```
> with(plots):Digits:=100:interface(displayprecision=10):with
  (linalg):


> N:=6;
  KK:=2;
  ### Change of notation: Now the indices K[i] are in order not
  alpha's
  # vector U shows  if the branch is up (1) or down (0)

  for j from 1 to N do
  U[j]:=0;
  od;
  alpha[1]:=0.4;K[1]:=1;U[K[1]]:=1;
  alpha[2]:=0.5;K[2]:=6;U[K[2]]:=0;#
  alpha[3]:=0.7;K[3]:=4;U[K[3]]:=0;  #
  alpha[4]:=0.7;K[4]:=5;U[K[4]]:=1;
  i:='i':
  beta:=N-KK+sum(alpha[i],i=1..KK);i:='i':
  delta1:=(xw,yw)-> piecewise(xw<=yw,0,1);
  for j from 1 to N do
  b[j]:=(j-1-sum((1-alpha[i])*delta1( j,K[i]) ,i=1..KK))/beta;
   od;  i:='i':
  b[N+1]:=1;
  for j from 1 to N do
  a[j]:=(j-1-sum((1-alpha[i])*delta1( j,K[i]-U[j]) ,i=1..KK));
   od;


  for j from 1 to KK do
  if U[K[j]]=0 then   c[j]:=b[K[j]+1];
               else  c[j]:=b[K[j]];fi;
   print(`c[`,j,`] = `,c[j]) ;
  od:

>
  maa:=a[2]-a[1]:# maximum a[i+1]-a[i]
  for i from 3 to N do
  if (a[i]-a[i-1])>maa then maa:=(a[i]-a[i-1]) fi
  od;#
  maa;
  beta_max:=evalf(1+(a[N]-a[1])/maa);
> if beta> beta_max then print("ERROR") fi;


>
  uint_of_x:=x->piecewise(x<b[2],1,# This function needs additions
```

```
by hand for
                                        # N>9 . Automathic procedure
causes plotting problems
                                        # but is used in other
programs
                            x<b[3],2,
                            x<b[4],3,
                            x<b[5],4,
                            x<b[6],5,
                            x<b[7],6,
                            x<b[8],7,
                            x<b[9],8,
                                  9);
int_of_x:=x->piecewise(x<=b[2],1,# This function needs additions
by hand for
                                        # N>9 . Automathic procedure
causes plotting problems
                                        # but is used in other
programs
                            x<=b[3],2,
                            x<=b[4],3,
                            x<=b[5],4,
                            x<=b[6],5,
                            x<=b[7],6,
                            x<=b[8],7,
                            x<=b[9],8,
                                  9);
x:='x':
uT:=x->beta*x-a[uint_of_x(x)];
T:=x->beta*x-a[int_of_x(x)];
for j from 1 to KK do
if U[K[j]]=0 then    Tc:=T(c[j]);
             else    Tc:=uT(c[j])fi;
print(`T(c[`,j,`])  =`,  Tc)
od;


plot([uT(x),x,0,1,1-alpha[1],1-alpha[2]],x=0..1,thickness=[2,1,
1,1,1,1,1]);
plot([T(x),x,0,1,alpha[1],alpha[2]],x=0..1,thickness=[2,1,1,1,1,
1,1]);
```

$$N := 6$$

$$KK := 2$$

$$U_1 := 0$$

$$U_2 := 0$$

$$U_3 := 0$$

$$U_4 := 0$$

$$U_5 := 0$$

$$U_6 := 0$$

$$\alpha_1 := 0.4000000000$$

$$K_1 := 1$$

$$U_1 := 1$$

$$\alpha_2 := 0.5000000000$$

$$K_2 := 6$$

$$U_6 := 0$$

$$\alpha_3 := 0.7000000000$$

$$K_3 := 4$$

$$U_4 := 0$$

$$\alpha_4 := 0.7000000000$$

$$K_4 := 5$$

$$U_5 := 1$$

$$\beta := 4.9000000000$$

$$\delta 1 := (xw, yw) \rightarrow piecewise(xw \leq yw, 0, 1)$$

$$b_1 := 0.0000000000$$

$$b_2 := 0.0816326531$$

$$b_3 := 0.2857142857$$

$$b_4 := 0.4897959184$$

$$b_5 := 0.6938775510$$

$$b_6 := 0.8979591837$$

$$b_7 := 1$$

$$a_1 := -0.6000000000$$

$$a_2 := 0.4000000000$$

$$a_3 := 1.4000000000$$

$$a_4 := 2.4000000000$$

$$a_5 := 3.4000000000$$

$$a_6 := 4.4000000000$$

$$c[, 1,] = , 0.0000000000$$

$$c[,2,] = ,1$$

1.0000000000

$$\beta max := 6.0000000000$$

$$uint\_of\_x := x \rightarrow piecewise\left(x < b_2, 1, x < b_3, 2, x < b_4, 3, x < b_5, 4, x < b_6, 5, x < b_7, 6, x < b_8, 7, x < b_9, 8, 9\right)$$

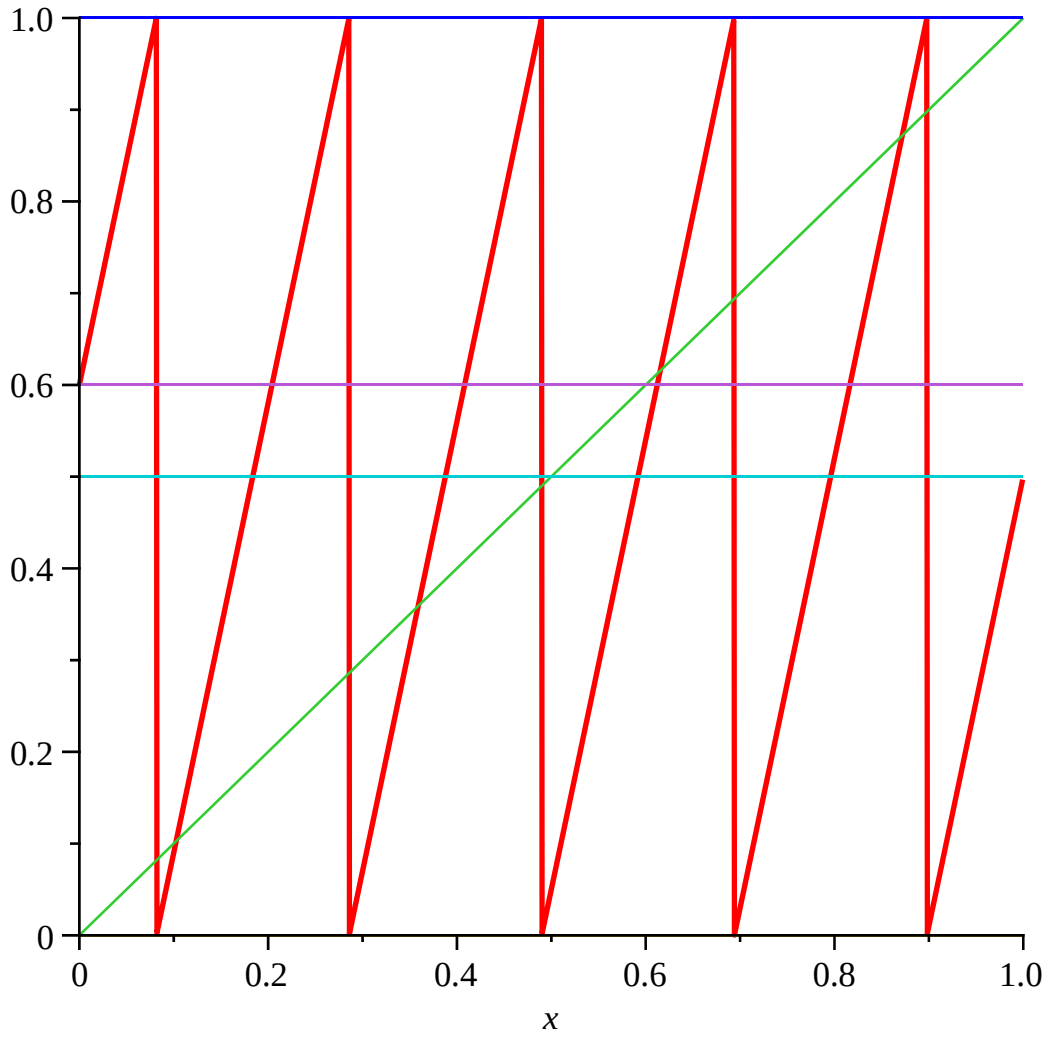
$$int\_of\_x := x \rightarrow piecewise\left(x \leq b_2, 1, x \leq b_3, 2, x \leq b_4, 3, x \leq b_5, 4, x \leq b_6, 5, x \leq b_7, 6, x \leq b_8, 7, x \leq b_9, 8, 9\right)$$

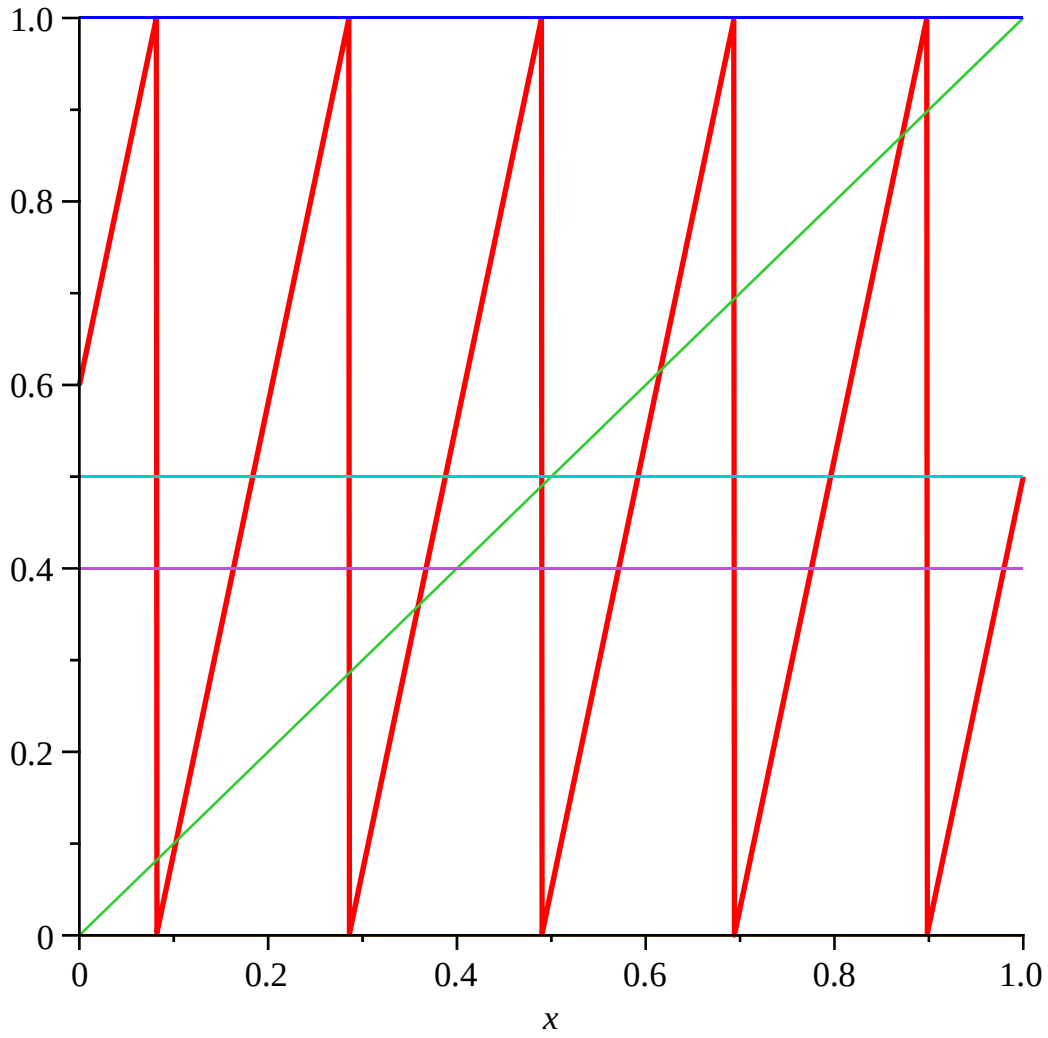
$$uT := x \rightarrow \beta x - a_{uint\_of\_x(x)}$$

$$T := x \rightarrow \beta x - a_{int\_of\_x(x)}$$

$$T(c[,1,]) =, 0.6000000000$$

$$T(c[,2,]) =, 0.5000000000$$






```
>
> ud:=vector(50):Digits:=100;NN:=50;
  d:=vector(50):
  xx:=evalf(rand()/10^12);
  xxt:=xx:
  for i from 1 to NN do
  ud[i]:=a[uint_of_x(xxt)];
  xxt:=uT(xxt);
  od:
  xxt:=xx:
  for i from 1 to NN do
  d[i]:=a[int_of_x(xxt)];
  xxt:=T(xxt);
  od:
  print(ud);
  uls_it_x:=evalf(sum(ud[j1]/beta^j1,j1=1..NN));
  print(d);
  ls_it_x:=evalf(sum(d[j1]/beta^j1,j1=1..NN));
  terr:=xx-uls_it_x;
  err:=xx-ls_it_x;
```

$$Digits := 100$$

$$NN := 50$$

$$xx := 0.3957188605$$

[1.4000000000, 2.4000000000, 0.4000000000, 3.4000000000, 1.4000000000, 3.4000000000,
0.4000000000, 0.4000000000, -0.6000000000, 2.4000000000, 4.4000000000,
1.4000000000, 3.4000000000, 2.4000000000, 2.4000000000, 1.4000000000,
1.4000000000, -0.6000000000, 3.4000000000, 4.4000000000, 1.4000000000,
0.4000000000, 3.4000000000, 4.4000000000, 0.4000000000, 4.4000000000,
1.4000000000, 3.4000000000, 0.4000000000, 4.4000000000, 1.4000000000,
3.4000000000, 1.4000000000, 4.4000000000, 0.4000000000, -0.6000000000,
2.4000000000, 3.4000000000, 4.4000000000, 0.4000000000, 4.4000000000,
0.4000000000, 4.4000000000, -0.6000000000, 3.4000000000, 0.4000000000,
2.4000000000, 3.4000000000, 0.4000000000, 2.4000000000]

$$uIs\_it\_x := 0.3957188605$$

[1.4000000000, 2.4000000000, 0.4000000000, 3.4000000000, 1.4000000000, 3.4000000000,
0.4000000000, 0.4000000000, -0.6000000000, 2.4000000000, 4.4000000000,
1.4000000000, 3.4000000000, 2.4000000000, 2.4000000000, 1.4000000000,
1.4000000000, -0.6000000000, 3.4000000000, 4.4000000000, 1.4000000000,
0.4000000000, 3.4000000000, 4.4000000000, 0.4000000000, 4.4000000000,
1.4000000000, 3.4000000000, 0.4000000000, 4.4000000000, 1.4000000000,
3.4000000000, 1.4000000000, 4.4000000000, 0.4000000000, -0.6000000000,
2.4000000000, 3.4000000000, 4.4000000000, 0.4000000000, 4.4000000000,
0.4000000000, 4.4000000000, -0.6000000000, 3.4000000000, 0.4000000000,
2.4000000000, 3.4000000000, 0.4000000000, 2.4000000000]

$$Is\_it\_x := 0.3957188605$$

$$terr := 2.106709284\ 10^{-35}$$

$$err := 2.106709284\ 10^{-35} \quad \textbf{(1)}$$

```
>
> NN:=50; chi:=(x1,x2,t)->piecewise(t<x1,0,t<=x2,1,0);
  uchi:=(x1,x2,t)->piecewise(t<x1,0,t<x2,1,0);

  #Expansion of c1, c2 ... and all the S's

  for i from 1 to KK do
  xxt:=c[i]; upflag:=U[K[i]];
       for n from 1 to NN+1 do
        if upflag=1 then intx:=uint_of_x(xxt) else intx:=int_of_x
  (xxt) fi;
          dc[i,n]:=a[intx];
          ic[i,n]:=intx-1;
         if upflag=0 then
              for ii from 1 to KK do
                   if xxt>c[ii] then cc[i,ii,n]:=1 else cc[i,ii,n]
  :=0 fi;
```

```
                od;
                        else
                for ii from 1 to KK do
                     if xxt<c[ii] then cc[i,ii,n]:=1 else cc[i,ii,n]
:=0 fi;
                od;
        fi;
        valc[i,n]:=xxt;
        if upflag=0 then xxt:=T(xxt) else xxt:=uT(xxt)fi;
        od:
Is_it_x:=sum(dc[i,j1]/beta^j1,j1=1..NN);
S[i]:=sum(ic[i,j1+1]/beta^(j1+1),j1=1..NN);
od;
for i from 1 to KK do
for j from 1 to KK do
SS[i,j]:=sum(cc[i,j,j1+1]/beta^(j1+1),j1=1..NN);

print(`SS[`,i,j,`] =`,SS[i,j]);
od;od;
```

$$NN := 50$$

$$\chi := (x1, x2, t) \rightarrow piecewise(t < x1, 0, t \leq x2, 1, 0)$$

$$uchi := (x1, x2, t) \rightarrow piecewise(t < x1, 0, t < x2, 1, 0)$$

$$xxt := 0.0000000000$$

$$upflag := 1$$

$$Is\_it\_x := -1.462141812\ 10^{-35}$$

$$S_1 := 0.1538461538$$

$$xxt := 1$$

$$upflag := 0$$

$$Is\_it\_x := 1.0000000000$$

$$S_2 := 0.1334379906$$

$$SS[, 1, 1,] =, 0.0000000000$$

$$SS[, 1, 2,] =, 0.0523286238$$

$$SS[, 2, 1,] =, 0.0523286238$$

$$SS[, 2, 2,] =, 0.0000000000$$

```
>
MM:=matrix(KK,KK,[]):
for i from 1 to KK do
for j from 1 to KK do

MM[j,i]:=-SS[i,j];
od;od;
print(`MM = `,MM);
```

```
print(`1/beta =`, 1/beta);

print(`eigenvalues MM =`, eigenvalues(MM));

ve:=vector(KK,[]):
for i from 1 to KK do
ve[i]:=1/beta;

MM[i,i]:=MM[i,i]+1/beta;
od:

print(MM);
print(ve);



DD:=linsolve(MM,ve);
```

$$MM = , \begin{bmatrix} -0.0000000000 & -0.0523286238 \\ -0.0523286238 & -0.0000000000 \end{bmatrix}$$

$$1/beta =, 0.2040816327$$

$$eigenvalues\ MM =, -0.0523286238, 0.0523286238$$

$$\begin{bmatrix} 0.2040816327 & -0.0523286238 \\ -0.0523286238 & 0.2040816327 \end{bmatrix}$$

$$\begin{bmatrix} 0.2040816327 & 0.2040816327 \end{bmatrix}$$

$$DD := \begin{bmatrix} 1.3448275862 & 1.3448275862 \end{bmatrix} \quad (2)$$

```
>
density:=proc(t) local j, den;
                  den:=1/beta;
                for j from 1 to KK do
                if U[K[j]]=0 then
                  den:=den+ DD[j]*sum((chi(0,valc[j,i1+1],t))
/beta^(i1+1),i1=1..50)
                                  else
                  den:=den+ DD[j]*sum((uchi(valc[j,i1+1],1,t))
/beta^(i1+1),i1=1..50)
                fi;
                od;
                return den;
          end proc;
P_density:=t->1+ sum((chi(0,valc[2,i1+1],t))/beta^(i1),i1=1.
.50)
```

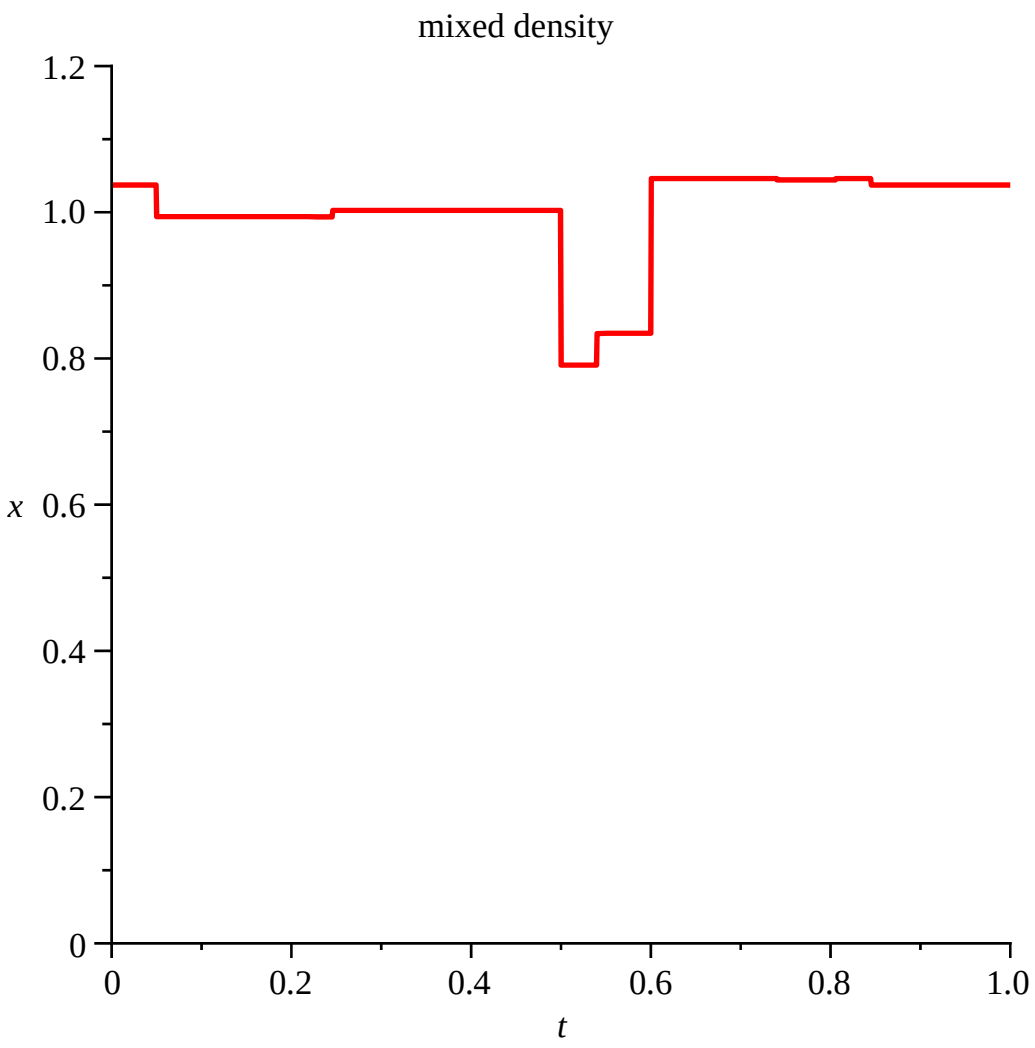
```
    - sum( ( chi ( 0, val c[ 1, i 1+1] , t ) ) / bet a^( i 1) , i 1=1. . 50) ;

    #Nor mal i zi ng  f act or
    t NC: =1/ bet a:
    f or  j  f r om 1  t o  KK  do
     t NC: =t NC+t DD[ j ] * sum( ( 1- t val c[ j , i 1+1] ) / bet a^( i 1+1) , i 1=1. . 50)
     od:
    NC: =1/ bet a:
    f or  j  f r om 1  t o  KK  do
    i f  U[ K[ j ] ] =0  t hen
     NC: =NC+DD[ j ] * sum( ( val c[ j , i 1+1] ) / bet a^( i 1+1) , i 1=1. . 50)
    el se
    NC: =NC+DD[ j ] * sum( ( 1- val c[ j , i 1+1] ) / bet a^( i 1+1) , i 1=1. . 50)
    f i ;
     od:
    P_NC: =  1+  sum( ( val c[ 2, i 1+1] ) / bet a^( i 1+1) , i 1=1. . 50)
    - sum( val c[ 1, i 1+1] / bet a^( i 1+1) , i 1=1. . 50) ;

    pr i nt ( `NC = `, NC) ;

    pl ot ( [ ( 1/ NC) * densi t y( t ) ] , t =0. . 1- 0. 00000000001, x=0. . 1. 2, t i t l e=
    "mi xed  densi t y", t hi ckness=2) ;
    pl ot ( [ ( 1/ P_NC) * P_densi t y( t ) ] , t =0. . 1- 0. 00000000001, x=0. . 1. 2,
    t i t l e="Par r y' s  densi t y", t hi ckness=2) ;
```

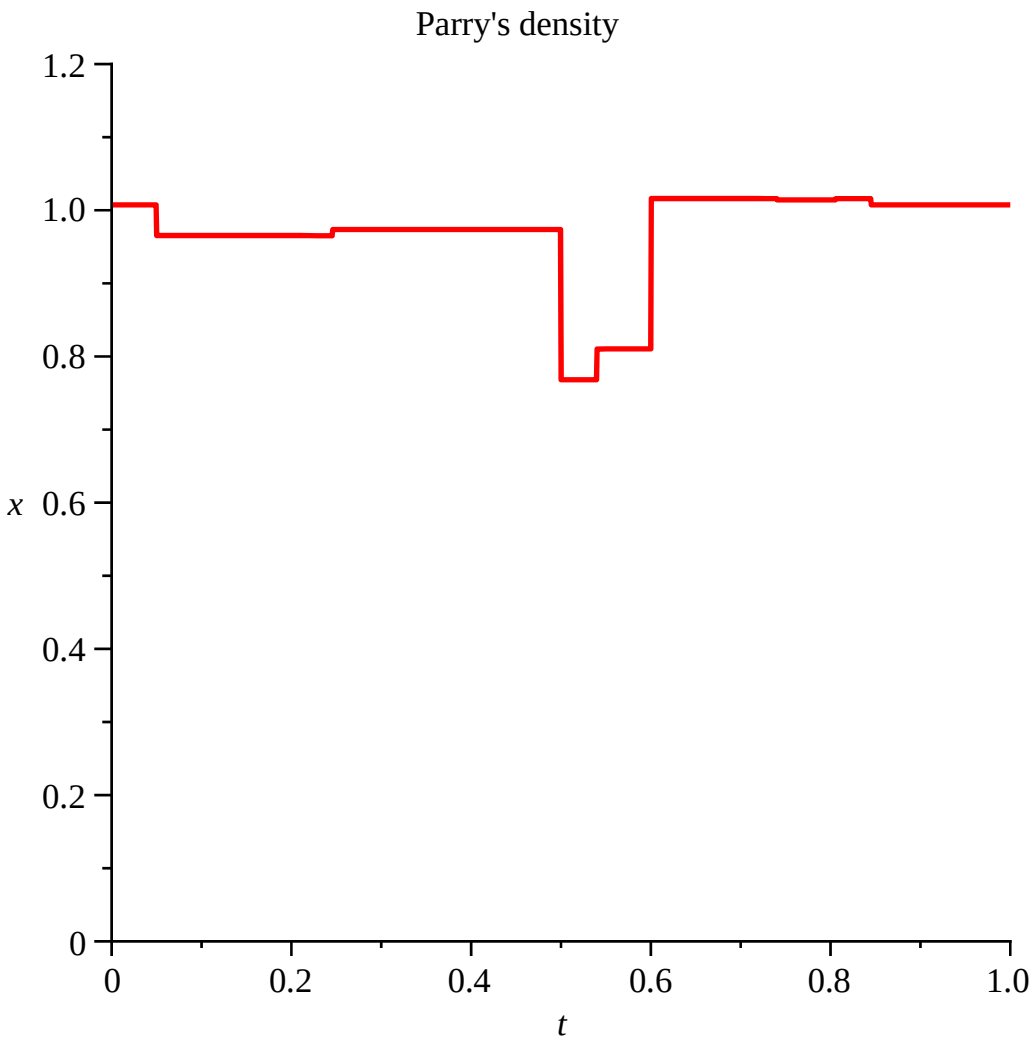
```
density := proc(t)
    local j, den;
    den := 1 / beta;
    for j to KK do
        if U[K[j]] = 0 then
            den := den + DD[j] * (sum(chi(0, valc[j, i1 + 1], t) / beta^(i1 + 1), i1 = 1 ..50))
        else
            den := den + DD[j] * (sum(uchi(valc[j, i1 + 1], 1, t) / beta^(i1 + 1), i1 = 1 ..50))
        end if
    end do;
    return den
end proc
```

$$P\_density := t \rightarrow 1 + \sum_{i1=1}^{50} \frac{\chi(0, valc_{2,\, i1+1}, t)}{\beta^{i1}} - \left(\sum_{i1=1}^{50} \frac{\chi(0, valc_{1,\, i1+1}, t)}{\beta^{i1}} \right)$$

$$P\_NC := 0.9926672798$$

$$NC = , 0.2645933650$$


mixed density
1.2
1.0
0.8
x 0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1.0
t




```
> #check density Parry's
  #preimages
  for j6 from 1 to KK-1 do
  y[j6]:=alpha[j6]+(alpha[j6+1]-alpha[j6])*rand()/10^12;
  od:
  y[0]:=alpha[1]*rand()/10^12:
  y[KK]:=alpha[KK]+(1-alpha[KK])*rand()/10^12;
  for j6 from 0 to KK do
  for i3 from 1 to N do
  pre[i3]:=(y[j6]+a[i3])/beta;
  od;
  #plot([T(t),0,1,y[j6],tT(tc[1]),tT(tc[2])],t=0..1,color=[red,
  black,black,green,yellow,yellow]);
  su:=0:
  for i3 from 1 to N do
  if (pre[i3]>=b[i3] and pre[i3]<=b[i3+1]) then
   su:=su+P_density(pre[i3])/beta;
  print(i3);
  fi;
  od;
  err[j6]:=P_density(y[j6])-su;
  od;
```

```
for j6 from 0 to KK do
print(`y =`,y[j6]);
print(`err[`,j6,`]=`,err[j6]);
od;
>
#check density greedy
#preimages
for j6 from 1 to KK-1 do
y[j6]:=alpha[j6]+(alpha[j6+1]-alpha[j6])*rand()/10^12;
od:
y[0]:=alpha[1]*rand()/10^12:
y[KK]:=alpha[KK]+(1-alpha[KK])*rand()/10^12;
for j6 from 0 to KK do
for i3 from 1 to N do
pre[i3]:=(y[j6]+a[i3])/beta;
od;
#plot([T(t),0,1,y[j6],tT(tc[1]),tT(tc[2])],t=0..1,color=[red,
black,black,green,yellow,yellow]);
su:=0:
for i3 from 1 to N do
if (pre[i3]>=b[i3] and pre[i3]<=b[i3+1]) then
 su:=su+density(pre[i3])/beta;
print(i3);
fi;
od;
err[j6]:=density(y[j6])-su;
od;

for j6 from 0 to KK do
print(`y =`,y[j6]);
print(`err[`,j6,`]=`,err[j6]);
od;
```

$$y_2 := 0.9000937422$$

$$su := 0$$

2

3

4

5

6

$$err_0 := -2.819758411\ 10^{-36}$$

$$su := 0$$

2

3

4

5

6

$err_1 := -2.819758411\ 10^{-36}$

$su := 0$

1

2

3

4

5

$err_2 := 3.489813850\ 10^{-36}$

$y =, 0.0089696682$

$err[, 0,]=, -2.819758411\ 10^{-36}$

$y =, 0.4193139816$

$err[, 1,]=, -2.819758411\ 10^{-36}$

$y =, 0.9000937422$

$err[, 2,]=, 3.489813850\ 10^{-36}$

$y_2 := 0.7061431429$

$su := 0$

2

3

4

5

6

$err_0 := -7.738956935\ 10^{-37}$

$su := 0$

2

3

4

5

6

$err_1 := -7.738956935\ 10^{-37}$

$su := 0$

1

2

3

4

5

$$err_2 := 9.577954972\ 10^{-37}$$

$$y =, 0.3370490738$$

$$err[, 0,]=, -7.738956935\ 10^{-37}$$

$$y =, 0.4427552057$$

$$err[, 1,]=, -7.738956935\ 10^{-37}$$

$$y =, 0.7061431429$$

$$err[, 2,]=, 9.577954972\ 10^{-37}$$

```
>
#check density greedy
#preimages
y:=0.50100400000000;
for i2 from 1 to N do
pre[i2]:=(y+a[i2])/beta;
od;
plot([T(t),0,1,y,T(c[1]),T(c[2])],t=0..1,color=[red,black,black,
green,yellow,yellow]);
su:=0:
for i2 from 1 to N do
if (pre[i2]>=b[i2] and pre[i2]<=b[i2+1]) then
 su:=su+density(pre[i2])/beta;
print(i2);
fi;
od;
err2:=density(y)-su;
```

$$y := 0.5010040000$$

$$pre_1 := -0.0202032653$$

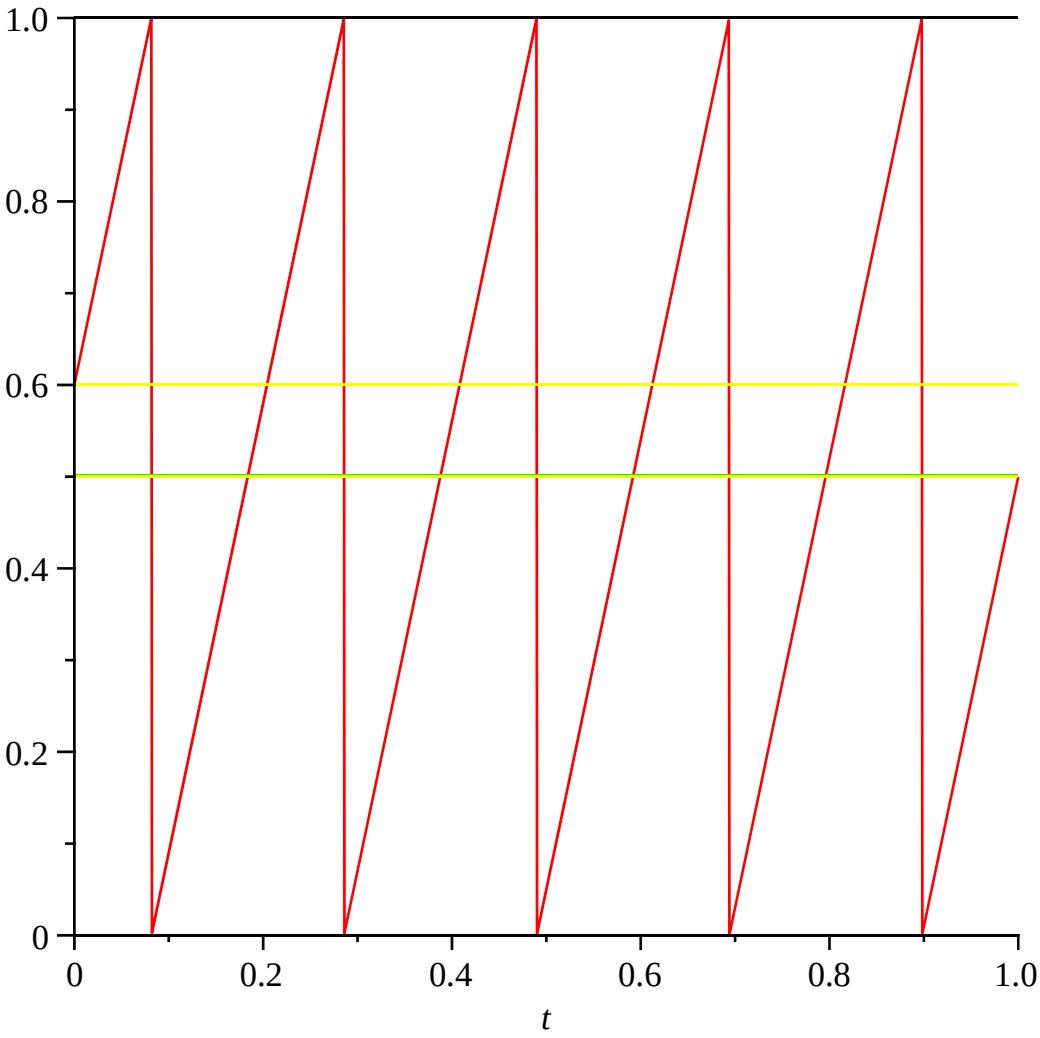
$$pre_2 := 0.1838783673$$

$$pre_3 := 0.3879600000$$

$$pre_4 := 0.5920416327$$

$$pre_5 := 0.7961232653$$

$$pre_6 := 1.0002048980$$



2

3

4

5

$err2 := 9.577954972\ 10^{-37}$

>

>