TMS050
Matematisk statistik och
simuleringsteknik Z, del B
Tommy Norberg
8 november 2002

Simuleringsuppgifter ht02

Alla kursdeltagare skall lämna sina egna redovisningar till minst 3 av de 5 uppgifterna på de följande sidorna. Glöm ej att ange namn, personnummer och gärna e-postadress på den första sidan i dina redovisningar och initialer på varje inlämnat ark papper. Du underlättar för oss som ska bedömma och ev kommentera din redovisning om du bemödar dig att skriva tydligt och kortfattat. Programkod behöver normalt ej redovisas. Det är dock viktigt att du ordentligt redogör för hur ditt simuleringsprogram arbetar, gärna med flödesschema. Ibland kan det vara svårt att bedömma korrektheten hos en redovisning. I sådana fall kan jag tänkas begära in programlistningar. De ska i så fall vara bra kommenterade (d.v.s lättlästa) och lämnas elektroniskt till e-postadressen `tommy@math.chalmers.se`. Det går bra att lämna redovisningarna elektroniskt till denna adress. Redovisningen ska då vara bilagd till e-brevet i formatet pdf eller ps. Enbart text direkt i e-brevet med bilagor i pdf eller ps-format accepteras också. Word-filer accepteras ej.

Glöm inte att statistiskt bedömma felet i dina simuleringssvar när så är möjligt med den kunskapsbank du förväntas ha efter genomgången grundkurs.

Detta är en obligatorisk praktisk del av kursen. Av hela kursens 6 p svarar denna simuleringsdel för ungefär 1 p. Jag kommer att rätta inlämnade redovisningar i takt med att de kommer in och lämnar icke godkända i retur om det hinns med innan vi skiljs åt sista föreläsningen torsdag den 12:e december. Räkna inte med att redovisningar inlämnade efter sista föreläsningen snabbrättas.

Betygsskala: 3 godkända redovisningar krävs för betyget 3, 4 godkända ger betyget 4 och 5 godkända ger betyget 5.

Innan du tar itu med simuleringsuppgifterna, läs de allmänna anvisningarna på de 4 sista sidorna i detta dokument. Där finns även en del programmeringstips som kan underlätta i det praktiska arbetet med uppgifterna.

Uppgift 1: Glassförsäljning

Glassförsäljningssäsongen på Avenyn är 120 dagar lång. Innehavaren av ett glass-stånd räknar med att hon, med nuvarande prissättning, under

- soliga dagar säljer för ca SEK 3 300
- molniga dagar säljer för ca SEK 1 300
- regndagar säljer för ca SEK 500

Hon tar kontakt med dig för att eventuellt kunna få en uppskattning av hur försäljningen, sedd över hela säsongen, varierar ifrån år till år. Speciellt vill hon veta sannolikheten att förtjänsten överstiger SEK 215 000, vilket är vad hon måste sälja för att få kostnadstäckning.

Hon vet att i snitt över en säsong är det 56 sol-, 31 moln- och 33 regndagar. Hennes medelförsäljning är alltså ca SEK

$$56 \cdot 3\,300 + 31 \cdot 1\,300 + 33 \cdot 500 = 241\,600 \approx 242\,000$$

Men detta räcker inte för att besvara hennes fråga.

Du inser att du behöver en (enkel) modell för väderförändringarna under säsongen. Efter flera kontakter med SMHI, får du reda på att soldagar följs av molndagar ca 1 dag av 5 och av regndagar också ca 1 dag av 5. Du får dessutom reda på att molndagar följs av molndagar i ca 1 fall av 5 och av regndagar i ca 2 fall av 5, samt att regndagar följs av soldagar ca 3 gånger av 10 och av molndagar ca 2 gånger av 5.

På basis av detta bestämmer du dig för modellera vädret med en Markovkedja. Skriv upp tillståndsrum och transitionsmatris, och försäkra dig om att din modell överensstämmer med innehavarens påstående att det under en säsong är ca 56 sol-, 31 moln- och 33 regndagar.

Simulera 250 säsonger och besvara med hjälp av dina simuleringsresultat innehavarens fråga. Glöm ej att göra en ordentlig statistisk analys av ditt simuleringsresultat.

Redovisa utöver ditt simuleringsresultat och den statistiska analysen av densamma

1) vald initialfördelning (inkl motivering)
2) genomsnittligt antal sol-, moln- och regndagar under en säsong
3) standardavvikelser för dessa variabler
4) medelvärde och standardavvikelse för försäljningen under en säsong

Till sist: Jag kan tänka mig två ganska öppenbara invändningar mot ovanstående simulering: dels så har vi inte tagit hänsyn till dag-till-dag-variationen i försäljningen; dels så är antagandet att vädret är Markovskt suspekt. Ibland har vi ju här i Göteborg typiskt varannandagsväder med en dags solsken mellan kortvariga regnväder ifrån väster, och ibland har vi ju ganska långvariga låg- resp högtryck.

Uppgift 2: Tillverkning av kretskort ur ett enhets- eller detalj-perspektiv

I denna uppgift ska vi medelst simulering lära oss mer om kretskortstillverkningen som studeras i övning 2.6 på sidorna 68-69 i F & V-F. I deluppgifterna (a) t.o.m (e) räknar vi fram flera intressanta fakta, bl.a väntevärdet $\mu_T = E[C|X_0 = T]$ av kostnaden C att tillverka ett kort, där betingningen $X_0 = T$ innebär att processandet av ett råaämne alltid påbörjas i "Tinning"-steget. Låt vidare F, I, S beteckna "Forming"-, "Insertion"- och "Solder"-stegen i tillverkningsprocessen. Vad jag vill veta är värdet av motsvarande betingade standardavvikelse $\sigma_T = \sqrt{\text{Var}[C|X_0 = T]}$. Det kan ju i vissa tillämpningar vara viktigt att veta hur kostnaden varierar från kort-till-kort.

Ett sätt att kontrollera rimligheten av sitt simuleringsresultat är att också låta programmet samtidigt simulera fram värden av kända storheter. Gör så för $E[C|X_0 = T]$, $R(T, i)$ för $i = T, F, I, S$ och sannolikheten $F(T, \text{scrap})$ att tillverkningen av ett kretskort misslyckas. Jämför med de teoretiskt uträknade värdena.

Uppgift 3: Stormskador enligt POT-modellen

Jag börjar med att kortfattat beskriva den s.k POT-modellen. POT är en förkortning av "Peaks Over Threshold". Vi tänker oss att en viss typ av händelser (t.ex olyckor, katastrofala stormar, e.dyl) inträffar enl en Poissonprocess med intensiteten λ. Man kan visa att Pareto-fördelningen är en rimlig stokastisk modell för de ekonomiska konsekvenserna av företeelser som inträffar sällan och får stora konsekvenser då de inträffar. Känt av oss alla är att Poissonprocessen är en bra modell för när i tiden ovanliga händelser inträffar.

Paretofördelningens fördelningsfunktion $F(x)$ ges av att

$$1 - F(x) = \left(1 + \frac{\xi x}{\sigma}\right)^{-1/\xi}, \quad x \geq 0,\ 1 + \xi x/\sigma > 0$$

Här är $-\infty < \xi < \infty$ en formparameter och $\sigma > 0$ en skalparameter. Då $\xi = 0$ ska $F(x)$ tolkas medelst gränsövergången $0 \neq \xi \to 0$, så att då är $1 - F(x) = \exp(-x/\sigma)$. Utfallsrummet för X definieras av att $x \geq 0$ och att $1 + \xi x/\sigma > 0$. Det är bara då $\xi < 0$ som det sistnämnda villkoret är en restriktion. Då gäller att $0 \leq x < -\sigma/\xi$. Annars gäller $0 \leq x < \infty$. Man kan visa att väntevärdet av en Pareto(ξ, σ)-variabel X är

$$E[X] = \frac{\sigma}{1 - \xi}$$

förutsatt att $\xi < 1$ (om $\xi \geq 1$ är väntevärdet oändligt).

Låt nu $u > 0$ vara en hög ekonomisk nivå. (Vi studerar bara händelser vars ekonomiska konsekvens är minst u). Låt $T_1, T_2, \ldots$ vara impulstidpunkterna i en Poissonprocess $N = \{N_t; t \geq 0\}$ med intensitet λ. Låt $X_1, X_2, \ldots$ vara oberoende och Pareto(ξ, σ)-fördelade.Vi ska tänka på T_i som tidpunkten då den i:te katastrofen inträffar. Motsvarande katastrofkostnad är $u + X_i$. Detta är POT-modellen.

Uppgiften består i att simulera fram ett försäkringsbolags totala kostnad C för de mycket stora stormar som kan tänkas inträffa de närmaste tre åren. Notera att

$$C = \sum_{T_i \leq 3} (u + X_i) = \sum_{i=1}^{N_3} (u + X_i)$$

Vi ska studera stormar, som leder till att det totala kompensationskravet från försäkringstagare är minst $u = 0.9$ Mkr (1 Mkr $= 10^6$ kr). Vi ska anta att det i snitt inträffar $\lambda = 3.8$ sådana stormar per år. Vi ska också anta att $\xi = 0.7$ och att $\sigma = 3.9$. Man kan visa att

$$E[C] = E[N_3]E[u + X_1] = 3\lambda(u + E[X_1]) = 3\lambda\left(u + \frac{\sigma}{1 - \xi}\right) = 158.46 \text{ Mkr}$$

Bolaget är inte speciellt intresserat av medelkostnaden $E[C]$. Däremot är man intresserad av stora kvantiler i fördelningen för C. För $0 < p < 1$ definierar vi därför c_p, så att

$$p = P(C \leq c_p)$$

Av intresse för försäkringsbolaget är kvantilerna $c_{0.50}$, $c_{0.75}$, $c_{0.90}$, $c_{0.95}$ och $c_{0.99}$. Den här typen av storheter benämns ofta "Value at Risk" (VaR) i ekonomiska sammanhang.

Din uppgift är att punktskatta $c_{0.90}$ medelst simulering.

Uppgift 4: En "steady-state" simulering

Låt $\{Y_t; t \geq 0\}$ vara den Markovprocess som definieras i exempel 4.1 i F & V-F, men med uthoppsintensiterna $\lambda(a) = 0.1$, $\lambda(b) = 0.2$ och $\lambda(c) = 0.15$ och förtjänstvektorn $\mathbf{f} = [\,80 \quad 100 \quad 125\,]^T$. Syftet med simuleringsuppgiften är att indirekt verifiera att Y_t, för stora t, är fördelad enligt den stationära fördelningen $\mathbf{p} = [\,9 \quad 3 \quad 4\,]/16$, genom att på nivån $\alpha = 0.05$ testa $H_0 : E[f(Y_t)] = \mathbf{pf}$ mot $H_1 : E[f(Y_t)] \neq \mathbf{pf}$.

Förhoppningsvis ska testet ej förkasta, men notera att det kommer att felaktigt göra så i ca 5% av simuleringarna. Råkar man ut för ett sådant oönskat förkastande, får man göra om simuleringen några gånger för att försäkra sig om att förkastandet var en tillfällighet.

I alla simuleringarna startar säljaren sin turné i stad a. Först ska du skaffa dig en uppfattning av längden av insvängningsförloppet genom att plotta det initiala förloppet av ett antal korta simuleringar. Skicka med en sådan plott i redovisningen och bestäm dig för ett värde $u > 0$, sådant att processen verkar vara stationär efter ca u tidsenheter.

Nu ska vi genom att göra en enda lång simulering studera gränsvärdet då $t \to \infty$ av den stokastiska variabeln $Z(u,t)/(t-u)$, där $Z(u,t) = \int_u^t f(Y_s)\,ds$. Självklart gäller att gränsvärdet, om det existerar, är oberoende av u.

Självklart räcker det inte som bevis av att H_0 är sann att vi i en enda lång simulering (d.v.s för ett stort t) erhåller $Z(u,t)/(t-u) \approx \mathbf{pf}$, utan vi måste också göra en statistisk analys, som som resultat ger att H_0 ej ska förkastas. Inte heller då kan vi vara övertygade om att H_0 är sann, eftersom man ju i statistiska test endast skyddar sig emot att felaktigt förkasta genom att välja nivån α litet. Dock ska vi försöka att åstadkomma ett test med stor detekteringssannolikhet, d.v.s ett test som är sådant att det med stor sannolikhet förkastar H_0 om alternativet H_1 är sant. Lyckas vi med detta kan vi känna oss tämligen säkra på att H_0 är sann om testet ej förkastar.

I den statistiska analysen ska vi använda oss av idéen med s.k "batch means" som diskuteras i F & V-F, avsnitt 6.2.2. Dela därför in tidsintervallet $[u,\, t]$ i n lika stora bitar $[t_0,\, t_1]$, $[t_1,\, t_2], \ldots,$ där $t_0 = u$, $t_1 = t_0 + (t-u)/n$, $t_2 = t_1 + (t-u)/n$, o.s.v, och låt

$$X_1 = \frac{Z(t_0,t_1)}{t_1 - t_0},\ X_2 = \frac{Z(t_1,t_2)}{t_2 - t_1}, \ldots,\ X_n = \frac{Z(t_{n-1},t_n)}{t_n - t_{n-1}}$$

Obs att $X_i = nZ(t_{i-1},t_i)/(t-u)$ (ty $t_i - t_{i-1} = (t-u)/n$) och att, under H_0,

$$E[X_i] = \frac{E\left[\int_{t_{i-1}}^{t_i} f(Y_s)\,ds\right]}{t_i - t_{i-1}} = \frac{\int_{t_{i-1}}^{t_i} E[f(Y_s)]\,ds}{t_i - t_{i-1}} = \mathbf{pf}$$

Om n ej är för stort, så är $X_1, \ldots, X_n$ approximativt oberoende och likafördelade. Vi kan nu testa H_0 så som vi lärt oss i del A av kursen, genom att beräkna medelvärde $\bar{x}$ och varians s^2, och låta testet förkasta H_0 då $|\bar{x} - \mathbf{pf}|/(s/\sqrt{n}) \geq t_{n-1,0.025}$. Alternativt kan man bestämma motsv 95% konfidensintervall och förkasta H_0 om $\mathbf{pf}$ ej ligger i det. Det är nu som det är viktigt att n ej heller är för litet, för då blir detekteringssannolikheten onödigt låg. Att välja ett bra värde på n är alltså kritiskt.

När du söker efter ett värde på n som varken är för stort eller onödigt litet, är det en god idé att plotta alla eller en del av observationerna $x_1, \ldots, x_n$ för att med ögats hjälp avgöra om de verkar oberoende. Det är svårt att säga hur en plott av oberoende observationer ska se ut annat än att den bör vara rätt kaotisk. Om observationerna ej är oberoende, så brukar man kunna se med ögat att de på något sätt hänger ihop. När du bestämt dig för ett lämpligt värde på n, skriv ut din plott och inkludera den i redovisningen av uppgiften.

Uppgift 5: Tillverkning av kretskort ur ett flödes-perspektiv

Syftet med denna uppgift är att simulera flödet av kort genom kretskortstillverkningen som definieras i övning 2.6 på sidorna 68-69 i F & V-F. J.f.r med uppgift 2. Vi ska tänka oss tillverkningsprocessen som ett nät av $M/M/c$-köer kallade T, F, I och S. Noderna T, F och S består av en betjäningsstation som i snitt klarar av att hantera $\mu_T = \mu_F = 30$ resp $\mu_S = 24$ kort per timma. Nod I består av två betjäningsstationer som vardera i medel kan hantera $\mu_I = 15$ kort per timma. Alla ankomster till nätet sker till nod T och i snitt ankommer $\gamma = 19$ råämnen per timma.

Tillverkning sker på arbetsdagar som nominellt är 8 timmar långa. Varje dag startar nätet tomt på kort (d.v.s alla noder är initialt tomma). Man kör nätet under $t < 8$ timmar. Därefter slutar man att ta emot externa råämnen vid nod T, men man fortsätter att köra nätet tills det är tomt igen. Först då avslutas dagens tillverkning.

Sök medelst simulering ett värde på t sådant att den genomsnittliga arbetsdagen blir ca 8 timmar lång. Upplösningen i t ska vara 6 minuter eller 1/10 timma. D.v.s, t ska ges med en decimal.

För det erhållna värdet på t, låt X vara antalet processade kort under en arbetsdag. Punkt- och intervallskatta medelst simulering X:s väntevärde μ och standardavvikelse σ samt punktskatta X:s median m.

TMS050
Matematisk statistik och
simuleringsteknik Z, del B
Tommy Norberg
8 november 2002

Allmänna anvisningar för simuleringsuppgifterna

1. Redovisningsmall

Redovisningen av en simuleringsuppgift bör bestå av fyra klart urskiljbara delar:

A Definition av den stokastiska processen som studeras

B Beskrivning av simuleringsalgoritmen

C Resultat och statistisk analys

D Svar på ev frågor med motiveringar baserade på resultatanalysen

Dessutom är det önskvärt att du, för att inte göra min arbetsbörda för tung, bemödar dig om att skriva kortfattat och avstår ifrån att redovisa uträkningar och enklare härledningar.

2. Matlab tipps

Matlab är det programmeringsspråk jag rekommenderar att ni använder till att utföra simuleringsuppgifterna ovan. För den som inte har någon vettig kort introduktion till Matlab, kan jag rekommendera en utmärkt sådan skriven av Jörgen Löfström. Du hittar den via Jörgens hemsida

```
http://www.math.chalmers.se/~jorgen
```

eller kursens

```
http://www.md.chalmers.se/Stat/Grundutb/Chalmers/TMS050/
```

Matlab är ett effektivt programspråk om man lyckas med att utnyttja vektoriseringsmöjligherna. T.ex gäller att om man vill generera $25 \cdot 2$ slumptal, så kan man göra det med instruktionen

```
u = rand(25,2)
```

Då erhåller du en matris med 25 rader och 2 kolumner fylld med slumptal. Vill du använda första kolumnen till observationer av Exp(1.2)-fördelningen, så gör du

```
u1 = u(1:25,1)
x = -1.2*log(u1)
```

Om du är ute efter impulstiderna i en Poissonprocess med intensiteten $1/1.2 = 5/6$, gör du sedan

```
t = cumsum(x)
```

Om du nu gör

```
u2 = u(1:25,2)
y = 2.*(u2 <= 0.75) + 3.*(0.75 < u2 & u2 <= 0.85);
y = y + 4.*(0.85 < u2 & u2 <= 0.92) + 5.*(0.92 < u2)
```

så erhåller du en kolumnvektor bestående av 25 observationer av en diskret stokastisk variabel med täthetsfunktionen

$$p(2) = 0.75 \quad p(3) = 0.10 \quad p(4) = 0.07 \quad p(5) = 0.08$$

Vill du istället ha en radvektor, så får du medelst transponering bilda

```
z = y'
```

Kommandot

```
obs = [t y]'
```

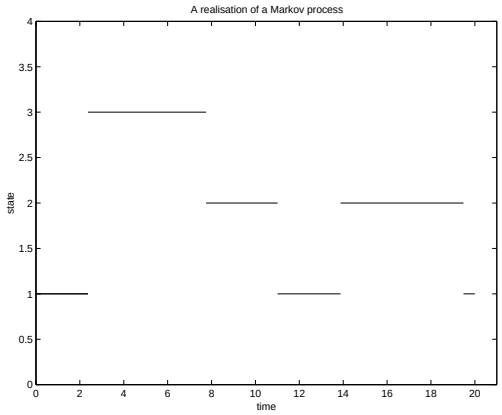
ger dig en observation (se figur 4 nedan) innehållande tiderna och storleken av de 25 första impulserna i en s.k märkt Poissonprocess (jämför med exempel 3.6 i F & V-F, där man bara vill åt vad som händer under en 15-minutersperiod). Plottar man processen som i figur 4 nedan är det brukligt att man kallar processen vi just simulerat för en märkt Poissonprocess. Annars kan man, som F & V-F gör i exempel 3.6, benämna den en sammansatt ("compound") Poissonprocess.

Ett par bra kommandon, som man behöver i den statistiska analysen är `mean` och `std`. Använd `help`-funktionen för att ta reda på exakt vad de gör. Det kan också vara bra att känna till att man med Matlab-kommandot `random` kan simulera observationer av flera olika fördelningar.

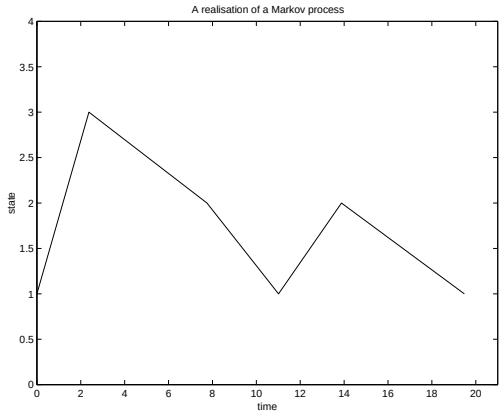
Strax följer en listning av ett program som ritar en realisering av en Markovprocess, som du lätt kan modifiera ifall du vill kunna se de realiseringar dina egna simuleringsprogram gör. Det kan ju vara vettigt att visuellt kolla att ens simulering ser ut att vara OK.

```
% Plottning av en
% realisering av en Markovprocess
% under tidsrymden [0, 20]
%
% Hopptidpunkter
t=[0 2.37 7.75 11.01 13.88 19.48];
% Sekvens av tillstånd
y=[1 3 2 1 2 1];
% Plotta första tillståndet
plot(t(1:2),[y(1) y(1)])
% Och sedan resten i samma plott
hold on
n=length(t); t=[t 20];
for i=1:n
  plot(t(i:(i+1)),[y(i) y(i)])
end
hold off
% Snygga till plotten
xlim([0 21]); ylim([0 4]);
title('A realisation of a Markov process');
xlabel('time'); ylabel('state');
% Skriv resultatet till en postscriptfil
print -dps exec0b.ps
```

Jag sparade koden i en script-fil som heter `exec0b.m`, och exekverade det genom att ge matlab kommandot `exec0b`. Resultatet visas i figur 1 nedan. Hur man skriver en script-fil berättar Jörgen om i kapitel 5 i sin korta och trevliga handledning.



Figur 1: Resultat av en exekvering av `exec0b`.



Figur 2: Resultat av en exekvering av `exec0c`.

Detta sätt att plotta lämpar sig inte så väl för långa realiseringar. Då är det ofta bättre att bara knyta ihop punkterna som i `exec0c.m`. Resultat är ej så snyggt. Se figur 2 ovan. Men som sagt, metodiken i `exec0c.m` är anpassad till långa realiseringar.

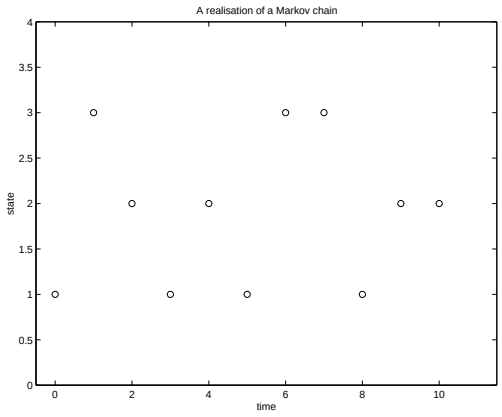
För den intresserade följer här en listning av `exec0c.m`:

```
% Plottning av en
% realisering av en Markovprocess
% under tidsrymden [0, 20]
%
% Hopptidpunkter
t=[0 2.37 7.75 11.01 13.88 19.48];
% Sekvens av tillstånd
y=[1 3 2 1 2 1];
% Plotta processen
plot(t,y);
% Snygga till plotten
xlim([0 21]); ylim([0 4]);
title('A realisation of a Markov process');
xlabel('time'); ylabel('state');
% Skriv resultatet till en postscriptfil
print -dps exec0c.ps
```

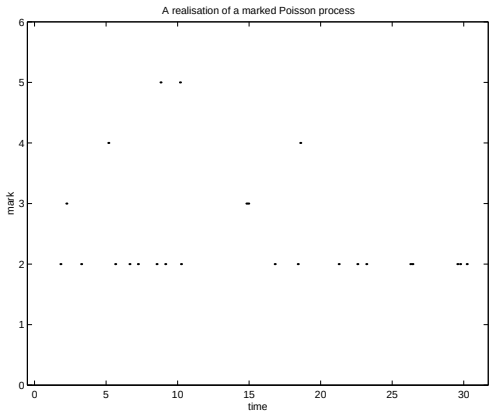
Att plotta sekvensen av tillstånd en Markovkedja genomlöper är enkelt. Här följer innehållet i `exec0a.m`:

```
% Plottning av en realisering
% av en Markovkedja
%
% Sekvens av tillstånd
x=[1 3 2 1 2 1 3 3 1 2 2];
% Motsvarande tidpunkter
n=1:length(x); n=n-1;
% Plotta kedjan
plot(n,x,'o')
% Snygga till plotten
xlim([-0.5 length(x)+0.5]); ylim([0 4]);
title('A realisation of a Markov chain');
xlabel('time'); ylabel('state');
% Skriv resultatet till en postscriptfil
print -dps exec0a.ps
```

Nedanstående figur 3 nedan visar resultatet av en körning. Testa själv att byta `'o'` i `plot(n,x,'o')` mot `'o-'` eller mot `'.'`.



Figur 3: Resultat av exekveringen av `exec0a`.



Figur 4: Realisering av en märkt Poissonprocess

Ibland behöver man dela upp en lång tidskontinuerlig realisering i mindre delar. Antag att hopptidpunkterna och sekvensen av tillstånd definieras av två vektorer `t` resp `y`. Scriptet nedan delar upp realiseringen i ett antal lika stora delar och beräkna deras respektive normerade integraler. Det skriver, efter det att en ny del är avskild, ut den och beräknar och skriver ut dess integral.

```
% Uppdelning av en realisering
% i delar som alla är ca tl tids-
% enheter långa samt beräkning av
% deras normerade integraler
%
% Hopptidpunkter och
% sekvens av tillstånd
t=[0 2.37 7.75 11.01 13.88 19.48];
y=[1 3 2 1 2 1];
tm=20; tl=7.5;
x=0; km=ceil(tm/tl); tl=tm/km
% Del k för k=1:km
i=1; n=1; z=y(i);
for k=1:km
  t1=(k-1)*tl; y1=z; j=2; s=t(i);
  while s<=k*tl & i<length(t)
    z=y(i);
    if i>1
      t1(j)=s; y1(j)=z;
    else
      j=j-1;
    end
    i=i+1; j=j+1; s=t(i);
  end
  if k==km & i==length(t)
    t1(j)=s; z=y(i); y1(j)=z;
  end
  [t1;y1]
% Integralen av del k dividerad
% med intervallets längd
  t2=[t1(2:length(t1)), k*tl]-t1;
  int=t2*y1'; x(n)=int/tl; x(n)
  n=n+1;
end
n=n-1;
% Plotta integralerna
plot(1:n,x,'o');
title('Batch means');
xlim([0,length(x)+0.5]);
ylim([0,ceil(max(x))]);
xlabel('obs no'); ylabel('value');
```

Scriptet sparar dessutom alla integralerna i vektorn `x` och avslutas med att de plottas mot $n = 1, 2, \ldots$.