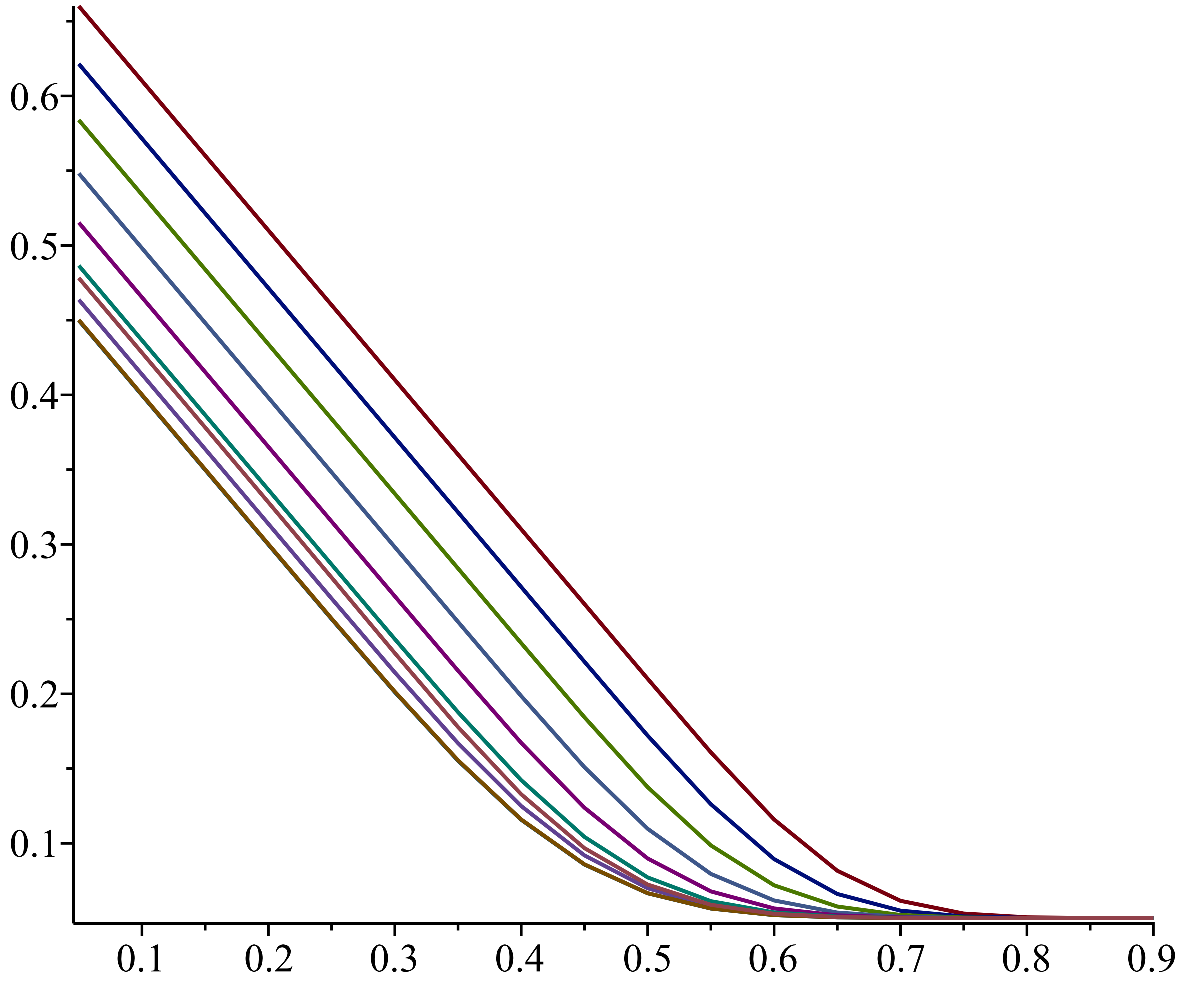
```
> restart:
  with(plots):
  with(LinearAlgebra):
  #-------------------------------------------------------------------
  -
  # ( ! )
  # ( ! ) ,              ,

  # ( ! )
    (       )
  #-------------------------------------------------------------------
  -

  #
  r0:=0.00000001: #      (   0   )
  R:=1: #
  alpha:=evalf(1/2): #
  Pnach:=1-x: #          (
    -       )


  #
  powFrac:=proc(chislo,stepen):
            evalf(sign(chislo)*abs(chislo)^stepen):
  end proc:

  #-------------------------------------------------------------------
  -
  #--------------------                        --------------------
  -
  #-------------------------------------------------------------------
  -
  Solver:= proc(nX,nT,K,n1) # -      ,   ,
  ,
            global r0,R,alpha,Pnach:
            local RightX, LeftX, RightT, LeftT, shagX, shagT, X, T,
  obshCoeff, a, b, A, Achert, B, E, SystemMatrix, Bright, Solution,
  p, c:
            local i,n,k,m:

             #
            RightX:=R:
            LeftX:=r0:
            X[0]:=LeftX:
            shagX:=evalf((RightX-LeftX)/nX):
            for i from 1 to nX do
                      X[i] := evalf(X[i-1]+shagX):
            end do:

             #
            RightT:=1:
            LeftT:=0:
            T[0]:=LeftX:
            shagT:=evalf((RightT-LeftT)/nT):
            for i from 1 to nT do
                      T[i] := evalf(T[i-1]+shagX):
```

```
          end do:

          # -
          for m from 1 to nX do
                    p[0,m]:=subs(x=X[m],Pnach):
          end do:

          # -
          for n from 1 to nT do
                    p[n,nX-n1]:=subs(x=X[nX-n1],Pnach):
          end do:



          #
 r ^ 2    x
          obshCoeff:=powFrac(X[1]-X[0],alpha/2)/GAMMA(alpha/2+2):
for n from 0 to nX do
          a[0,n]:=obshCoeff*(powFrac(n-1,alpha/2+1)-(n-1-alpha/2)
*powFrac(n,alpha/2)):
          for k from 1 to n-1 do
                    a[k,n]:=obshCoeff*(powFrac(n-k+1,alpha/2+1)
                    +powFrac(n-k-1,alpha/2+1)-2*powFrac(n-k,
alpha/2+1)):
          end do:
          a[n,n]:=obshCoeff*1:
end do:

#              x
R^2
obshCoeff:=powFrac(X[1]-X[0],alpha/2)/GAMMA(alpha/2+2):
for n from 0 to nX do
          b[n,n]:=obshCoeff*1:
          for k from n+1 to nX-1 do
                    b[k,n]:=obshCoeff*(powFrac(k+1-n,alpha/2+1)
                    +powFrac(k-1-n,alpha/2+1)-2*powFrac(k-n,
alpha/2+1)):
          end do:
          b[nX,n]:=obshCoeff*(powFrac(nX-n,alpha/2)*(alpha/2+n-
nX+1)+powFrac(nX-1-n,alpha/2+1)):
end do:

#
for i from 0 to nX do
          for n from 0 to nX do
                    if i>n then
                              A[i,n]:=0;
                    fi:
                    if i<=n then
                              A[i,n]:=2^(-alpha)*a[i,n]*add(b[j,
i],j=i..nX)*X[i]^(-alpha/2);
                    fi:
          end do:
end do:

#
for n from 0 to nX do
```

```
          for i from 0 to nX do
                    Achert[i,n]:=A[i,n]*sqrt(X[i])*sqrt(X[n]);
          end do:
end do:

print(X[n1]);
#
for n from 0 to nT do
          for m from 1 to nX-1-n1 do
                    p[n+1,m]:=p[n,m]+4*shagT/shagX^2*sqrt(X[n])*
add((Achert[k,n+1]-Achert[k,n])*(p[n,m+1]-p[n,m]),k=0..nX);
          end do:
end do:




seq([seq([X[m],p[n,m]],m=1..nX-1-n1)],n=1..nT);
end proc:
> 
SolN100:=[Solver(20,40,0.9,1)]:#   -        ,
,   ,


plot([seq(SolN100[k],k=10..19)]);
```

```
                         0.05000000950
```



```
> SolN100:=[Solver(20,60,0.9,1)]:#   -         ,
  ,   ,

  plot([seq(SolN100[k],k=10..19)]);
                              0.05000000950                              (1)
>
```