```
> with(plots):Digits:=100:interface(displayprecision=6):with
  (linalg):


>
  N:=8;

  bb:=vector(N+1,[]):#for  printing  only  =  b[]
  beta:=vector(N,[]):

  alpha:=vector(N,[]):
  gamm=vector(N,[]):#heiths  of  lower  ends  of  hanging  branches
                          #  alpha[i]+gamma[i]<1    !!!!!!!
                          #if  gamma[i]>0
  #Map  #1
  alpha[1]:=1:beta[1]:=2.9:gamm[1]:=0:

  alpha[2]:=0.5:beta[2]:=4:gamm[2]:=0.0:#
  alpha[3]:=0.5:beta[3]:=2.9:gamm[3]:=0.2:  #
  alpha[4]:=1:beta[4]:=2.5:gamm[4]:=0:  #




  i:='i':
  beta_const:=sum(alpha[i],i=1..N/2);
  i:='i':
  for  j  from 1  to  N/2  do
  beta[j]:=beta_const;
  od:
  #Map  #2
  for  i  from 1  to  N/2  do
  alpha[N-i+1]:=alpha[i];
  beta[N-i+1]:=beta[i];
  gamm[N-i+1]:=1-alpha[i]-gamm[i];
  od;
  for  i  from 1  to  N  do
  alpha[i]:=alpha[i]/2;
  gamm[i]:=gamm[i]/2;
  od;
  for  i  from N/2+1  to  N  do
  gamm[i]:=gamm[i]+0.5;
  od;

  print(`alpha  =`,alpha);
  print(`beta  =`,beta);
  print(`gamma  =`,gamm);
```

```
   gamm[6]:=gamm[6]-0.1;
   print(`gamma =`,gamm);


   b[1]:=0:
   for j from 1 to N do
   b[j+1]:=b[j]+alpha[j]/beta[j]:
    od: i:='i':
   b[N+1]:=1:
   ag:=vector(N,[]):
   al:=vector(N,[]):
   a:=vector(N,[]):
   c:=vector(2*N,[]):
   for j from 1 to N do
   bb[j]:=b[j];
   ag[j]:=beta[j]*b[j];
   al[j]:=-1+beta[j]*b[j+1];
   od:
   bb[N+1]:=1:
   for j from 1 to N do
   a[j]:=ag[j]-gamm[j];
    od:

   print(`b =`,bb);
   print(`ag =`,ag);
   print(`al =`,al);
   print(`a =`,a);
   print(`gamma =`,gamm);
>
>
> # ag shows maximal digit (greedy)
   #  al shows minimal digit (lazy)  #### if ag[j]=al[j] then j is
   onto branch and there is
                                                                    #
   no choice there
   # a shows digits assigned automatically using the vector U: U(j)
   =1 lazy
   #                                                          U(j)=
   0 greedy
   # we can assign digit a arbitrarily between minimum and maximum
   and then put 2 into vector U



   # Now we will name points c[i] (there is KK + number of 2's in U
   points c[i])
   # and create a vectors sidec[], ineqc[],signc[] which shows the
```

```
character of the point c[i]
Kc:=0:# new number of c points
for j from 1 to N do if alpha[j]<1 then Kc:=Kc+1 fi od:
for j from 1 to N do if (gamm[j]>0 and alpha[j]+gamm[j]<1) then
Kc:=Kc+1 fi od:
print(`Kc =`,Kc);
c:=vector(2*N,[]):
sidec:=vector(2*N,[]):# 1 left (use uT), 0 right (use T)
j_of_c:=vector(2*N,[]):# shows the index of the interval
associated with c

cj:=1:# this is the new index for c points
for j from 1 to N do
if (alpha[j]<1 and gamm[j]=0) then   c[cj]:=b[j+1]; sidec[cj]:=
0;j_of_c[cj]:=j;cj:=cj+1 fi;
if (alpha[j]<1 and gamm[j]+alpha[j]=1) then  c[cj]:=b[j]; sidec
[cj]:=1;j_of_c[cj]:=j;cj:=cj+1 fi;
if (gamm[j]>0 and gamm[j]+alpha[j]<1) then  c[cj]:=b[j]; sidec
[cj]:=1;j_of_c[cj]:=j;cj:=cj+1 ;
                        c[cj]:=b[j+1]; sidec[cj]:=0;j_of_c[cj]:=j;
cj:=cj+1 fi;
od:

print(`c =`,c);
print(`sidec =`,sidec);
print(`j_of_c =`,j_of_c);


>
>


>
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,
                          x<b[10],9,
```

```
                                x<b[11], 10,
                                    11);
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,
                                x<=b[10], 9,
                                x<=b[11], 10,
                                    11);
x:='x':
uT:=x->beta[uint_of_x(x)]*x-a[uint_of_x(x)];
T:=x->beta[int_of_x(x)]*x-a[int_of_x(x)];
Tc:=vector(Kc+2,[]):
for j from 1 to Kc do
if sidec[j]=0 then    Tc[j]:=T(c[j]);
              else    Tc[j]:=uT(c[j]) fi;
od:

print(`Tc = `, Tc);

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

$$N := 8$$

$$\beta const := 3.000000$$

$$\alpha_8 := 1$$

$$\beta_8 := 3.000000$$

$$gamm_8 := 0$$

$$\alpha_7 := 0.500000$$

$$\beta_7 := 3.000000$$

$$gamm_7 := 0.500000$$

$$\alpha_6 := 0.500000$$

$$\beta_6 := 3.000000$$

$$gamm_6 := 0.300000$$

$$\alpha_5 := 1$$

$$\beta_5 := 3.000000$$

$$gamm_5 := 0$$

$$\alpha_1 := \frac{1}{2}$$

$$gamm_1 := 0$$

$$\alpha_2 := 0.250000$$

$$gamm_2 := 0.000000$$

$$\alpha_3 := 0.250000$$

$$gamm_3 := 0.100000$$

$$\alpha_4 := \frac{1}{2}$$

$$gamm_4 := 0$$

$$\alpha_5 := \frac{1}{2}$$

$$gamm_5 := 0$$

$$\alpha_6 := 0.250000$$

$$gamm_6 := 0.150000$$

$$\alpha_7 := 0.250000$$

$$gamm_7 := 0.250000$$

$$\alpha_8 := \frac{1}{2}$$

$$gamm_8 := 0$$

$$gamm_5 := 0.500000$$

$$gamm_6 := 0.650000$$

$$gamm_7 := 0.750000$$

$$gamm_8 := 0.500000$$

$$alpha = \left[\begin{array}{cccccccc} \frac{1}{2} & 0.250000 & 0.250000 & \frac{1}{2} & \frac{1}{2} & 0.250000 & 0.250000 & \frac{1}{2} \end{array} \right]$$

$$beta = \left[\begin{array}{cccccccc} 3.000000 & 3.000000 & 3.000000 & 3.000000 & 3.000000 & 3.000000 & 3.000000 & 3.000000 \end{array} \right]$$

$$gamma =, \begin{bmatrix} 0 & 0.000000 & 0.100000 & 0 & 0.500000 & 0.650000 & 0.750000 & 0.500000 \end{bmatrix}$$

$$gamm_6 := 0.550000$$

$$gamma =, \begin{bmatrix} 0 & 0.000000 & 0.100000 & 0 & 0.500000 & 0.550000 & 0.750000 & 0.500000 \end{bmatrix}$$

$$b =, \begin{bmatrix} 0 & 0.166667 & 0.250000 & 0.333333 & 0.500000 & 0.666667 & 0.750000 & 0.833333 & 1 \end{bmatrix}$$

$$ag =, \begin{bmatrix} 0.000000 & 0.500000 & 0.750000 & 1.000000 & 1.500000 & 2.000000 & 2.250000 & 2.500000 \end{bmatrix}$$

$al =,$

$$\left[-0.500000, -0.250000, -1.\ 10^{-100}, 0.500000, 1.000000, 1.250000, 1.500000, 2.000000\right]$$

$$a =, \begin{bmatrix} 0.000000 & 0.500000 & 0.650000 & 1.000000 & 1.000000 & 1.450000 & 1.500000 & 2.000000 \end{bmatrix}$$

$$gamma =, \begin{bmatrix} 0 & 0.000000 & 0.100000 & 0 & 0.500000 & 0.550000 & 0.750000 & 0.500000 \end{bmatrix}$$

$$Kc =, 10$$

$$c =, \left[0.166667, 0.250000, 0.250000, 0.333333, 0.500000, 0.500000, 0.666667, 0.750000, 0.750000, 0.833333, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}\right]$$

$$sidec =, \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & sidec_{11} & sidec_{12} & sidec_{13} & sidec_{14} & sidec_{15} & sidec_{16} \end{bmatrix}$$

$j\_of\_c =,$

$$\begin{bmatrix} 1 & 2 & 3 & 3 & 4 & 5 & 6 & 6 & 7 & 8 & j\_of\_c_{11} & j\_of\_c_{12} & j\_of\_c_{13} & j\_of\_c_{14} & j\_of\_c_{15} & j\_of\_c_{16} \end{bmatrix}$$

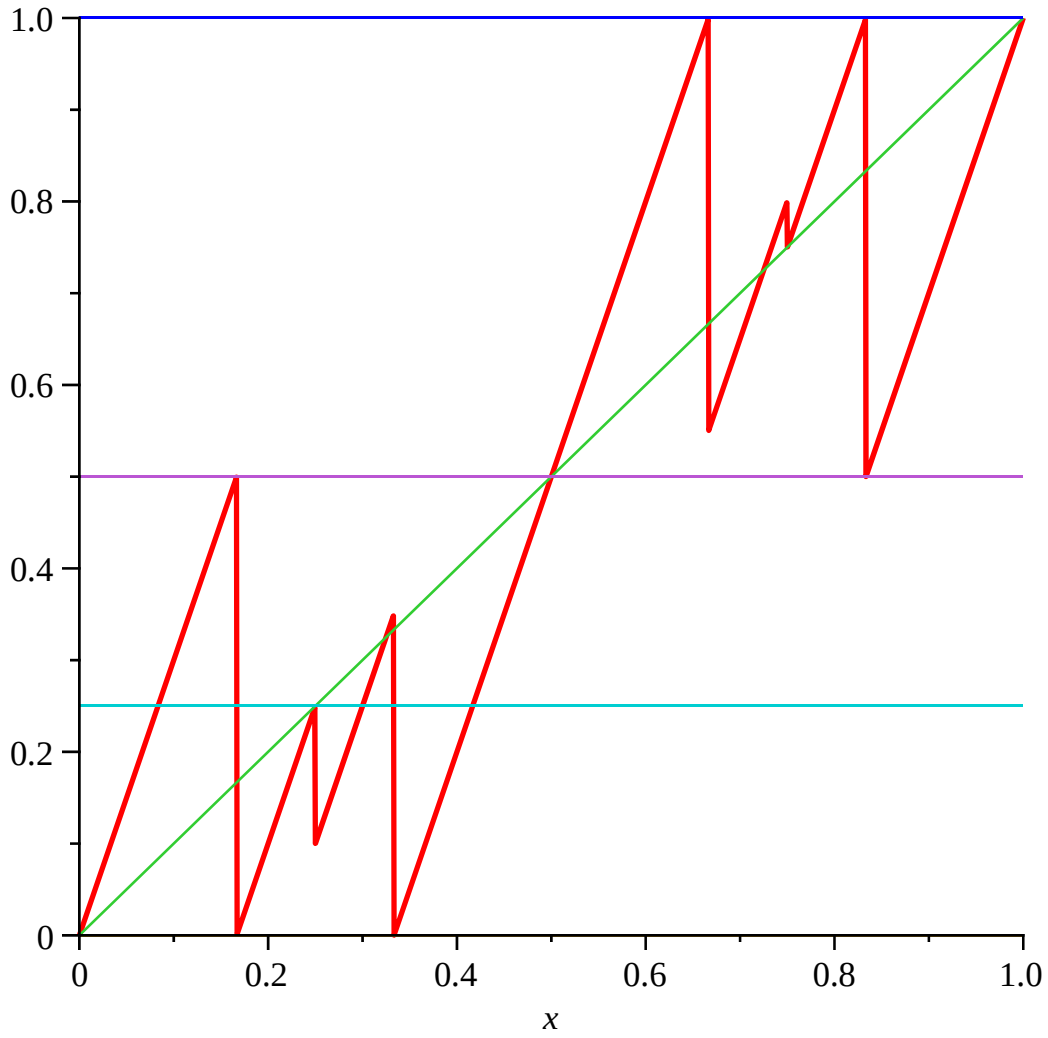
$$uint\_of\_x := x \rightarrow piecewise\left(x < b_2, 1, x < b_3, 2, x < b_4, 3, x < b_5, 4, x < b_6, 5, x < b_7, 6, x < b_8, 7, x < b_9, 8, x < b_{10}, 9, x < b_{11}, 10, 11\right)$$

$$int\_of\_x := x \rightarrow piecewise\left(x \le b_2, 1, x \le b_3, 2, x \le b_4, 3, x \le b_5, 4, x \le b_6, 5, x \le b_7, 6, x \le b_8, 7, x \le b_9, 8, x \le b_{10}, 9, x \le b_{11}, 10, 11\right)$$

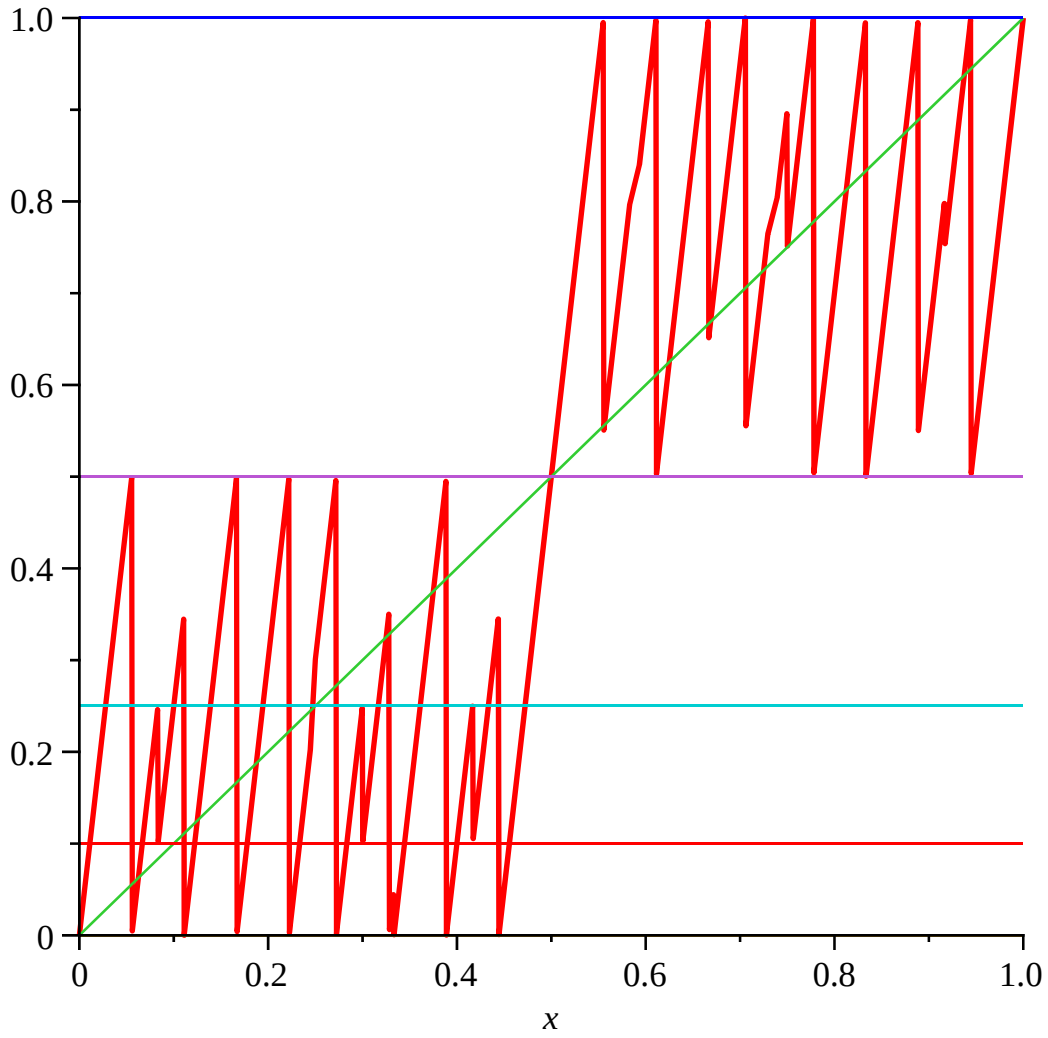
$$uT := x \rightarrow \beta_{uint\_of\_x(x)}\, x - a_{uint\_of\_x(x)}$$

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

$$Tc =, \left[0.500000, 0.250000, 0.100000, 0.350000, 0.500000, 0.500000, 0.550000, 0.800000, 0.750000, 0.500000, Tc_{11}, Tc_{12}\right]$$




```
> plot([' T(T(x))' , x, 0, 1, Tc[1] , Tc[2] , Tc[3] ] , x=0..1, thickness=[2, 1, 1, 1, 1, 1, 1, 1]);
T(T(c[2]));
```



0.250000 **(1)**

```
>
> ud:=vector(50):Digits:=100;NN:=50;#Expansion with variable
  slopes
  d:=vector(50):
  xx:=c[1];
  xxt:=xx:
  bet:=1:
  for i from 1 to NN do
  bet:=bet/beta[uint_of_x(xxt)];
  ud[i]:=a[uint_of_x(xxt)];
  udb[i]:=a[uint_of_x(xxt)]*bet;
  xxt:=uT(xxt);
  od:

  xxt:=xx:
  bet:=1:
  for i from 1 to NN do
  bet:=bet/beta[int_of_x(xxt)];
  d[i]:=a[int_of_x(xxt)];
  db[i]:=a[int_of_x(xxt)]*bet;
  xxt:=T(xxt);
```

```
od:
print(ud);
uls_it_x:=evalf(sum(udb[j1],j1=1..NN));
print(d);
ls_it_x:=evalf(sum(db[j1],j1=1..NN));
terr:=xx-uls_it_x;
err:=xx-ls_it_x;
```

$$Digits := 100$$

$$NN := 50$$

$$xx := 0.166667$$

[0.500000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,
0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,
0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,
0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,
0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,
0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,
0.000000]

$$uIs\_it\_x := 0.166667$$

[0.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000,
1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000,
1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000,
1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000,
1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000,
1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000,
1.000000]

$$Is\_it\_x := 0.166667$$

$$terr := 0.000000$$

$$err := 6.96478\ 10^{-25} \quad \textbf{(2)}$$

```
>
>
> NN:=70;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 Kc do
  xxt:=c[i];
  bet:=1:
       for n from 1 to NN+1 do
        if sidec[i]=1 then intx:=uint_of_x(xxt) else intx:=
  int_of_x(xxt) fi;
          bet_real:=bet;
          bet:=bet/beta[intx];
```

```
          dcb[i,n]:=a[intx]*bet;

        if sidec[i]=0 then
             for ii from 1 to Kc do
                  if xxt>c[ii]+10^(-12) then cc[i,ii,n]:=1*
bet_real else cc[i,ii,n]:=0 fi;
             od;
             if intx=1 then Sc[i,n]:= 0
                          else Sc[i,n]:=sum(1/(beta[j7]),j7=1..
intx-1)*bet_real fi;
             #bc[i,n]:=b[intx]*bet_real;
             ################
                    else
             for ii from 1 to Kc do
                  if xxt<c[ii]-10^(-12) then cc[i,ii,n]:=1*
bet_real else cc[i,ii,n]:=0 fi;
             od;
              if intx=N then Sc[i,n]:= 0
                          else Sc[i,n]:=sum(1/(beta[j8]),j8=
intx+1..N)*bet_real fi;
             #bc[i,n]:=b[intx+1]*bet_real;
      fi;
      valc[i,n]:=xxt;
      betc[i,n]:=bet_real;
      if sidec[i]=1 then
                    Rounding:=infinity:
                    xxt:=uT(xxt)
                   else
                     Rounding:=0;
                    xxt:=T(xxt)
      fi;
       Rounding:=nearest:
      od:
Is_it_x:=sum(dcb[i,j1],j1=1..NN);
od;

for i from 1 to Kc do
S[i]:=evalf(sum(Sc[i,j21+1],j21=1..NN));
#sum_b[i]:=evalf(sum(bc[i,j2+1],j2=1..NN));
od;
for i from 1 to Kc do
for j from 1 to Kc do
SS[i,j]:=evalf(sum(cc[i,j,j1+1],j1=1..NN));

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

$$NN := 70$$

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

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

$$xxt := 0.166667$$

$$bet := 1$$

$$Is\_it\_x := 0.166667$$

$$xxt := 0.250000$$

$$bet := 1$$

$$Is\_it\_x := 0.250000$$

$$xxt := 0.250000$$

$$bet := 1$$

$$Is\_it\_x := 0.250000$$

$$xxt := 0.333333$$

$$bet := 1$$

$$Is\_it\_x := 0.333333$$

$$xxt := 0.500000$$

$$bet := 1$$

$$Is\_it\_x := 0.500000$$

$$xxt := 0.500000$$

$$bet := 1$$

$$Is\_it\_x := 0.500000$$

$$xxt := 0.666667$$

$$bet := 1$$

$$Is\_it\_x := 0.666667$$

$$xxt := 0.750000$$

$$bet := 1$$

$$Is\_it\_x := 0.750000$$

$$xxt := 0.750000$$

$$bet := 1$$

$$Is\_it\_x := 0.750000$$

$$xxt := 0.833333$$

$$bet := 1$$

$$Is\_it\_x := 0.833333$$

$$S_1 := 0.625000$$

$$S_2 := 0.166667$$

$$S_3 := 1.064103$$

$$S_4 := 0.350000$$

$S_5 := 0.625000$

$S_6 := 0.500000$

$S_7 := 0.450000$

$S_8 := 1.017901$

$S_9 := 0.166667$

$S_{10} := 0.500000$

$SS[, 1, 1,] =, 0.500000$

$SS[, 1, 2,] =, 0.500000$

$SS[, 1, 3,] =, 0.500000$

$SS[, 1, 4,] =, 0.500000$

$SS[, 1, 5,] =, 0.000000$

$SS[, 1, 6,] =, 0.000000$

$SS[, 1, 7,] =, 0.000000$

$SS[, 1, 8,] =, 0.000000$

$SS[, 1, 9,] =, 0.000000$

$SS[, 1, 10,] =, 0.000000$

$SS[, 2, 1,] =, 0.500000$

$SS[, 2, 2,] =, 0.000000$

$SS[, 2, 3,] =, 0.000000$

$SS[, 2, 4,] =, 0.000000$

$SS[, 2, 5,] =, 0.000000$

$SS[, 2, 6,] =, 0.000000$

$SS[, 2, 7,] =, 0.000000$

$SS[, 2, 8,] =, 0.000000$

$SS[, 2, 9,] =, 0.000000$

$SS[, 2, 10,] =, 0.000000$

$SS[, 3, 1,] =, 0.346154$

$SS[, 3, 2,] =, 0.346154$

$SS[, 3, 3,] =, 0.346154$

$SS[, 3, 4,] =, 0.500000$

$SS[, 3, 5,] =, 0.500000$

$SS[, 3, 6,] =, 0.500000$

$SS[, 3, 7,] =, 0.500000$

$SS[, 3, 8,] =, 0.500000$

$SS[, 3, 9,] =, 0.500000$

$SS[, 3, 10,] =, 0.500000$

$SS[, 4, 1,] =, 0.350000$

$SS[, 4, 2,] =, 0.350000$

$SS[, 4, 3,] =, 0.350000$

SS[, 4, 4,] =, 0.350000
SS[, 4, 5,] =, 0.000000
SS[, 4, 6,] =, 0.000000
SS[, 4, 7,] =, 0.000000
SS[, 4, 8,] =, 0.000000
SS[, 4, 9,] =, 0.000000
SS[, 4, 10,] =, 0.000000
SS[, 5, 1,] =, 0.500000
SS[, 5, 2,] =, 0.500000
SS[, 5, 3,] =, 0.500000
SS[, 5, 4,] =, 0.500000
SS[, 5, 5,] =, 0.000000
SS[, 5, 6,] =, 0.000000
SS[, 5, 7,] =, 0.000000
SS[, 5, 8,] =, 0.000000
SS[, 5, 9,] =, 0.000000
SS[, 5, 10,] =, 0.000000
SS[, 6, 1,] =, 0.000000
SS[, 6, 2,] =, 0.000000
SS[, 6, 3,] =, 0.000000
SS[, 6, 4,] =, 0.000000
SS[, 6, 5,] =, 0.000000
SS[, 6, 6,] =, 0.000000
SS[, 6, 7,] =, 0.500000
SS[, 6, 8,] =, 0.500000
SS[, 6, 9,] =, 0.500000
SS[, 6, 10,] =, 0.500000
SS[, 7, 1,] =, 0.000000
SS[, 7, 2,] =, 0.000000
SS[, 7, 3,] =, 0.000000
SS[, 7, 4,] =, 0.000000
SS[, 7, 5,] =, 0.000000
SS[, 7, 6,] =, 0.000000
SS[, 7, 7,] =, 0.450000
SS[, 7, 8,] =, 0.450000
SS[, 7, 9,] =, 0.450000
SS[, 7, 10,] =, 0.450000
SS[, 8, 1,] =, 0.500000
SS[, 8, 2,] =, 0.500000
SS[, 8, 3,] =, 0.500000

$$SS[,\ 8,\ 4,\ ]\ =,\ 0.500000$$
$$SS[,\ 8,\ 5,\ ]\ =,\ 0.500000$$
$$SS[,\ 8,\ 6,\ ]\ =,\ 0.500000$$
$$SS[,\ 8,\ 7,\ ]\ =,\ 0.487037$$
$$SS[,\ 8,\ 8,\ ]\ =,\ 0.450000$$
$$SS[,\ 8,\ 9,\ ]\ =,\ 0.450000$$
$$SS[,\ 8,\ 10,\ ]\ =,\ 0.116667$$
$$SS[,\ 9,\ 1,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 2,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 3,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 4,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 5,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 6,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 7,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 8,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 9,\ ]\ =,\ 0.000000$$
$$SS[,\ 9,\ 10,\ ]\ =,\ 0.500000$$
$$SS[,\ 10,\ 1,\ ]\ =,\ 0.000000$$
$$SS[,\ 10,\ 2,\ ]\ =,\ 0.000000$$
$$SS[,\ 10,\ 3,\ ]\ =,\ 0.000000$$
$$SS[,\ 10,\ 4,\ ]\ =,\ 0.000000$$
$$SS[,\ 10,\ 5,\ ]\ =,\ 0.000000$$
$$SS[,\ 10,\ 6,\ ]\ =,\ 0.000000$$
$$SS[,\ 10,\ 7,\ ]\ =,\ 0.500000$$
$$SS[,\ 10,\ 8,\ ]\ =,\ 0.500000$$
$$SS[,\ 10,\ 9,\ ]\ =,\ 0.500000$$
$$SS[,\ 10,\ 10,\ ]\ =,\ 0.500000 \qquad \textbf{(3)}$$

>

>

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

MM[i,j]:=-SS[j,i];
MMM[i,j]:=-SS[j,i];
od;od;
print(`MM = `,MM);
print(`det MM = `,det(MM));
print(1/beta[j_of_c[5]]);
```

```
print(`eigenvalues MM =`,eigenvalues(MM));
print(`1/ average beta = `,1/ave_beta);
eigenvectors(MM);
ve:=vector(Kc,[]):
for i from 1 to Kc do
ve[i]:=1;

MMM[i,i]:=MMM[i,i]+1.0;
od:

print(`MMM =`,MMM);
print(`det MMM = `,det(MMM));
print(ve);_t:='_t';



DD:=linsolve(MMM,ve);
sum((S[ii7]-1/beta[j_of_c[ii7]])*DD[ii7],ii7=1..Kc)-(1-sum
(1/beta[i8],i8=1..N));
MMMv:=concat(MMM,ve);
Mg:=gausselim(MMMv);
```

$$MM = , \begin{bmatrix} [-0.500000, -0.500000, -0.346154, -0.350000, -0.500000, -0.000000, \\ -0.000000, -0.500000, -0.000000, -0.000000], \\ [-0.500000, -0.000000, -0.346154, -0.350000, -0.500000, -0.000000, -0.000000, \\ -0.500000, -0.000000, -0.000000], \\ [-0.500000, -0.000000, -0.346154, -0.350000, -0.500000, -0.000000, -0.000000, \\ -0.500000, -0.000000, -0.000000], \\ [-0.500000, -0.000000, -0.500000, -0.350000, -0.500000, -0.000000, -0.000000, \\ -0.500000, -0.000000, -0.000000], \\ [-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.000000, -0.000000, \\ -0.500000, -0.000000, -0.000000], \\ [-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.000000, -0.000000, \\ -0.500000, -0.000000, -0.000000], \\ [-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.500000, -0.450000, \\ -0.487037, -0.000000, -0.500000], \\ [-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.500000, -0.450000, \\ -0.450000, -0.000000, -0.500000], \\ [-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.500000, -0.450000, \\ -0.450000, -0.000000, -0.500000], \\ [-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.500000, -0.450000, \\ -0.116667, -0.500000, -0.500000] \end{bmatrix}$$

$$det\ MM = , 0$$

0.333333

eigenvalues MM =, -2.129984, 0.364784, $1.06887\ 10^{-13}$, 0.169046, -1.000000, $-2.63418\ 10^{-9}$, $2.63418\ 10^{-9}$, $9.02699\ 10^{-17}$, 0.000000, 0.000000

$$\textit{1/ average beta} = , \frac{1}{ave\_beta}$$

$\Big[0.000000, 3, \Big\{\big[-4.533276, -2.25000\ 10^{-98}, -2.25000\ 10^{-98}, 3.974304, 1.751264, 1.751264, -0.324308, -1.26000\ 10^{-98}, -1.26000\ 10^{-98}, -1.459387\big], \big[4.533276, 2.25000\ 10^{-98}, 2.25000\ 10^{-98}, -3.974304, -1.751264, -1.751264, 0.324308, 1.26000\ 10^{-98}, 1.26000\ 10^{-98}, 1.459387\big], \big[-0.920208, -5.80449\ 10^{-99}, -4.72495\ 10^{-99}, 0.565040, 0.524680, 0.524680, -0.097163, -3.58527\ 10^{-99}, -3.58527\ 10^{-99}, -0.437233\big]\Big\}\Big], \Big[-2.72727\ 10^{-101}, 1, \big\{\big[-0.693638, -3.45300\ 10^{-99}, -3.45300\ 10^{-99}, 1.024483, -0.023500, -0.023500, 0.403395, -2.08760\ 10^{-99}, -2.08760\ 10^{-99}, -0.339555\big]\big\}\Big], [0.364784, 1, \{[0.139420, -0.376126, -0.376126, -0.217496, 0.634194, 0.634194, -0.077774, -0.086562, -0.086562, -0.047013]\}], [0.169046, 1, \{[-0.630856, 0.322231, 0.322231, 0.028974, 0.907647, 0.907647, -0.491267, -0.629100, -0.629100, -0.008855]\}], \Big[-1.000000, 1, \big\{\big[-0.943423, -0.628949, -0.628949, -0.725710, -2.73835\ 10^{-35}, -2.73835\ 10^{-35}, 0.652243, 0.628949, 0.628949, 0.733774\big]\big\}\Big], \big[1.95811\ 10^{-100}, 1, \big\{\big[0.920208, 5.30000\ 10^{-99}, 5.30000\ 10^{-99}, -0.565040, -0.524680, -0.524680, 0.097163, 3.00000\ 10^{-99}, 3.00000\ 10^{-99}, 0.437233\big]\big\}\big], [-2.129984, 1, \{[-0.686373, -0.555883, -0.555883, -0.596034, -0.272908, -0.272908, -0.617246, -0.606697, -0.606697, -0.654170]\}]$

MMM =, [[0.500000, -0.500000, -0.346154, -0.350000, -0.500000, -0.000000, -0.000000, -0.500000, -0.000000, -0.000000],
[-0.500000, 1.000000, -0.346154, -0.350000, -0.500000, -0.000000, -0.000000, -0.500000, -0.000000, -0.000000],
[-0.500000, -0.000000, 0.653846, -0.350000, -0.500000, -0.000000, -0.000000, -0.500000, -0.000000, -0.000000],
[-0.500000, -0.000000, -0.500000, 0.650000, -0.500000, -0.000000, -0.000000, -0.500000, -0.000000, -0.000000],
[-0.000000, -0.000000, -0.500000, -0.000000, 1.000000, -0.000000, -0.000000, -0.500000, -0.000000, -0.000000],
[-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, 1.000000, -0.000000, -0.500000, -0.000000, -0.000000],
[-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.500000, 0.550000, -0.487037, -0.000000, -0.500000],
[-0.000000, -0.000000, -0.500000, -0.000000, -0.000000, -0.500000, -0.450000, 0.550000, -0.000000, -0.500000],

[−0.000000, −0.000000, −0.500000, −0.000000, −0.000000, −0.500000, −0.450000, −0.450000, 1.000000, −0.500000],
[−0.000000, −0.000000, −0.500000, −0.000000, −0.000000, −0.500000, −0.450000, −0.116667, −0.500000, 0.500000]]

$$det\ MMM = , -3.91984\ 10^{-34}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\_t := \_t$$

$$DD := [-0.997931, -0.665287, -0.665287, -0.767639, -1.54697\ 10^{-34}, -1.54697\ 10^{-34}, -1.384146, -1.334713, -1.334713, -1.557165]$$

$$-0.124741$$

$$MMMv := [[0.500000, -0.500000, -0.346154, -0.350000, -0.500000, 0.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1],$$
$$[-0.500000, 1.000000, -0.346154, -0.350000, -0.500000, 0.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1],$$
$$[-0.500000, 0.000000, 0.653846, -0.350000, -0.500000, 0.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1],$$
$$[-0.500000, 0.000000, -0.500000, 0.650000, -0.500000, 0.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1],$$
$$[0.000000, 0.000000, -0.500000, 0.000000, 1.000000, 0.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1],$$
$$[0.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1],$$
$$[0.000000, 0.000000, -0.500000, 0.000000, 0.000000, -0.500000, 0.550000, -0.487037, 0.000000, -0.500000, 1],$$
$$[0.000000, 0.000000, -0.500000, 0.000000, 0.000000, -0.500000, -0.450000, 0.550000, 0.000000, -0.500000, 1],$$
$$[0.000000, 0.000000, -0.500000, 0.000000, 0.000000, -0.500000, -0.450000, -0.450000, 1.000000, -0.500000, 1],$$
$$[0.000000, 0.000000, -0.500000, 0.000000, 0.000000, -0.500000, -0.450000, -0.116667, -0.500000, 0.500000, 1]]$$

$$Mg := [[0.500000, -0.500000, -0.346154, -0.350000, -0.500000, 0.000000, 0.000000, -0.500000, 0.000000, 0.000000, 1.000000],$$
$$[0, 0.500000, -0.692308, -0.700000, -1.000000, 0.000000, 0.000000, -1.000000, 0.000000, 0.000000, 2.000000],$$
$$[0, 0, -1.538462, -0.400000, -2.000000, 0.000000, 0.000000, -2.000000, 0.000000, 0.000000, 4.000000],$$
$$[0, 0, 0, -1.300000, -1.500000, 0.000000, 0.000000, -1.500000, 0.000000, 0.000000,$$

(4)

3.000000],

[0.000000, 0.000000, 0, 0, 1.500000, 0.000000, 0.000000, $3.48791\ 10^{-34}$, 0.000000,

0.000000, $-6.97581\ 10^{-34}$],

[0.000000, 0.000000, 0, 0, 0, 1.000000, 0.000000, $2.32527\ 10^{-34}$, 0.000000, 0.000000,

$-4.65054\ 10^{-34}$],

[0.000000, 0.000000, 0, 0, 0, 0, 0.550000, 0.012963, 0.000000, −0.500000, $-6.97581\ 10^{-34}$

],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 1.060606, 0.000000, −0.909091, $-1.26833\ 10^{-33}$],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 0, 1.000000, −0.857143, $-1.19585\ 10^{-33}$],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 0, 0, $8.95963\ 10^{-34}$, $-1.39516\ 10^{-33}$]]

```
> M2:=submatrix(Mg, 1..Kc, 1..Kc); v2:=col(Mg, Kc+1);
  M2[Kc, Kc]:=0: M2[5, 8]:=0: M2[6, 8]:=0: #M2[7, 8]:=0:
  for i from 5 to Kc do v2[i]:=0; od:
  print(M2, v2);
  eigenvalues(M2);


  DDo:=linsolve(M2, v2);
```

$M2 :=$ [[0.500000, −0.500000, −0.346154, −0.350000, −0.500000, 0.000000, 0.000000,

−0.500000, 0.000000, 0.000000],

[0, 0.500000, −0.692308, −0.700000, −1.000000, 0.000000, 0.000000, −1.000000,

0.000000, 0.000000],

[0, 0, −1.538462, −0.400000, −2.000000, 0.000000, 0.000000, −2.000000, 0.000000,

0.000000],

[0, 0, 0, −1.300000, −1.500000, 0.000000, 0.000000, −1.500000, 0.000000, 0.000000],

[0.000000, 0.000000, 0, 0, 1.500000, 0.000000, 0.000000, $3.48791\ 10^{-34}$, 0.000000,

0.000000],

[0.000000, 0.000000, 0, 0, 0, 1.000000, 0.000000, $2.32527\ 10^{-34}$, 0.000000, 0.000000],

[0.000000, 0.000000, 0, 0, 0, 0, 0.550000, 0.012963, 0.000000, −0.500000],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 1.060606, 0.000000, −0.909091],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 0, 1.000000, −0.857143],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 0, 0, $8.95963\ 10^{-34}$]]

$v2 :=$ [1.000000, 2.000000, 4.000000, 3.000000, $-6.97581\ 10^{-34}$, $-4.65054\ 10^{-34}$,

$-6.97581\ 10^{-34}$, $-1.26833\ 10^{-33}$, $-1.19585\ 10^{-33}$, $-1.39516\ 10^{-33}$]

[[0.500000, −0.500000, −0.346154, −0.350000, −0.500000, 0.000000, 0.000000, −0.500000,

0.000000, 0.000000],

[0, 0.500000, −0.692308, −0.700000, −1.000000, 0.000000, 0.000000, −1.000000, 0.000000, 0.000000],

[0, 0, −1.538462, −0.400000, −2.000000, 0.000000, 0.000000, −2.000000, 0.000000, 0.000000],

[0, 0, 0, −1.300000, −1.500000, 0.000000, 0.000000, −1.500000, 0.000000, 0.000000],

[0.000000, 0.000000, 0, 0, 1.500000, 0.000000, 0.000000, 0, 0.000000, 0.000000],

[0.000000, 0.000000, 0, 0, 0, 1.000000, 0.000000, 0, 0.000000, 0.000000],

[0.000000, 0.000000, 0, 0, 0, 0, 0.550000, 0.012963, 0.000000, −0.500000],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 1.060606, 0.000000, −0.909091],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 0, 1.000000, −0.857143],

[0.000000, 0.000000, 0, 0, 0, 0, 0, 0, 0, 0]],

$$\begin{bmatrix} 1.000000 & 2.000000 & 4.000000 & 3.000000 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

0.500000, 0.500000, −1.538462, −1.300000, 1.500000, 1.000000, 0.550000, 1.060606, 1.000000, 0.000000

$$DDo := \left[\_t_1,\ 0.666667\ \_t_1,\ 1.\ 10^{-99} + 0.666667\ \_t_1,\ 0.769231\ \_t_1,\ -0.000000,\ -0.000000,\ -0.691358\ \_t_1 - 2.074074,\ -0.666667\ \_t_1 - 2.000000,\ -0.666667\ \_t_1 - 2.000000,\ -0.777778\ \_t_1 - 2.333333 \right] \quad \textbf{(5)}$$

```
> _t[1]:=-0.5;
  DDo1 := Vector[row](10, {(1) = _t[1], (2) = .666667*_t[1], (3) =
  .666667*_t[1], (4) = .769231*_t[1], (5) = -0., (6) = -0., (7) =
  -.691358*_t[1]-2.074074, (8) = -.666667*_t[1]-2.000000, (9) =
  -.666667*_t[1]-2.000000, (10) = -.777778*_t[1]-2.333333});
  _t[1]:=-1.9;
  DDo2 := Vector[row](10, {(1) = _t[1], (2) = .666667*_t[1], (3) =
  .666667*_t[1], (4) = .769231*_t[1], (5) = -0., (6) = -0., (7) =
  -.691358*_t[1]-2.074074, (8) = -.666667*_t[1]-2.000000, (9) =
  -.666667*_t[1]-2.000000, (10) = -.777778*_t[1]-2.333333});
  DDo3:=vector(Kc,[-0.943423,-0.628949,-0.628949,-0.725710,0,0,
  0.652243,0.628949,0.628949,0.733774]);
```

$$\_t_1 := -0.500000$$

$$DDo1 := [-0.500000, -0.333334, -0.333334, -0.384616, -0.000000, -0.000000, -1.728395, -1.666667, -1.666667, -1.944444]$$

$$\_t_1 := -1.900000$$

$$DDo2 := [-1.900000, -1.266667, -1.266667, -1.461539, -0.000000, -0.000000, -0.760494, -0.733333, -0.733333, -0.855555]$$

$$DDo3 := [-0.943423, -0.628949, -0.628949, -0.725710, 0, 0, 0.652243, 0.628949, 0.628949, 0.733774] \quad \textbf{(6)}$$

```
> AA:=submatrix(MM,1..Kc/2,1..Kc/2);
```

```
eigenvalues(AA);
AA:=submatrix(MM,Kc/2+1..Kc,Kc/2+1..Kc);
eigenvalues(AA);
```

$$AA := \begin{bmatrix} -0.500000 & -0.500000 & -0.346154 & -0.350000 & -0.500000 \\ -0.500000 & -0.000000 & -0.346154 & -0.350000 & -0.500000 \\ -0.500000 & -0.000000 & -0.346154 & -0.350000 & -0.500000 \\ -0.500000 & -0.000000 & -0.500000 & -0.350000 & -0.500000 \\ -0.000000 & -0.000000 & -0.500000 & -0.000000 & -0.000000 \end{bmatrix}$$

$$-1.552824,\ 2.22045\ 10^{-16},\ 0.356670,\ 0.000000,\ 0.000000$$

$$AA := \begin{bmatrix} -0.000000 & -0.000000 & -0.500000 & -0.000000 & -0.000000 \\ -0.500000 & -0.450000 & -0.487037 & -0.000000 & -0.500000 \\ -0.500000 & -0.450000 & -0.450000 & -0.000000 & -0.500000 \\ -0.500000 & -0.450000 & -0.450000 & -0.000000 & -0.500000 \\ -0.500000 & -0.450000 & -0.116667 & -0.500000 & -0.500000 \end{bmatrix}$$

$$-1.616515,\ 0.216515,\ 1.40507\ 10^{-16},\ -2.87053\ 10^{-9},\ 2.87053\ 10^{-9} \qquad (7)$$

```
>
> density1:=proc(t) local j,i, den;
                    i:='i':

                  den:=1:
                  for j from 1 to Kc do
                  if sidec[j]=0 then
                    den:=den+ DDo1[j]*sum((chi(0,valc[j,i1+1],t))*
betc[j,i1+1],i1=1..50) fi;

                    if sidec[j]=1 then
                     den:=den+ DDo1[j]*sum((uchi(valc[j,i1+1],1,t))*
betc[j,i1+1],i1=1..50) fi;


                    od;
                  return den;
            end proc;
density2:=proc(t) local j,i, den;
                    i:='i':
                    #den:=sum(chi(b[i],b[i+1],t)/beta[i],i=1..N);
                  den:=1:
                  for j from 1 to Kc do
                  if sidec[j]=0 then
                    den:=den+ DDo2[j]*sum((chi(0,valc[j,i1+1],t))*
betc[j,i1+1],i1=1..50) fi;

                    if sidec[j]=1 then
```

```
                    den:=den+ DDo2[j]*sum((uchi(valc[j,i1+1],1,t))*
betc[j,i1+1],i1=1..50) fi;


                  od;
                return den;
          end proc;
density3:=proc(t) local j,i, den;
                  i:='i':
                  #den:=sum(chi(b[i],b[i+1],t)/beta[i],i=1..N);
                den:=0:
                for j from 1 to Kc do
                if sidec[j]=0 then
                  den:=den+ DDo3[j]*sum((chi(0,valc[j,i1+1],t))*
betc[j,i1+1],i1=1..50) fi;

                  if sidec[j]=1 then
                  den:=den+ DDo3[j]*sum((uchi(valc[j,i1+1],1,t))*
betc[j,i1+1],i1=1..50) fi;


                  od;
                return den;
          end proc;

#Normalizing factor

NC:=1:
for j from 1 to Kc do
if sidec[j]=0 then
 NC:=NC+DDo1[j]*sum((valc[j,i1+1])*betc[j,i1+1],i1=1..50) fi;
if sidec[j]=1 then
NC:=NC+DDo1[j]*sum((1-valc[j,i1+1])*betc[j,i1+1],i1=1..50) fi;

 od:
NC2:=1:
for j from 1 to Kc do
if sidec[j]=0 then
 NC2:=NC2+DDo2[j]*sum((valc[j,i1+1])*betc[j,i1+1],i1=1..50) fi;
if sidec[j]=1 then
NC2:=NC2+DDo2[j]*sum((1-valc[j,i1+1])*betc[j,i1+1],i1=1..50) fi;

 od:
NC3:=0:
for j from 1 to Kc do
if sidec[j]=0 then
 NC3:=NC3+DDo3[j]*sum((valc[j,i1+1])*betc[j,i1+1],i1=1..50) fi;
```

```
    if  sidec[j]=1  then
    NC3:=NC3+DDo3[j]*sum((1-valc[j,i1+1])*betc[j,i1+1],i1=1..50)  fi;

     od:
    print(`NC  =  `,NC);
    print(`NC2  =  `,NC2);
    print(`NC3  =  `,NC3);
    NC:=int(abs(density1(t)),t=0..1);
    NC2:=int(abs(density2(t)),t=0..1);
    NC3:=int(abs(density3(t)),t=0..1);


    plot([-(1/NC)*density1(t),-(1/NC2)*density2(t),(1/NC3)*density3
    (t)]
    ,t=0..1-0.000001,color=[black,grey,gray],thickness=2,linestyle=
    [solid,solid,dashdot]);
```

```
density1 := proc(t)
    local j, i, den;
    i := 'i';
    den := 1;
    for j to Kc do
        if sidec[j] = 0 then
            den := den + DDo1[j] * (sum(chi(0, valc[j, i1 + 1], t) * betc[j, i1 + 1], i1 = 1
            ..50))
        end if;
        if sidec[j] = 1 then
            den := den + DDo1[j] * (sum(uchi(valc[j, i1 + 1], 1, t) * betc[j, i1 + 1], i1 = 1
            ..50))
        end if
    end do;
    return den
end proc
density2 := proc(t)
    local j, i, den;
    i := 'i';
    den := 1;
    for j to Kc do
        if sidec[j] = 0 then
            den := den + DDo2[j] * (sum(chi(0, valc[j, i1 + 1], t) * betc[j, i1 + 1], i1 = 1
            ..50))
        end if;
```

```
        if sidec[j] = 1 then
            den := den + DDo2[j] * (sum(uchi(valc[j, i1 + 1], 1, t) * betc[j, i1 + 1], i1 = 1
            ..50))
        end if
    end do;
    return den
end proc
density3 := proc(t)
    local j, i, den;
    i := 'i';
    den := 0;
    for j to Kc do
        if sidec[j] = 0 then
            den := den + DDo3[j] * (sum(chi(0, valc[j, i1 + 1], t) * betc[j, i1 + 1], i1 = 1
            ..50))
        end if;
        if sidec[j] = 1 then
            den := den + DDo3[j] * (sum(uchi(valc[j, i1 + 1], 1, t) * betc[j, i1 + 1], i1 = 1
            ..50))
        end if
    end do;
    return den
end proc
```

$$NC = , -1.067308$$

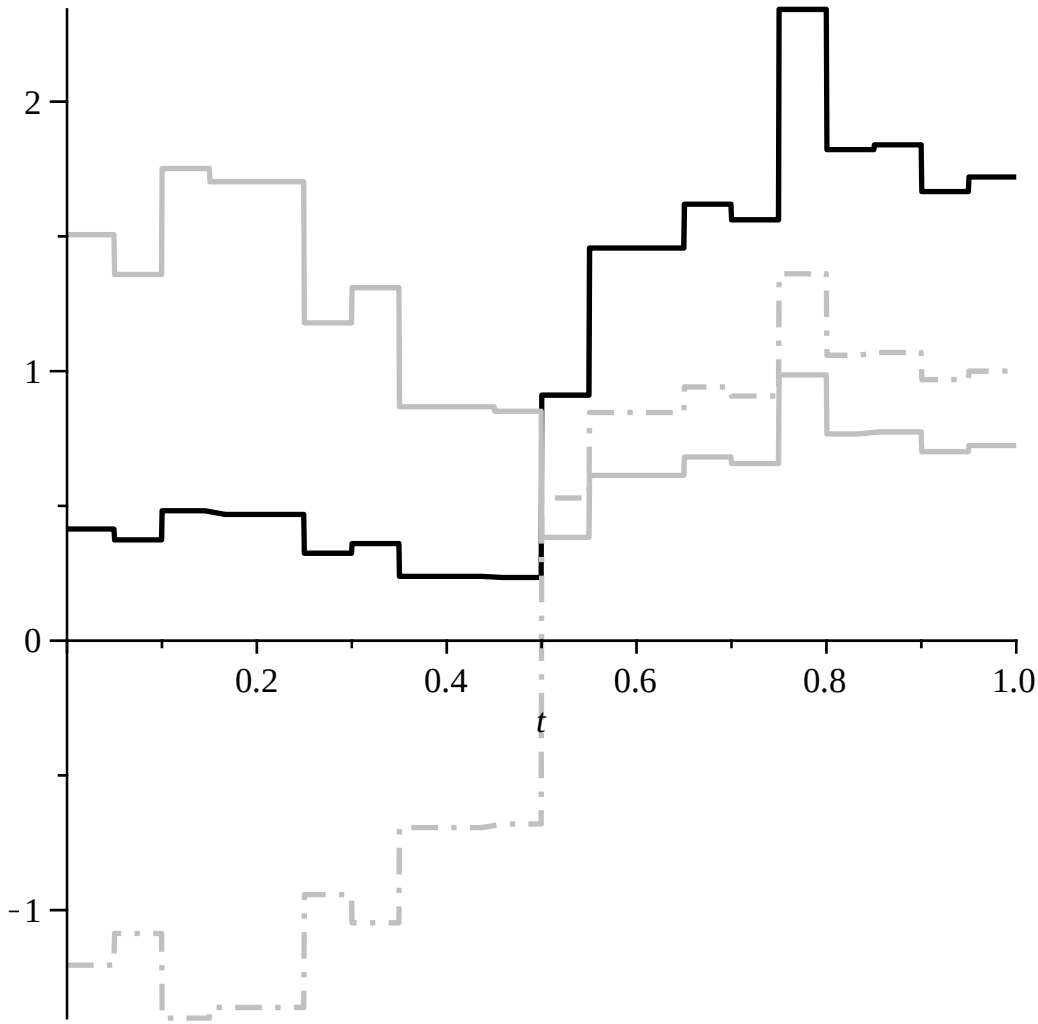
$$NC2 = , -1.115769$$

$$NC3 = , -0.032657$$

$$NC := 1.067308$$

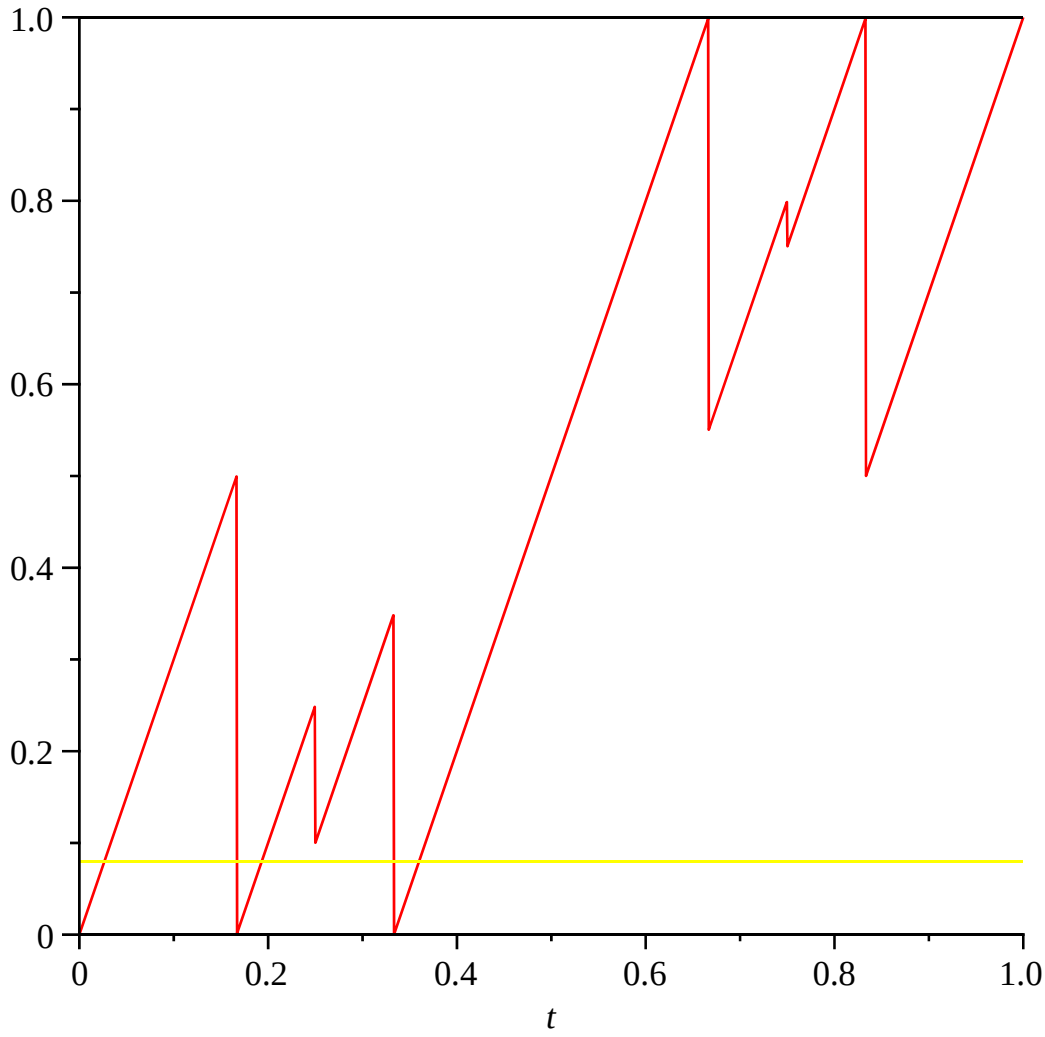
$$NC2 := 1.115769$$

$$NC3 := 0.693053$$




```
> 
> 
#check density1
#preimages
for j6 from 0 to 9 do
y[j6]:=j6/10+(0.1)*rand()/10^12;
od:
for j6 from 0 to 9 do
for i3 from 1 to N do
pre[i3]:=(y[j6]+a[i3])/beta[i3];
od;
plot([T(t),0,1,y[j6]],t=0..1,
color=[red,black,black,yellow]);
su:=0:
for i3 from 1 to N do
if (pre[i3]>=b[i3] and pre[i3]<=b[i3+1]) then
 su:=su+evalf(density1(pre[i3])/beta[i3]);
print(i3);
fi;
od;
err[j6]:=evalf(density1(y[j6])-su);
od;
```

```
for j6 from 0 to 9 do
print(`y =`,y[j6]);
print(`err[`,j6,`]=`,err[j6]);
od;
```



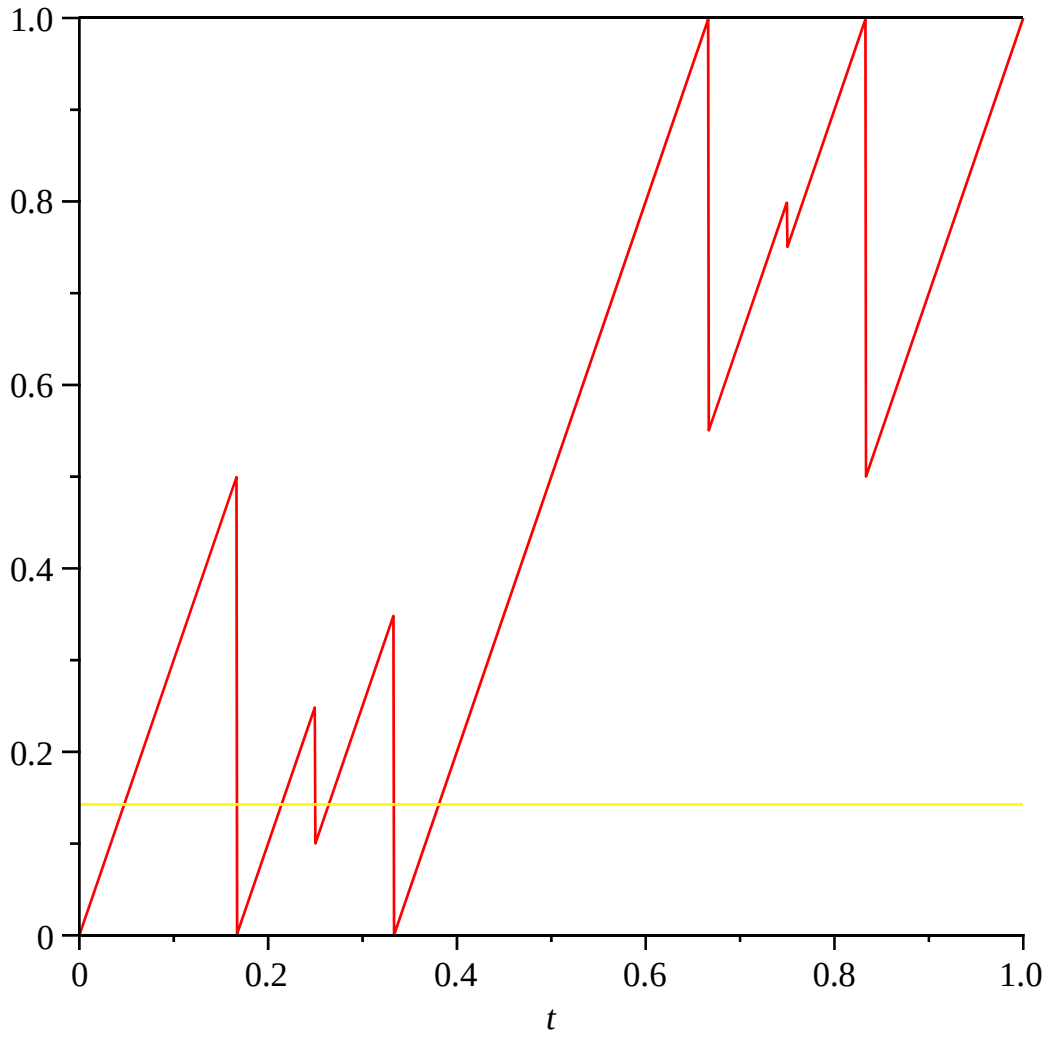
$su := 0$

1

2

4

$err_0 := 7.69231\ 10^{-9}$



$su := 0$

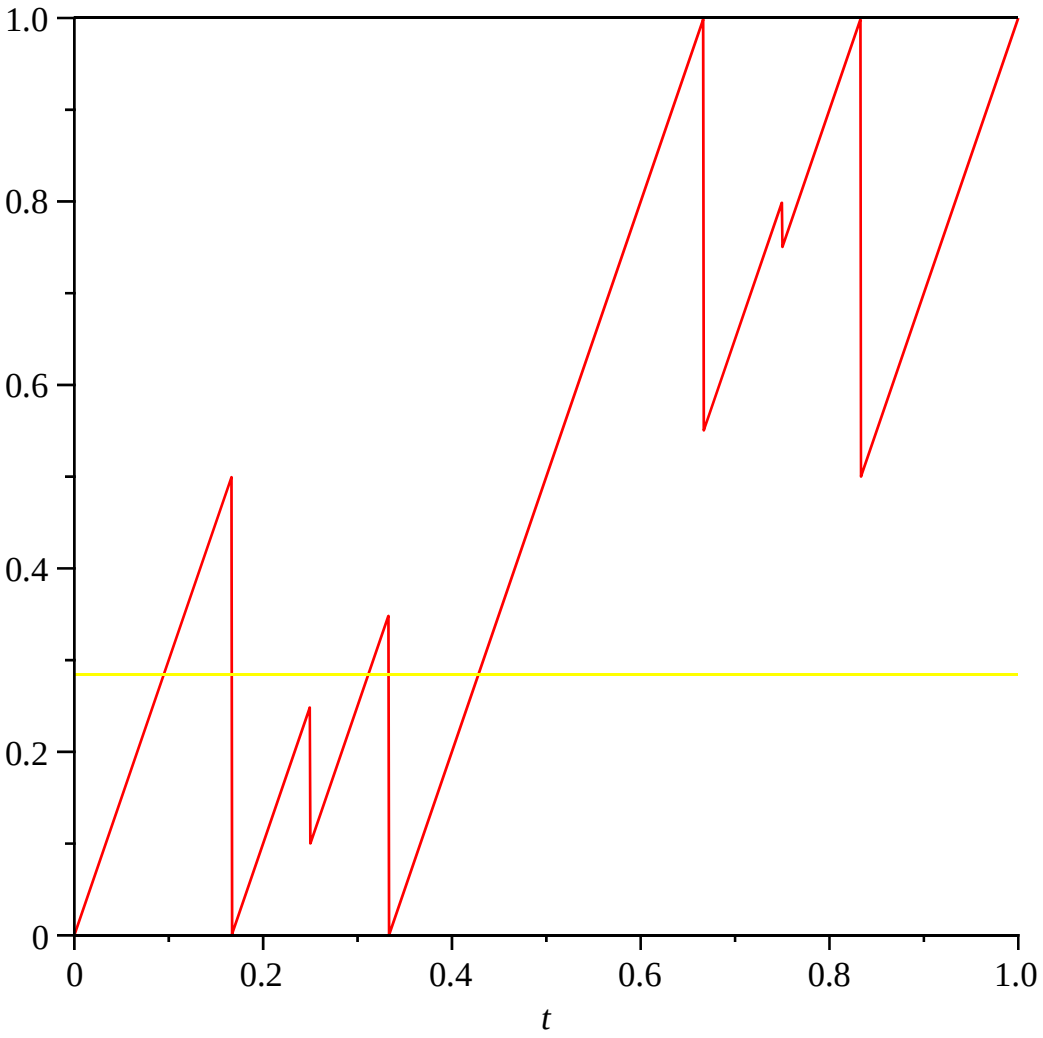
1

2

3

4

$err_1 := -4.29487\ 10^{-8}$



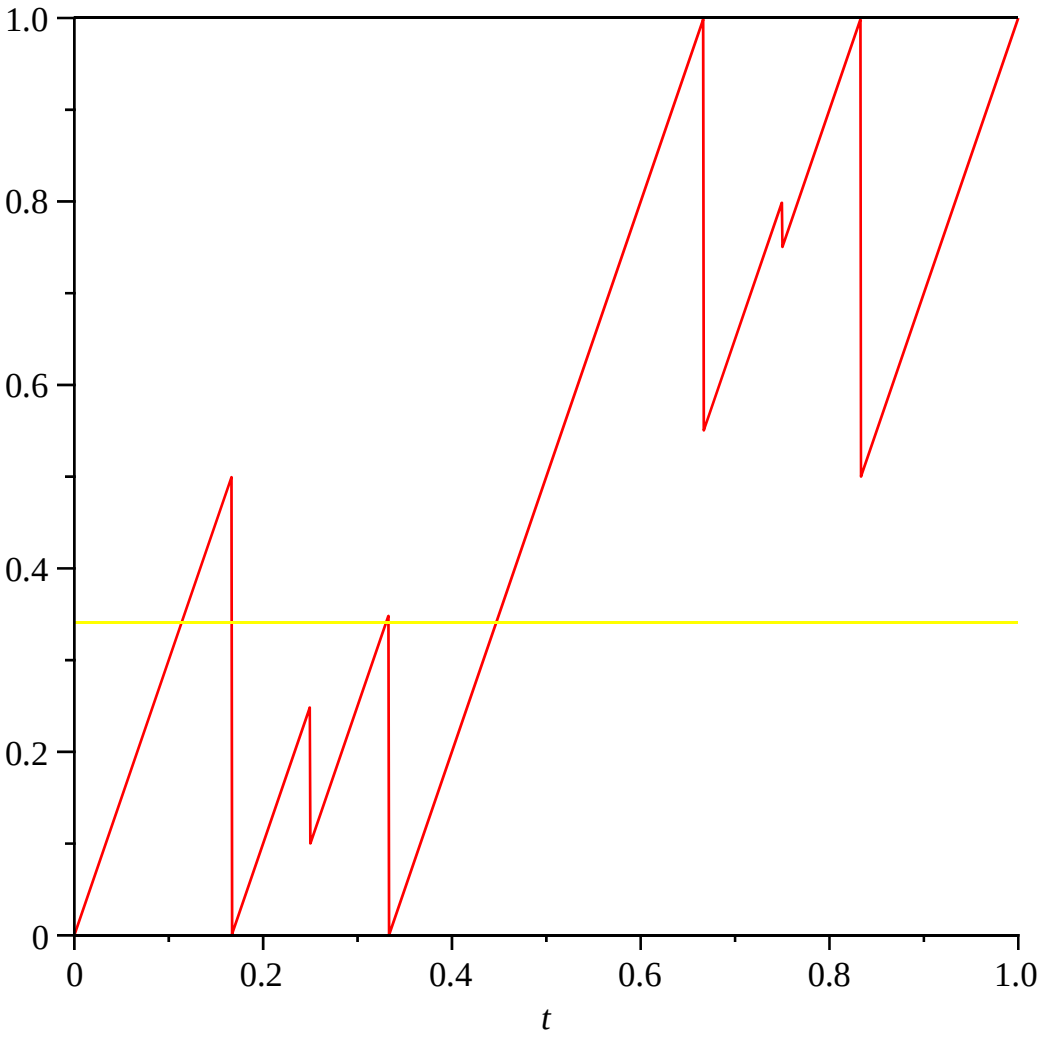
$su := 0$

1

3

4

$err_2 := 7.69231\ 10^{-9}$



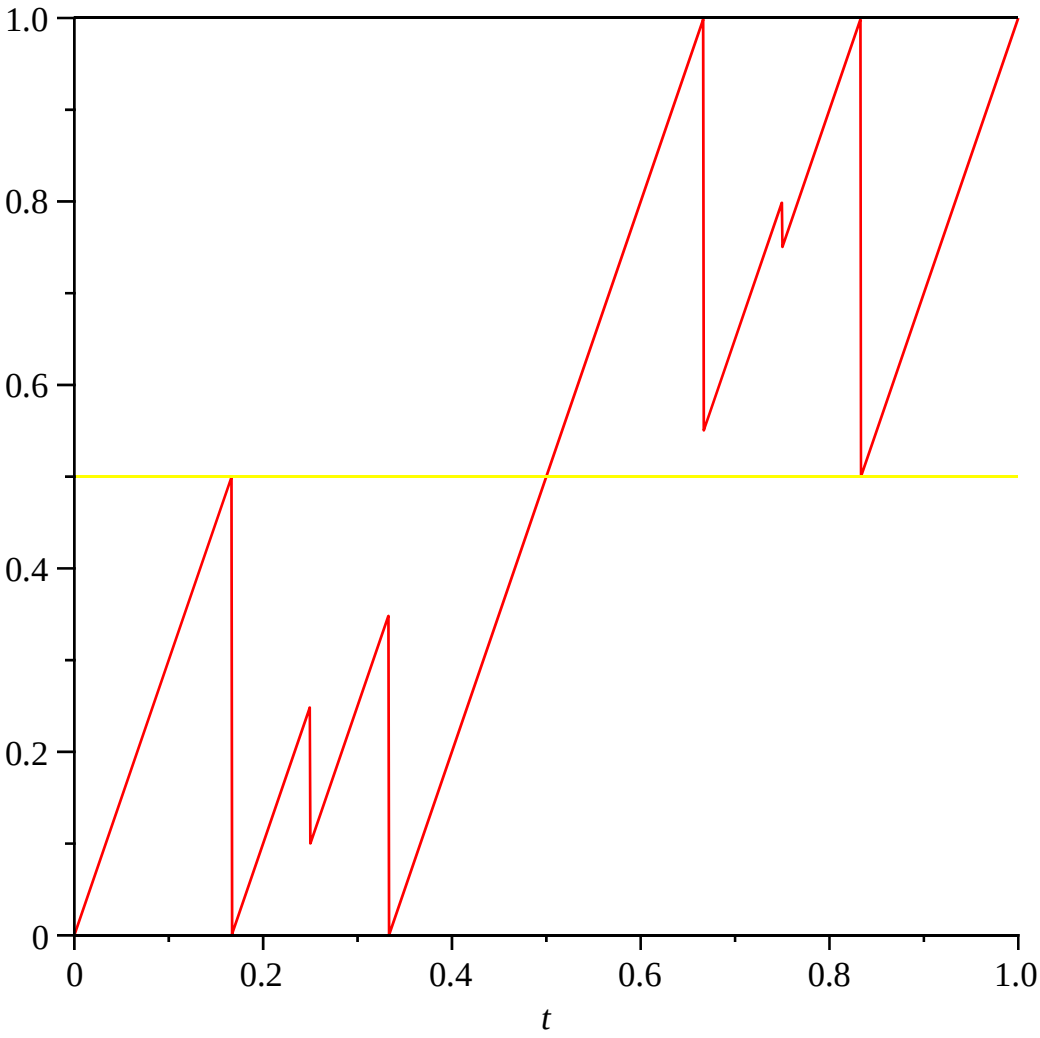
$su := 0$

1

3

4

$err_3 := 7.69231\ 10^{-9}$

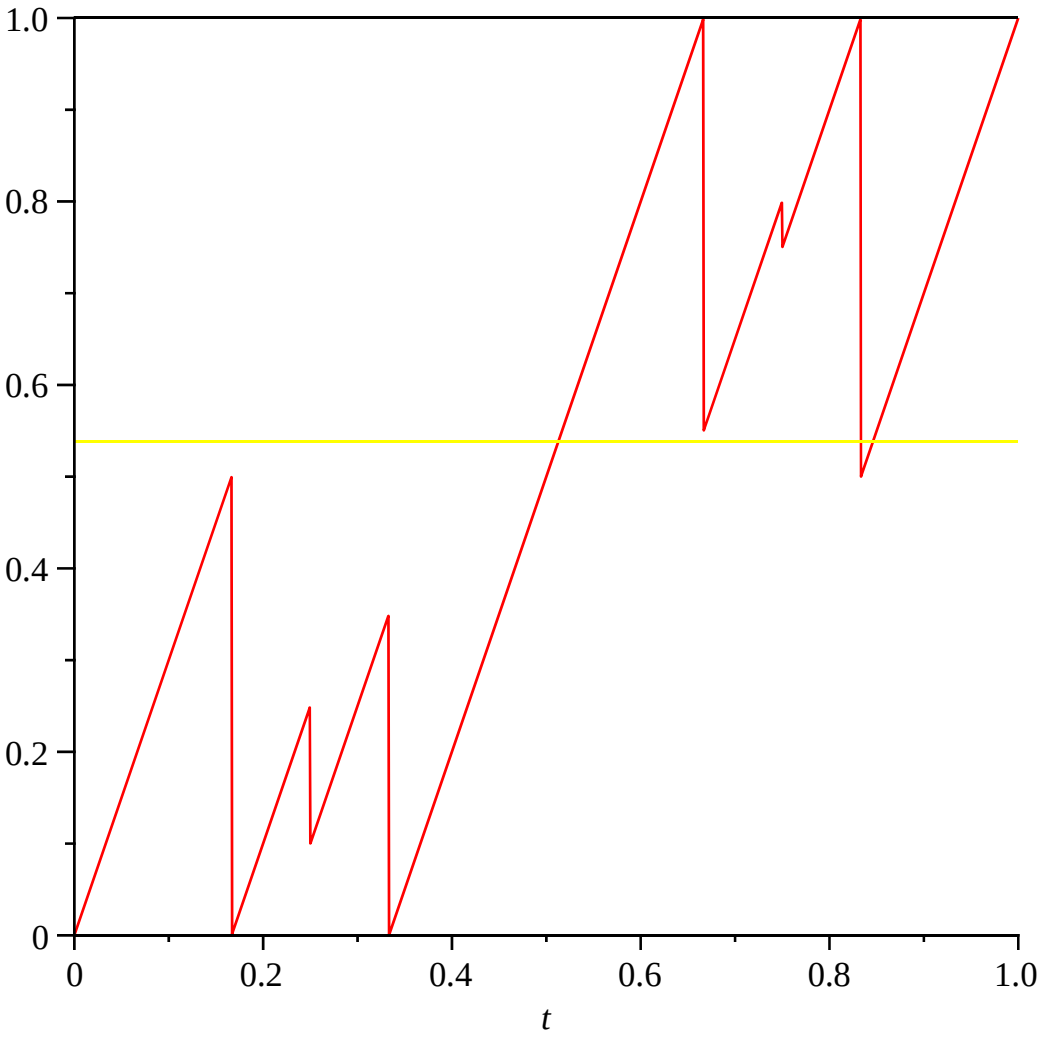


$su := 0$

1

4

$err_4 := 3.26923\ 10^{-8}$

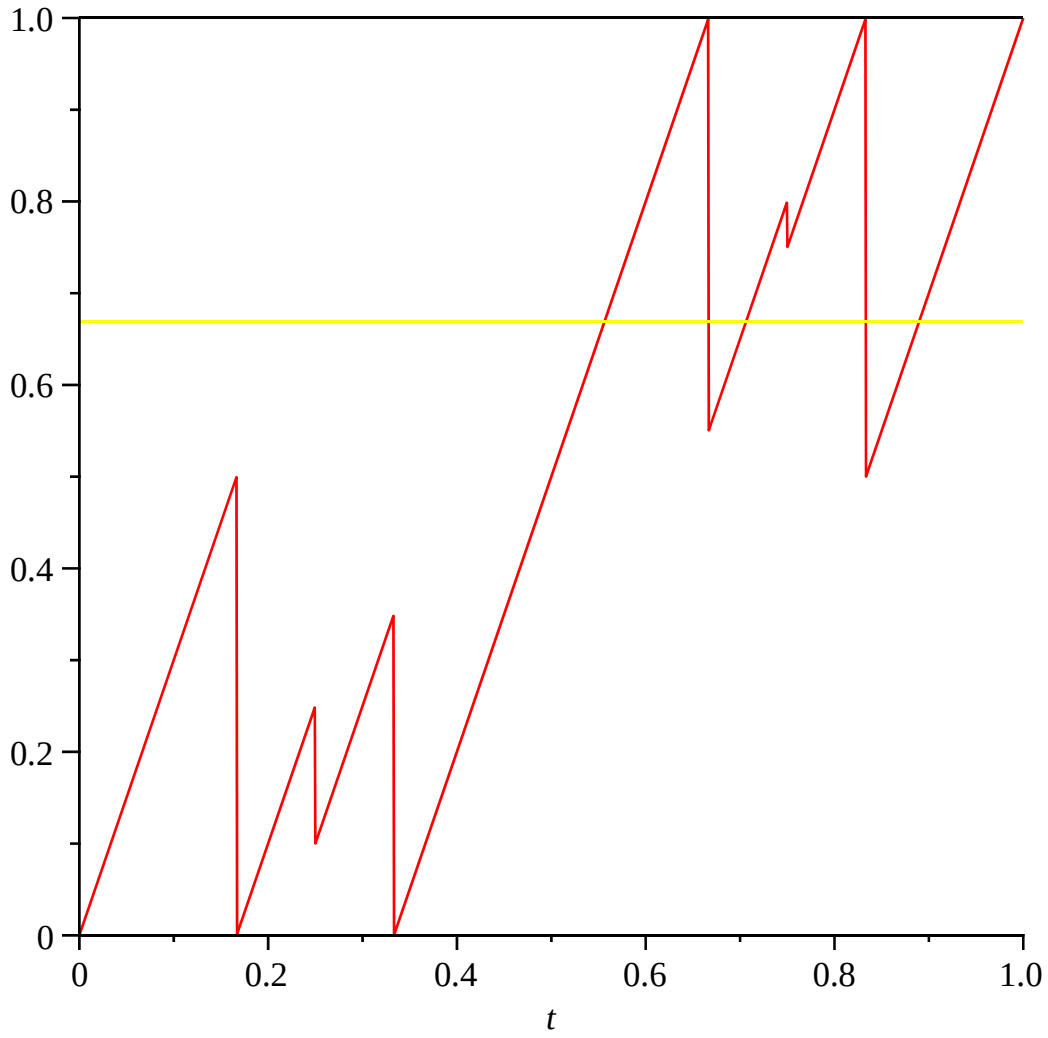


$su := 0$

5

8

$err_5 := 5.83333\ 10^{-8}$



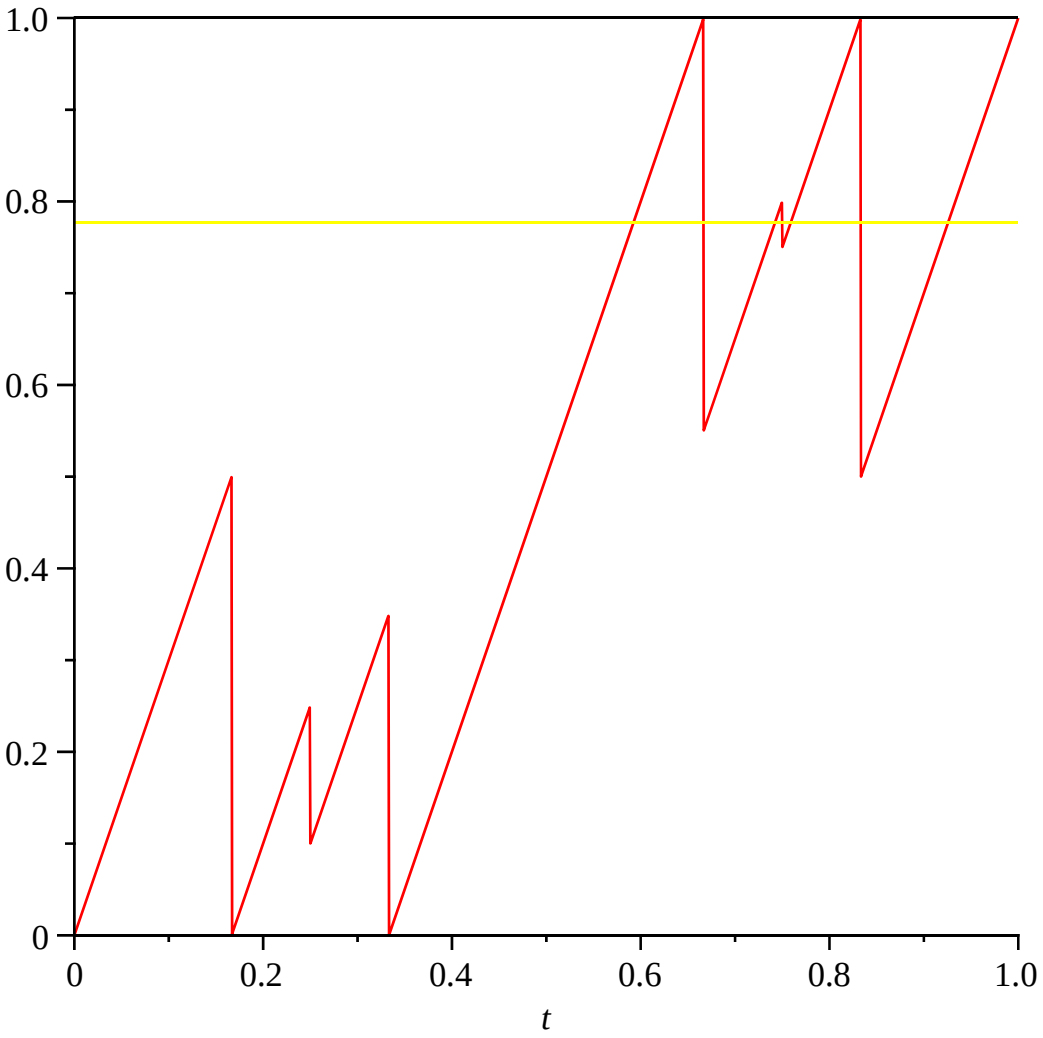
$su := 0$

5

6

8

$err_6 := -3.70370\ 10^{-9}$



$su := 0$

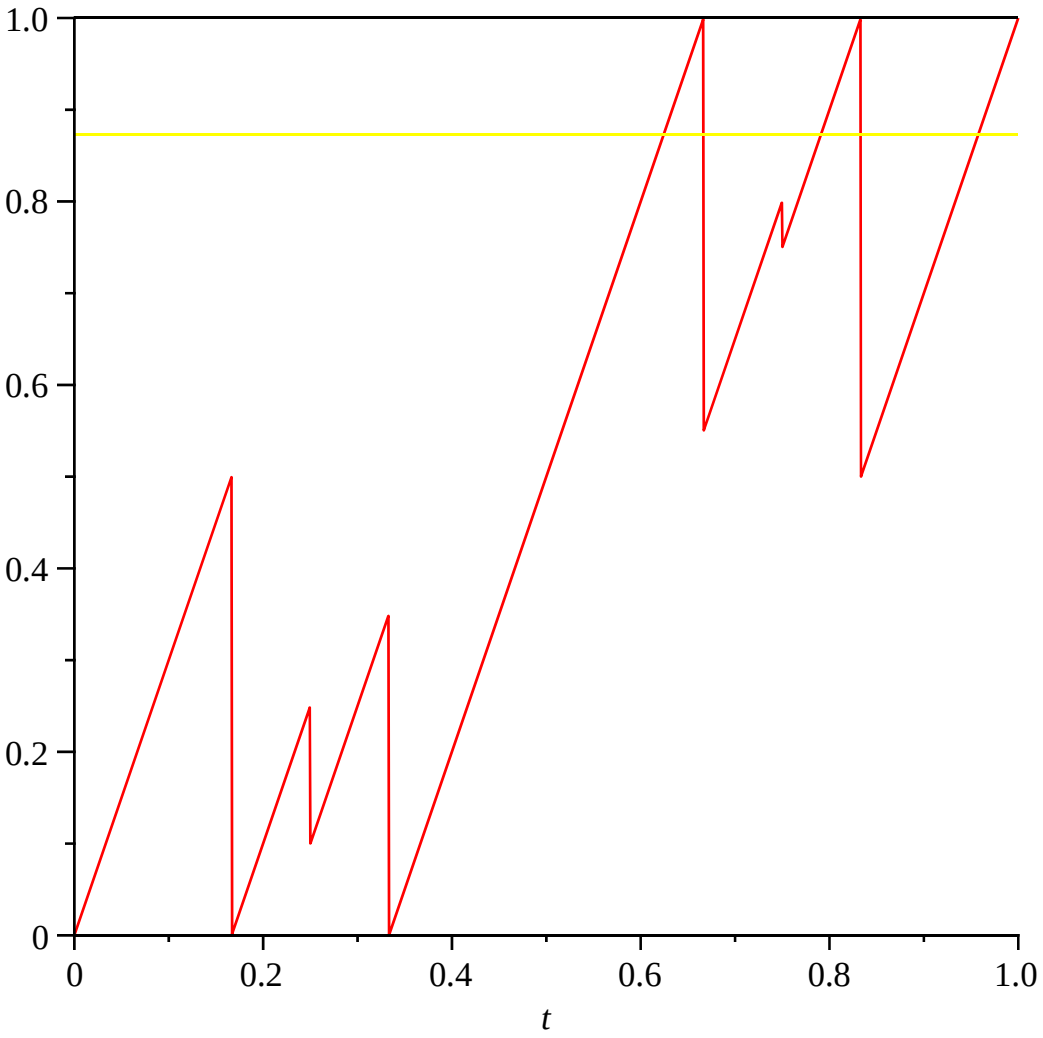
5

6

7

8

$err_7 := -2.87037\ 10^{-8}$



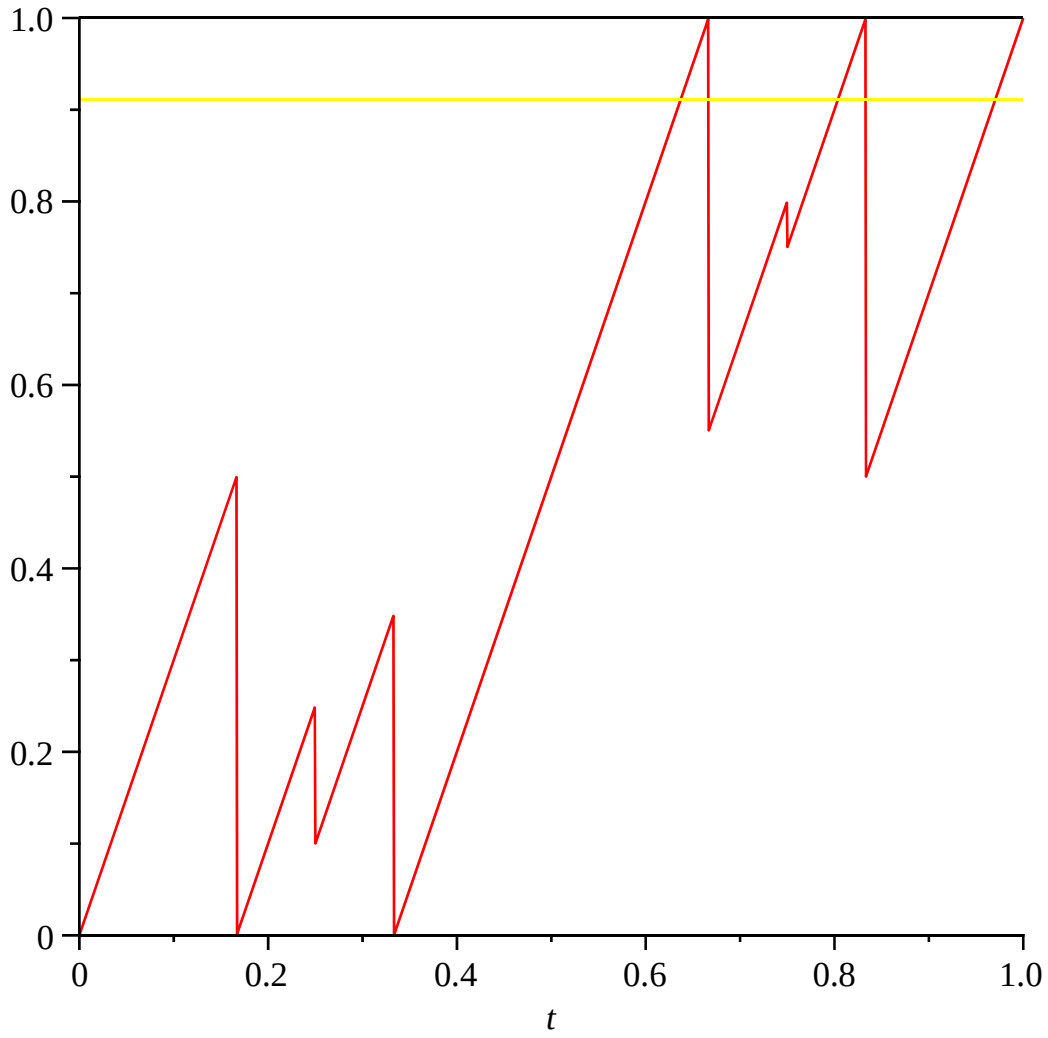
$su := 0$

5

7

8

$err_8 := -3.70370\ 10^{-9}$



$su := 0$

5

7

8

$err_9 := -3.70370\ 10^{-9}$

$y =, 0.080019$

$err[, 0,] =, 7.69231\ 10^{-9}$

$y =, 0.142755$

$err[, 1,] =, -4.29487\ 10^{-8}$

$y =, 0.284262$

$err[, 2,] =, 7.69231\ 10^{-9}$

$y =, 0.341229$

$err[, 3,] =, 7.69231\ 10^{-9}$

$y =, 0.499642$

$err[, 4,] =, 3.26923\ 10^{-8}$

$y =, 0.538641$

$err[, 5,] =, 5.83333\ 10^{-8}$

$y =, 0.669461$

$$err[, 6,]=, -3.70370\ 10^{-9}$$

$$y =, 0.777301$$

$$err[, 7,]=, -2.87037\ 10^{-8}$$

$$y =, 0.873062$$

$$err[, 8,]=, -3.70370\ 10^{-9}$$

$$y =, 0.910651$$

$$err[, 9,]=, -3.70370\ 10^{-9}$$

>