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


> N:=6;
  KK:=4;
  ### 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]:=3;U[K[2]]:=1;#
  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]);
```

$$N := 6$$

$$KK := 4$$

$$U_1 := 0$$

$$U_2 := 0$$

$$U_3 := 0$$

$$U_4 := 0$$

$$U_5 := 0$$

$$U_6 := 0$$

$$\alpha_1 := 0.4000000000$$

$$K_1 := 1$$

$$U_1 := 1$$

$$\alpha_2 := 0.5000000000$$

$$K_2 := 3$$

$$U_3 := 1$$

$$\alpha_3 := 0.7000000000$$

$$K_3 := 4$$

$$U_4 := 0$$

$$\alpha_4 := 0.7000000000$$

$$K_4 := 5$$

$$U_5 := 1$$

$$\beta := 4.3000000000$$

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

$$b_1 := 0.0000000000$$

$$b_2 := 0.0930232558$$

$$b_3 := 0.3255813953$$

$$b_4 := 0.4418604651$$

$$b_5 := 0.6046511628$$

$$b_6 := 0.7674418605$$

$$b_7 := 1$$

$$a_1 := -0.6000000000$$

$$a_2 := 0.4000000000$$

$$a_3 := 0.9000000000$$

$$a_4 := 1.9000000000$$

$$a_5 := 2.3000000000$$

$$a_6 := 3.3000000000$$

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

$c[, 2,] = , 0.3255813953$

$c[, 3,] = , 0.6046511628$

$c[, 4,] = , 0.6046511628$

1.0000000000

$\beta max := 4.9000000000$

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

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

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

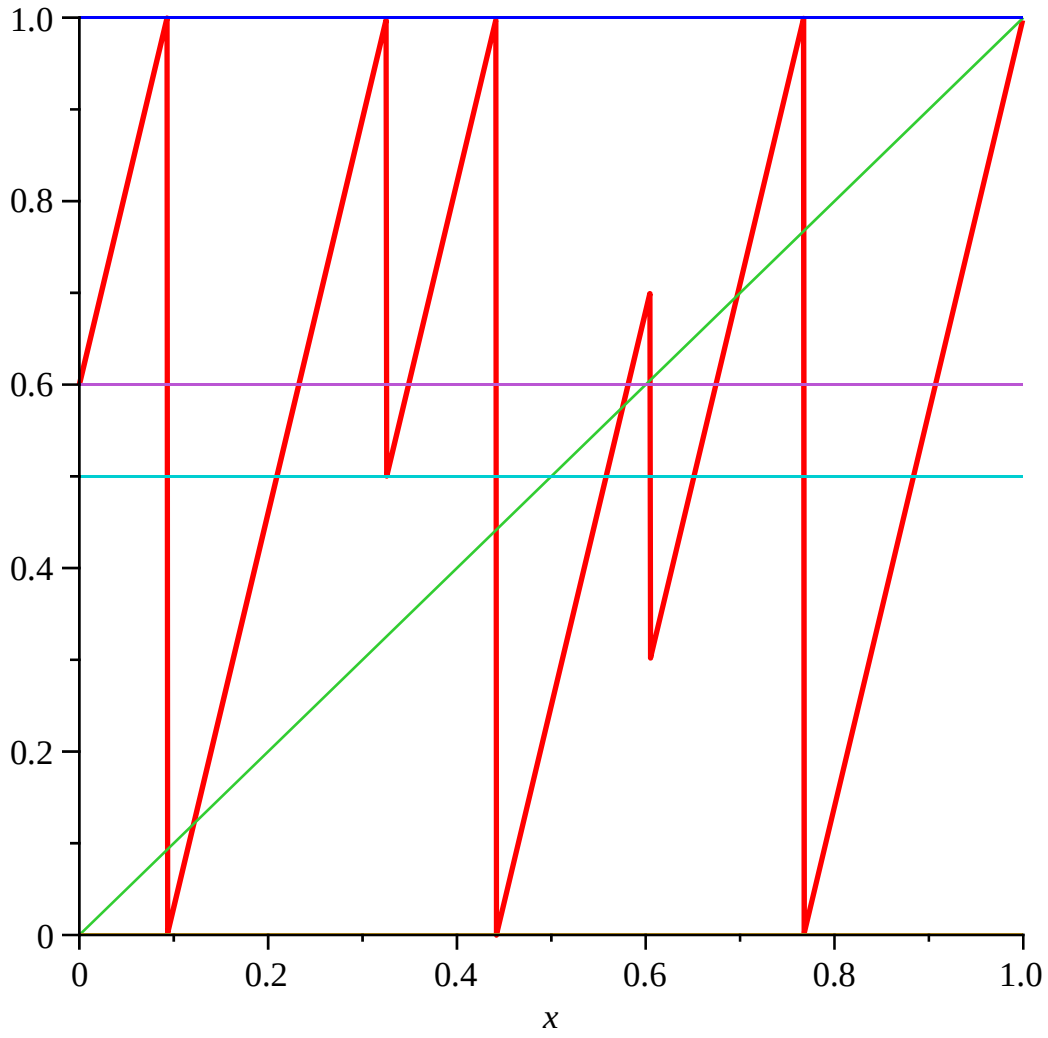
$T := x \rightarrow \beta x - a_{int\_of\_x(x)}$

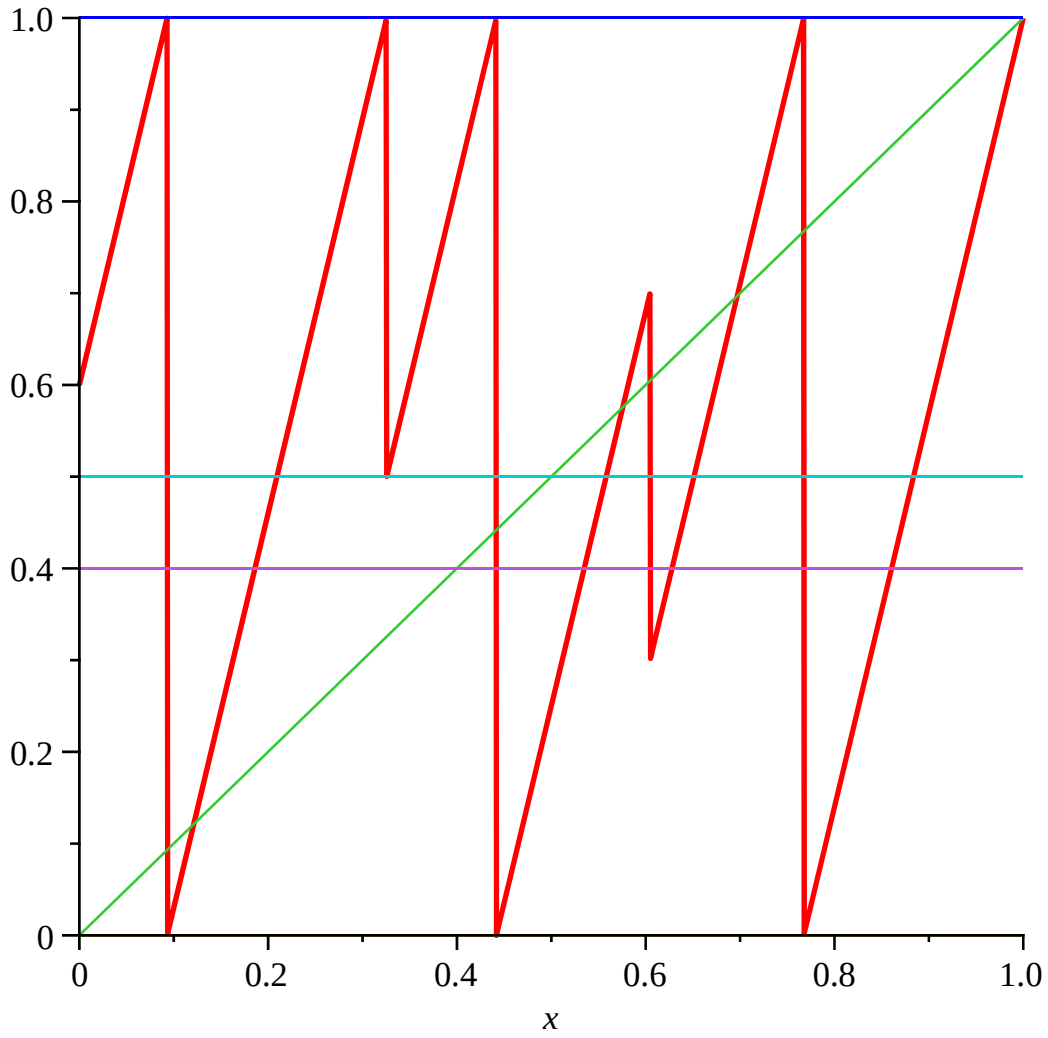
$T(c[, 1,]) =, 0.6000000000$

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

$T(c[, 3,]) =, 0.7000000000$

$T(c[, 4,]) =, 0.3000000000$






```
>
> ud:=vector(50):Digits:=100;NN:=50;
  d:=vector(50):
  xx:=c[3];#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;
```

$$Digits := 100$$

$$NN := 50$$

$$xx := 0.6046511628$$

[2.3000000000, 0.4000000000, 3.3000000000, 1.9000000000, 0.9000000000, 2.3000000000, 1.9000000000, 2.3000000000, 2.3000000000, 0.4000000000, 3.3000000000, 3.3000000000, 1.9000000000, -0.6000000000, 3.3000000000, 0.4000000000, 3.3000000000, 0.4000000000, 0.9000000000, 3.3000000000, 0.4000000000, 0.4000000000, 2.3000000000, 2.3000000000, 2.3000000000, 3.3000000000, 1.9000000000, -0.6000000000, 2.3000000000, 0.9000000000, 3.3000000000, 0.4000000000, -0.6000000000, 3.3000000000, 1.9000000000, 0.4000000000, 0.9000000000, 3.3000000000, 3.3000000000, 3.3000000000, 2.3000000000, 2.3000000000, 3.3000000000, 1.9000000000, 1.9000000000, 0.4000000000, -0.6000000000, 2.3000000000, 2.3000000000, 1.9000000000]

$$uIs\_it\_x := 0.6046511628$$

[1.9000000000, 2.3000000000, 2.3000000000, 2.3000000000, 3.3000000000, 2.3000000000, 3.3000000000, 0.9000000000, 2.3000000000, 1.9000000000, 0.9000000000, 3.3000000000, 0.4000000000, 1.9000000000, 0.9000000000, 2.3000000000, 2.3000000000, 3.3000000000, -0.6000000000, 2.3000000000, 0.9000000000, 3.3000000000, 2.3000000000, 3.3000000000, 2.3000000000, 1.9000000000, -0.6000000000, 2.3000000000, 3.3000000000, 3.3000000000, 3.3000000000, 3.3000000000, 0.4000000000, -0.6000000000, 2.3000000000, 3.3000000000, -0.6000000000, 3.3000000000, 1.9000000000, 1.9000000000, 0.4000000000, 0.4000000000, 0.9000000000, 2.3000000000, 1.9000000000, 2.3000000000, 1.9000000000, -0.6000000000, 1.9000000000, 2.3000000000]

$$Is\_it\_x := 0.6046511628$$

$$terr := 8.459788981\ 10^{-33}$$

$$err := 1.362493092\ 10^{-32} \quad \textbf{(1)}$$

```
>
> 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;
```

$$NN := 50$$

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

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

$$xxt := 0.0000000000$$

$$upflag := 1$$

$$Is\_it\_x := -8.094992520\ 10^{-33}$$

$$S_1 := 0.2264782321$$

$$xxt := 0.3255813953$$

$$upflag := 1$$

$$Is\_it\_x := 0.3255813953$$

$$S_2 := 0.1893820292$$

$$xxt := 0.6046511628$$

$$upflag := 0$$

$$Is\_it\_x := 0.6046511628$$

$$S_3 := 0.2825878926$$

$$xxt := 0.6046511628$$

$$upflag := 1$$

$$Is\_it\_x := 0.6046511628$$

$$S_4 := 0.1278927740$$

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

$$SS[, 1, 2,] =, 7.174245674\ 10^{-9}$$

$$SS[, 1, 3,] =, 0.0547655433$$

$SS[,1,4,] =, 0.0547655433$

$SS[,2,1,] =, 0.0000000000$

$SS[,2,2,] =, 0.0125775089$

$SS[,2,3,] =, 0.0673516067$

$SS[,2,4,] =, 0.0673516067$

$SS[,3,1,] =, 0.0704721635$

$SS[,3,2,] =, 0.0704721577$

$SS[,3,3,] =, 0.0704630300$

$SS[,3,4,] =, 0.0704630300$

$SS[,4,1,] =, 0.0000000000$

$SS[,4,2,] =, 0.0540837524$

$SS[,4,3,] =, 0.0577257826$

$SS[,4,4,] =, 0.0577257826$

```
>
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);
```

$$MM = \begin{bmatrix} -0.0000000000 & -0.0000000000 & -0.0704721635 & -0.0000000000 \\ -7.174245674\,10^{-9} & -0.0125775089 & -0.0704721577 & -0.0540837524 \\ -0.0547655433 & -0.0673516067 & -0.0704630300 & -0.0577257826 \\ -0.0547655433 & -0.0673516067 & -0.0704630300 & -0.0577257826 \end{bmatrix}$$

$$1/beta = 0.2325581395$$

$$eigenvalues\ MM = -0.0043240201,\ -0.1942381960,\ 0.0577958946,\ 0.0000000000$$

$$\begin{bmatrix} 0.2325581395 & -0.0000000000 & -0.0704721635 & -0.0000000000 \\ -7.174245674\,10^{-9} & 0.2199806306 & -0.0704721577 & -0.0540837524 \\ -0.0547655433 & -0.0673516067 & 0.1620951095 & -0.0577257826 \\ -0.0547655433 & -0.0673516067 & -0.0704630300 & 0.1748323570 \end{bmatrix}$$

$$\begin{bmatrix} 0.2325581395 & 0.2325581395 & 0.2325581395 & 0.2325581395 \end{bmatrix}$$

$$DD := \begin{bmatrix} 3.1887169333 & 5.1468001371 & 7.2227658799 & 7.2227658799 \end{bmatrix} \quad \textbf{(2)}$$

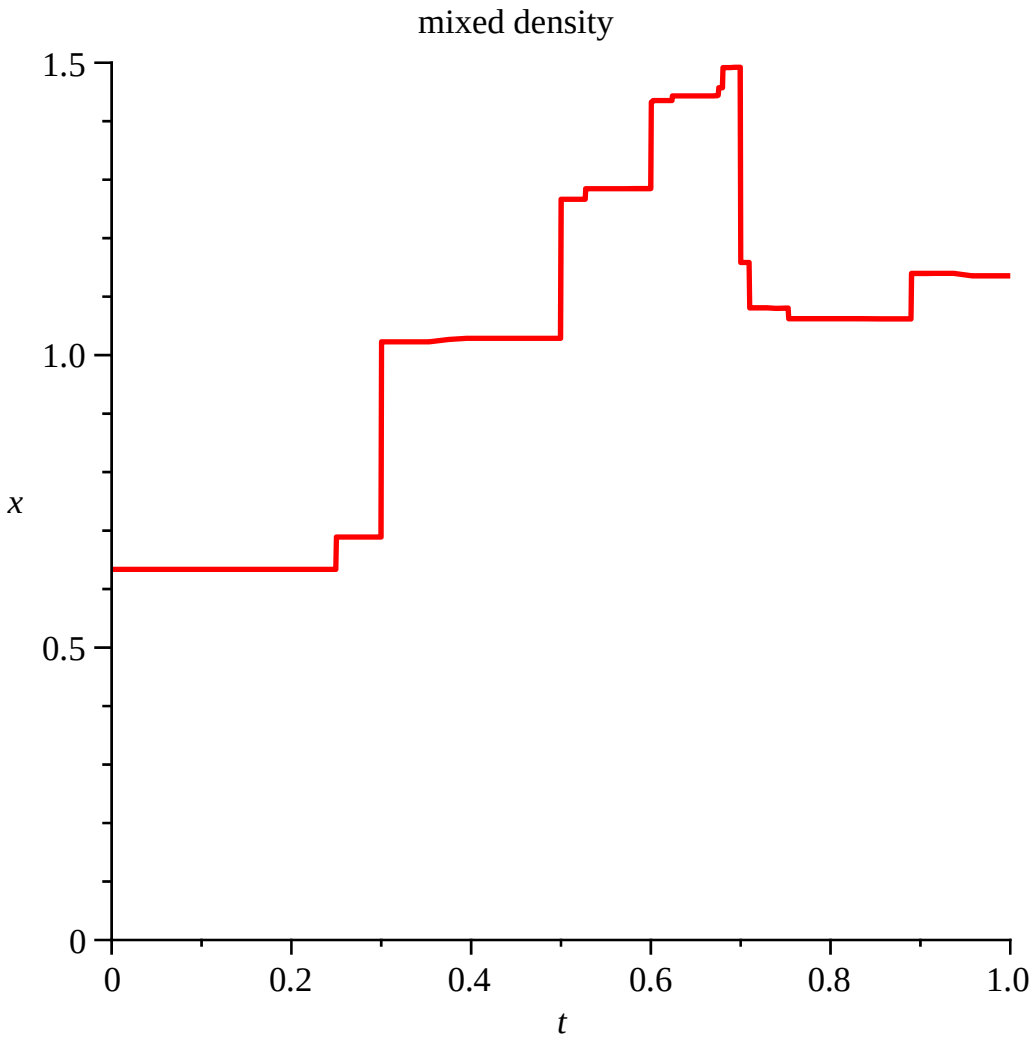
>

```
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;
#Normalizing factor
tNC:=1/beta:
for j from 1 to KK do
 tNC:=tNC+tDD[j]*sum((1-tvalc[j,i1+1])/beta^(i1+1),i1=1..50)
 od:
NC:=1/beta:
for j from 1 to KK do
if U[K[j]]=0 then
 NC:=NC+DD[j]*sum((valc[j,i1+1])/beta^(i1+1),i1=1..50)
else
NC:=NC+DD[j]*sum((1-valc[j,i1+1])/beta^(i1+1),i1=1..50)
fi;
 od:

print(`NC = `,NC);
```

```
    plot([(1/NC)*density(t)], t=0..1-0.00000000001, x=0..1.5, title=
    "mixed density", thickness=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
```

$NC = , 1.1703247925$




```
>
>
    #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;
```

$$y_4 := 0.8282656171$$

$$su := 0$$

2

4

5

6

$$err_0 := -2.542734052\ 10^{-34}$$

$$su := 0$$

2

4

5

6

$$err_1 := -2.542734052\ 10^{-34}$$

$$su := 0$$

2

3

4

5

$$6$$

$$err_2 := 8.031705964\ 10^{-33}$$

$$su := 0$$

$$1$$

$$2$$

$$3$$

$$4$$

$$5$$

$$6$$

$$err_3 := 8.031705964\ 10^{-33}$$

$$su := 0$$

$$1$$

$$2$$

$$3$$

$$5$$

$$6$$

$$err_4 := -9.816804384\ 10^{-33}$$

$$y =, 0.3200749938$$

$$err[, 0,] =, -2.542734052\ 10^{-34}$$

$$y =, 0.4395718861$$

$$err[, 1,] =, -2.542734052\ 10^{-34}$$

$$y =, 0.5386279633$$

$$err[, 2,] =, 8.031705964\ 10^{-33}$$

$$y =, 0.7000000000$$

$$err[, 3,] =, 8.031705964\ 10^{-33}$$

$$y =, 0.8282656171$$

$$err[, 4,] =, -9.816804384\ 10^{-33}$$

```
>
#check density greedy
#preimages
y:=0.500000000000;
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;
```

$$y := 0.5000000000$$

$$pre_1 := -0.0232558140$$

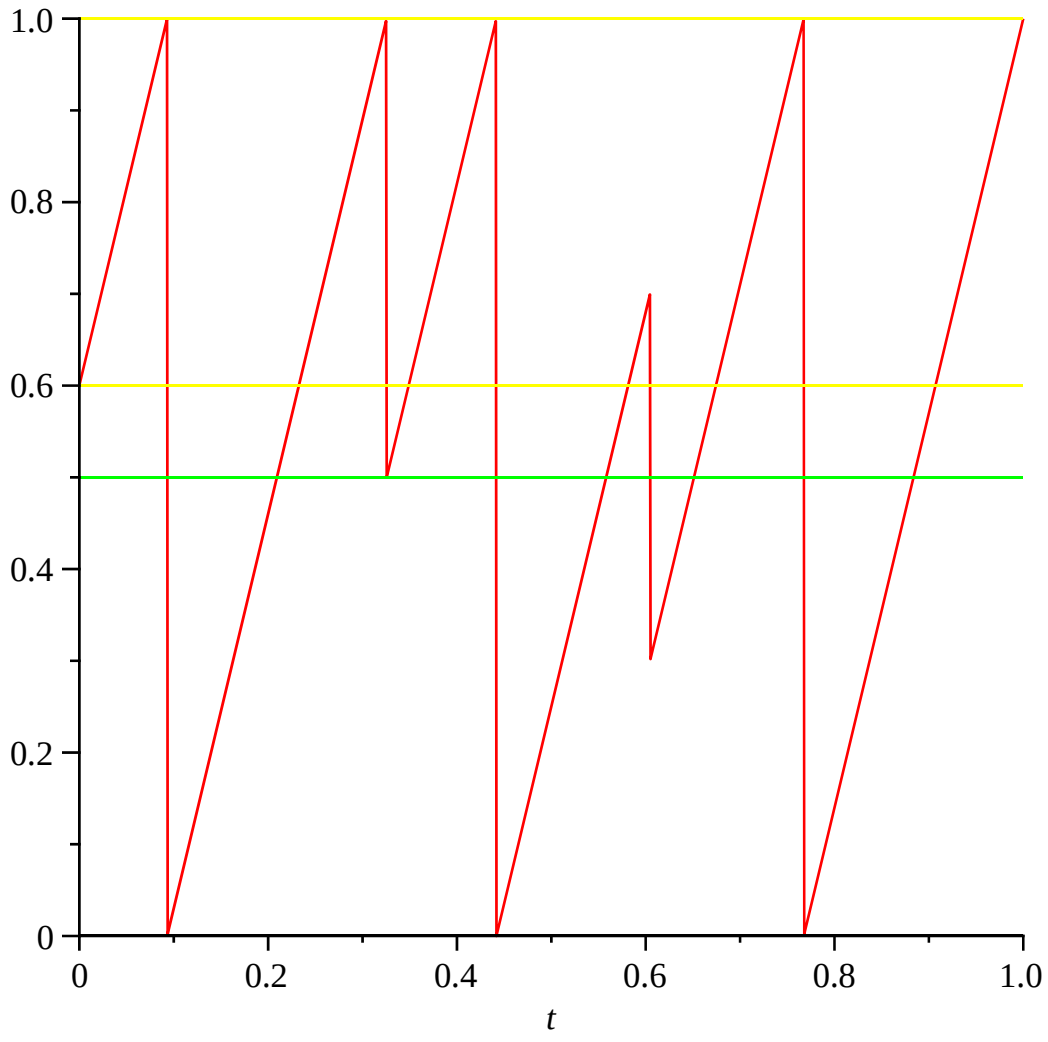
$$pre_2 := 0.2093023256$$

$$pre_3 := 0.3255813953$$

$$pre_4 := 0.5581395349$$

$$pre_5 := 0.6511627907$$

$$pre_6 := 0.8837209302$$



2

3

4

5

6

$$err2 := 8.031705964\ 10^{-33}$$

>

>