

Warteschlangen, Simulationen

Stochastik mit MuPAD,

Prof. Dr. Dörte Haftendorn 25.11.04 Version MuPAD 4 vom 28. April 08

<http://haftendorn.uni-lueneburg.de>

www.mathematik-verstehen.de

+++++

Achtung, den theoretischen Teil habe ich inzwischen verbessert.

lamb

ist der Erwartungswert für die Anzahl der ankommenden Leute in einer (vernünftigen) Zeiteinheit, z.B 1/4 Stunde.

my

ist der Erwartungswert für die Anzahl der bedienten Leute in einer (vernünftigen) Zeiteinheit, z.B 1/4 Stunde.

h ist ein so kleiner Zeittakt, dass im Takt h niemals zwei Leute ankommen oder zwei bedient werden.

$\rho = \text{lamb} \cdot h$ ist dann die Wahrscheinlichkeit, dass im (kleinen) Zeittakt h eine Person ankommt.

$q = \text{my} \cdot h$ ist dann die Wahrscheinlichkeit, dass im (kleinen) Zeittakt h eine Person bedient wird.

Zustand i bedeutet: i Personen sind in der Warteschlange. $i=0,1,2,3,4,\dots$ hier bis 5, eigentlich weiter.

Bei dem Ansatz für die Übergangswahrscheinlichkeiten werden Terme mit h^2 vernachlässigt, denn h macht man hinreichend klein.

```
delete lamb, my, hz:
```

```
U:=matrix([[1-lamb*hz, lamb*hz, 0, 0, 0, 0, 0], [my*hz, 1-lamb*hz-my*hz, lamb*hz, 0, 0, 0], [0, my*hz, 1-lamb*hz-my*hz, lamb*hz, 0, 0], [0, 0, my*hz, 1-lamb*hz-my*hz, lamb*hz, 0], [0, 0, 0, my*hz, 1-lamb*hz-my*hz, lamb*hz], [0, 0, 0, 0, my*hz, 1-lamb*hz-my*hz]])
```

$$\begin{pmatrix} 1 - hz \cdot \text{lamb} & hz \cdot \text{lamb} & 0 & 0 & 0 & 0 \\ hz \cdot \text{my} & 1 - hz \cdot \text{my} - hz \cdot \text{lamb} & hz \cdot \text{lamb} & 0 & 0 & 0 \\ 0 & hz \cdot \text{my} & 1 - hz \cdot \text{my} - hz \cdot \text{lamb} & hz \cdot \text{lamb} & 0 & 0 \\ 0 & 0 & hz \cdot \text{my} & 1 - hz \cdot \text{my} - hz \cdot \text{lamb} & hz \cdot \text{lamb} & 0 \\ 0 & 0 & 0 & hz \cdot \text{my} & 1 - hz \cdot \text{my} - hz \cdot \text{lamb} & hz \cdot \text{lamb} \\ 0 & 0 & 0 & 0 & hz \cdot \text{my} & 1 - hz \cdot \text{my} - hz \cdot \text{lamb} \end{pmatrix}$$

```
lamb:=3: my:=4: rho:=lamb/my
```

$$\frac{3}{4}$$

```
hz:=1/10; h:=hz*1.0;
```

$$\frac{1}{10}$$

0.1

1

```
U:=matrix([[1-lamb*h, lamb*h, 0, 0, 0, 0, 0], [my*h, 1-lamb*h-my*h, lamb*h, 0, 0, 0], [0, my*h, 1-lamb*h-my*h, lamb*h, 0, 0], [0, 0, my*h, 1-lamb*h-my*h, lamb*h, 0], [0, 0, 0, my*h, 1-lamb*h-my*h, lamb*h], [0, 0, 0, 0, my*h, 1-lamb*h-my*h]])
```

```
[0,0,0,my*h,1-lamb*h-my*h,lamb*h,0],
[0,0,0,0,my*h,1-lamb*h-my*h,lamb*h],
[0,0,0,0,0,my*h,1-lamb*h-my*h]])
```

$$\begin{pmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.3 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.3 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.3 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.3 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0.3 \end{pmatrix}$$

Für die Bestimmung des Eigenvektors zum Eigenwert 1:

```
U-U^0
```

$$\begin{pmatrix} -0.3 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & -0.7 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & -0.7 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & -0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & -0.7 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & -0.7 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & -0.7 \end{pmatrix}$$

```
ev:=matrix([[ev1,ev2,ev3,ev4,ev5,ev6,ev7]])
```

```
( ev1 ev2 ev3 ev4 ev5 ev6 ev7 )
```

```
ev*(U-U^0)
```

```
( 0.4·ev2 - ev1·0.3 0.3·ev1 - ev2·0.7 + 0.4·ev3 0.3·ev2 - ev3·0.7 + 0.4·ev4 0.3·ev3
```

Wenn man dieses gleich Nullvektor setzt und analysiert, folgt

```
delete lamb,my:
```

```
rhoz:=lamb/my;
```

$$\frac{\text{lamb}}{\text{my}}$$

```
ev1:=1-rhoz
```

$$1 - \frac{\text{lamb}}{\text{my}}$$

```
vvalg:=x->rhoz*x
```

```
x → rhoz · x
```

```
start:=ev1:n:=7: evalg:=matrix([[start,(vvalg@@i)(start) $ i=1..n-1]])
```

$$\left(1 - \frac{\text{lamb}}{\text{my}} \quad - \frac{\text{lamb} \cdot \left(\frac{\text{lamb}}{\text{my}} - 1\right)}{\text{my}} \quad - \frac{\text{lamb}^2 \cdot \left(\frac{\text{lamb}}{\text{my}} - 1\right)}{\text{my}^2} \quad - \frac{\text{lamb}^3 \cdot \left(\frac{\text{lamb}}{\text{my}} - 1\right)}{\text{my}^3} \quad - \frac{\text{lamb}^4 \cdot \left(\frac{\text{lamb}}{\text{my}} - 1\right)}{\text{my}^4} \quad - \frac{\text{lamb}^5 \cdot \left(\frac{\text{lamb}}{\text{my}} - 1\right)}{\text{my}^5} \right)$$

Die Komponenten des Eigenvektors bilden eine geometrische Folge mit dem Faktor rhoz.

Das bekommt man am besten "von Hand" heraus.

2

Entscheidend ist, dass die unendliche Summe = 1/(1-rhoz) ist und der Startwert 1-rhoz.

```
lamb:=3: my:=4: rho:=lamb/my
```

```
lamb:=3: my:=4: rho:=lamb/my
```

$$\frac{3}{4}$$

```
vv1:=1-rho
```

$$\frac{1}{4}$$

```
vv:=x->rho*x
```

$x \rightarrow \rho \cdot x$

```
start:=vv1:n:=7:
```

```
evz:=matrix([[start,(vv@@i)(start) $ i=1..n-1]])
```

$$\left(\frac{1}{4} \quad \frac{3}{16} \quad \frac{9}{64} \quad \frac{27}{256} \quad \frac{81}{1024} \quad \frac{243}{4096} \quad \frac{729}{16384} \right)$$

```
start:=vv1*1.0:n:=7:
```

```
evv:=matrix([[start,(vv@@i)(start) $ i=1..n-1]])
```

$$(0.25 \quad 0.1875 \quad 0.140625 \quad 0.10546875 \quad 0.0791015625 \quad 0.05932617187 \quad 0.04449462891)$$

```
Uz:=matrix([[1-lamb*hz,lamb*hz,0,0,0,0,0], [my*hz,1-lamb*hz-my*hz,lamb*hz,
[0,my*hz,1-lamb*hz-my*hz,lamb*hz,0,0,0], [0,0,my*hz,1-lamb*hz-my*hz,lamb*h
[0,0,0,my*hz,1-lamb*hz-my*hz,lamb*hz,0],
[0,0,0,0,my*hz,1-lamb*hz-my*hz,lamb*hz],
[0,0,0,0,0,my*hz,1-lamb*hz-my*hz]])
```

$$\begin{pmatrix} \frac{7}{10} & \frac{3}{10} & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{5} & \frac{3}{10} & \frac{3}{10} & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{5} & \frac{3}{10} & \frac{3}{10} & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{5} & \frac{3}{10} & \frac{3}{10} & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{5} & \frac{3}{10} & \frac{3}{10} & 0 \\ 0 & 0 & 0 & 0 & \frac{2}{5} & \frac{3}{10} & \frac{3}{10} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{5} & \frac{3}{10} \end{pmatrix}$$

```
evz;evz*Uz;evv;evv*U;
```

$$\left(\frac{1}{4} \quad \frac{3}{16} \quad \frac{9}{64} \quad \frac{27}{256} \quad \frac{81}{1024} \quad \frac{243}{4096} \quad \frac{729}{16384} \right)$$

$$\left(\frac{1}{4} \quad \frac{3}{16} \quad \frac{9}{64} \quad \frac{27}{256} \quad \frac{81}{1024} \quad \frac{243}{4096} \quad \frac{5103}{163840} \right)$$

$$(0.25 \quad 0.1875 \quad 0.140625 \quad 0.10546875 \quad 0.0791015625 \quad 0.05932617187 \quad 0.04449462891)$$

$$(0.25 \quad 0.1875 \quad 0.140625 \quad 0.10546875 \quad 0.0791015625 \quad 0.05932617187 \quad 0.03114624023)$$

```
evz*Uz^2;evv*U^2
```

$$\left(\frac{1}{4} \quad \frac{3}{16} \quad \frac{9}{64} \quad \frac{27}{256} \quad \frac{81}{1024} \quad \frac{22113}{409600} \quad \frac{44469}{1638400} \right)$$

$$(0.25 \quad 0.1875 \quad 0.140625 \quad 0.10546875 \quad 0.0791015625 \quad 0.05398681641 \quad 0.02714172363)$$

(0.25 0.1875 0.140625 0.10546875 0.0791015625 0.05398681641 0.02714172363)

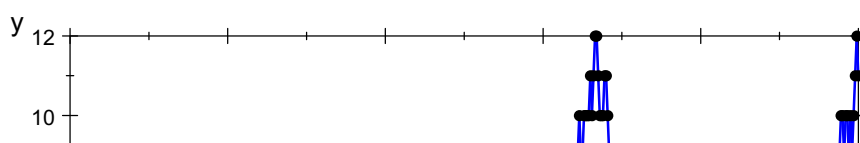
Man sieht, dass wirklich Eigenvektoren entstanden sind.

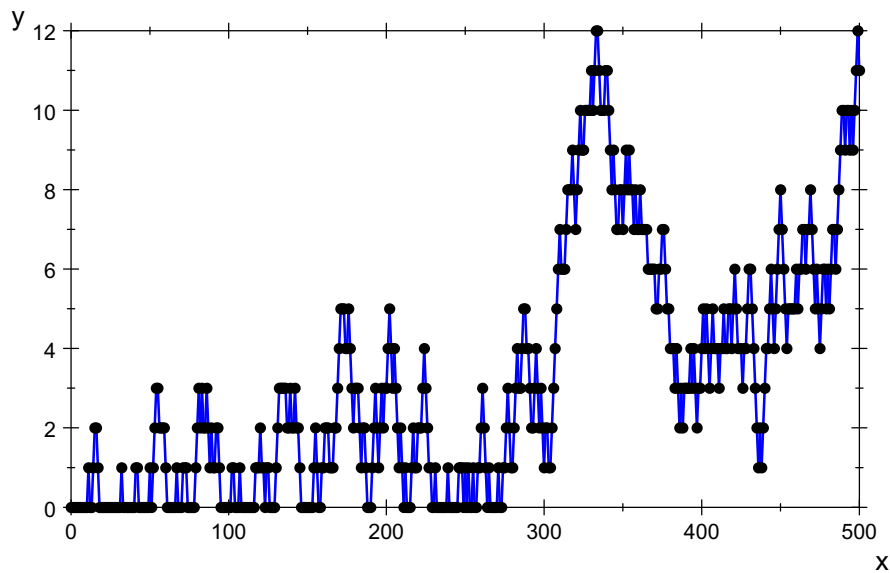
#####

Simulation

```
module("util") :
simu:=proc(lam,my,h,n)
  local rho,p,q,i,li,pkt,zz,zf,zu;
  begin
    i:=1: p:=lam*h: q:=my*h: rho:=lam/my:
    pkt:=[0,0]:li:=[pkt]:zz:=frandom(util::time());
    for i from 1 to n do
      zf:=zz();
      zu:=pkt[2]:
      if zu=0 then
        if zf <=p then pkt:=[i,zu+1];
        else pkt:=[i,zu];
        end_if:
      else /* zu <> 0 */
        if zf <=p then pkt:=[i,zu+1]
        else if zf<=q+p then pkt:=[i,zu-1];;
        else pkt:=[i,zu];
        end_if;
      end_if;
    end_if:
    li:=li.[pkt]:
  end_for;
  return(li);
end_proc:
```

```
liroh:=simu(3,4,0.1,500):
liroh:=plot::Listplot(liroh): plot(liroh):
mittlereLaenge:=float(_plus(liroh[i][2] $
i=1..500)/500);
```





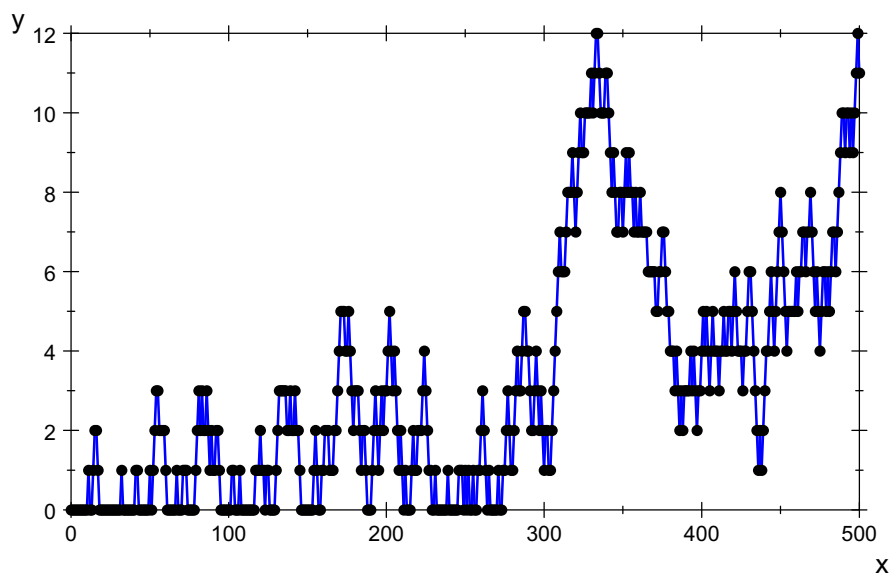
3.136

Aufheben der guten Beispiele

```
liste:=array(1..20):mittelLang:=array(1..20):
liste[9]:=liroh:
mittelLang[9]:=mittlereLaenge:
```

```
//liste[4]:
mittelLang[4];
plot(plot::Listplot(liste[4]));
//liste[9]:
mittelLang[9];
plot(plot::Listplot(liste[9]));
```

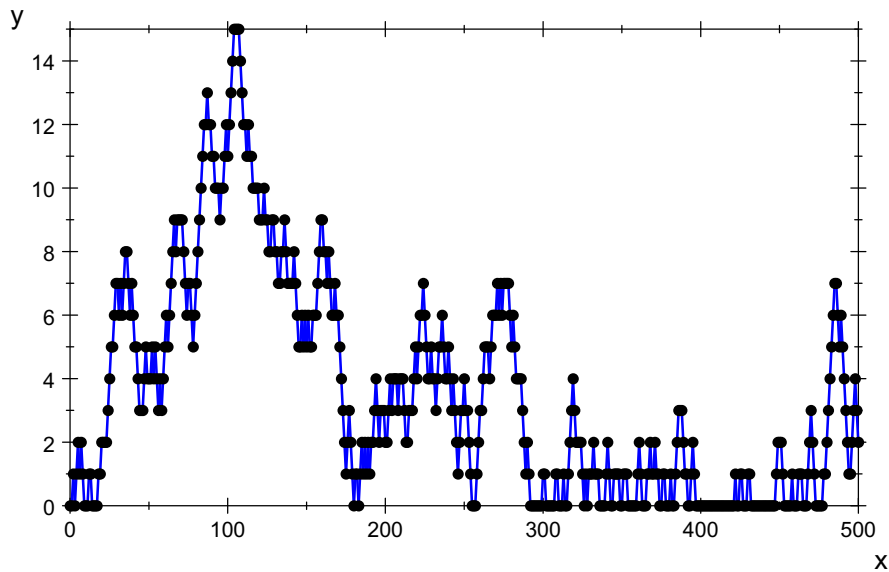
3.136



```
//liste[8]:
```

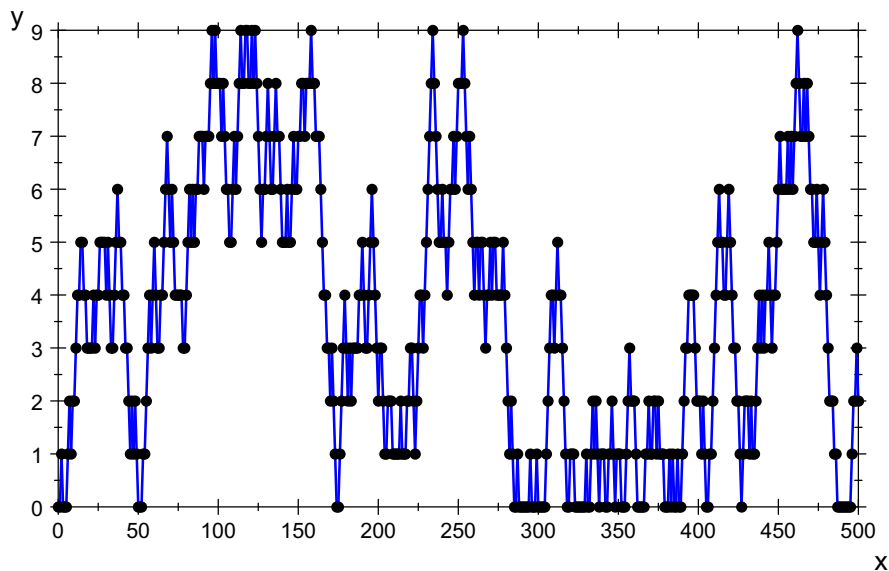
```
//liste[8]:  
mittelLang[8];  
plot(plot::Listplot(liste[8]));
```

3.56



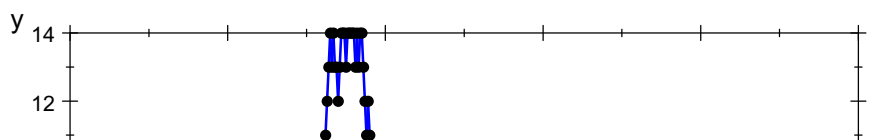
```
//liste[7]:  
mittelLang[7];  
plot(plot::Listplot(liste[7]));
```

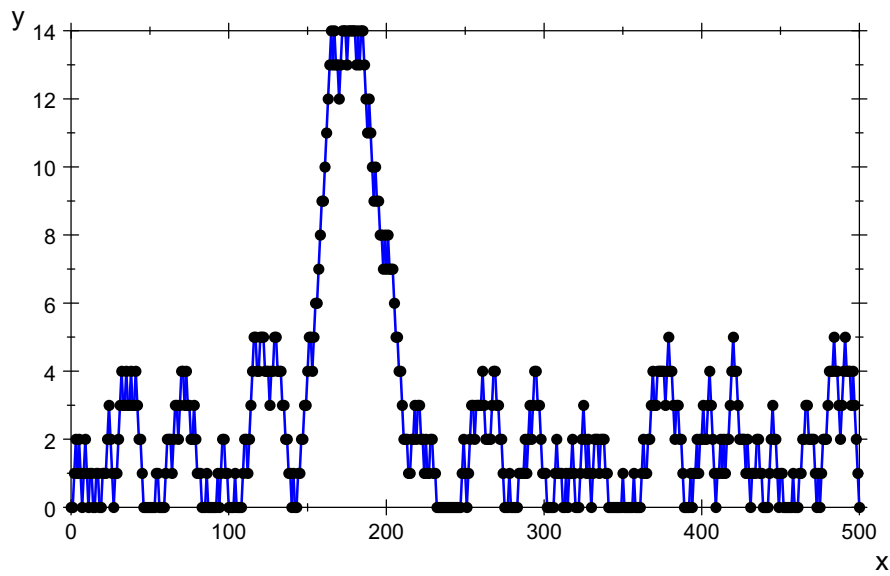
3.586



```
//liste[6]:  
mittelLang[6];  
plot(plot::Listplot(liste[6]));
```

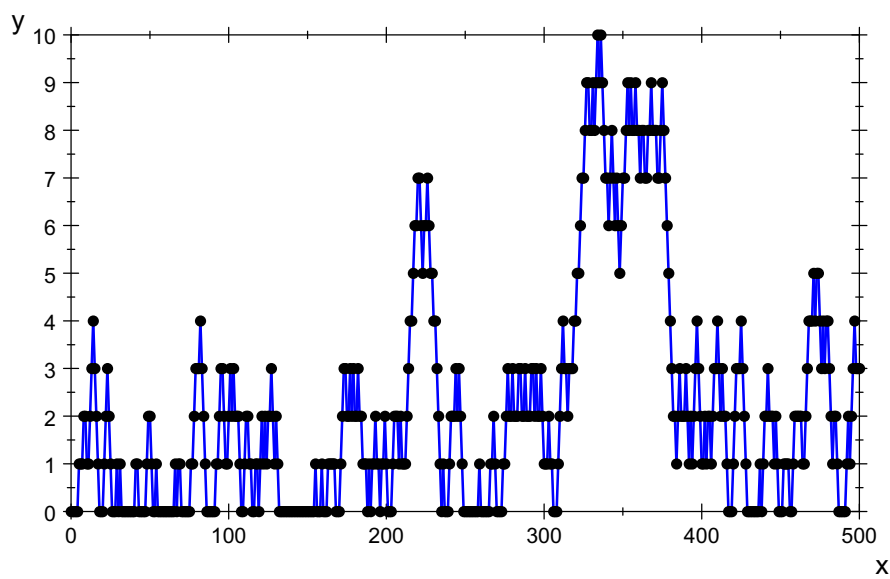
2.632





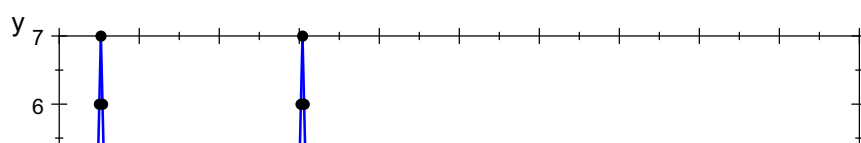
```
//liste[5]:
mittelLang[5];
plot(plot::Listplot(liste[5]));
```

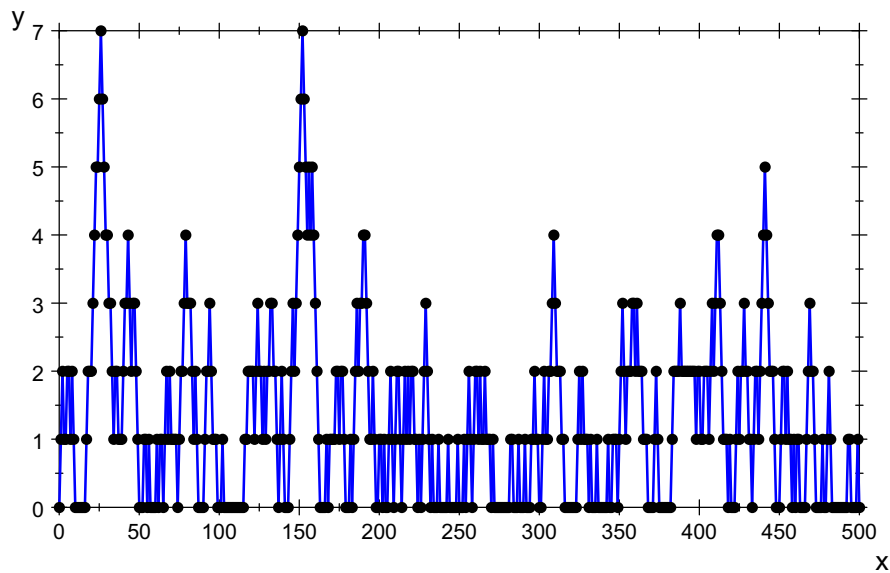
2.204



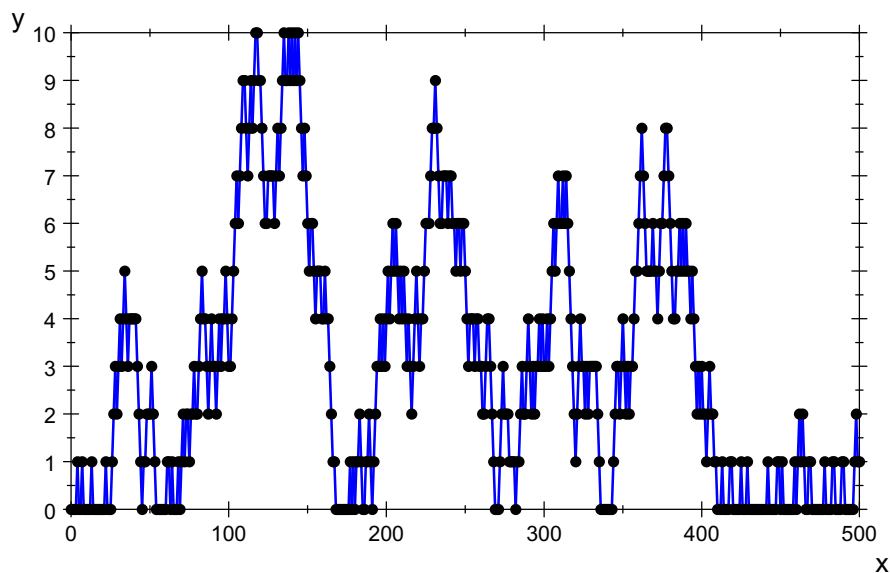
```
//liste[4]:
mittelLang[4];
plot(plot::Listplot(liste[4]));
```

1.274



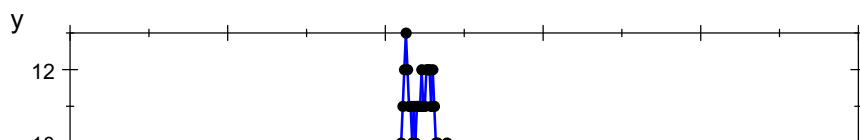


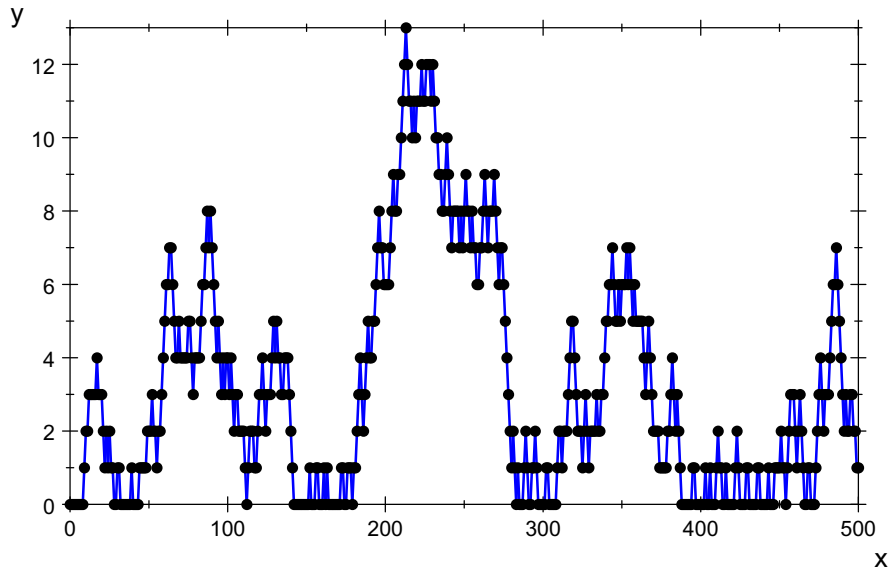
```
//liste[3]:
mittelLang[3]:
plot(plot::Listplot(liste[3]));
```



```
//liste[2]:=lirh:
mittelLang[2];
plot(plot::Listplot(liste[2]));
```

3.2





```

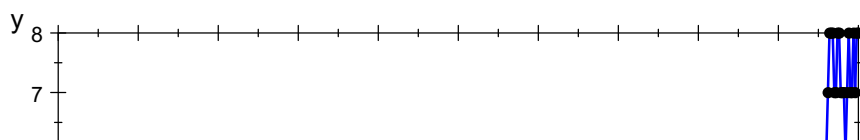
bspl := [[0, 0], [1, 1], [2, 1], [3, 2], [4, 2], [5, 1], [6, 0], [7, 0], [8, 1], [9, 2], [10, 1], [11, 1], [12, 1], [13, 0],
[14, 1], [15, 0], [16, 0], [17, 0], [18, 1], [19, 2], [20, 1], [21, 1], [22, 2], [23, 1], [24, 2], [25, 2], [26, 2], [27, 2], [28,
2], [29, 2], [30, 1], [31, 2], [32, 1], [33, 2], [34, 2], [35, 1], [36, 2], [37, 3], [38, 4], [39, 3], [40, 2], [41, 2], [42, 2],
[43, 1], [44, 0], [45, 0], [46, 0], [47, 0], [48, 1], [49, 0], [50, 0], [51, 0], [52, 0], [53, 0], [54, 1], [55, 2], [56, 2], [57,
3], [58, 2], [59, 3], [60, 4], [61, 5], [62, 6], [63, 5], [64, 5], [65, 5], [66, 4], [67, 3], [68, 2], [69, 2], [70, 1], [71, 0],
[72, 0], [73, 0], [74, 0], [75, 0], [76, 0], [77, 1], [78, 0], [79, 1], [80, 1], [81, 1], [82, 0], [83, 0], [84, 0], [85, 1], [86,
2], [87, 2], [88, 1], [89, 2], [90, 1], [91, 0], [92, 0], [93, 0], [94, 0], [95, 1], [96, 0], [97, 1], [98, 2], [99, 3], [100, 3],
[101, 4], [102, 3], [103, 2], [104, 1], [105, 0], [106, 0], [107, 0], [108, 0], [109, 0], [110, 0], [111, 0], [112, 1], [113, 1],
[114, 2], [115, 1], [116, 2], [117, 3], [118, 2], [119, 3], [120, 4], [121, 5], [122, 5], [123, 4], [124, 3], [125, 2], [126, 3],
[127, 4], [128, 5], [129, 4], [130, 3], [131, 2], [132, 2], [133, 2], [134, 2], [135, 1], [136, 0], [137, 0], [138, 0], [139, 0],
[140, 0], [141, 0], [142, 0], [143, 0], [144, 1], [145, 1], [146, 2], [147, 1], [148, 2], [149, 1], [150, 2], [151, 2], [152, 1],
[153, 0], [154, 0], [155, 1], [156, 0], [157, 1], [158, 0], [159, 0], [160, 0], [161, 0], [162, 0], [163, 0], [164, 1], [165, 1],
[166, 1], [167, 2], [168, 1], [169, 2], [170, 1], [171, 2], [172, 2], [173, 1], [174, 1], [175, 1], [176, 2], [177, 2], [178, 1],
[179, 0], [180, 0], [181, 1], [182, 1], [183, 0], [184, 0], [185, 0], [186, 1], [187, 0], [188, 0], [189, 1], [190, 2], [191, 1],
[192, 0], [193, 1], [194, 1], [195, 2], [196, 1], [197, 2], [198, 2], [199, 3], [200, 4], [201, 3], [202, 2], [203, 2], [204, 2],
[205, 3], [206, 4], [207, 5], [208, 5], [209, 4], [210, 3], [211, 3], [212, 3], [213, 4], [214, 3], [215, 4], [216, 3], [217, 3],
[218, 3], [219, 2], [220, 1], [221, 0], [222, 1], [223, 2], [224, 3], [225, 4], [226, 5], [227, 4], [228, 3], [229, 2], [230, 1],
[231, 2], [232, 3], [233, 2], [234, 1], [235, 0], [236, 0], [237, 0], [238, 0], [239, 0], [240, 0], [241, 1], [242, 2], [243, 1],
[244, 2], [245, 2], [246, 3], [247, 3], [248, 3], [249, 2], [250, 1], [251, 2], [252, 2], [253, 1], [254, 2], [255, 2], [256, 1],
[257, 0], [258, 0], [259, 1], [260, 1], [261, 2], [262, 1], [263, 0], [264, 1], [265, 1], [266, 1], [267, 2], [268, 3], [269, 4],
[270, 3], [271, 4], [272, 3], [273, 2], [274, 1], [275, 0], [276, 1], [277, 1], [278, 2], [279, 1], [280, 2], [281, 1], [282, 0],
[283, 0], [284, 0], [285, 0], [286, 1], [287, 1], [288, 2], [289, 2], [290, 2], [291, 2], [292, 1], [293, 0], [294, 0], [295, 0],
[296, 0], [297, 0], [298, 1], [299, 0], [300, 0], [301, 0], [302, 1], [303, 2], [304, 1], [305, 0], [306, 0], [307, 1], [308, 0],
[309, 1], [310, 1], [311, 1], [312, 1], [313, 2], [314, 3], [315, 3], [316, 4], [317, 3], [318, 3], [319, 4], [320, 4], [321, 3],
[322, 4], [323, 5], [324, 4], [325, 3], [326, 3], [327, 2], [328, 1], [329, 0], [330, 0], [331, 0], [332, 0], [333, 1], [334, 1],
[335, 0], [336, 0], [337, 0], [338, 0], [339, 1], [340, 1], [341, 1], [342, 0], [343, 0], [344, 0], [345, 1], [346, 0], [347, 0],
[348, 0], [349, 0], [350, 0], [351, 1], [352, 0], [353, 0], [354, 0], [355, 0], [356, 0], [357, 0], [358, 0], [359, 1], [360, 0],
[361, 0], [362, 0], [363, 1], [364, 2], [365, 3], [366, 4], [367, 4], [368, 5], [369, 4], [370, 3], [371, 3], [372, 2], [373, 1],
[374, 0], [375, 1], [376, 2], [377, 2], [378, 1], [379, 1], [380, 0], [381, 0], [382, 0], [383, 0], [384, 0], [385, 0], [386, 1],
[387, 0], [388, 0], [389, 0], [390, 0], [391, 1], [392, 0], [393, 0], [394, 0], [395, 0], [396, 0], [397, 0], [398, 0], [399, 0],
[400, 0], [401, 1], [402, 0], [403, 1], [404, 2], [405, 2], [406, 2], [407, 1], [408, 0], [409, 0], [410, 0], [411, 0], [412, 0],
[413, 0], [414, 0], [415, 0], [416, 1], [417, 2], [418, 1], [419, 0], [420, 0], [421, 1], [422, 1], [423, 0], [424, 0], [425, 0],
[426, 0], [427, 1], [428, 0], [429, 0], [430, 0], [431, 0], [432, 1], [433, 2], [434, 2], [435, 3], [436, 2], [437, 1], [438, 2],
[439, 1], [440, 1], [441, 1], [442, 1], [443, 1], [444, 1], [445, 1], [446, 2], [447, 2], [448, 2], [449, 1], [450, 0], [451, 1],
[452, 1], [453, 0], [454, 0], [455, 0], [456, 0], [457, 0], [458, 0], [459, 0], [460, 0], [461, 1], [462, 2], [463, 3], [464, 4],
[465, 3], [466, 4], [467, 3], [468, 2], [469, 3], [470, 3], [471, 4], [472, 5], [473, 4], [474, 3], [475, 2], [476, 2], [477, 3],
[478, 4], [479, 5], [480, 6], [481, 7], [482, 8], [483, 8], [484, 8], [485, 7], [486, 7], [487, 8], [488, 8], [489, 7], [490, 7],
[491, 7], [492, 6], [493, 7], [494, 8], [495, 7], [496, 7], [497, 8], [498, 7], [499, 8], [500, 8]]:

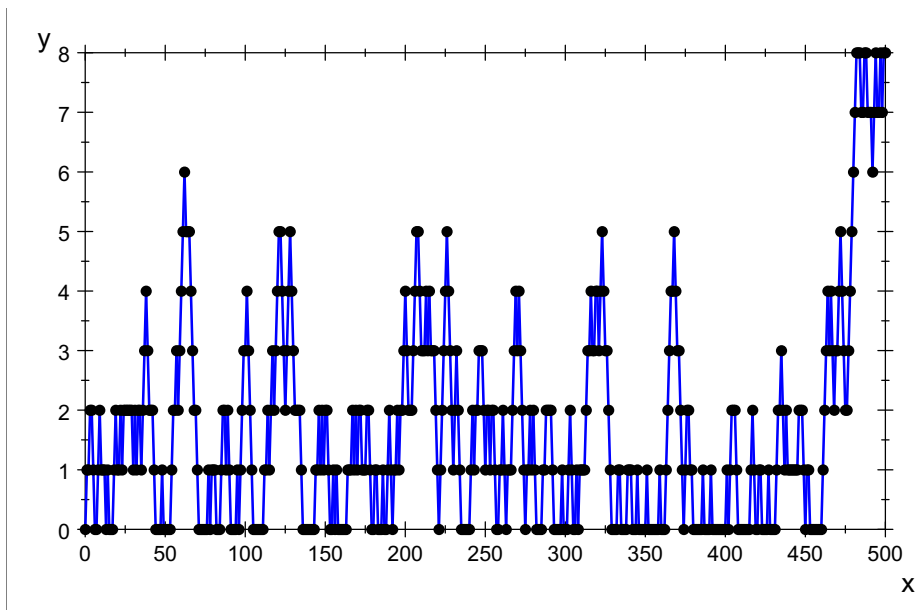
```

```

bsplp:=lip:
plot(bsplp):

```





```

liste[1]:=bspl:
mittelLang[1]:=float(_plus(bspl[i][2] $ i=1..500)/500)
1.622

```

[