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


> N:=5;
  KK:=3;

  alpha[1]:=0.4;tK[1]:=1;
  alpha[2]:=0.5;tK[2]:=3;# here we need alpha1 <= alpha2
  alpha[3]:=0.7;tK[3]:=2; # order of K1, K2 is arbitrary
  alpha[4]:=0.7;tK[4]:=6;
  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
  tb[j]:=(j-1-sum((1-alpha[i])*delta1( j,tK[i]) ,i=1..KK))/beta;
   od; i:='i':
  tb[N+1]:=1;
  for j from 1 to N do
  ta[j]:=(j-1-sum((1-alpha[i])*delta1( j,tK[i]-1) ,i=1..KK));
   od;

  for j from 1 to N do # alternative formula for a[i]
  aa[j]:=beta*(tb[j+1]-tb[1])-1;
   od:
  for j from 1 to N do
  ta[j]-aa[j];
  od:
  for j from 1 to KK do
  tc[j]:=tb[tK[j]];
  od;
>
  maa:=ta[2]-ta[1]:# maximum a[i+1]-a[i]
  for i from 3 to N do
  if (ta[i]-ta[i-1])>maa then maa:=(ta[i]-ta[i-1]) fi
  od;#
  maa;
  beta_max:=evalf(1+(ta[N]-ta[1])/maa);
> if beta> beta_max then print("ERROR") fi;

  ##### Conjugated part
  for j from 1 to N do
  a[j]:=ta[N]-ta[N-j+1];
  od;
  for j from 1 to N do
  b[j]:=a[j]/beta;
  od;
```

```
b[N+1]:=1;
for i from 1 to KK do
K[i]:=N-tK[i]+1;
c[i]:=b[K[i]+1];
od;
>
tint_of_x:=x->piecewise(x<tb[2],1,# This function needs
additions by hand for
                                    # N>9 . Automathic procedure
causes plotting problems
                                    # but is used in other
programs
                          x<tb[3],2,
                          x<tb[4],3,
                          x<tb[5],4,
                          x<tb[6],5,
                          x<tb[7],6,
                          x<tb[8],7,
                          x<tb[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':
tT:=x->beta*x-ta[tint_of_x(x)];
T:=x->beta*x-a[int_of_x(x)];
for j from 1 to KK do
print(`tT(tc[`,j,`]) =`, tT(tc[j]))
od;
for j from 1 to KK do
print(`T(c[`,j,`]) =`, T(c[j]))
od;

plot([tT(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]) ;

$$N := 5$$

$$KK := 3$$

$$\alpha_1 := 0.4000000000$$

$$tK_1 := 1$$

$$\alpha_2 := 0.5000000000$$

$$tK_2 := 3$$

$$\alpha_3 := 0.7000000000$$

$$tK_3 := 2$$

$$\alpha_4 := 0.7000000000$$

$$tK_4 := 6$$

$$\beta := 3.6000000000$$

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

$$tb_1 := 0.0000000000$$

$$tb_2 := 0.1111111111$$

$$tb_3 := 0.3055555556$$

$$tb_4 := 0.4444444444$$

$$tb_5 := 0.7222222222$$

$$tb_6 := 1$$

$$ta_1 := -0.6000000000$$

$$ta_2 := 0.1000000000$$

$$ta_3 := 0.6000000000$$

$$ta_4 := 1.6000000000$$

$$ta_5 := 2.6000000000$$

$$tc_1 := 0.0000000000$$

$$tc_2 := 0.3055555556$$

$$tc_3 := 0.1111111111$$

$$1.0000000000$$

$$\beta max := 4.2000000000$$

$$a_1 := 0.0000000000$$

$$a_2 := 1.0000000000$$

$$a_3 := 2.0000000000$$

$$a_4 := 2.5000000000$$

$$a_5 := 3.2000000000$$

$$b_1 := 0.0000000000$$

$$b_2 := 0.2777777778$$

$$b_3 := 0.5555555556$$

$$b_4 := 0.6944444444$$

$$b_5 := 0.8888888889$$

$$b_6 := 1$$

$$K_1 := 5$$

$$c_1 := 1$$

$$K_2 := 3$$

$$c_2 := 0.6944444444$$

$$K_3 := 4$$

$$c_3 := 0.8888888889$$

$$tint\_of\_x := x \rightarrow piecewise\left(x < tb_2, 1, x < tb_3, 2, x < tb_4, 3, x < tb_5, 4, x < tb_6, 5, x < tb_7, 6, x < tb_8, 7, x < tb_9, 8, 9\right)$$

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

$$tT := x \rightarrow \beta\, x - ta_{tint\_of\_x(x)}$$

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

$$tT(tc[, 1,]) =, 0.6000000000$$

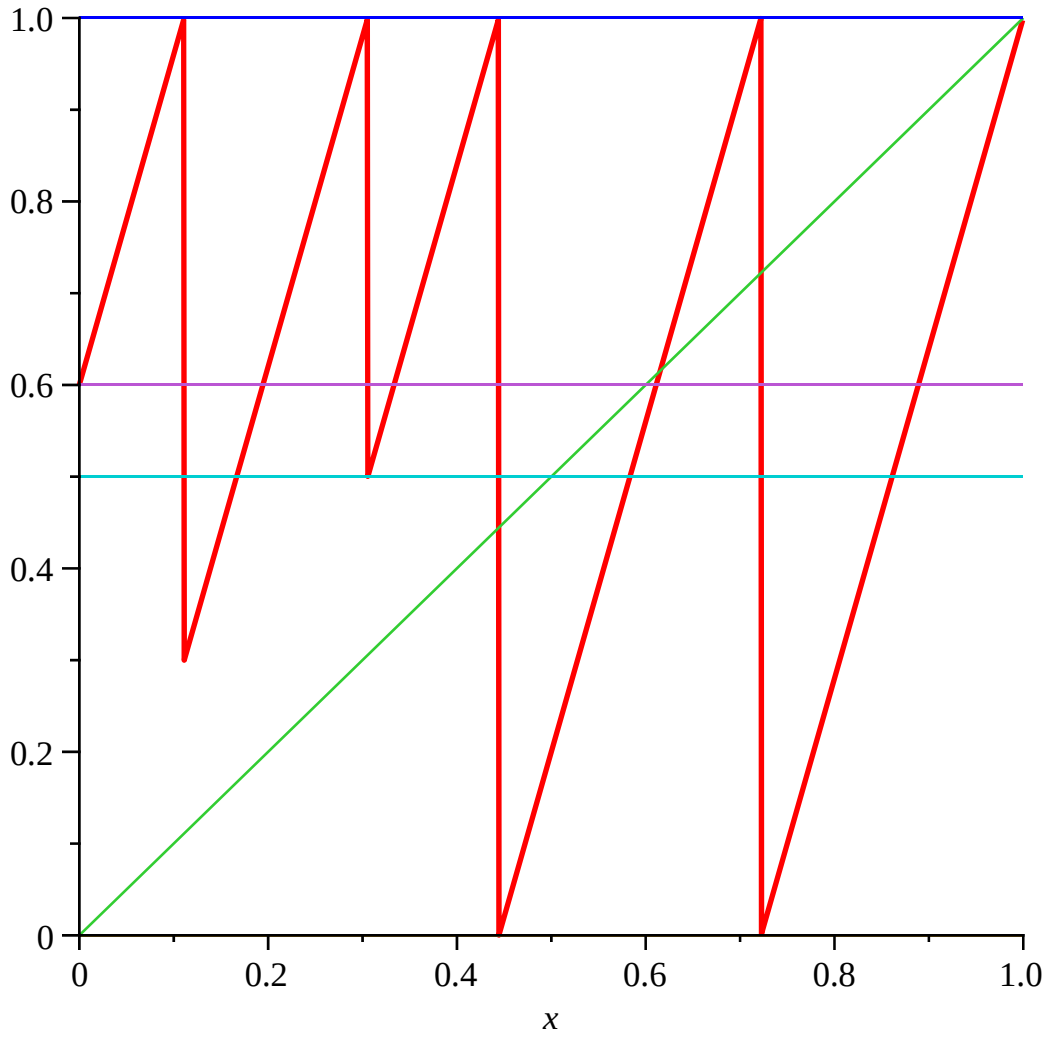
$$tT(tc[, 2,]) =, 0.5000000000$$

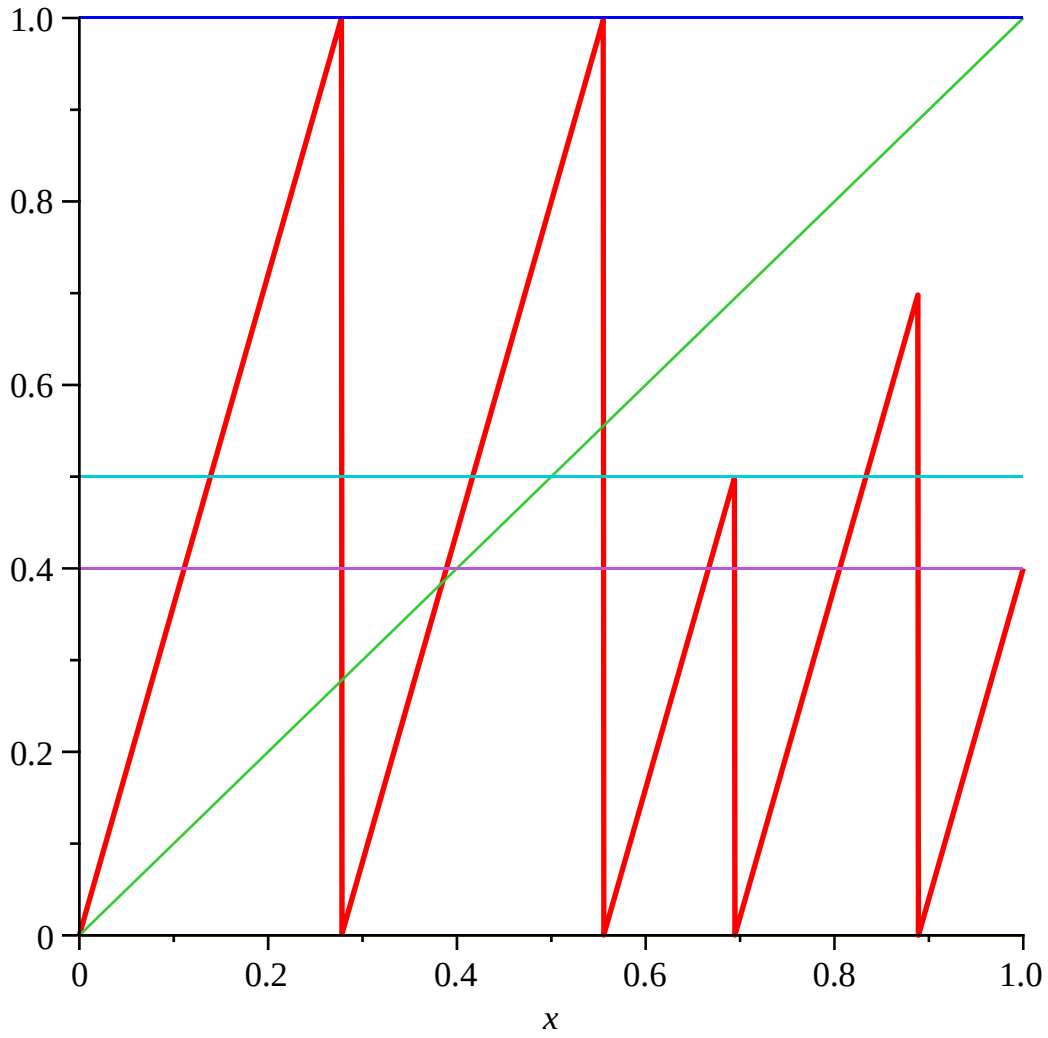
$$tT(tc[, 3,]) =, 0.3000000000$$

$$T(c[, 1,]) =, 0.4000000000$$

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

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


1.0
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1.0
x




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

$$Digits := 100$$

$$NN := 50$$

$$xx := 0.3957188605$$

[0.6000000000, 2.6000000000, 0.6000000000, 2.6000000000, −0.6000000000, 1.6000000000,
2.6000000000, 0.1000000000, 1.6000000000, 0.1000000000, 0.6000000000,
1.6000000000, 2.6000000000, 2.6000000000, 2.6000000000, 1.6000000000,
0.6000000000, 2.6000000000, 2.6000000000, 1.6000000000, 0.6000000000,
2.6000000000, 2.6000000000, 2.6000000000, 2.6000000000, 2.6000000000,
1.6000000000, 1.6000000000, 2.6000000000, 0.6000000000, 1.6000000000,
2.6000000000, 2.6000000000, 0.1000000000, 1.6000000000, 2.6000000000,
2.6000000000, 2.6000000000, 0.1000000000, 2.6000000000, 0.6000000000,
2.6000000000, 0.1000000000, 2.6000000000, 0.1000000000, 2.6000000000,
1.6000000000, 1.6000000000, 2.6000000000, 1.6000000000]

$$tIs\_it\_x := 0.3957188605$$

[1.0000000000, 1.0000000000, 1.0000000000, 3.2000000000, 0.0000000000, 0.0000000000,
2.0000000000, 1.0000000000, 0.0000000000, 1.0000000000, 2.0000000000,
1.0000000000, 0.0000000000, 0.0000000000, 2.5000000000, 1.0000000000,
1.0000000000, 3.2000000000, 0.0000000000, 1.0000000000, 2.0000000000,
1.0000000000, 1.0000000000, 1.0000000000, 1.0000000000, 1.0000000000,
1.0000000000, 2.0000000000, 1.0000000000, 2.0000000000, 1.0000000000,
1.0000000000, 2.5000000000, 0.0000000000, 0.0000000000, 0.0000000000,
0.0000000000, 2.5000000000, 0.0000000000, 1.0000000000, 0.0000000000,
2.5000000000, 0.0000000000, 1.0000000000, 0.0000000000, 1.0000000000,
2.5000000000, 1.0000000000, 3.2000000000, 0.0000000000]

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

$$terr := 1.420595522\ 10^{-28}$$

$$err := 1.025994608\ 10^{-28} \quad \textbf{(1)}$$

```
>
> NN:=50; chi:=(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:=tc[i];
       for n from 1 to NN+1 do
          dc[i,n]:=ta[tint_of_x(xxt)];
          ic[i,n]:=tint_of_x(xxt)-1;
          for ii from 1 to KK do
              if xxt<tc[ii] then cc[i,ii,n]:=1 else cc[i,ii,n]:=0
  fi;
          od;
       tvalc[i,n]:=xxt;
       xxt:=tT(xxt);
       od:
  Is_it_x:=sum(dc[i,j1]/beta^j1,j1=1..NN);
```

```
tS[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
tSS[i,j]:=sum(cc[i,j,j1+1]/beta^(j1+1),j1=1..NN);
#print(`tSS[`,i,j,`] =`,tSS[i,j]);
od;od;
for i from 1 to KK do
xxt:=c[i];
      for n from 1 to NN+1 do
         dc[i,n]:=a[int_of_x(xxt)];
         ic[i,n]:=int_of_x(xxt)-1;
         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;
      valc[i,n]:=xxt;
      xxt:=T(xxt);
      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(`tSS[`,i,j,`] =`,tSS[i,j]);
print(`SS[`,i,j,`] =`,SS[i,j]);
od;od;
```

$$NN := 50$$

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

$$xxt := 0.0000000000$$

$$Is\_it\_x := -9.327743860\ 10^{-30}$$

$$tS_1 := 0.3162393162$$

$$xxt := 0.3055555556$$

$$Is\_it\_x := 0.3055555556$$

$$tS_2 := 0.2777434356$$

$$xxt := 0.1111111111$$

$$Is\_it\_x := 0.1111111111$$

$$tS_3 := 0.1939578389$$

$$xxt := 1$$

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

$$S_1 := 0.1111111111$$

$$xxt := 0.6944444444$$

$$Is\_it\_x := 0.6944444444$$

$$S_2 := 0.1496069917$$

$$xxt := 0.8888888889$$

$$Is\_it\_x := 0.8888888889$$

$$S_3 := 0.2333925884$$

$$tSS[, 1, 1,] =, 0.0000000000$$

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

$$tSS[, 1, 2,] =, 2.651734690\ 10^{-16}$$

$$SS[, 1, 2,] =, 2.651734690\ 10^{-16}$$

$$tSS[, 1, 3,] =, 4.251796410\ 10^{-29}$$

$$SS[, 1, 3,] =, 4.251796410\ 10^{-29}$$

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

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

$$tSS[, 2, 2,] =, 0.0214362056$$

$$SS[, 2, 2,] =, 0.0214362056$$

$$tSS[, 2, 3,] =, 7.653545582\ 10^{-12}$$

$$SS[, 2, 3,] =, 7.653545582\ 10^{-12}$$

$$tSS[, 3, 1,] =, 0.0000000000$$

$$SS[, 3, 1,] =, 0.0000000000$$

$$tSS[, 3, 2,] =, 0.0776206518$$

$$SS[, 3, 2,] =, 0.0776206518$$

$$tSS[, 3, 3,] =, 0.0004593937$$

$$SS[, 3, 3,] =, 0.0004593937$$

```
> tMM:=matrix(KK,KK,[]):
  MM:=matrix(KK,KK,[]):
  for i from 1 to KK do
  for j from 1 to KK do
  tMM[j,i]:=-tSS[i,j];
  MM[j,i]:=-SS[i,j];
  od;od;
  print(`tMM = `, tMM);print(`MM = `,MM);
  print(`1/beta =`, 1/beta);
  print(`eigenvalues tMM =`,eigenvalues(tMM));
  print(`eigenvalues MM =`,eigenvalues(MM));

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

```
od:
print(tMM);
print(MM);
print(ve);


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

$$tMM = , \begin{bmatrix} -0.0000000000 & -0.0000000000 & -0.0000000000 \\ -2.651734690\ 10^{-16} & -0.0214362056 & -0.0776206518 \\ -4.251796410\ 10^{-29} & -7.653545582\ 10^{-12} & -0.0004593937 \end{bmatrix}$$

$$MM = , \begin{bmatrix} -0.0000000000 & -0.0000000000 & -0.0000000000 \\ -2.651734690\ 10^{-16} & -0.0214362056 & -0.0776206518 \\ -4.251796410\ 10^{-29} & -7.653545582\ 10^{-12} & -0.0004593937 \end{bmatrix}$$

$$1/beta =, 0.2777777778$$

$$eigenvalues\ tMM =, -0.0004593936, -0.0214362057, -0.0000000000$$

$$eigenvalues\ MM =, -0.0004593936, -0.0214362057, -0.0000000000$$

$$\begin{bmatrix} 0.2777777778 & -0.0000000000 & -0.0000000000 \\ -2.651734690\ 10^{-16} & 0.2563415722 & -0.0776206518 \\ -4.251796410\ 10^{-29} & -7.653545582\ 10^{-12} & 0.2773183841 \end{bmatrix}$$

$$\begin{bmatrix} 0.2777777778 & -0.0000000000 & -0.0000000000 \\ -2.651734690\ 10^{-16} & 0.2563415722 & -0.0776206518 \\ -4.251796410\ 10^{-29} & -7.653545582\ 10^{-12} & 0.2773183841 \end{bmatrix}$$

$$\begin{bmatrix} 0.2777777778 & 0.2777777778 & 0.2777777778 \end{bmatrix}$$

$$tDD := \begin{bmatrix} 1.0000000000 & 1.3869268632 & 1.0016565568 \end{bmatrix}$$

$$DD := \begin{bmatrix} 1.0000000000 & 1.3869268632 & 1.0016565568 \end{bmatrix} \quad \textbf{(2)}$$

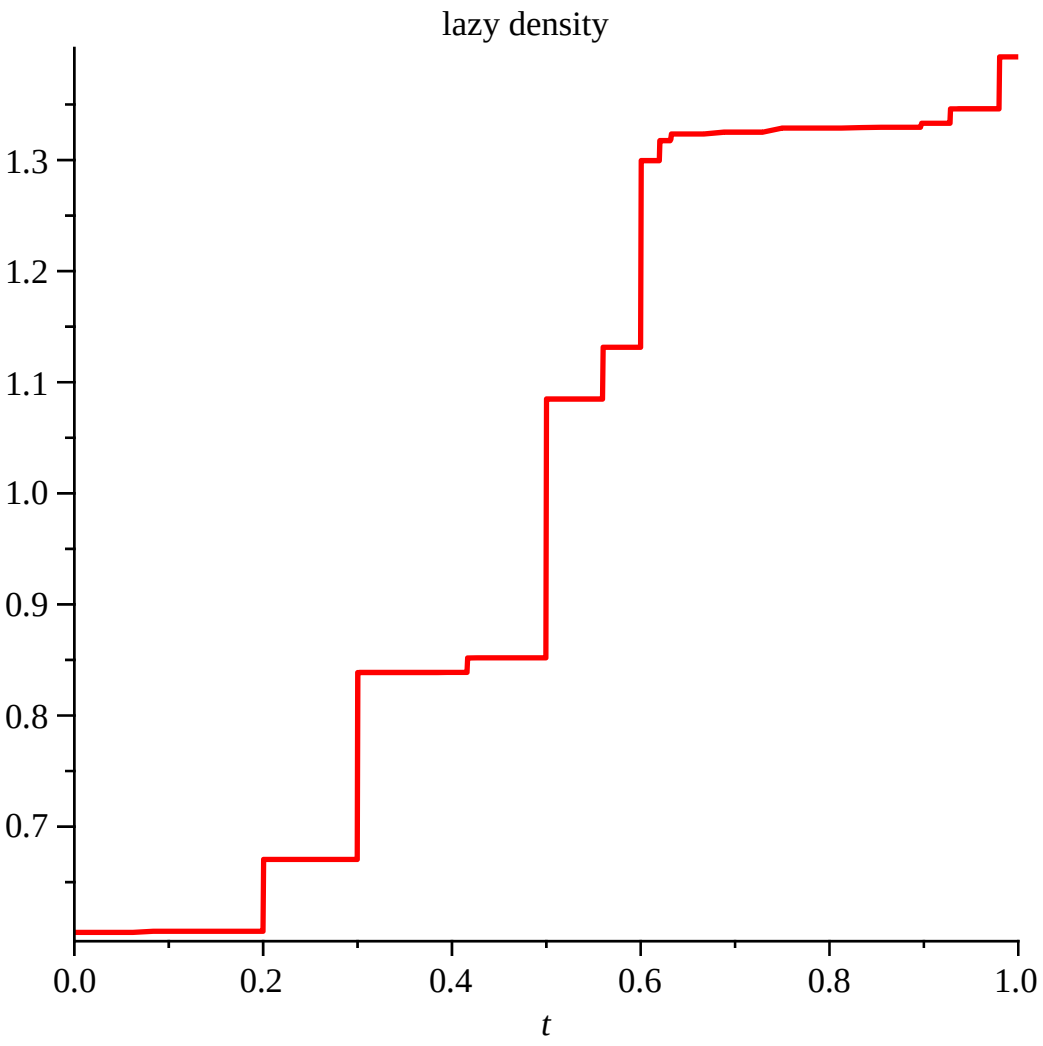
```
> tdensity:=proc(t) local j, den;
                    den:=1/beta;
                  for j from 1 to KK do
                    den:=den+tDD[j]*sum((chi(tvalc[j,i1+1],1,t))
/beta^(i1+1),i1=1..50)
                    od;
                  return den;
            end proc;
      density:=proc(t) local j, den;
                    den:=1/beta;
```

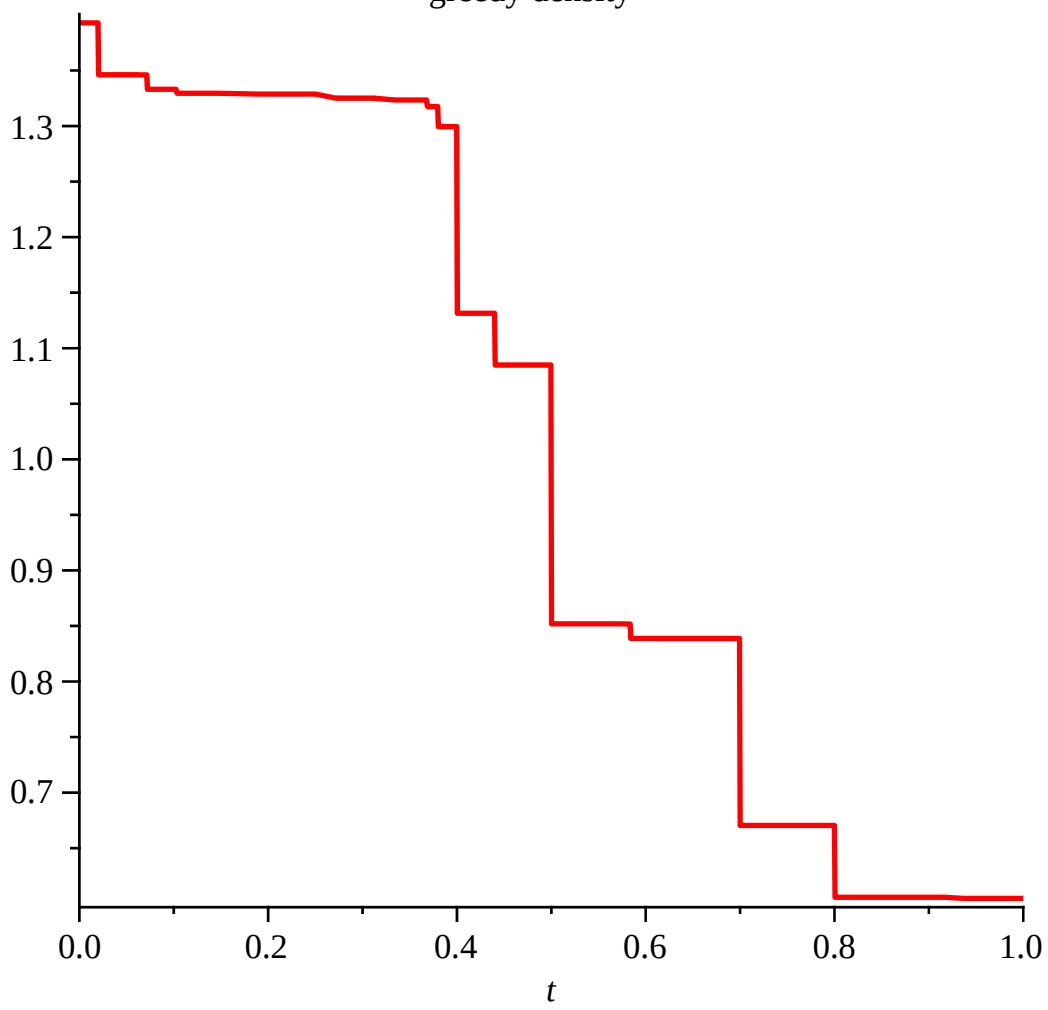
```
                    for j from 1 to KK do
                      den:=den+ DD[j]*sum((chi(0,valc[j,i1+1],t))
    /beta^(i1+1),i1=1..50)
                     od;
                    return den;
                end proc;
    #Normalizing factor
    tNC:=1/beta:
    for j from 1 to KK do
     tNC:=tNC+tDD[j]*sum((1-tvalc[j,i1+1])/beta^(i1+1),i1=1..50)
     od:
    NC:=1/beta:
    for j from 1 to KK do
     NC:=NC+DD[j]*sum((valc[j,i1+1])/beta^(i1+1),i1=1..50)
     od:
    print(`tNC = `,tNC);

    print(`NC = `,NC);
    plot([(1/tNC)*tdensity(t)],t=0..1-0.00001,title="lazy density",
    thickness=2);
    plot([(1/NC)*density(t)],t=0..1,title="greedy density",
    thickness=2);
```

```
tdensity := proc(t)
    local j, den;
    den := 1/beta;
    for j to KK do
        den := den + tDD[j]*(sum(chi(tvalc[j, i1+1], 1, t)/beta^(i1+1), i1=1..50))
    end do;
    return den
end proc
density := proc(t)
    local j, den;
    den := 1/beta;
    for j to KK do
        den := den + DD[j]*(sum(chi(0, valc[j, i1+1], t)/beta^(i1+1), i1=1..50))
    end do;
    return den
end proc
```

$$tNC = , 0.4593491897$$

$$NC = , 0.4593491897$$


lazy density
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.0
0.2
0.4
0.6
0.8
1.0
t




```
>
> #check density lazy
  #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]+ta[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]>=tb[i3] and pre[i3]<=tb[i3+1]) then
   su:=su+tdensity(pre[i3])/beta;
  print(i3);
  fi;
  od;
  terr[j6]:=tdensity(y[j6])-su;
```

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

$y_1 :=$

0.4996417214180000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000

$y_2 :=$

0.5772816614900000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000

$y_0 :=$

0.2778428757060000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000

$y_3 :=$

0.9319038940069000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000

$su := 0$

4

5

$terr_0 :=$

$7.976312751243623645307416447974723106602641923042057210599263409217633810^{-30}$

$su := 0$

2

4

5

$terr_1 :=$

$7.976312751243623645307416447974723106602641923042057210599263409217633710^{-30}$

$su := 0$

2

3

4

5

$terr_2 :=$

$7.976312751243623645307416447974723106602641923042057210599263409217633710^{-30}$

$su := 0$

1

2

3

4

5

$terr_3 :=$

$-3.204470619075617616088133709930153792551880210186688071879776408530509110^{-29}$

$y_1 :=$

0.473061629294600000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000

$y_2 :=$

0.521301410731400000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000

$y_0 :=$

0.158565089201200000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000

$y_3 :=$

0.983474005008700000000000000000000000000000000000000000000000000000000000000000\

00000000000000000000000000

$su := 0$

1

2

3

4

5

$err_0 :=$

$-3.204470619075617616088133709930153792551880210186688071879776408530509 10^{-29}$

$su := 0$

1

2

3

4

$err_1 :=$

$7.976312751243623645307416447974723106602641923042057210599263409217633 8 10^{-30}$

$su := 0$

1

2

4

$err_2 :=$

$7.976312751243623645307416447974723106602641923042057210599263409217633 7 10^{-30}$

$su := 0$

1

2

$err_3 :=$

$7.976312751243623645307416447974723106602641923042057210599263409217633 8 10^{-30}$

$terr[, 0,]=,$

$-3.204470619075617616088133709930153792551880210186688071879776408530509 10^{-29}$

$terr[, 1,]=,$

$7.976312751243623645307416447974723106602641923042057210599263409217633 8 10^{-30}$

$terr[, 2,]=,$

7.976312751243623645307416447974723106602641923042057210599263409217633710^-30

terr[, 3,]=,

7.976312751243623645307416447974723106602641923042057210599263409217633810^-30

err[, 0,]=,

-3.204470619075617616088133709930153792551880210186688071879776408530509010^-29

err[, 1,]=,

7.976312751243623645307416447974723106602641923042057210599263409217633810^-30

err[, 2,]=,

7.976312751243623645307416447974723106602641923042057210599263409217633710^-30

err[, 3,]=,

7.976312751243623645307416447974723106602641923042057210599263409217633810^-30

```
> #check density lazy
  #preimages
  y:=alpha[1]*rand()/10^12;i3:='i3':
  for i3 from 1 to N do
  pre[i3]:=(y+ta[i3])/beta;
  od;i3:='i3':
  plot([tT(t),0,1,y,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]>=tb[i3] and pre[i3]<=tb[i3+1]) then
   su:=su+tdensity(pre[i3])/beta;
  print(i3);
  fi;
  od;
  err1:=tdensity(y)-su;
  print(`_____________________________________________________`);
  #check density greedy
  #preimages
  y:=alpha[1]+(alpha[2]-alpha[1])*rand()/10^12;
  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;
```

$y :=$

0.24629282769920000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000

$pre_1 :=$

−0.09825199230577777777777777777777777777777777777777777777777777777777777777777\
77777777777777777777777777778

$pre_2 :=$

0.09619245213866666666666666666666666666666666666666666666666666666666666666666\
66666666666666666666666666667

$pre_3 :=$

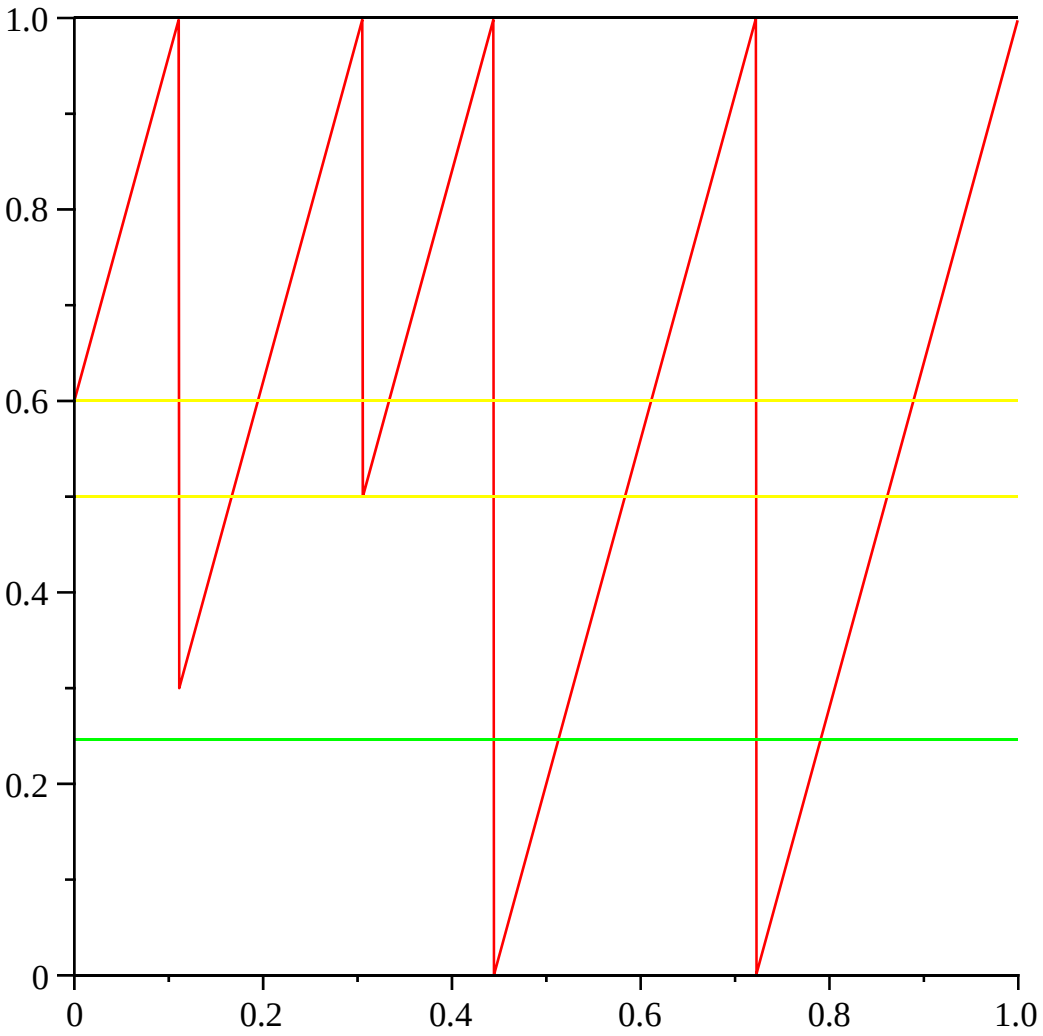
0.23508134102755555555555555555555555555555555555555555555555555555555555555555\
55555555555555555555555555556

$pre_4 :=$

0.51285911880533333333333333333333333333333333333333333333333333333333333333333\
33333333333333333333333333333

$pre_5 :=$

0.79063689658311111111111111111111111111111111111111111111111111111111111111111\
11111111111111111111111111111



4

5

err1 :=

7.976312751243623645307416447974723106602641923042057210599263409217633910^-30

---

y :=

0.484701876538800000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000

pre_1 :=

0.134639410149666666666666666666666666666666666666666666666666666666666666666666\
666666666666666666666666667

pre_2 :=

0.412417187927444444444444444444444444444444444444444444444444444444444444444444\
444444444444444444444444444

pre_3 :=

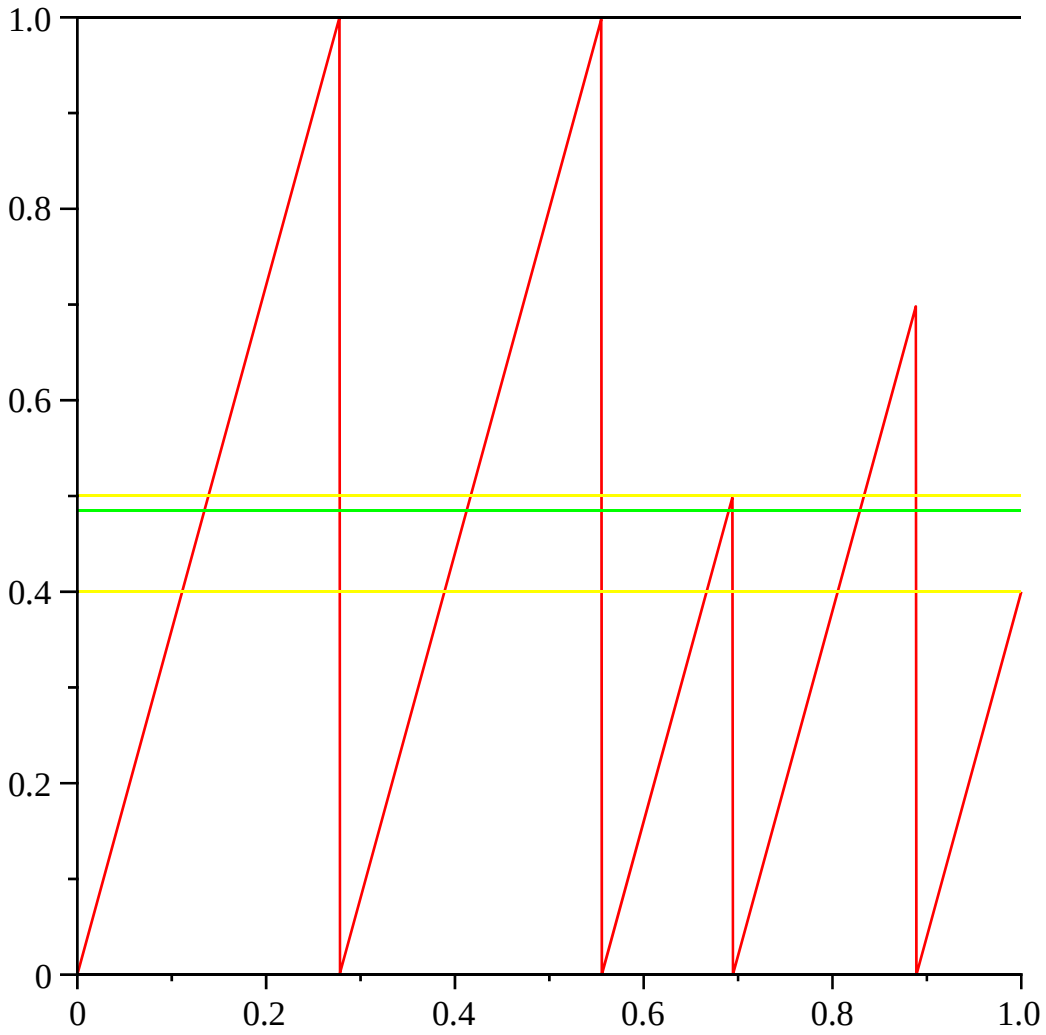
0.690194965705222222222222222222222222222222222222222222222222222222222222222222\
222222222222222222222222222

$pre_4 :=$

0.829083854594111111111111111111111111111111111111111111111111111111111111111\
11111111111111111111111111111

$pre_5 :=$

1.023528299038555555555555555555555555555555555555555555555555555555555555555\
55555555555555555555555555556



1

2

3

4

$err2 :=$

7.976312751243623645307416447974723106602641923042057210599263409217633 8

10^{-30}

>

>