$ ise $x = y$ and return

* $t(x) = q$ olsun.

$$c = \int_a^b t(x)\,dx = \int_a^b q\,dx = qx = q(b-a)$$

$$r(x) = \frac{t(x)}{c} = \frac{q}{q(b-a)} = \frac{1}{b-a}$$

$$r(x) = \begin{cases} \dfrac{1}{b-a} & a \leq x \leq b \\ 0 & \text{d.d} \end{cases}$$

$$R(x) = \int_a^y r(x)\,dx = u$$

$$= \int_a^y \frac{1}{b-a}\,dx = u$$

$$u = \frac{y-a}{b-a} \Rightarrow u\cdot(b-a) + a = y$$

Algoritması :

1. $U_1 \sim U(0,1)$ üret . $y = a + U_1(b-a)$
2. $U_2 \sim U(0,1)$ üret
3. $U_2 \le \dfrac{f(y)}{t(y)}$ ise , $x = y$

Return

değilse go to 1

① $m = 2^b$ $c \neq 0$

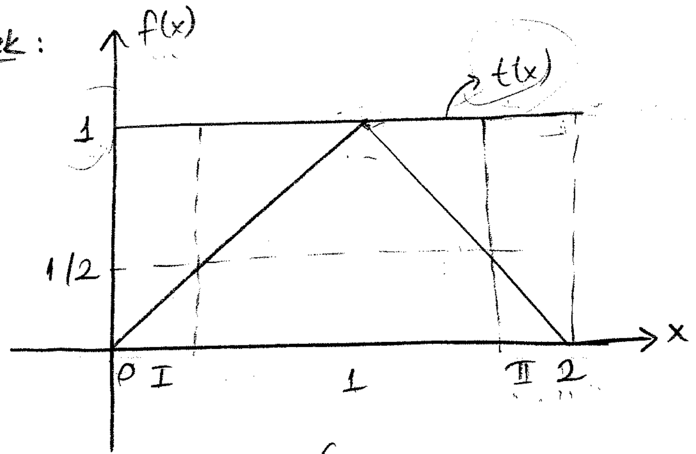
m c asal

$a = 1 + 4k$

m $a - 1 = 4k$

②

Örnek :



$$f(x) = \begin{cases} x & , 0 \le x \le 1 \\ 2 - x & , 1 < x \le 2 \\ 0 & , \text{d.d} \end{cases}$$

Reddetme tekniği ile çözünüz.

Çözüm : $t(x) = \begin{cases} 1 & , 0 \le x \le 2 \\ 0 & , \text{d.d.} \end{cases}$

$c = \int_0^2 t(x)\,dx = \int_0^2 1\,dx$

$c = 2$

$r(x) = \dfrac{t(x)}{c} = \dfrac{1}{2}$

$r(x) = \begin{cases} \frac{1}{2} & , 0 \le x \le 2 \\ 0 & , \text{dd} \end{cases}$

$R(x) = u = \int_0^x \frac{1}{2}\,dx$

$u = \frac{x}{2} \Rightarrow x = 2u$

$0 \le 2u \le 2$

$0 \le u \le 1$

Algoritma

1. $U_1 \sim U(0,1)$ üret . $y = x = 2u_1$
2. $U_2 \sim U(0,1)$ üret
3. $y \le 1$ ve $U_2 \le \dfrac{y}{1} \Rightarrow x = y$

   $y > 1$ ve $U_2 \le \dfrac{(2-y)}{1} \Rightarrow x = y$

   ve Return

   değilse goto 1

SLAYT-2
DİNAMİK SİSTEMLER

* Sürekli sistemlerin matematiksel modelleri sıklıkla diferansiyel denklemler ile ifade edilir.
* Diferansiyel denklemler belirli bir duyarlılık ile sürekli dinamik ortamları tanımlayabilir.
* Bu derste bilgisayar kullanımı ile tahmin edilebilen modeller üzerinde durulacaktır.

Başlangıç Değer Problemleri

* Modellerin genel bir sınıfı dinamik sistemlerdir.
* Dinamik sistemler kendi sistem durumu ile tanımlanır ve sıklıkla diferansiyel denklemlerin bir kümesi ile tanımlanır.

1. dereceden başlangıç değer problemi ile başlayacağız;

$$\frac{dx}{dt} = F(t, x)$$

$$X(t_0) = X_0$$

$X(t) = [x_1(t), x_2(t), \ldots, x_n(t)]$ → **Sistem durum vektörü**

$X(0) = [x_1(0), x_2(0), \ldots, x_n(0)]$ → **ilgili başlangıç durumları**

Örnek: Aşağıdaki denklemler ile gösterilen bir sistem düşünelim;

$$\alpha'' + 2\beta'\alpha + \beta^2\alpha = \cos t$$

$$\beta' + \alpha\beta = 4$$

başlangıç durumlarına bağlı;

$$\alpha(0) = 2$$

$$\alpha'(0) = -1$$

$$\beta(0) = 1$$

NOT: İki dinamik durum değişkenleri ve birinci, ikinci dereceden diferansiyel denklemler olduğu için **üçüncü dereceden bir sistemdir.** Bu yüzden üç tane birinci dereceden bir diferansiyel denklem sistemi olarak tekrar tanımlamak mümkündür.

$$X_1 = \alpha(t), \quad X_2 = \alpha'(t) \quad X_3 = \beta(t)$$

$$\left.\begin{aligned} & x_2' + 2x_3'x_1 + x_3^2 x_1 = \cos t \\ & x_3' + x_1 \cdot x_3 = 4 \\ & x_1' = x_2 \\ & x_1(0) = 2 \\ & x_2(0) = -1 \\ & x_3(0) = 1 \end{aligned}\right\} \quad \begin{aligned} & x_2' = \cos t - 2x_3'x_1 + x_3^2 x_1 \\ & x_3' = 4 - x_1 x_3 \\ & x_1' = x_2 \\ & x_1(0) = 2 \\ & x_2(0) = -1. \\ & x_3(0) = 1 \end{aligned}$$

$$F = [x_2, \cos t - 2x_3'x_1 + x_3^2 x_1]$$

$$x(0) = [2, -1, 1]$$

$$t_0 = 0$$

EULER YÖNTEMİ

$$\frac{dx}{dt} = F(t,x)$$

Türev tanımını kullanacak olursak;

$$\lim_{h \to 0} \frac{x(t+h) - x(t)}{h} = F(t,x)$$

$$x(t+h) = x(t) + h.F(t,x)$$

$t_0 \leq t \leq t_n$ ve $k = 0,1,2,\dots,n$ için;

* $t \equiv t_k = h.k + t_0$ $\longrightarrow$ sürekli zamanlı t yerine yeni bir ayrık k değişkeni tanımlamak uygundur.

$$t_k = h.(k+1) + t_0$$

$$x(h(k+1) + t_0) = [x(hk+t_0) + h.F(hk+t_0,\ x(hk+t_0))]$$

değişkenini de şu şekilde gösterebiliriz.

* Yeni bir ayrık $x(k)$

$$\boxed{x(k+1) = [x(k) + h.F(t(k),\ x(k))] \qquad k = 0,1,2,\dots,n}$$

* Zaman değişkeni t ise; sinyal sürekli ya da analog olarak alınmıştır ve durum $x(t)$ dir.
* Zaman ayrık (kesikli) ise; durum değişkeni $x(k)$'dır. Sürekli zamanın ayrık zamanla yerdeğiştirdiği bu işleme **ayrıklaştırma** denir.
* Denklemin sağ tarafında değişkenler k, sol tarafında ise $k+1$'dir. Bunu x değişkeninin güncelleştirmesi olarak ifade ederiz.

Örnek: $x' = x^2 t$ $\quad x(1) = 3$ şeklinde tanımlanmış bir sistemi düşünelim.
$t_0 = 1$
$x(t_0) = 3$

Gerçek Çözüm:

$$\frac{dx}{dt} = x^2 t \quad \text{ve} \quad x(1) = 3$$

$$\int \frac{dx}{x^2} = \int dt.t$$

$$\int_3^x \frac{dx}{x^2} = \int_1^t t.dt$$

$$-\frac{1}{x} + \frac{1}{3} = \frac{t^2}{2} - \frac{1}{2}$$

$$x(t) = \frac{6}{5 - 3t^2} \quad //$$

Fakat, açık çözümün bilinmediğini farz edersek euler yöntemini çıkarabiliriz. Keyfi verilmiş $h = 0{,}05$ adım uzunluğu ile eşit ayrık sistem başlangıç koşulları ile karakterize edilmiştir.

Euler çözüm: $t_0 = 1$
$x(t_0) = 3$

$$t_{k+1} = t_k + h \Rightarrow t_{k+1} = t_k + \frac{1}{20}$$

$$x(k+1) = x(k) + h.F(t(k),\ x(k))$$

$$x(k+1) = x(k) + \frac{1}{20}.x^2(k).t_k, \qquad k = 1,2,\dots,n$$

NOT: t'nin eski değeri x güncellemesine ihtiyaç duyar fakat x t'nin güncellemesine ihtiyaç duymaz

Sözde Kod:

```
t=1
x=3
print t,x
   for k=1 to n
      x= x+ h x^2 t
      t = t+h
      print t, x
   next k
```

Matlab Kodu:

```
to=1;
xo=3;
x=[xo];
t = [to];
xg= [6/(5-3*to^2)];
h=0,05;
   for i=1:5
      xyeni= xo+ h* xo ^2 *to;
      to= to+h;
      xg= [xg  6/(5-3*to^2)];
      x= [x  xyeni];
      t= [t  to];
      xo= xyeni;
   end
   plot (t,x, 'r-d')
   hold on
   plot ( t, xg , 'k-s')
```

* $k=0$ için , $h=0{,}05$

$t_0=1$

$t_1=1{,}05$

$x(1)=3$

1.adım

$$x_g=\frac{6}{5-3t^2}=\frac{6}{5-3.1^2}=\frac{6}{2}=3$$

$$x_{euler}=x(k+1)=x(k)+\frac{1}{20}x^2(k).t_k$$

$x(1)=3$

$t_{k+1}=t_k+h \Rightarrow t_2=1{,}05+0{,}05=1{,}10$

2.adım:

$k=1$ için;

$t_{k+1}=t_k+h=t_2=1{,}05+0{,}05=1{,}10$

$$x_g=\frac{6}{5-3t^2}=\frac{6}{5-3.(1{,}05)^2}=\underline{\underline{3{,}55}}$$

$$x_{euler}=x(k+1)=x(k)+\frac{1}{20}x^2(k)\,t_k$$

$$x(2)=x(1)+\frac{1}{20}.x^2(1)\,t_1$$

$$x(2)=3+\frac{9}{20}.1=\underline{\underline{3{,}45}}$$

3.adım: $k=2$ için;

$t_{k+1}=t_k+h=1{,}05+0{,}05=1{,}10$

$$x_g=\frac{6}{5-3t^2}=\frac{6}{5-3.(1{,}10)^2}=4{,}38$$

$$x_{euler}=x(k+1)=x(k)+\frac{1}{20}x^2(k)\,t_k$$

$$=x(3)=3{,}45+\frac{1}{20}.(3{,}45)^2.$$

Sayfa 14

k	0	1	...	6
t_k	1	1,05		1,30
$x(t)$	3	3,55		-85,71
$x(k)$	3	3,45		13,83

* Yaklaşık çözüm tam olarak $t_0=1$'de başlamasına rağmen sıralı adım gerçek çözümden daha fazla uzaklaşır. Bu yüzden euler yönteminin uygulanması başlangıç noktasından daha çok sapmaması için önemlidir, aynı zamanda h adım uzunluğunun hassas seçilmesi gereklidir.

* Doğru çözüm elde etmenin yolu h adım uzunluğunun azaltılmasıdır.
h değerinin azaltılması iki büyük engele sahiptir.

1) Verilen bir noktadaki çözümü tahmin etmek için daha fazla hesaplama gerekecektir.

2) Veri gösteriminde bir sonraki makine sınırlamalarından dolayı, h çok fazla küçük olabilir.

* Her hesaplamanın sonucunda bir prosedürün sonuçlarının çıktısını almak yerine hepsinin, periyodik sonuç çıktılarının alınmasını sağlayan **kontrol break** işlemi uygulanır. Bir kontrol break iç içe döngüler yoluyla çalışır.

```
t=1
x=3
print t,x
  for i=1 to n
    for j=1 to m
      x= x + h.x^2.t
      t=t+h
      next j
    print t,x
  next i
```

Not: Dış döngü i (i=1,2,...n) indeksi kullanarak kontrol eder, iç döngü j (j=1,2,...m) indeksi kullanır.

TAYLOR YÖNTEMİ

* Euler yöntemi, Taylor teoreminin özel bir durumu olarak düşünülebilir.

$$X(t+h) = \sum_{i=0}^{\infty} \frac{x^{\wedge} i(t)}{i!} h^{\wedge} i$$

$$x' = F(t,x)$$

Euler güncelleştirme formülü;

$$X(t+h) = X(t) + h.F(t,x)$$

bu formül daha yüksek mertebeli yaklaşım kullanılarak genişletilebilir. Mesela ikinci dereceden yaklaşım şu formülü verir;

$$\boxed{X(t+h) = X(t) + h.X'(t) + \frac{1}{2}h^2 x''(t)} \rightarrow \text{taylor yöntemi}$$

euler yöntemi

$$* \quad X'' = \frac{d}{dF} F(t,x) = \frac{\partial F}{\partial t} + \frac{\partial F}{\partial x}\frac{dx}{dt} = \frac{\partial F}{\partial t} + \frac{\partial F}{\partial x} F(t,x)$$

$$X(t+h) = X(t) + h.x'(t) + \frac{1}{2}h^2\left(\frac{\partial F}{\partial t} + \frac{\partial F}{\partial x}F(t,x)\right)$$

Bu formülü aşağıdaki şekilde Euler gibi ayrıklaştırabiliriz;

$$t_{k+1} = t_k + h$$

$$X(t+h) = X(t) + h.F + \frac{1}{2}h^2\left(\frac{\partial F}{\partial t} + \frac{\partial F}{\partial x}F\right)$$

Örnek: $X' = X^2 t$ taylor tekniğini uygulayınız. $(h=0{,}05)$

$$\frac{\partial F}{\partial t} = x^2 \qquad \frac{\partial F}{\partial x} = 2xt$$

$$X(k+1) = X(k) + h.X^2(k)\,t_k + \frac{1}{2}h^2\left(x^2(k) + (2X(k)\,t_k).x^2(k).t_k\right]$$

$$X(k+1) = X(k) + h.x^2(k)\,t_k + h^2 x^2(k)\left(\frac{1}{2} + X(k)t_k^2\right)$$

k=0 için:

$t_0 = 1$

$X(t) = 3$

$X(k) = 3$

k=1 için;

$t_{k+1} = t_k + h$

$t_1 = t_0 + h$

$t_1 = 1 + 0{,}5 = 1{,}05$

$$X_g = \frac{6}{5-3t^2} = \frac{6}{5-3.(1{,}05)^2} = 3{,}55$$

$$X(k+1) = X(k) + \frac{1}{20}.X'(k) + \frac{1}{2}\left(\frac{1}{20}\right)^2 \left(x^2 + 2xt.x^2t\right)$$

$$= 3 + \frac{1}{20}.3^2.1 + \frac{1}{2}.\frac{1}{400}\left(9 + 6.9\right) = 3{,}53$$

k=2 için:

$t_{k+1} = t_k + h$

$t_3 = t_2 + h$

$t_3 = 1,1$

$X_g = \dfrac{6}{5-3t^2} = 4,38$

$$X(k+1) = X(k) + h.X'(k) + \frac{1}{2}h^2\left(\frac{\partial F}{\partial t} + \frac{\partial F}{\partial x}F(t,x)\right)$$

$$= 3,53 + \frac{1}{20}\cdot(3,53)^2 \cdot 1,05 + \frac{1}{2}\left(\frac{1}{20}\right)^2\left((3,53)^2 + 2\times 3,53 \cdot 1,05 \cdot (3,53)^2 \cdot 1,05\right)\right)$$

$$= 4,30$$

K	0	1	2	3	4	5	6
t_k	1.00	1.05	1,10	1,15	1,20	1,25	1,30
X(t)	3.00	3,55	4,38	5,81	8,82	19,20	-85,71
X(k)	3.00	3,53	4,32	5,61	8,05	13,89	36,66

* Taylor, Euler yöntemine göre daha doğru sonuçlar elde eder.

RUNGE-KUTTA YÖNTEMİ

* Daha önceki örneklerde birinci dereceden denklemlere Euler, ikinci dereceden denklemlere Taylor teknikleri uygulanıp karşılaştırıldı.

* Dördüncü dereceden bir yaklaşım sıklıkla şu şekilde kullanılır.

$$X(t+h) = X(t) + h.X'(t) + \frac{1}{2}h^2 X''(t) + \frac{1}{6}h^3 X^{(3)}(t) + \frac{1}{24}h^4 X^{(4)}(t)$$

→ Bununla birlikte bu formülü uygulamak için, Taylor için yapıldığı gibi ilk önce F'i birkaç kez ayırmak gereklidir. F için bir analitik formül kullanılamıyor olabileceği için çoğu zaman bu imkansızdır ve en iyi ihtimalle sıkıcı olabilir.

$K_1 = F(t_k, X(k))$,

$K_2 = F(t_k + \frac{1}{2}h,\ X(k) + \frac{1}{2}hK_1)$,

$K_3 = F(t_k + \frac{1}{2}h,\ X(k) + \frac{1}{2}hK_2)$,

$K_4 = F(t_k + h,\ X(k) + hK_3)$,

$t_{k+1} = t_k + h$,

$X(k+1) = X(k) + \frac{1}{6}h(K_1 + 2K_2 + 2K_3 + K_4)$

Bu klasik bir runge-kutta algoritmasıdır. İki avantajı vardır:

1) Doğru çözümdür. (4. dereceden Taylor'a eşdeğerdir.)

2) Türev hesaplaması gerektirmediğinden kullanımı kolaydır.

Örnek: $x' = x^2 t$ runge-kutta yöntemini uygulayınız. $(h=0{,}05)$

$x(1)=3$

Çözüm: $k=0$ için;

$t_0 = 1$

$X_g = 3$

$X(k) = 3$

$t_{k+1} = t_k + h$

$\underline{t_1 = 1{,}05}$

$k=1$ için;

$$X_g = \frac{6}{5-3.(1{,}05)^2} = \underline{3{,}55}$$

$K_1 = F(1,\ 3) \rightarrow 9$

$K_2 = F(1{,}025,\ 3{,}225) \rightarrow 10{,}66$

$K_3 = F(t_k + \frac{1}{2}h,\ X(k) + \frac{1}{2}h\,K_2)$

$F(1{,}025,\ 3{,}2665) \rightarrow 10{,}93$

$K_4 = F(t_k + h,\ X(k) + h.K_3)$

$F(1{,}05,\ 3{,}5465) \rightarrow 13{,}20$

$$X(k+1) = X(k) + \frac{1}{6}h\,(K_1 + 2K_2 + 2K_3 + K_4)$$

$$= 3 + \frac{1}{6}.0{,}05\ (9 + 2.(10{,}66) + 2.(10{,}93) + 13{,}20)$$

$$= \underline{3{,}544}$$

Sözde Kodu:

```
input t, x
print t, x
 for k=1 to n
    K1 = t x^2
    K2 = (t + 1/2 h) (x + 1/2 h K1)^2
    K3 = (t + 1/2 h) (x + 1/2 h K2)^2
    K4 = (t+h) (x + h K3)^2
   x = x + 1/6 h (K1 + 2K2 + 2K3 + K4)
   t = t+h
   print t, x
 next
```

Matlab Kodu:

```
to=1;
xo=3;
x=[xo];
t=[to];
xg=[6/(5-3*to^2)];
h=0,05;
  for i=1:5
     K1= to* xo^2;
     K2= (to+1/2*h)(xo+1/2*h*K1)^2
     K3= (to+1/2*h)(xo+1/2*h*K2)^2
     K4= (to+h)(xo+h*K3)^2
xyeni= xo+1/6*h(K1+2*K2+2*K3+K4)
 to=to+h;
xg=[xg 6/(5-3*to^2)];
x=[x xyeni];
t=[t to];
xo=xyeni;
end
plot(t,x,'r-d')        plot(t,xg,'k-s');
hold on
```

NOT: Runge-kutta yöntemi ; euler, taylor yöntemlerine göre daha iyi sonuç verir. Çünkü gerçek çözüme yakın sonuç runge-kutta ile elde edilir.

ADAPTİF RUNGE-KUTTA YÖNTEMİ

* Daha popüler heuristiklerden (çözümlerden) biri dördüncü ve beşinci dereceden farklı mertebelerin kullanımı ile bir çözüm hesaplamaktır.
* Eğer sonuçlar benzer ise h değeri büyütülür. $(3 \times h)$
* Eğer sonuçlar benzer değilse h değeri küçültülür. $(h/10)$

Sözde Kodu:

```
t = to;
X = Xo;
print t,x
  for k=1 to n
      4.dereceden Runge-Kutta ile X'i bul.
      5.dereceden Runge-kutta ile Y'yi bul
  if |x-y| < ε then
      x=y
      t=t+h
      h=3h
  else
      h=h/10
      go to [1]
  end if
      print t,x
   next k
```

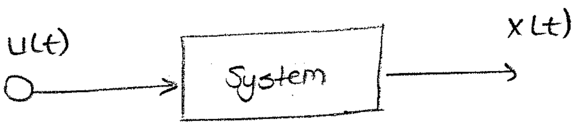
YÜKSEK DERECEDEN SİSTEMLER

* Herhangi bir diferansiyel denklemler toplamını birinci dereceden denklemler kümesine eşdeğer hale getirmek mümkündür.

$$x' = f(t, x, y) \qquad X(t_0) = X_0$$
$$y' = G(t, x, y) \qquad Y(t_0) = y_0$$

Daha önce anlatıldığı gibi x değişkenini K_1, K_2, K_3 ve K_4 değerleriyle, y değişkenini $\lambda_1, \lambda_2, \lambda_3$ ve λ_4 ile ilişkilendirerek bu sisteme Runge-Kutta yöntemini uygulamak basittir.

ÖRNEK:



Yandaki blok diyagramı ile gösterilen sistemi düşünelim.

Euler yöntemini uygulayınız.

$x'' + 3xx' = u(t)$

$u(t) = t \qquad t \geq 0$

$X(0) = 2,$

$x'(0) = 1;$

Çözüm:

$Y(t) = x'(t)$

$Y' + 3XY = t \Rightarrow Y' = t - 3XY$

$X(0) = 2$ $\qquad X' = Y$

$Y(0) = 1$ $\qquad X(0) = 2$

$\qquad Y(0) = 1$

Euler uygularsak;

$$t_{k+1} = t_k + h$$

$$X(k+1) = X(k) + h \cdot X'(k) = X(k) + h \cdot Y(k)$$

$$Y(k+1) = Y(k) + h \cdot Y'(k) = Y(k) + h \cdot [t_k - 3X(k)Y(k)]$$

Sözde Kodu:

```
t=0
x=2
Y=1
print  t,x,Y
for  k=1 to n
     x1= x+h.Y
     Y1= Y+h.(t-3xy)
     x=x1
     Y=Y1
     t=t+h
     print t,x,Y
   next k
```

OTONOM DİNAMİK SİSTEMLER

* Simulasyon açısından bakıldığında çoğu sistemler dinamiktir. Bu sistemler basit statik yapılardan çok, zaman ile değişirler.
* Sistemler ya endojen (iç) veya eksojen (dış) girişler tarafından sürülebilir.
* Sistem zaman odaklı ise bu girişler eş zamanlıdır, olay tabanlı ise eş zamanlı değildir.

Örn: $x'' + 2\tau w x' + w^2 x = r(t)$ τ, w sabitlerine sahiptir.

$r(t)$ girişine, $x(t)$ çıkışına ve τ, w sabitlerine sahiptir. $r(t)$ girişine, $x(t)$ çıkışına ve τ, w sabitlerine sahiptir.

* $r(t) = 4 - t$ gibi bir deterministik zaman fonksiyonu ise sistem **zaman odaklıdır.**

* Eğer $r(t)$ aşağıdaki şekilde rastgele bir işlem gibi stokastik olarak tanımlanırsa model **olay odaklıdır.**

$$r(t) = \begin{cases} 1 , & \text{olay oluşur} \\ 0 , & \text{olay oluşmaz} \end{cases}$$

* Bir sistemin hiçbir girişe sahip olmaması mümkündür. Böyle sistemler dış etkilerden bağımsız olduğu için **otonom** olarak adlandırılır.
* Buna göre otonom sistemin çözümü, sistemin doğal tepkisi olarak adlandırılır.
* Sistem lineer yani doğrusal ise; üç tip doğal tepkiden bahsedebiliriz.

1) Kararlı: Kısa bir geçiş evresinden sonra çıktı sıfıra doğru yaklaşır.

2) Kararsız: Sınırlama olmaksızın doğal tepki artar.

3) Marjinal: Tepki periyodik ve sınırlıdır.

* Belirli bir türün popülasyon dinamiğini modelleyen bir sistem düşünün.

→ Bir popülasyonu kontrol eden çok sayıda faktör olmasına rağmen, bir popülasyon büyüklüğünün artma oranı, herhangi bir zamandaki popülasyon büyüklüğü ile kabaca orantılıdır.

→ $X(t)$, t zamanındaki popülasyon büyüklüğünü ve λ sabit orantıyı gösterir.

$x' = \lambda . x$ $t=0$ 'da başlangıç büyüklüğü X_0

$t>=0$ $\underline{x(t) = X_0 . e^{\lambda t}}$ 'dir.

* Açıktır ki bu popülasyon zaman içinde katlanarak artar. Bu model **Malthusian modeli** olarak adlandırılır.

X_m → Sistemin desteklediği maksimum popülasyon büyüklüğü

$\left(\frac{X(t)}{X_m}\right)$ → Sistemin doluluk oranı

$\cancel{1 - \frac{X(t)}{X_m}}$ →

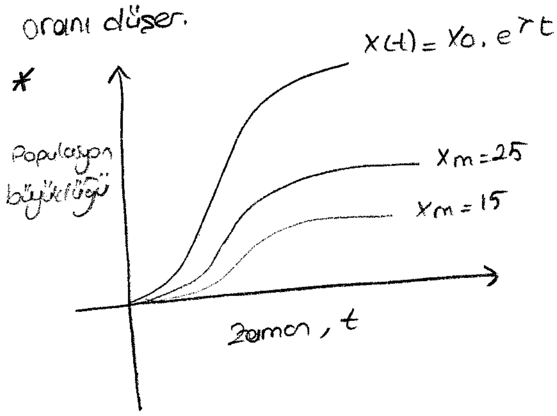
$1 - \frac{X(t)}{X_m}$ → Büyüme için geriye kalan kullanılabilir sistem oranıdır.

$$X' = \lambda . X \left(1 - \frac{X}{X_m}\right)$$ → **lojistik denklem** (doğrusal değildir.)

* $X(t)$ küçükse; $1 - \frac{X(t)}{X_m}$ olduğu için; $1 - \frac{X(t)}{X_m} \approx 1$ olduğu için denklem **Malthusian'dır.**

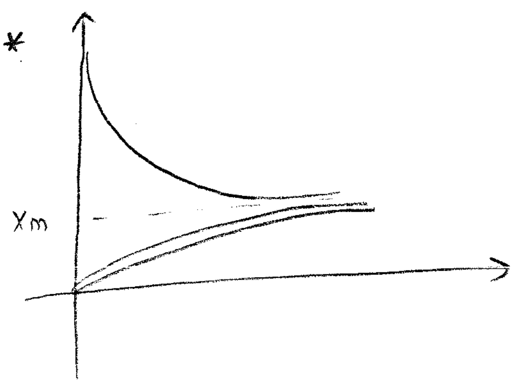
* Popülasyon büyüklüğü kapasiteye yakın olduğunda; $1 - \frac{X(t)}{X_m} \approx 0$ dır ve büyüme oranı düşer.

*



* Farklı başlangıç büyüklükleri ve $X(m) = 25$ için $X(t)$ değerini gösterir. X_0, X_m'den daha küçük ise $X(t)$ büyümesi asimtotik olarak X_m taşıma kapasitesine yaklaşırken, X_0, X_m'den daha büyük ise eğri yukarıdan X_m'e yaklaşır.

*



* Farklı X_m değerleri için $X_0 = 1$'de sabitli X_0 için $X(t)$ değerini gösterir. Başlangıç değerine bakılmaksızın $X(t)$, X_m taşıma kapasitesine her durumda yaklaşır.

* Lojistik model sınırlı yiyecek kaynağının ve bir popülasyonun olduğu ortamlarda iyi performans sergiler.

* Bununla birlikte, yiyerek ve yenilmeyerek hayatta kalan av ve bir avcının var olduğu ortam daha tipik durumlardan biridir.

→ $X(t)$ ve $Y(t)$ sırasıyla av ve avcı popülasyon büyüklüğünü göstersin

→ Av-avcı etkileşim sayısı $X(t).Y(t)$ çarpımı ile orantılıdır.

Av-avcı popülasyonunun büyüme oranı;

$$\left.\begin{aligned} X' &= \alpha_1 . X \left(1 - \frac{Y}{\beta_1}\right) \\ Y' &= \alpha_2 . Y \left(-1 + \frac{X}{\beta_2}\right) \end{aligned}\right\}$$

$\alpha_1, \alpha_2, \beta_1, \beta_2 \longrightarrow$ pozitif kapasite sabitleri

→ **Lotka-Volterra denklemi**

* Bu denklem doğrusal değildir ve bilinen analitik bir çözümü yoktur. Bu yüzden daha önce anlattığımız sayısal teknikler kullanılır.

* Euler uygularsak; iki denklem çifti olduğu için geçici değişkenler kullanılır, değişkenler hesaplandıktan sonra güncellenir.

~~Hatlatsak~~

Sözde Kodu:

```
read  t, X, Y
print t, X, Y
  for k=1 to n
      X1 = X(1 + α1.h - α1 h Y / β1)
      Y1 = Y(1 - α2 h + α2 h X / β2)

      X = X1
      Y = Y1
      t = t+h
      print t, X, Y
  next k
```

Sonuçların $[t_0, t_n]$ aralığında ve h aralığı olduğunu düşünürsek; $n = (t_n - t_0)/h$ oldukça büyük olur.

Örneğin; $t_0 = 0$ $t_n = 10$ $h = 0.001$

$n = 10\ 000$ örnek gerekir.

Muhtemelen sadece 50 tanesi yeterli olacaktır.

* Bu problem kontrol break ile çözülebilir. Böylece $[t_0, t_n]$ aralığında $m \times n$ hesaplama yapılır.

$m = (t_n - t_0)/n.h$

Sözde Kodu:

```
read t, x, y
print t, x, y
for i=1 to n
    for j=1 to m
        X1 = X(1 - α1 h + α1 h X / β1)
        Y1 = Y(1 - α2 h + α2 h X / β2)
        X = X1
        Y = Y1
        t = t+h
    next j
    print t, x, y
next i
```

Örnek:

$$X' = 3x\left(1 - \frac{1}{10}Y\right)$$

$$Y' = 1{,}2\,Y\left(-1 + \frac{1}{25}x\right)$$

$$X(0) = 10$$

$$Y(0) = 5$$

Lotka-Voltera sisteminin $[0,5]$ aralığında sayısal çözümünü veriniz. ($h = 0{,}001$)

Çözüm: Grafik 5 birim zamana kadar olacak herbir aralıkta 10 örnek yeterlidir. Dolayısıyla $5 \times 10 = 50$ örnek yeterlidir.

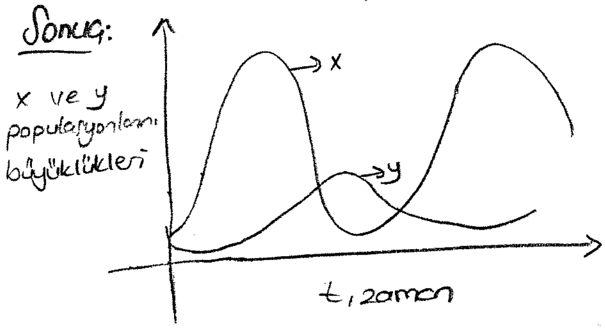
$m = (5-0) / 50 \times 0.001 = \underline{100}$

Toplamda $\rightarrow$ $50 \times 100 = 5000$ güncelleme olur.

Matlab Kodu:

```
function [t,x,y] = lotka_voltera (h, xo, yo, n)
x = [xo];
y = [yo];
t=0;
  for i=1:n
     for j=1:100
        xx= xo * (1+3*h - 3*h * yo/10);
        yy= yo * (1- 1.2*h + 1.2* xo/25);
        xo = xx;
        yo = yy;
        x = [x ; xx];
        y = [y ; yy];
        t = t+h;
     end
  end
  plot(x)
  hold on
  plot (y, 'r-')
```

Sonuç:



* Avcının yiyecek kaynağı arttıkça avcı popülasyonu artmaktadır. Fakat avın harcanmasıyla avcı popülasyonu düşer. Dolayısıyla bir popülasyon bol bir besin kaynağına ulaştığında kendi basarısının kurbanı olur.

ÖRNEKLER

1) $x' = x+t$ $x(0) = 2$ euler yöntemiyle çözünüz. ($h=0.5$)

$x(k+1) = x(k) + h.\ F(t,x)$

$x(1) = x(0) + 0.5\ .(x+t)$

$x(1) = x(0) + 0.5\ (2+0)$

$x(1) = 2 + 0.5 . 2 = 3$

$t_{k+1} = t_k + h$

$t_1 = 0 + 0.5 \Rightarrow t_1 = 0.5$

$x(2) = x(1) + h.\ (x+t)$

$x(2) = 3 + 0.5(3+0.5)$

$x(2) = 4.75$

2) $x' = x+t$ $x(0) = 2$ taylor yöntemini uygulayınız. ($h=0.5$)

$$x(k+1) = x(k) + h.\ F(t,x) + \frac{1}{2}h^2\left(\frac{\partial F}{\partial t} + \frac{\partial F}{\partial x} F(t,x)\right)$$

$$x(1) = x(0) + 0.5.(x+t) + \frac{1}{2}.(0.5)^2\left((1+x) + (1+t).(x+t)\right)$$

$$= 2 + 0.5(2+0) + \frac{1}{2}(0.5)^2\left((1+2) + 1.2\right)$$

$$= 2 + 0.5 . 2 + \frac{1}{2}.(0.5)^2 . 5 = 3.625$$

$t_1 = 0 + 0.5 \Rightarrow t_1 = 0.5$

$$x(2) = 3.625 + 0.5(3.625 + 0.5) + \frac{1}{2}.(0.5)^2 \underbrace{\left((1+3.625) + (1+0.5).(3.625+0.5)\right)}_{1.351}$$

$= 7.03$

3) $x' = x+t$ $x(0)=2$ runge-kutta yöntemi uygulayınız. ($h=0.5$)

$K_1 = F(t_k, x_k) = 2$

$K_2 = F(t_k + \frac{1}{2}h,\ x(k) + \frac{1}{2}hK_1) \Rightarrow F(0.25,\ 2.5) = 2.75$

$K_3 = F(t_k + \frac{1}{2}h,\ x(k) + \frac{1}{2}hK_2) \Rightarrow F(0.25,\ 2.68) = 2.93$

$K_4 = F(t_k + h,\ x(k) + hK_3) \Rightarrow F(0.5,\ 3.465) = 3.96$

$x(k+1) = 2 + \frac{1}{6}.0.5\left(2 + 2.2.75 + 2.2.93 + 3.96\right) = \underline{\underline{3.44}}$

4) $\frac{dy}{dx} = -\frac{y}{x}$ $y(4) = 0.75$ $y(7) = ?$ euler yöntemi ile çözünüz. $(h=1)$

$y' = -\frac{y}{x}$ $y(4) = 0.75$

$y(5) = y(4) + 1.(-0.187) = 0.75 - 0.187 = \underline{\underline{0.563}}$

$y(6) = y(5) + 1.(-0.1126) = 0.563 - 0.1126 = \underline{0.45}$

5) $\frac{dx}{dt} = x^2 - 4t$ $X(0) = 0.5$ $(h = 0.01)$

$X(k+1) = X(k) + h.\ F(t, x)$

$X(1) = X(0) + 0.01(x^2(k) - 4t_k)$

$X(1) = 0.5 + 0.01(0.25 - 0)$

$= 0.5025$

$t_{k+1} = t_k + 0.01 \Rightarrow 0 + 0.01 = 0.01$

$X(2) = X(1) + 0.01(0.212)$

$= 0.\underline{\underline{5046}}$

Örnek:

a) Değişken değiştirme yöntemiyle 1. dereceden diferensiyel denkleme dönüştürünüz.
b) Euler yöntemini uygulayarak algoritmasını oluşturunuz.

$$\alpha'' + 2\beta\alpha' + \beta^2\alpha = \cos t$$

$$\beta' + \alpha\beta = 4$$

$$\alpha(0) = 2$$

$$\alpha'(0) = -1$$

$$\beta(0) = 1$$

Çözüm: a)

$$\left.\begin{aligned} &\alpha' = Y \\ &Y' + 2\beta Y + \beta^2\alpha = \cos t \\ &\beta' + \alpha\beta = 4 \\ &\alpha(0) = 2 \\ &Y(0) = -1 \\ &\beta(0) = 1 \end{aligned}\right\} \quad \begin{aligned} &\alpha' = Y \\ &Y' = \cos t - 2\beta Y + \beta^2\alpha \\ &\beta' = 4 - \alpha\beta \\ &\alpha(0) = 2 \\ &Y(0) = -1 \\ &\beta(0) = 1 \end{aligned}$$

b)

$$\alpha(k+1) = \alpha(k) + h \cdot Y(k)$$

$$Y(k+1) = Y(k) + h \cdot (\cos t_k - 2\beta(k)\, Y(k) + \beta^2(k)\, \alpha(k))$$

$$\beta(k+1) = \beta(k) + h\, (4 - \alpha(k) \cdot \beta(k))$$

Sözde kodu:

```
t=0
α=2
Y=-1
β=1
h=0.1
for k=1 to n
    α1= α + h.Y
    Y1= Y + h.(cost - 2βY + β²α)
    β1= β + h.(4-α.β)
    α=α1
    Y=Y1
    β=β1
    t=t+h
print t, α, Y, β
end
```

Örnek: $x'' + 2x' + 5y = 0$

$x' + 2y = y'$

$x(0) = 0$

$x'(0) = 0$

$y(0) = 1$

a) Değişken değiştirme yöntemiyle 1. dereceden bir diferansiyel denkleme dönüştürünüz.

b) Euler yöntemini kullanarak algoritmasını oluşturunuz.

c) Taylor yöntemini kullanarak algoritmasını oluşturunuz. $(h = 0.1)$

a) $\boxed{x' = z}$

$$\left.\begin{aligned} & z' + 2z + 5y = 0 \\ & z + 2y = y' \\ & x(0) = 0 \\ & z(0) = 0 \\ & y(0) = 1 \end{aligned}\right\} \quad \begin{aligned} & x' = z \\ & z' = -2z - 5y \\ & y' = z + 2y \\ & x(0) = 0 \\ & z(0) = 0 \\ & y(0) = 1 \end{aligned}$$

b) $x(k+1) = x(k) + h.z(k)$

$y(k+1) = y(k) + h.(z(k) + 2y(k))$

$z(k+1) = z(k) + h.(-2z(k) - 5y(k))$

Algoritması

```
t=0
x=0
y=1
z=0
h=0.1
for k=1 to n
    x1= x+ h.z
    y1= y+h.(z+2y)
    z1= z+ h.(-2z-5y)
    x=x1
    y=y1
    z=z1

  t=t+h
  print t,x,y,z

  end
```

taylor açılımı

c) $\boxed{x(k+1) = x(k) + h.F + \frac{1}{2}h^2\left[\frac{\partial F}{\partial t} + \frac{\partial F}{\partial x}.F\right]}$

$x(k+1) = x(k) + h.z(k) + \frac{1}{2}h^2(1)$

$y(k+1) = y(k) + h.(z+2y) + \frac{1}{2}h^2((1+2y) + (z+2).(z+2y))$

$z(k+1) = z(k) + h.(-2z-5y) + \frac{1}{2}h^2((-2z-5) + (-2-5y).(-2z-5y))$

Sistemin Performans Ölçütleri

1) Çevrim Zamanı
2) Doluluk Oranı
3) Bekleme Zamanı
4) Kalite
5) Maliyet

SİSTEMLER

KESİKLİ SİSTEM

* Sistem durum değişkenleri zaman içinde kesikli noktalarda değişir.

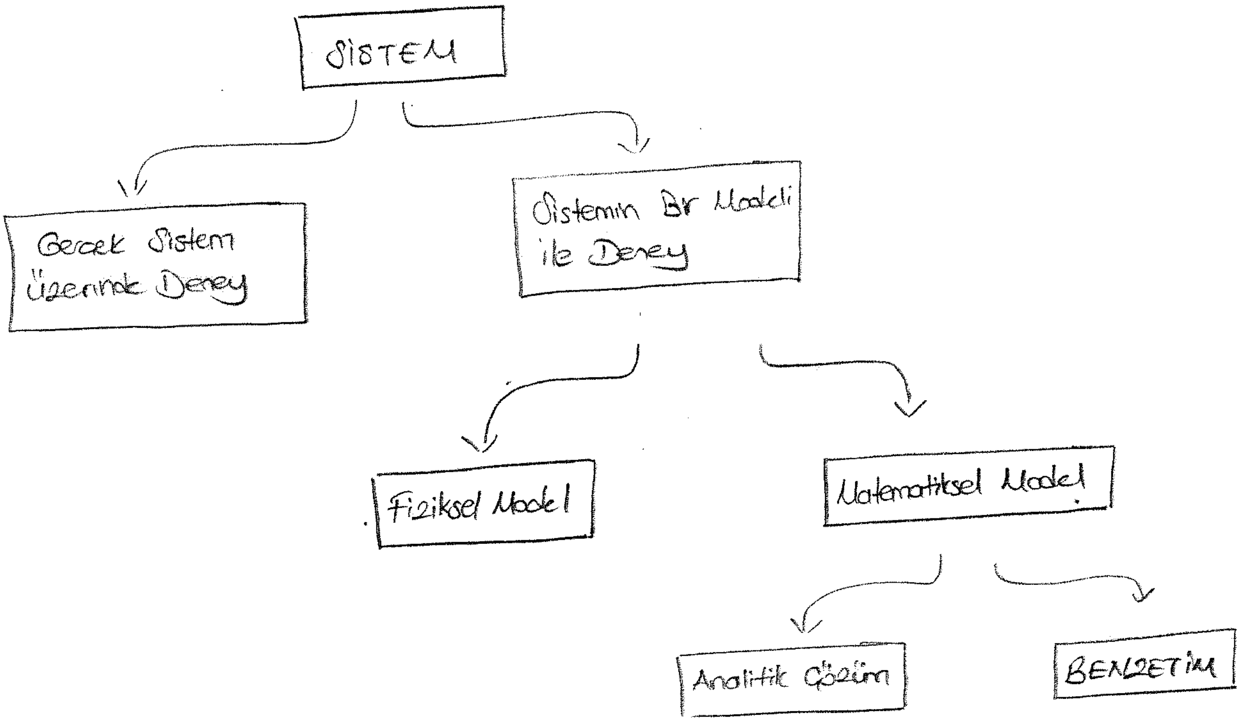
Ör: Banka kesikli sisteme bir örnektir. Çünkü müşteri sayısı yeni bir müşteri geldiğinde ya da müşteri servisi tamamladığında değişir.

SÜREKLİ SİSTEM

* Sistem durum değişkenleri zaman içinde sürekli olarak değişir.

Ör: Uçağın hareketi sürekli sisteme örnektir. Uçağın hızı ve pozisyonu sürekli olarak değişir.

SİSTEMLERİN ÇÖZÜMÜ



MODEL

* Bir sistemin gösterimi olarak tanımlanabilir.
* Bir model, gercek sistem hakkında gerekli sonucları cıkarmaya için verecek detaya sahip olmalıdır.

FIZIKSEL MODEL
→ Gercek bir sisteme benzer.

MATEMATIKSEL MODEL
→ Bir sistemi göstermek için sembolik notasyonlar ve matematiksel esitsizlikler kullanılır.

BENZETIM MODELLERI 3 ANA GRUPTA TOPLANABILIR

→ **Statik veya Dinamik** (Static - Dynamic)
→ **Belirli veya Olasılıklı** (Deterministic - Stochastic)
→ **Kesikli veya Sürekli** (Discrete - Continuous)

* STATIK BENZETIM MODELI

• Sistemin belirli bir anındaki gösterimidir.
• Monte Carlo benzetim modeli bu türe uygun modellerdir.

* DINAMIK BENZETIM MODELI

• Sistemin calısma zamanına göre yapılan modellemedir. (Belirli bir aralık veya tüm calısma zamanı)
Örnek: Bir banka için kurulan benzetim modeli 8 saatlik bir calısma zamanı dikkate alınarak calıstırılır.

(DETERMINISTIC)
* BELIRLI BENZETIM MODELI

• Rassal değişken icermeyen benzetim modelidir.
• Bu benzetim modelinde verilen bir girdi seti için bir cıktı seti vardır.

(STOCHASTIC)
* OLASILIKLI BENZETIM MODELI

• Bir veya daha fazla rassal değisken iceren benzetim modelleridir.
• Stochastic benzetim modeli kullanarak elde edilen cıktı rassal olup, modelin karakteristiklerinin tahminidir.
Örn: Banka örneğinde; varıslar arası zaman aralığı ve servis zamanı rassal değiskenlerdir.

ZAMAN DILIMLEME

→ Dinamik benzetimin temeli, sistem durum değiskenlerinin zaman boyunca modellenmesidir.
→ Benzetimde zaman akısını kontrol etmenin en basit yolu, esit zaman aralıklarında ilerlemektir. (zaman dilimleme)
→ Zaman dilimleme ne kadar genis olursa; ortaya cıkan durum değismelerinin basılarının modellemesini yapmak olanaksız olacağından, gercek sistemden daha kaba olacaktır.
→ Zaman dilimleme ne kadar küçük olursa; model gereksiz yere sıkca incelenir, bu da asırı bilgisayar calıstırmasına yol acar.

ÖRNEK:

İş Numarası	Yığın Büyüklüğü	Beklenen Sipariş Günü
1	200	1
2	400	8
3	100	14
4	200	18

Makine A: (Yığın büyüklüğü / 50 + 1) gün
Makine B: (Yığın büyüklüğü / 100 + 3) gün

Herbir iş önce makine A'da yığın olarak bitirildikten sonra, makine B'de yığın olarak başlar ve biter. Bu atölyede yukarıda verilen 4 siparişi kabul ederse son yığın ne zaman tamamlanır?

Çözüm: İş süreleri

İş Numarası	Makine A	Makine B
1	$(200/50)+1=\underline{5}$	$(200/100)+3=\underline{5}$
2	$(400/50)+1=\underline{9}$	$(400/100)+3=\underline{7}$
3	$(100/50)+1=\underline{3}$	$(100/100)+3=\underline{4}$
4	$(200/50)+1=\underline{5}$	$(200/100)+3=\underline{5}$

Gün	Kuyruktaki iş: Makine A	Kuyruktaki iş: Makine B	Çalıştırılan iş: Makine A	Çalıştırılan iş: Makine B
1	—	—	1	—
2	—	—	1	—
3	—	—	1	—
4	—	—	1	—
5	—	—	1	—
6	—	—	—	1
7	—	—	—	1
8	—	—	2	1
9	—	—	2	1
10	—	—	2	1
11	—	—	2	—
12	—	—	2	—
13	—	—	2	—
14	3	—	2	—
15	3	—	2	—
16	3	—	2	—
17	—	—	3	2
18	4	—	3	2
19	4	—	3	2
20	—	3	4	2
21	—	3	4	2
22	—	3	4	2
23	—	3	4	2
24	—	—	4	3
25	—	4	—	3
26	—	4	—	3

Gün	Kuyruktaki İş		Çalıştırılan İş	
	Makine A	Makine B	Makine A	Makine B
27	—	4	—	3
28	—	—	—	4
29	—	—	—	4
30	—	—	—	4
31	—	—	—	4
32	—	—		4

SONRAKİ OLAY TEKNİĞİ

* Bu yaklaşımda, yalnızca bir durum değişmesinin olacağı bilindiğinde yaklanır ve güncellenir.
* Bu durum değişmeleri genellikle olay olarak adlandırılır ve zaman olaydan olaya aktarıldığı için sonraki olay tekniği olarak adlandırılır.

Yukarıdaki örneği sonraki olay tekniği ile çözümlersek;

İş Numarası	Geliş Zamanı	Makine A Başlama	Makine A Bitiş	Makine B Başlama	Makine B Bitiş
1	1	1	5	6	10
2	8	8	16	17	23
3	14	17	19	24	27
4	18	20	24	28	32

Zaman Dilimleme mi Sonraki Olay Tekniği mi?

Sonraki olay tekniği zaman dilimlemeye göre iki avantaja sahiptir.

Zaman artımı yüksek ya da düşük faaliyet dönemlerini otomatik olarak ayarlar, böylece gereksiz ve yararsız modelin durumunun kontrollerinden kaçınmış olur.

Önemli olayların, benzetiminde ne zaman olduğunu açıklıkla ortaya koyar.

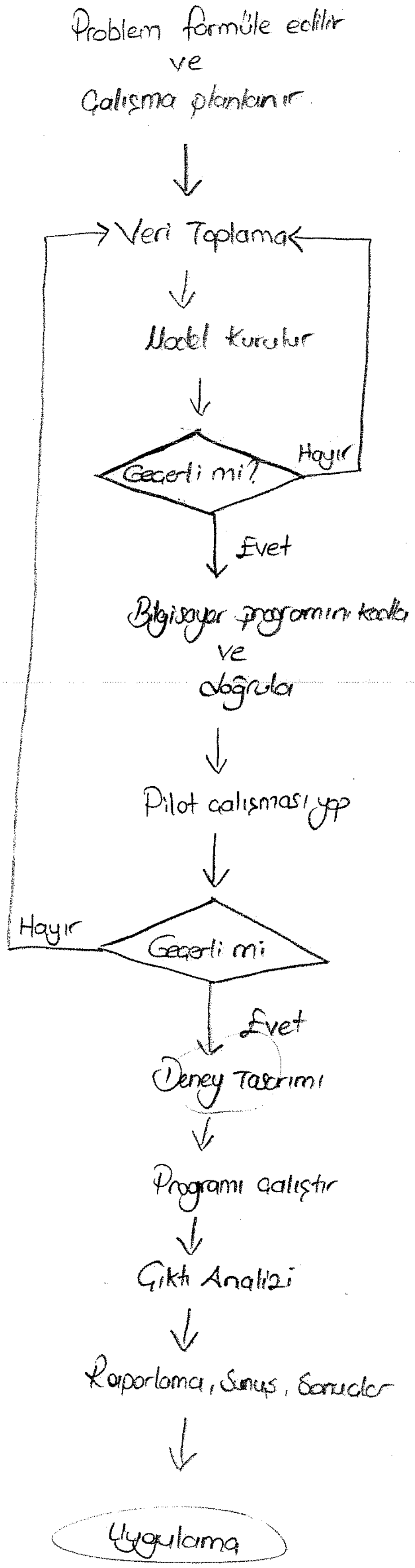
Stokastik Mi? Deterministik Mi?

→ Bir sistemin; eğer davranışı tümüyle tahmin edilebilirse deterministiktir.

→ Eğer bir sistemin davranışı tümüyle tahmin edilebilir değilse stokastiktir.

* Deterministik benzetim modeli hiçbir stokastik eleman içermez.
* Stokastik benzetim modellerinde olasılık dağılımları kullanılır.

BENZETİMİN AŞAMALARI



0 – 0.1 → 10

0.1 – 0.2 → 9

0.01 → 1

0.1 → 0.2 → 10

MONTE CARLO BENZETİM METODU

→ Monte Carlo benzetim modeli; olasılık teorisi üzerine kurulu bir sistemdir.

→ Monte Carlo metodunda; istatistiksel ve matematiksel tekniklerle bir deneyi ya da çözülmesi gereken fiziksel bir olayı tesadüfi sayıları defalarca kullanarak simule edip çözmek esastır.

→ Günümüzde bu metot, fizik ve matematik problemlerinin çözümünde, MCNP (Monte Carlo N Parçacık Taşınım) kodunu kullanarak nükleer transport hesaplamalarında iyi sonuçlar vermektedir.

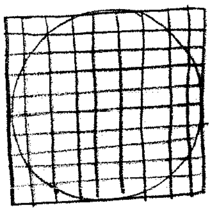
→ (0,1) aralığında düzgün, rassal sayılar kullanılarak, zaman faktörünün önemli olmadığı, stokastik ve deterministik problemlerin çözümünde kullanılan bir tekniktir.

→ Monte Carlo benzetimi genellikle statik benzetim modellerinde kullanılır.

Statik Monte Carlo Benzetimi Tanımı

* Monte Carlo yöntemi direkt analitik yaklaşımın mümkün olmadığı fonksiyonların integralinin sayısal elde edilmesinin bir yoludur.

Örnek: π sayısı bilinmeden dairenin alanı hesaplanmaya çalışılır.



→ Dairenin içinde yer alan karelerin sayılması bize π sayısının hesaplanmasına olanak verir.

→ Geniş kare içinde → n tane kare

→ Dairenin içinde → m tane kare

varsa dairenin alanı m/n ile karenin alanının çarpımı olacaktır.

Karenin Alanı $= 4r^2$

Dairenin Alanı $= \pi r^2$

$$\text{Dairenin Alanı} = \text{Karenin Alanı} \times \frac{m}{n}$$

$$\pi \cancel{r^2} = 4 \cancel{r^2} \times \frac{m}{n}$$

$$\boxed{\pi = 4 \cdot \frac{m}{n}}$$

Subjektif Olasılıklar: Bunlar olasılığı değişik şekillerde tanımlamayı sağlar.

→ Önsel Sav: Tüm sonuçlar hakkında bilgi sahibi olduğumuz durum. 6 yüzlü bir zarın herbir çıktısının olasılığı 1/6'dır.

→ Göreceli Sıklık Savı: Çıktıları üreten süreci anlamadığımız fakat onların göreceli sıklıklarını hesaplamak için yeterli veriye sahip olduğumuz durum.

→ [illegible]: Önsel + Göreceli

Örneğin; bozuk para atıldığında yazı gelme olasılığı 0.5, tura gelme olasılığı 0.5'tir.

Monte Carlo Benzetimi Ortalama Metodu

Örnek: $I = \int_a^b g(x)\,dx$ integralini çözmek istiyoruz.

(g(x), analitik çözümü olmayan bir fonksiyon olsun.)

Çözüm:
* Yeni bir rassal değişken olarak Y tanımlansın.

$$Y = (b-a) \cdot x \qquad a \leq x \leq b$$

* X, [a,b] aralığında düzgün dağılıma sahip sürekli bir rassal değişkendir.

$$f(x) = \frac{1}{b-a} \qquad a \leq x \leq b$$

$$E(Y) = E[(b-a) \cdot g(x)]$$

$$E(Y) = (b-a)\, E[g(x)]$$

$$E(Y) = (b-a) \int_a^b g(x) \cdot f(x)\,dx$$

$$E(Y) = (b-a) \int_a^b g(x) \cdot \frac{1}{(b-a)} \cdot dx$$

$$\boxed{E(Y) = \int_a^b g(x)\,dx}$$

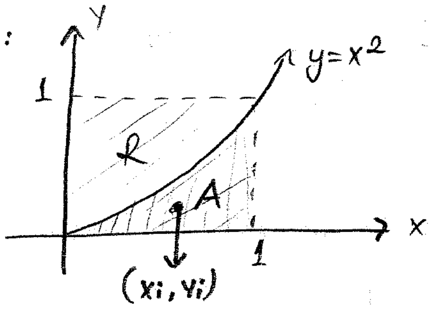
* Aranılan integralin değeri Y'nin beklenen değerine eşit çıktı. Buradan yararlanarak;

$I = \int_a^b g(x)\,dx$ 'in değerini Monte Carlo benzetimi ile bulabiliriz.

$$E(Y) = Y_{at} = \frac{\sum_{i=1}^{n} y_i}{n} = (b-a) \frac{\sum_{i=1}^{n} g(x_i)}{n}$$

Burada $X_1, X_2, X_3, \ldots, X_n \sim U(a,b)$ rassal değişkenlerdir.

ÖRNEK:



Şekilde görülen $y=x^2$ eğrisi ile x ekseni arasında kalan alanı bulmak için monte-carlo metodunu kullanınız.

Çözüm: 1) Eğer kare içerisinde rastgele noktalar (x_i, y_i) işaretleyip, bu noktaların eğrinin altında olup olmadıklarını belirler ve bunu toplam nokta sayısına oranlarsak, A alanının R karesine olan oranını yaklaşık olarak elde edebiliriz.

$$\Theta(x_i, y_i) = \frac{\text{Eğrinin altında olma oranı}}{\text{Karenin Alanı}} = \frac{\int_0^1 x^2 dx}{1.1} = \frac{\left.\frac{x^3}{3}\right|_0^1}{1} = \frac{1}{3}$$

2) Toplam nokta sayısı $= n$

Eğrinin altında kalan nokta sayısı $= m$

$\frac{m}{n} \rightarrow$ aranan noktanın eğrinin altında kalma olasılığı

3) İki denklemi eşitlersek; $\boxed{\frac{m}{n} = \frac{1}{3}}$

Matlab Kodu:

```
function egrialan = X kare(n)
    egri=0;
    sayac=0;
    for i=1:n
        x=rand;
        y=rand;
        if ( y < x * x)          → eğrinin altındaysa (A alanı içinde)
                                    egri=egri+1
            egri=egri+1;
        end
        sayac=sayac+1;           → değilse sayacı artır.
    end
    egrialan=egri/sayac;         → eğrinin altında kalma olasılığı
```

Örnek: 0 ile 100 arasında bulunan sayılar içinden rastgele seçilen bir sayının 11'e tam bölünebilme olasılığını monte-carlo yöntemi ile çözünüz.

Çözüm: * Analitik çözüm; 0 ile 100 arasında 11'e tam bölünebilen sayıları bul. Kaç tane ise onu tüm sayılara böl.

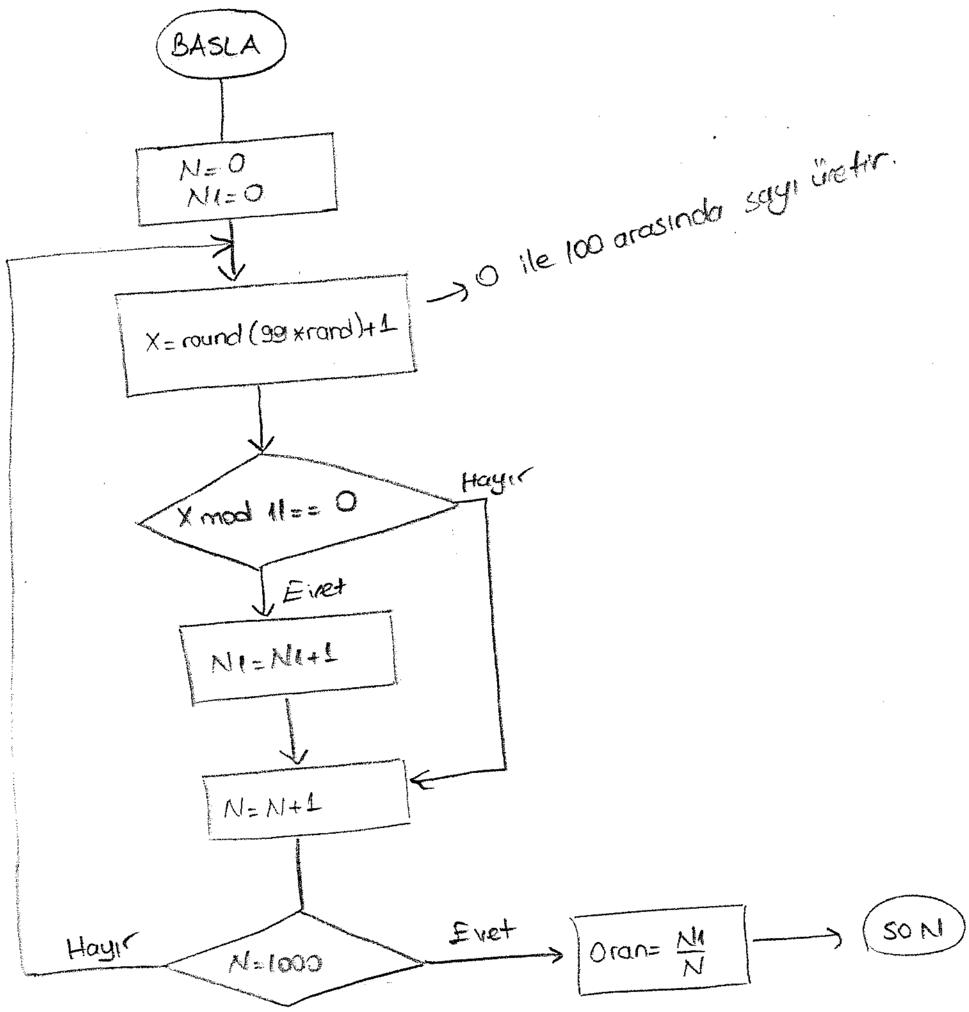
11, 22, 33, 44, 55, 66, 77, 88, 99 → 9 tane

0 ile 100 arasında → 100 tane sayı var. (0,100]

$$\text{Olasılık} = \frac{9}{100} = 0{,}09 \text{ 'dur.}$$

* Monte Carlo benzetimi ile çözüm

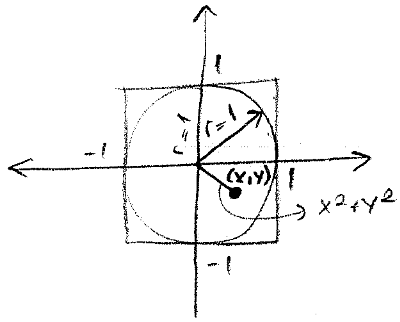
1) 0-100 arasında Xl adet sayı seçmemizi ister.
2) Bunların N1 tanesi 11'e bölünür.
3) Yani istenen olasılık $\frac{N1}{Xl}$ 'dir.
4) Programlama esnasında seçilecek her sayı için N değeri artırılır. 11'e bölünebilme kontrolü yapılır. Bölünebiliyorsa N1 değeri artırılır.



Matlab Kodu :

```
function sonuc = bolunebilme (n)
  bol=0;  % 11'e bölünebilen sayısı
  sayac=0; % toplam seçilen sayı sayısı
  for i=1:n
     x = round ( 99 * rand) + 1
     sonuc= mod (x, 11);
     if ( sonuc = = 0)
        bol= bol + 1
     end
       sayac = sayac+1
    end
    sonuc = bol /sayac;
```

Çemberde π sayısının Hesabı (Matlab Kodu)



$x^2+y^2<1$ olup olmadığı dağılığı = ?

Neden 1 ? Çünkü yarıçapı = 1, çemberin içinde olup olmadığını bulacağız.

Çemberin Alanı $= \pi r^2$

Karenin Alanı $= 2r \times 2r = 4r^2$

$$P(x^2+y^2<=1) = \frac{\text{Çemberin Alanı}}{\text{Karenin Alanı}} = \frac{\pi r^2}{4r} = \frac{\pi}{4}$$

$$\downarrow$$

$$\frac{m}{n}$$

$$\frac{m}{n} = \frac{\pi}{4} \Rightarrow \boxed{\pi = \frac{4m}{n}}$$

Karenin içine n adet nokta koyulur (100 000 – 1000 000)

Çemberin içinde m adet nokta vardır.

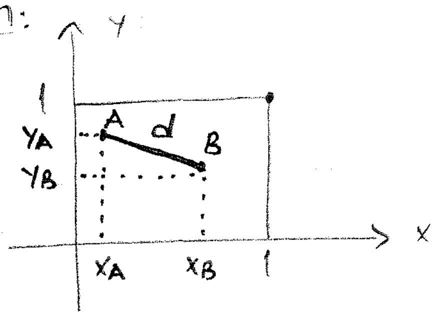
Matlab Kodu :

```
function pi= montecarlo (n)
  Cember = 0;  % x^2+y^2<=1 'olan yani çemberin içinde olan noktaların sayısı
  Sayac=0 ;%  Karenin içinde olan noktaların sayısı
  for i=1:n
     x=rand;
     y=rand;
     if((x^2 +y^2)< =1)
        cember =cember +1 ;
     end
      sayac =sayac +1 ;
    end
    pi= 4 * cember / sayac;
```

ÖRNEK: Kenarları 1 br uzunlukta olan bir kare düşününüz. Bu kare içinden rassal seçilen A ve B noktaları olsun. A ve B arası d uzunluğundadır. d'nin 0.8'den küçük olma olasılığı nedir?

Açıklama: Monte Carlo tekniği ile rassal olarak 1000 tane A ve B noktaları üreterek d'nin 0.8'den küçük olma olasılığını bulunuz. Kullanacağınız yaklaşımı açıklayarak akış şemasını çiziniz.

Çözüm:



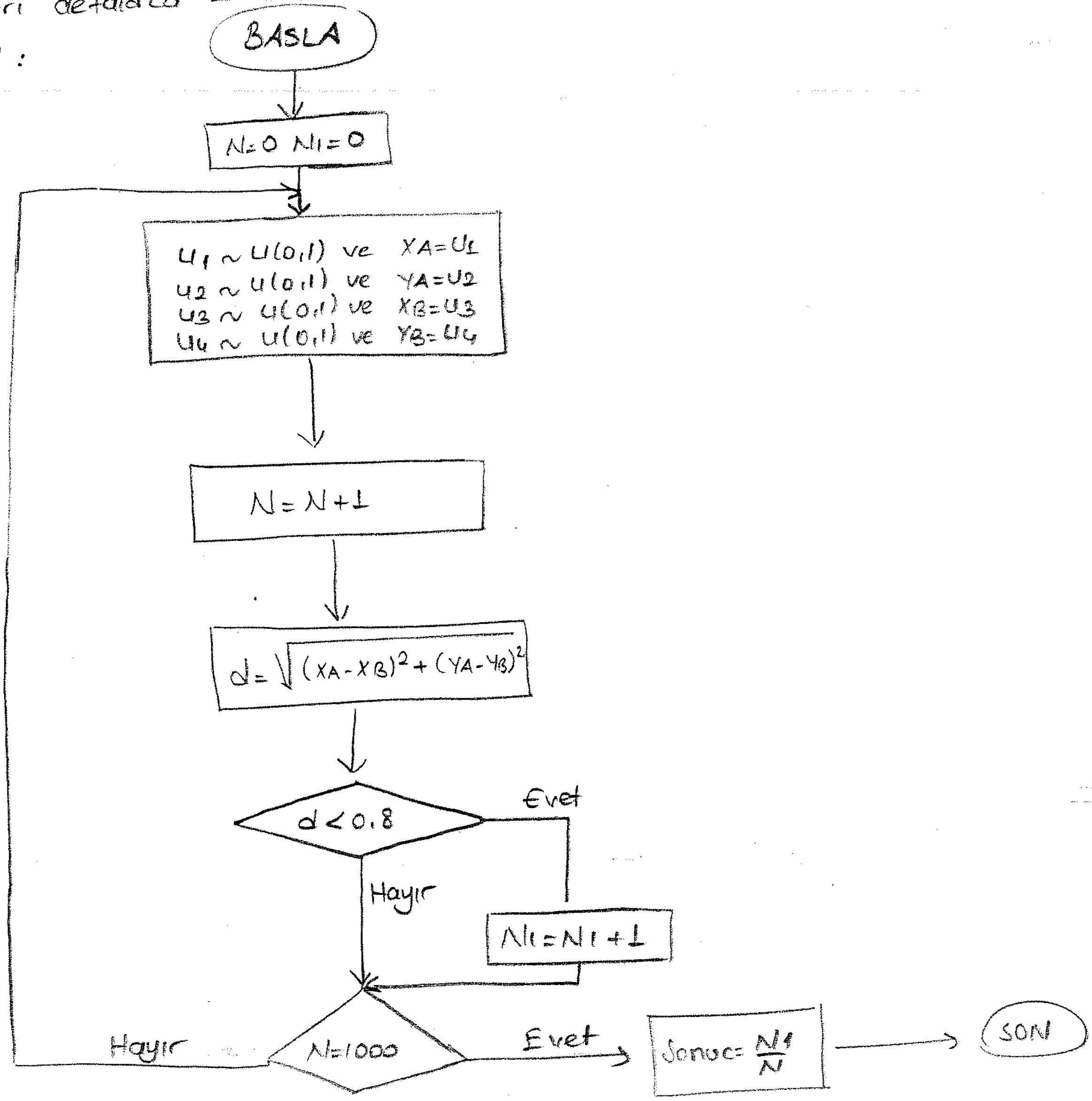
$A = (X_A, Y_A)$ $B = (X_B, Y_B)$ olsun.

Bu durumda;

$$d = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$$

* Kullanacağımız yaklaşım Monte Carlo benzetim modelidir. Monte Carlo benzetim modeli, olasılık teorisi üzerine kurulu bir sistemdir. Bu metotda istatistiksel ve matematiksel tekniklerle bir deneyi veya çözülmek istenen bir fiziksel olayı tesadüfi sayıları defalarca kullanarak simulasyon edip çözmek esastır.

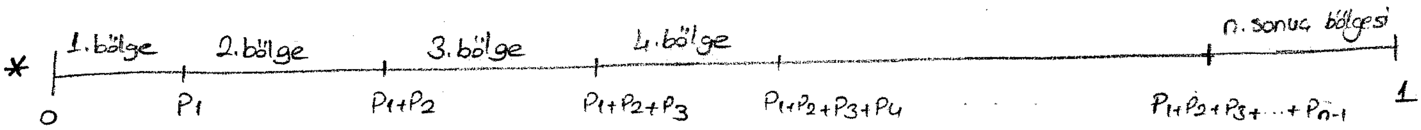
Akış Şeması:



Matlab Kodu:

```
function sonuc = hesapla(n)
    N1=0; % d<0.8 olanların sayısı
    N=0; % bütün d değerlerinin sayısı
  for i=1:n
      XA= rand;     YA= rand;
      XB= rand;     YB= rand;

      d= sqrt( (XA-XB)^2 + (YA-YB)^2) );
      if( d <0.8)
         N1 = N1+1
      end
         N= N+1
      end
      sonuc= N1/N ;
```



q sayısı

$0 < q < P_1 \rightarrow$ 1. sonuç

$P_1 < q \leq P_1 + P_2 \rightarrow$ 2. sonuç

$P_1 + P_2 + P_3 + \dots + P_{n-1} \leq q < 1 \rightarrow$ n. sonuç bölgesi

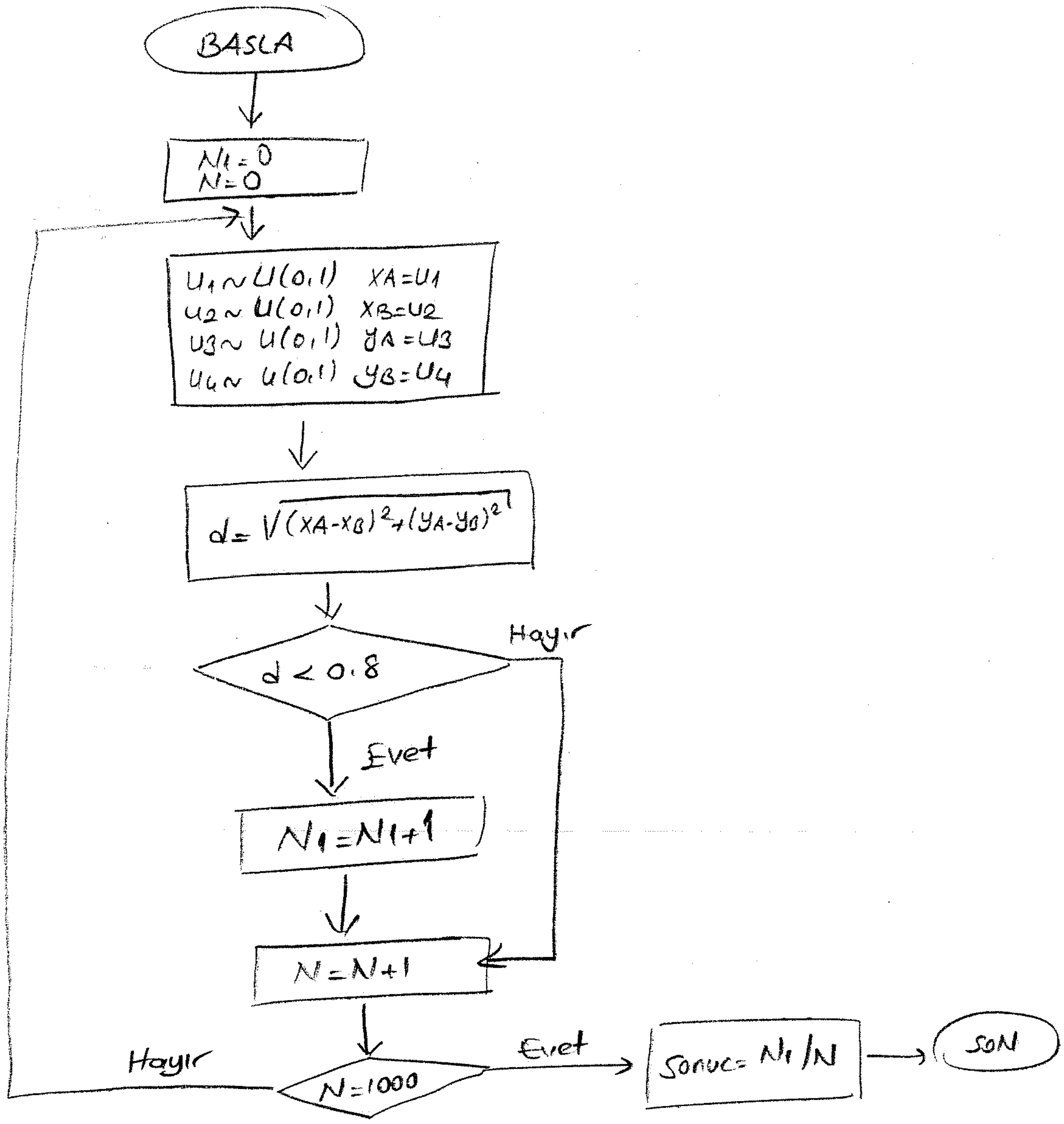
SORU: 4. sınıfta olan üniversite öğrencisine ailesinin 3 yıl içinde aylık gönderdiği para miktarı yukarıda gösterilmiştir. Monte Carlo benzetim modelini kullanarak öğrenciye 4. yılında gönderilecek para miktarını tespit ediniz.

Gönderilen Miktar	Frekans	Olasılık
100	3	0.0833
200	8	0.222
300	10	0.277
400	11	0.305
500	4	0.111
	36	1

Miktar	Kümülatif Olasılık	
100	0.0833	→ 0 – 0.0833
200	0.305	→ 0.0833 – 0.305
300	0.583	→ 0.305 – 0.583
400	0.888	→ 0.583 – 0.888
500	1	→ 0.888 – 1

```
function para = hesaplama (n)
    F1=0    F3=0    F5=0
    F2=0    F4=0

    for i=1 : n
        x=rand;
        if (0 < x < 0.0833)
            F1=F1+1
        end
        if (0.0833 < x < 0.305)
            F2=F2+1
        end
        if (0.305 < x < 0.583)
            F3=F3+1
        end
        if (0.583 < x < 0.888)
            F4=F4+1
        end
        if (0.888 < x < 1)
            F5=F5+1
        end
    end
    para =(F1+F2+F3+F4+F5) *100,
```


BASLA
N1=0
N=0
U1~U(0,1) XA=U1
U2~U(0,1) XB=U2
U3~U(0,1) YA=U3
U4~U(0,1) YB=U4
d= √((XA-XB)²+(YA-YB)²)
d < 0.8
Hayır
Evet
N1=N1+1
N=N+1
N=1000
Hayır
Evet
Sonuç= N1/N
SON

GENEL TEKRAR

1) DİNAMİK SİSTEMLER

Başlangıç Değer Problemi

$$\frac{dx}{dt} = F(x,t)$$

$X(t_0) = X_0$

$X(t) = [\,x_1(t), x_2(t), \ldots, x_n(t)\,]$ → sistem durum vektörü

$X(0) = [\,x_1(0), x_2(0), \ldots, x_n(0)\,]$ → ilgili başlangıç durumları

Örnek: $\alpha'' + 2\beta'\alpha + \beta^2\alpha = \cos t$

$\beta' + \alpha\beta = 4$

başlangıç durumlarına bağlı;

$\alpha(0) = 2$

$\alpha'(0) = -1$

$\beta(0) = 1$

Çözüm: $x_1 = \alpha(t) \quad x_2 = \alpha'(t) \quad x_3 = \beta(t)$

$\boxed{x_1' = x_2}$

$x_2' + 2x_3'x_1 + x_3^2x_1 = \cos t \Rightarrow \boxed{x_2' = \cos t - 2x_3'x_1 - x_3^2x_1}$

$x_3' + x_1.x_3 = 4 \Rightarrow \boxed{x_3' = 4 - x_1.x_3}$

$\boxed{x_1' = x_2}$

$\boxed{\begin{array}{l} x_1(0) = 2 \\ x_2(0) = -1 \\ x_3(0) = 1 \end{array}}$

$X(t) = [\,x_2, \cos t - 2x_3'x_1 - x_3^2x_1, 4 - x_1.x_3\,]$

$X(0) = [\,2, -1, 1\,]$

$t_0 = 0$

EULER YÖNTEMİ

$X(t)$ → sürekli zamanlı t

$X(k)$ → ayrık zamanlı k

Sürekli zamanlı t yerine ayrık zamanlı k değişkeni tanımlamak uygundur.

Sürekli zamanın ayrık zamanla yerdeğiştirdiği bu işleme ayrıklaştırma denir.

$\boxed{X(k+1) = X(k) + h.F(t_k, X(k))}$ → euler yöntemi $\boxed{t_{k+1} = t_k + h}$

Örn: $x' = x^2t \qquad x(1) = 3$ euler yöntemi uygulayınız.

$\frac{dx}{dt} = F(t,x)$

$x(t_0) = 3$

$t_0 = 1$

$\int \frac{dx}{dt} = \int x^2t$

$\int_3^x \frac{dx}{x^2} = \int_1^t t.dt \Rightarrow -\frac{1}{x} + \frac{1}{3} = \frac{t^2}{2} - \frac{1}{2}$

$\boxed{x(t) = \frac{6}{5 - 3t^2}}$ → gerçek çözüm

Euler uygularsak: $h=0.05$

$$X(k+1) = X(k) + h.\,F(t_k, X(k))$$

$$X(1) = 3$$

k=1 için

$$X(2) = X(1) + 0.05.\ (X^2(k).\,t_k)$$

$$X(2) = 3 + 0.05\ (9.1) = 3.45$$

$$t_1 = t_0 + 0.05$$

$$t_1 = 1.05$$

$$X(3) = X(2) + 0.05.(3.45)^2.(1.05)$$

$$= 3.45 + 0.05\ (3.45)^2.(1.05)$$

$$= 4.07$$

Sözde Kod:

```
t=1
X=3
print t,x
    for k=1 to n
        X= X + h. x^2 t
        t= t+h
    print t,x
    next k
```

Matlab Kodu:

```
to=1;
Xo=3;
X=[Xo];
t=[to];
Xg=[6/5-3*to^2];
h=0.05;
 for i=1:5
   Xyeni= Xo+h*Xo^2*to;
   to=to+h;
  Xg=[Xg 6/5-3*to^2];
  X=[X Xyeni];
  t=[t to];
  Xo=Xyeni;
  end.
   plot(t,x,'r-d')
   hold on
    plot(t,Xg,'k-s')
```

TAYLOR YÖNTEMİ

$$\boxed{\underbrace{X(t+h) = X(t) + h.X'(t)}_{\text{euler yöntemi}} + \frac{1}{2}h^2 X''(t)}$$ → taylor yöntemi

$$X(k+1) = X(k) + h.F(t_k, x(k)) + \frac{1}{2}h^2\left(\frac{\partial F}{\partial t} + \frac{\partial F}{\partial x}.F(t,x)\right)$$

Örn: $X' = x^2 t$ taylor tekniğini uygulayınız. $(h=0.05)$ $X(1)=3$

$$X(k+1) = X(k) + h.F(t_k, X(k)) + \frac{1}{2}h^2\left(\frac{\partial F}{\partial t} + \frac{\partial F}{\partial x}.F(t,x)\right)$$

$$\frac{\partial F}{\partial t} = x^2 \qquad \frac{\partial F}{\partial x} = 2xt \qquad F(t,x) = x^2 t$$

$$X(2) = X(1) + 0.05.(9.1) + \frac{1}{2}.(0.05)^2(9+6.9)$$

$$X(2) = 3 + 0.05.9 + \frac{1}{2}(0.05)^2.(63)$$

$$= 3 + 0.45 + 0.07875 = \underline{\underline{3,53}}$$

$t_2 = 1 + 0.05 = 1.05$

$$X(3) = X(2) + 0.05(3.53)^2.(1.05) + \frac{1}{2}.(0.05)^2.((3.53)^2 + 7.41.13.03)$$

$$= 3,53 + 0.6515 + 0.136 = \underline{\underline{4,32}}$$

* Taylor Euler yöntemine göre daha doğru sonuçlar elde eder.

Sözde Kodu:

```
t = 1
X = 3
print t, x
  for k=1 to n
     X = X + h. x²t + ½h². (x2 + 2xt. x²t)
     t = t+h
     print t, x
     next k
```

Matlab Kodu :

```
to=1;
Xo=3;
t=[to];
X=[Xo];
Xg=[6/5-3*to^2];
h=0.05
for i=1:5
    Xyeni = Xo + h* Xo^2 *to + 1/2 *h^2 (Xo^2 + 2*Xo*t * Xo^2*to);
    to=to+h;
    Xg=[Xg  6/5-3*to^2];
    X=[X  Xyeni];
    t=[t  to];
    Xo=Xyeni;
end.
plot (t,X, 'r-d');
hold on
plot (t,X, 'K-s');
```

RUNGE-KUTTA YÖNTEMİ

Euler → 1. dereceden
Taylor → 2. dereceden
Runge-kutta → 4. dereceden

Bu klasik bir runge-kutta algoritmasıdır. İki avantajı var =
1) Doğru çözümlerdir. (4. dereceden Taylor'la eşdeğer.)
2) Türev hesaplaması gerektirmediğinden kullanımı kolaydır.

$$K_1 = F(t_k, x(k))$$
$$K_2 = F(t_k+\tfrac{1}{2}h, \quad x(k)+\tfrac{1}{2}hK_1)$$
$$K_3 = F(t_k+\tfrac{1}{2}h, \quad x(k)+\tfrac{1}{2}h.K_2)$$
$$K_4 = F(t_k+h, \quad x(k)+hK_3)$$
$$t_{k+1} = t_k + h$$
$$x(k+1) = x(k) + \tfrac{1}{6}h(K_1+2K_2+2K_3+K_4)$$

Örn: $x' = x^2t \qquad X(1)=3$ runge-kutta yöntemi ile bulunuz. ($h=0.05$)

$K_1 = F(1, 3)$

$K_1 = x^2t = 9$

$K_2 = F(1+\frac{1}{2}.0.05, \quad 3+\frac{1}{2}.0.05.9)$

$= F(1.025, \quad 3.225)$

$K_2 = x^2t = 10.66$

$K_3 = F(1+\frac{1}{2}.0.05, \quad 3+\frac{1}{2}.0.05.10.66)$

$= F(1.025, \quad 3.2665)$

$K_3 = x^2t = 10.93$

$K_4 = F(1+0.05, \quad 3+0.05.10.93)$

$F = (1,05, \quad 3.5465)$

$K_4 = x^2t = 13,20$

$$X(k+1) = X(k) + \frac{1}{6}h(K_1+2K_2+2K_3+K_4)$$

$$X(2) = X(1) + \frac{1}{6}h(K_1+2K_2+2K_3+K_4)$$

$$X(2) = 3+\frac{1}{6}.0.05(9+21.32+21.86+13.20)$$

$$= \underline{\underline{3,544}}$$

Sözde Kodu:

```
t=1
x=3
print t, x
  for k=1 to n
     K1= tx^2
     K2= (t+1/2 h) (x+1/2 h.K1)^2
     K3= (t+1/2 h) (x+1/2 h.K2)^2
     K4= (t+h) . (x+h.K3)^2
     X= X+ 1/6 .h (K1+2K2+2K3+K4)
     t=t+h
     print t, x
     next k
```

Matlab Kodu:

```
to=1;
xo=3;
t=[to];
x=[xo];
xg= [6/5-3*to^2];
h=0.05;
 for i=1:5
    K1= to* xo^2;
    K2= (to+1/2*h).(xo+1/2*h*K1)^2
    K3= (to+1/2*h). (xo+1/2*h*K2)^2
    K4= (to+h) . (xo+h*K3)^2
    Xyeni = Xo+ 1/6*h*(K1+2*K2+2*K3+K4)
    to=to+h;
    Xg= [xg 6/5-3*to^2];
    X= [x Xyeni];
    t= [t to];
    Xo=Xyeni;
 end
 plot(t, x, 'r-d');
 hold on
 plot(t, xg,'k-s';
```

* Runge-kutta yöntemi; euler, taylor yöntemlerine göre daha iyi sonuç verir. Çünkü gerçek çözüme yakın sonuç runge-kutta ile elde edilir.

YÜKSEK DERECEDEN SİSTEMLER

* Herhangi bir diferansiyel denklemler toplamını birinci dereceden denklemler kümesine eşdeğer hale getirmek mümkündür.

* $X' = F(t, x, y)$ $X(t_0) = X_0$
  $Y' = G(t, X, y)$ $Y(t_0) = y_0$

  Daha önce anlatıldığı gibi X değişkenini k_1, k_2, k_3 ve k_4 değerleriyle, y değişkenini τ_1, τ_2, τ_3 ve τ_4 ile ilişkilendirerek Runge-kutta yöntemini uygulamak basittir.

Örnek:

$u(t) \longrightarrow \boxed{\text{System}} \longrightarrow x(t)$

$$\left.\begin{aligned} &X'' + 3XX' = u(t) \\ &u(t) = t \quad t \geq 0 \\ &X(0) = 2; \\ &X'(0) = 1; \end{aligned}\right\}$$ Euler yöntemini uygulayınız.

Çözüm: $y(t) = x'(t)$

$$y' + 3xy = t$$
$$x' = y$$
$$x(0) = 2;$$
$$y(0) = 1;$$

$$\boxed{\begin{aligned} &y' = t - 3xy \\ &x' = y \\ &x(0) = 2 \\ &y(0) = 1 \end{aligned}}$$

$t_{k+1} = t_k + h$

$$X(k+1) = X(k) + h.\,F(t, x) = X(k) + h.\,y(k)$$
$$y(k+1) = y(k) + h.\,F(t, y) = y(k) + h.\,(t - 3x(k)y(k))$$

Sözde Kodu:

```
t=0
x=2
y=1
 print t,x,y
    for k=1 to n

        x = x + h.y
        y = y+h.(t-3xy)

        t=t+h;
    print t,x,y
    next k
```

Sistem
- Girişi Olan
  - Zaman Odaklı
  - Olay odaklı
- Girişi Olmayan
  - Otonom
    - Kararlı
    - Kararsız
    - Marjinal

Otonom Sistemler: Bir sistemin hiçbir girişinin olmaması mümkündür. Böyle sistemler dış tepkilerden bağımsız olduğu için otonom olarak adlandırılır. Otonom sistemin çözümü, sistemin doğal tepkisi olarak adlandırılır.

Kararlı: Kısa bir geçiş evresinden sonra çıktı sıfıra doğru yaklaşır.

Kararsız: Sınırlama olmaksızın doğal tepki artar.

Marjinal: Tepki periyodik ve sınırlıdır.

Zaman Odaklı: $r(t) = 4 - t$ gibi deterministik bir zaman fonksiyonu ise **zaman odaklıdır.**
Olay Odaklı: $r(t)$ aşağıdaki şekilde rastgele bir işlem gibi stokastik olarak tanımlanırsa model **olay odaklıdır.**

$$r(t) = \begin{cases} 1 & \text{, olay oluşur} \\ 0 & \text{, olay oluşmaz.} \end{cases}$$

* $x(t)$, t anındaki popülasyon büyüklüğü

$x_m \rightarrow$ sistemin desteklediği maksimum popülasyon büyüklüğü

$\dfrac{x(t)}{x_m} \rightarrow$ Sistemin doluluk oranı

$1 - \dfrac{x(t)}{x_m} \rightarrow$ Büyüme için geriye kalan kullanılabilir sistem oranı.

$x(t)$ küçükse; $1 - \dfrac{x(t)}{x_m}$ olduğu için ; $1 - \dfrac{x(t)}{x_m} \approx 1$ olur. Denklem Malthusian'dır. (Büyüme artar)

$x(t)$ büyüklüğü kapasiteye yakınsa; $1 - \dfrac{x(t)}{x_m} \approx 0$ olur ve büyüme oranı düşer.

* $x(t) \rightarrow$ Av Popülasyon büyüklüğü $\quad y(t) \rightarrow$ Avcı popülasyon büyüklüğü

Av-Avcı Popülasyon büyüme oranı;

$$\left.\begin{aligned} x' &= \alpha_1 . x \left(1 - \frac{y}{\beta_1}\right) \\ y' &= \alpha_2 . y . \left(-1 + \frac{x}{\beta_2}\right) \end{aligned}\right\} \rightarrow \text{Lotka - Volterra denklemi}$$

Bu denklem doğrusal olmadığı için bilinen analitik bir çözümü yoktur. Önce anlattığımız sayısal teknikler kullanılır.

Sözde Kodu:

```
read t, x, y
print t, x, y
  for k=1 to n
    x1 = x(1+hα1 - hα1 y/β1)
    y1 = y(1-hα2 + hα2 x/β2)
    x = x1
    y = y1
    t = t+h
    print t, x, y
  next k
```

Euler ile; $X(k+1) = X(k) + h.(\alpha_1 . x(1 - \frac{y}{\beta_1}))$

$= X(k) + h.\alpha_1 x - h\alpha_1 x y/\beta_1$

$= x(1 + h\alpha_1 - h\alpha_1 y/\beta_1)$

$y(k+1) = y(k) + h.(\alpha_2 . y . (-1 + \frac{x}{\beta_2}))$

$= y(k) - h.\alpha_2 . y + h\alpha_2 y . x/\beta_2$

$= y(1 - h\alpha_2 + h\alpha_2 x/\beta_2)$

NOT: $[t_0, t_n]$ aralığında ve h aralığı olduğunu düşünürsek; $n = (t_n - t_0)/h$ oldukça büyük dür. $t_0=0$ $t_n=10$ $h=0.001$ ise $n=10000$ örnek gerekir. Fakat 50 tanesi yeterli olacaktır. Bu problem **kontrol break** ile çözülebilir. Böylece $[t_0, t_n]$ aralığında $m \times n$ hesaplama yapılır. $m = (t_n - t_0)/n.h$

Sözde Kodu:

```
read t, x, y
print t, x, y
  for i=1 to n
    for j=1 to m
      x1 = x(1+hα1 - hα1 y/β1)
      y1 = y(1-hα2 + hα2 x/β2)
      x = x1
      y = y1
      t = t+h
    next j
  print t, x, y
  next i
```

Örnek:

$$\left.\begin{aligned} X' &= 3x\left(1 - \tfrac{1}{10}y\right) \\ Y' &= 1{,}2Y\left(-1 + \tfrac{1}{25}x\right) \\ X(0) &= 10 \\ y(0) &= 5 \end{aligned}\right\}$$

Lotka-voltera sisteminin $[0,5]$ aralığında sayısal çözümünü veriniz. $(h=0.001)$

Her bir aralıkta 10 örnek alırsak; $n = 10 \times 5 = 50$

$m = (5-0)/50 \times 0.001 = 100$

Toplamda; $m \times n$ döner. $100 \times 50 = 5000$ güncelleme olur.

Euler uygularsak;

$$X(k+1) = X(k) + h.\left(\alpha_1 . x - \left(1 - \frac{y}{B_1}\right)\right)$$
$$y(k+1) = y(k) + h.\left(\alpha_2 y - \left(-1 + \frac{x}{B_2}\right)\right)$$

$$x(k+1) = x + h.\alpha_1 x - h\alpha_1 x y / B_1 \Rightarrow x(1 + h\alpha_1 - h\alpha_1 y / B_1)$$
$$y(k+1) = y - h\alpha_2 y + h\alpha_2 y x / B_2 \Rightarrow y(1 - h\alpha_2 + h\alpha_2 x / B_2)$$

Matlab Kodu:

```
function [t,x,y] = lotka_voltera(h, xo, yo, n)
  x = [xo];
  y = [yo];
  t=0;
    for i = 1:n
      for j= 1:100
        X1 = Xo * (1+h*3 - h*3*yo/10);
        Y1 = Yo* (1- h*1.2 - h*1.2*Xo/25);
        Xo = XX;
        Yo = YY;
        X = [x; xx]
        y=[y; yy]
        t=t+h;
      end
    end
    plot(x)
    hold on
    plot(y,'r-')
```

→ matlab'da yaz.

SİSTEMİN PERFORMANS ÖLÇÜTLERİ

1. Çevrim Zamanı
2. Doluluk Oranı
3. Bekleme Zamanı
4. Kalite
5. Maliyet

SİSTEMLER

Kesikli Sistem

* Sistem durum değişkenleri zaman içinde kesikli noktalarda değişir.
* Banka kesikli sisteme bir örnektir. Çünkü müşteri sayısı bir müşteri geldiğinde ya da bir müşteri servisi tamamladığında değişir.

Sürekli Sistem

* Sistem durum değişkenleri zaman içerisinde sürekli olarak değişir.
* Uçağın hareketi sürekli sisteme örnektir. Hız ve pozisyonu sürekli olarak değişir.

SİSTEMLERİN ÇÖZÜMÜ

- SİSTEM
  - Gerçek Sistem Üzerinde Deney
  - Sistemin Bir Modeli ile Deney
    - Fiziksel Model
    - Matematiksel Model
      - Analitik Çözüm
      - BENZETİM

Model: Bir sistemin gösterimi olarak tanımlanabilir.

Fiziksel Model

Gerçek sisteme benzer.

Matematiksel Model

Bir sistemi göstermek için sembolik notasyonlar ve matematiksel eşitsizlikler kullanılır.

BENZETİM MODELLERİ

- Statik veya Dinamik
- Deterministik veya Stokastik
- Kesikli veya Sürekli

Statik veya Dinamik

1) Statik Benzetim Modeli

* Sistemin belirli bir andaki gösterimidir.
* Monte Carlo benzetim modeli bu türe uygun modellerdir.

2) Dinamik Benzetim Modeli

* Sistemin çalışma zamanına göre yapılan modellemedir.
* Bir banka için kurulan benzetim modeli 8 saatlik çalışma zamanı dikkate alınarak çalıştırılır.

Deterministik veya Stokastik

1) Deterministik Benzetim Modeli

* Rassal değişken içermeyen benzetim modelidir.
* Bu benzetim modelinde verilen bir girdi seti için çıktı seti vardır.

2) Stokastik Benzetim Modeli

* Bir veya daha fazla rassal değişken içeren benzetim modelidir.
* Banka örneğinde, varışlar arası zaman aralığı ve servis zamanı rassal değişkenlerdir.

Kesikli veya Sürekli

EXCEL

* Excel'de rastgele sayı üretimi için `=S_SAYI_URET()` fonksiyonu kullanılır.
* 0 veya 1 üretilmek isteniyorsa; `=EGER(S_SAYI_URET()<=0.5, -0, -1)`
* Belirli bir aralıkta rastgele sayı üretmek için `=RASTGELEARADA(alt, -üst)` metodu kullanılır.
* Rnd 01() = 0 ile 1 aralığında rastgele sayı üretir.
* Discrete Uniform (min, max) = min ile max arasında rastgele sayı üretir.

1) MADENİ PARA BENZETİMİ

* Hilesiz bir paranın 10 defa atıldığını düşünelim. (Yazı tura gelme olasılığı eşit = 0.5)
* Bu bir Monte Carlo benzetimidir. Çünkü bir zaman veya olay yoktur. Böyle sistemler stokastik sistemlerdir.

`=EĞER(C15<=$C$7; H; T)` (C7 = 0.5 => H'ın gelme olasılığı)

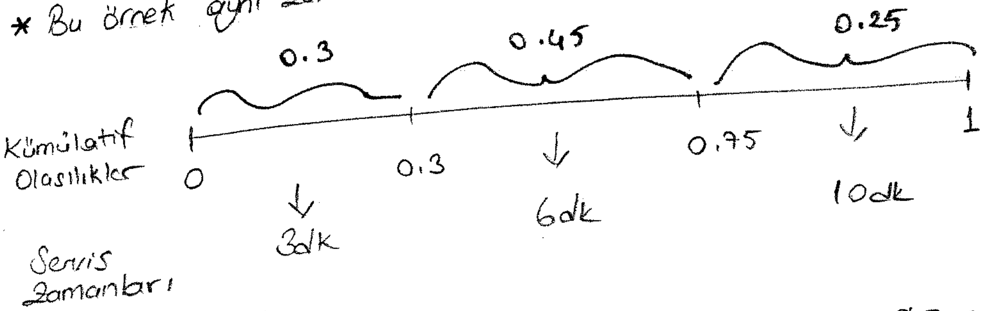
→ `$C$7`'nin anlamı; C7 hücresinin içeriğinin alınmasıdır. Yani denklemde sürekli olarak C7'nin değeri kullanılsın.

→ Frekans herbirinin kaç defa geldiğini belirlemek için kullanılır.

Frekansı bulmak için `=EĞERSAY($I$15:$I$24; H)` kullanılır. I15 ile I24 arasında (10 hücrede) H'ları say. Aynı şekilde; `=EĞERSAY($I$15:$I$24; T)`. T'leri de sayarız. Frekanslarını bulup olasılıklarını hesaplarız.

2) RASTGELE SERVİS ZAMANLARININ BENZETİMİ

* Otomatikleştirilmiş bir telefon bilgi servisi herbir çağrıya 3, 6 ve 10 dk harcamaktadır. Herbir servisin çağrı oranı %30, %45 ve %25'tir. Amacımız bu servis zamanlarını EXCEL'de simule etmektir.
* Bu örnek aynı zamanda Monte-Carlo benzetimidir. Çözüm şöyle olmalıdır.



`=DiscreteEmp($D$7:$D$9; $B$7:$B$9)` fonksiyonunda

`$D$7:$D$9` → kümülatif olasılıkları içeren hücrelerin aralığı

`$B$7:$B$9` → servis zamanları için istenen değerleri içeren hücrelerin aralığı

	B	C	D
	Servis Time	Probability	Cumulative Probability
7	3	0.30	0.30
8	6	0.45	0.75
9	10	0.25	1.00

	Simulation Table	
	Step	Service Time
16	1	3
17	2	6
18	3	6
19	4	10
⋮	⋮	⋮
40	27	10

= DiscreteEmp ($D$7 : $D$9 ; $B$7 : $B$9)

Frekansları; = EĞERSAY ($C$16 : $C$40 ; 3)

= EĞERSAY ($C$16 : $C$40 ; 6)

= EĞERSAY ($C$16 : $C$40 ; 10)

3) RASTGELE VARIŞ ZAMANLARININ BENZETİMİ

* Bir telefon bilgi servisine göre ; telefon çağrıları varışlar arası zamanı 1,2,3,4 dakikaya sahip varışlar arası zaman ile rastgele zamanlardan oluşur. Her birinin olasılığı eşittir.
* Amaç varış zamanları ile varışlar arası zamanı nasıl üreteceğimizi belirlemektir.
* Bu, önceki varış olayı gibi sadece bir olaya sahiptir. Dolayısıyla bu örnek aynı zamanda ilk dinamik olay tabanlı örneğimizdir.
* Ayrık bir unifor dağılım ile zamanlar üretilir.

= DiscreteUniform ($C$6 ; $C$7) fonksiyonu kullanılır.

Veya = DiscreteUniform (1,4)

low high

Bu örnek için;

= DiscreteUniform = 1 + Int (4 * Rnd()) komutu kullanılır.

	B	C
	Interarrival Times (minutes)	
6	Minimum	1
7	Maximum	4

Simulation Table		
Step	Interarrival Time	Arrival Time
		0
1	2	1
2	1	5
3	4	9
4	4	10
5	1	13
6	3	

= DiscreteUniform ($C$6 ; $C$7)