```
> restart:
  with(Statistics):
  with(plots):

  #          ,          0 . 1   0 . 0 5
  0.01  0.005
  ChZnPoZap:=6:
  N:=10^ChZnPoZap:
  ShagT:=evalf[1](1/10^ChZnPoZap);



  SubdivizionTrue005:=[0, 0.002, 0.005432, 0.007021, 0.009403,
  0.022044, 0.02735, 0.035111, 0.036276, 0.0406, 0.053353,
  0.055209, 0.06885, 0.069522, 0.071398, 0.072472, 0.080219,
  0.081417, 0.106, 0.107365, 0.11135, 0.112779, 0.113114, 0.137649,
  0.139081, 0.14266, 0.14502, 0.14532, 0.1524, 0.16015, 0.16112,
  0.16592, 0.178368, 0.1789, 0.180737, 0.181805, 0.193463,
  0.194508, 0.196266, 0.196957, 0.209433, 0.20999, 0.21388,
  0.214486, 0.215524, 0.21561, 0.22257, 0.22363, 0.22754, 0.22814,
  0.24173, 0.24283, 0.257561, 0.258381, 0.259303, 0.272062, 0.2764,
  0.277756, 0.28443, 0.286677, 0.288702, 0.337439, 0.340146,
  0.347801, 0.354836, 0.357106, 0.357465, 0.361458, 0.362671,
  0.387008, 0.38876, 0.392441, 0.39447, 0.418343, 0.41958, 0.41993,
  0.427913, 0.428805, 0.431159, 0.444826, 0.447399, 0.455219,
  0.456858, 0.481319, 0.483195, 0.485141, 0.514796, 0.516835,
  0.518573, 0.543711, 0.545046, 0.552578, 0.553826, 0.555206,
  0.56999, 0.571647, 0.572281, 0.580212, 0.581792, 0.606287,
  0.607756, 0.610953, 0.61255, 0.637492, 0.639096, 0.642709,
  0.6453, 0.652508, 0.659425, 0.661763, 0.662616, 0.711453,
  0.713873, 0.714882, 0.730498, 0.73197, 0.73244, 0.736349,
  0.739811, 0.741402, 0.74271, 0.75628, 0.757027, 0.759005,
  0.759815, 0.767634, 0.769335, 0.793211, 0.795056, 0.802749,
  0.805194, 0.819847, 0.822389, 0.831511, 0.856348, 0.857871,
  0.861182, 0.863188, 0.886765, 0.887408, 0.88884, 0.892727,
  0.893085, 0.895377, 0.9024, 0.91, 0.911, 0.91508, 0.915755,
  0.928233, 0.929113, 0.93042, 0.93123, 0.943985, 0.946044,
  0.94693, 0.9591, 0.959964, 0.963969, 0.965048, 0.972781,
  0.978426, 0.990752, 0.995075, 0.99752, 0.9992, 1]:
  N005:=numelems(%);

  #      = 0 . 1 ,   9 7
  SubdivizionTrue01:=[0, 0.023719, 0.027577, 0.03505, 0.036389,
  0.038614, 0.086441, 0.088532, 0.090704, 0.097553, 0.098511,
  0.104947, 0.107061, 0.10762, 0.110933, 0.112439, 0.137714,
  0.142702, 0.145291, 0.15247, 0.160043, 0.1615, 0.163582,
  0.213014, 0.215944, 0.221817, 0.224756, 0.275041, 0.275883,
  0.278083, 0.285061, 0.286268, 0.290041, 0.290541, 0.304696,
  0.319835, 0.322512, 0.326147, 0.328, 0.422, 0.423154, 0.424327,
  0.427378, 0.428712, 0.430484, 0.431305, 0.443939, 0.448149,
  0.450805, 0.453, 0.547, 0.548946, 0.552339, 0.55377, 0.555211,
  0.569832, 0.571132, 0.572297, 0.580122, 0.581937, 0.605841,
  0.60686, 0.608035, 0.610869, 0.612319, 0.637505, 0.638785,
  0.642581, 0.645134, 0.652345, 0.659917, 0.661383, 0.663588,
  0.711579, 0.713599, 0.715636, 0.721734, 0.724735, 0.775513,
  0.778181, 0.785648, 0.789079, 0.820448, 0.821801, 0.822396,
```

```
0.826185, 0.828, 0.922, 0.923946, 0.927494, 0.930318, 0.944998,
0.94739, 0.956859, 0.98, 1.0]:
N01:=numelems(%);

sigma005:=evalf(GAMMA(ln(3)/ln(2)+1)*[0, 0.001712292217,
0.003187258397, 0.00426834073, 0.005666180747, 0.01076693522,
0.01439504481, 0.01766065087, 0.01867200149, 0.02113725979,
0.02612612901, 0.02758684733, 0.03307467239, 0.03350817579,
0.03443439764, 0.03513000896, 0.0386638205, 0.03969157334,
0.04946102197, 0.05026357509, 0.05227714248, 0.05330042672,
0.05356585003, 0.06266609002, 0.06361193797, 0.06546591233,
0.06684316085, 0.06706629948, 0.07012388608, 0.07337788875,
0.07413401016, 0.07690605779, 0.0819134435, 0.08233250532,
0.08324013084, 0.0839982878, 0.08822684399, 0.08900261776,
0.08973678768, 0.09022147968, 0.09521630249, 0.09569338287,
0.09741323565, 0.09783571952, 0.09833500986, 0.09848329209,
0.1012310528, 0.1020681591, 0.1040287458, 0.1044660382,
0.1099752748, 0.1108301968, 0.1168815945, 0.1175748864,
0.1180278519, 0.1231651531, 0.1254996448,
0.1265724311, 0.1293351275, 0.1305837193, 0.1320255153,
0.1502271534, 0.1524950656, 0.1557631002, 0.1588899112,
0.1600546757, 0.1604614694, 0.162173628, 0.1632531729,
0.1723608065, 0.1737151767, 0.1754324156, 0.1768236056,
0.1855913037, 0.1863391627, 0.1867494219, 0.1902060918,
0.1910856934, 0.1924853126, 0.1979755074, 0.2001319982,
0.2035895312, 0.2050186159, 0.2141496819, 0.215114857,
0.2165770313, 0.2287200565, 0.2304148415, 0.2312407824,
0.240906638, 0.2419859587, 0.2455810322, 0.2464178365,
0.2473204322, 0.2532710764, 0.2546550578, 0.2551152191,
0.2585625969, 0.2599344507, 0.2691264418, 0.2701566479,
0.2715178414, 0.2727719617, 0.282152579, 0.2832041899,
0.2849327391, 0.2868021134, 0.289844962, 0.2928797345,
0.2941396461, 0.2949555956,
0.3132318419, 0.3146787882, 0.3155291811, 0.3224152067,
0.3234215117, 0.3238456195, 0.325587708, 0.3271114128,
0.3281331878, 0.3290782993, 0.3346409622, 0.3351862867,
0.3360905001, 0.3368455421, 0.3403162727, 0.3416852545,
0.3502669211, 0.3518626406, 0.3553006389, 0.3574302543,
0.3634027572, 0.3655267, 0.3706533182, 0.3801517644,
0.3812645174, 0.3828666521, 0.3842572158,
0.3927619682, 0.3931800765, 0.3942098599, 0.3959502724,
0.3963573172, 0.3976711532, 0.4006936672, 0.4039498932,
0.4047089066, 0.4066708314, 0.4071048928, 0.4120942777,
0.4126041114, 0.4131206195, 0.4136914614, 0.4187086583,
0.4200763827, 0.4205143282, 0.4252363655, 0.4259807333,
0.4277843619, 0.4286141912, 0.4319625277, 0.4356472973,
0.4406898461, 0.4431375476, 0.4447407188, 0.4455845394,
0.4464167501]);



sigma01:=evalf(GAMMA(ln(3)/ln(2)+1)*[0, 0.01701452824,
0.02003996187, 0.02317104403, 0.02438677733, 0.02562136429,
0.04321602114, 0.04467434255, 0.04590812097, 0.04866697337,
0.0493804554, 0.05202275582, 0.05313458841, 0.05356338545,
0.0550645239, 0.05601559221, 0.06602515008, 0.0689198108,
0.07079866303, 0.07383309055, 0.07703609084, 0.0782652198,
```

```
0.07949670827, 0.09834641289, 0.1003972657, 0.1024555288,
0.1045183788, 0.1239293337, 0.1247582191, 0.1259960032,
0.1287574144, 0.1298189545, 0.1315310868, 0.1319591413,
0.1380371964, 0.1444528901, 0.1465400791, 0.1482546668,
0.1497356467, 0.182843107, 0.1839129326, 0.184648229,
0.1856764081, 0.1867038134, 0.1874541381, 0.1879306917,
0.1930687969, 0.1954409078, 0.1969406299, 0.198690943,
0.2317984032, 0.2335129819, 0.2350100235, 0.2359815453,
0.2369759204, 0.2428863228, 0.2437041016, 0.2447039495,
0.2481359401, 0.2495282315, 0.258305485, 0.2590275492,
0.2597622902, 0.260814567, 0.2618220383, 0.2715202625,
0.2725529141, 0.2742707021, 0.2758554371,
 0.2792141522, 0.2824233591, 0.2836470517, 0.2848843726,
0.3024668792, 0.3038599138, 0.3051621717, 0.3074486766,
0.3094887837, 0.3294388444, 0.3313502034, 0.3349828832,
0.3377129545, 0.3514417248, 0.3524836225, 0.3528946558,
0.3546141683, 0.3560858409, 0.3891933011, 0.3909078798,
0.3926383399, 0.3946981068, 0.4006756374, 0.4028019501,
0.4083924682, 0.4176833501, 0.4341889575]);
sigma01[N01]:=GAMMA(ln(3)/ln(2)+1)*0.4341889575:
```

$$ShagT := 1.\ 10^{-6}$$

$$N005 := 167$$

$$N01 := 96$$

$$\sigma 005 := [0., 0.002420572187, 0.004505649768, 0.006033915680, 0.008009964297,$$
$$0.01522061695, 0.02034947350, 0.02496587900, 0.02639556907, 0.02988056750,$$
$$0.03693305419, 0.03899799037, 0.04675582316, 0.04736864278, 0.04867799105,$$
$$0.04966133805, 0.05465689070, 0.05610976768, 0.06992029336, 0.07105481803,$$
$$0.07390128616, 0.07534784613, 0.07572306028, 0.08858756297, 0.08992465556,$$
$$0.09254551590, 0.09449245547, 0.09480789413, 0.09913023409, 0.1037302365,$$
$$0.1047991233, 0.1087178127, 0.1157964751, 0.1163888795, 0.1176719392,$$
$$0.1187437034, 0.1247213779, 0.1258180461, 0.1268559013, 0.1275410834,$$
$$0.1346019863, 0.1352764083, 0.1377076684, 0.1383049104, 0.1390107293,$$
$$0.1392203477, 0.1431047040, 0.1442880746, 0.1470596469, 0.1476778228,$$
$$0.1554659239, 0.1566744795, 0.1652290036, 0.1662090717, 0.1668494039,$$
$$0.1741117206, 0.1774118615, 0.1789283998, 0.1828338699, 0.1845989346,$$
$$0.1866371214, 0.2123677639, 0.2155737851, 0.2201936237, 0.2246138224,$$
$$0.2262603851, 0.2268354467, 0.2292558300, 0.2307819226, 0.2436568772,$$
$$0.2455714749, 0.2479990400, 0.2499656878, 0.2623600945, 0.2634173012,$$
$$0.2639972618, 0.2688837636, 0.2701272074, 0.2721057712, 0.2798669539,$$
$$0.2829154649, 0.2878031868, 0.2898234043, 0.3027314840, 0.3040958983,$$
$$0.3061628927, 0.3233288115, 0.3257246347, 0.3268922214, 0.3405563033,$$
$$0.3420820789, 0.3471642342, 0.3483471779, 0.3496231272, 0.3580352216,$$
$$0.3599916791, 0.3606421835, 0.3655155496, 0.3674548629, 0.3804490690,$$
$$0.3819054141, 0.3838296575, 0.3856025376, 0.3988633941, 0.4003499979,$$
$$0.4027935516, 0.4054361820, 0.4097376877, 0.4140277766, 0.4158088435,$$
$$0.4169623059, 0.4427984179, 0.4448438853, 0.4460460384, 0.4557804295,$$
$$0.4572029869, 0.4578025244, 0.4602652180, 0.4624191946, 0.4638636211,$$

0.4651996726, 0.4730632995, 0.4738341944, 0.4751124305, 0.4761797913,
0.4810861700, 0.4830214234, 0.4951528477, 0.4974086275, 0.5022687343,
0.5052792531, 0.5137222479, 0.5167247476, 0.5239719624, 0.5373993870,
0.5389724240, 0.5412372727, 0.5432030351, 0.5552257302, 0.5558167866,
0.5572725341, 0.5597328583, 0.5603082749, 0.5621655717, 0.5664383315,
0.5710414769, 0.5721144519, 0.5748879158, 0.5755015242, 0.5825547399,
0.5832754634, 0.5840056222, 0.5848125896, 0.5919051216, 0.5938385974,
0.5944576966, 0.6011329778, 0.6021852492, 0.6047349385, 0.6059080220,
0.6106413789, 0.6158503326, 0.6229787031, 0.6264388825, 0.6287051963,
0.6298980585, 0.6310745085]

$\sigma 01 :=$ [0., 0.02405249141, 0.02832937851, 0.03275561506, 0.03447422955, **(1)**
0.03621949641, 0.06109208334, 0.06315363113, 0.06489775500, 0.06879779106,
0.06980640089, 0.07354167390, 0.07511340974, 0.07571957624, 0.07784165211,
0.07918612443, 0.09333607918, 0.09742809990, 0.1000841287, 0.1043737300,
0.1089016332, 0.1106391844, 0.1123800711, 0.1390268492, 0.1419260256,
0.1448356776, 0.1477518138, 0.1751920002, 0.1763637493, 0.1781135358,
0.1820171891, 0.1835178292, 0.1859381753, 0.1865432922, 0.1951355003,
0.2042050093, 0.2071555522, 0.2095793694, 0.2116729483, 0.2584751219,
0.2599874748, 0.2610269223, 0.2624804019, 0.2639327876, 0.2649934798,
0.2656671571, 0.2729306104, 0.2762839316, 0.2784040052, 0.2808783254,
0.3276804989, 0.3301043033, 0.3322205877, 0.3335939740, 0.3349996667,
0.3433548735, 0.3445109219, 0.3459243512, 0.3507759653, 0.3527441701,
0.3651520848, 0.3661728268, 0.3672114892, 0.3686990343, 0.3701232404,
0.3838330801, 0.3852928822, 0.3877212235, 0.3899614751, 0.3947095037,
0.3992461808, 0.4009760469, 0.4027251786, 0.4275805894, 0.4295498449,
0.4313907744, 0.4346230791, 0.4375070649, 0.4657093551, 0.4684113369,
0.4735466542, 0.4774060040, 0.4968136023, 0.4982864751, 0.4988675300,
0.5012983091, 0.5033787307, 0.5501809042, 0.5526047086, 0.5550509637,
0.5579627415, 0.5664128438, 0.5694186937, 0.5773216979, 0.5904557004,
0.6137887588]

```
> with(Statistics):

  #
  LinearInterpol:=proc(A)
            option remember:
            local N,i,eq,Ahelp:
            Ahelp:=A:
            N:=Count(Ahelp)[1];
            for i from 1 to N-1 do
                      eq[i]:=(A[i+1,2]-A[i,2])/(A[i+1,1]-A[i,1])*
  (xxx-A[i,1])+A[i,2]:
            od:

            Matrix([[seq(xxx>=A[i,1] and xxx<=A[i+1,1], i = 1 ..
  N-1)], [seq(eq[i],i=1..N-1)]]):
```

```
            piecewise(seq(i, i in %)):
end proc:



#
PowInterpol:=proc(a, LocalLestnica)
            local i,N:
            local X, b, AA, StupenPow:
            option remember;
            N:=numelems(LocalLestnica):

#
            X[1]:=LocalLestnica[1,1]:
            for i from 2 to N do
                        X[i]:=LocalLestnica[i,1]:
                        b[i]:=LocalLestnica[i,2]/(X[i])^a:
            end do:
            print(LocalLestnica[1,2],LocalLestnica[1,1]);
            #   -
            AA := evalf[5](Matrix([[x>=0 and x<=X[3], seq(x>=X[i]
and x<=X[i+1], i = 3 .. N-1)], [seq(b[i]*x^a,i=2..N-1)]])):
            StupenPow:=piecewise(seq(i, i in AA)):

            StupenPow;
end proc:
```

```
> #
MeasureLinearKoxa005:=subs(x=xxx,LinearInterpol([[0,0],seq(
[SubdivizionTrue005[j],sigma005[j]],j=2..N005)])):
MeasureLinearKoxa01:=subs(x=xxx,LinearInterpol([[0,0],seq(
[SubdivizionTrue01[j],sigma01[j]],j=2..N01)])):
```

```
> #
eq01:=evalf(int((a*x^(ln(4)/ln(3))-MeasureLinearKoxa01)^2,x=0..1)
):
eq001:=evalf(int((a*x^(ln(4)/ln(3))-MeasureLinearKoxa001)^2,x=0.
.1)):
minimize(eq01, a = 0 .. 1,location);
minimize(eq001, a = 0 .. 1,location);
```

Warning, computation interrupted

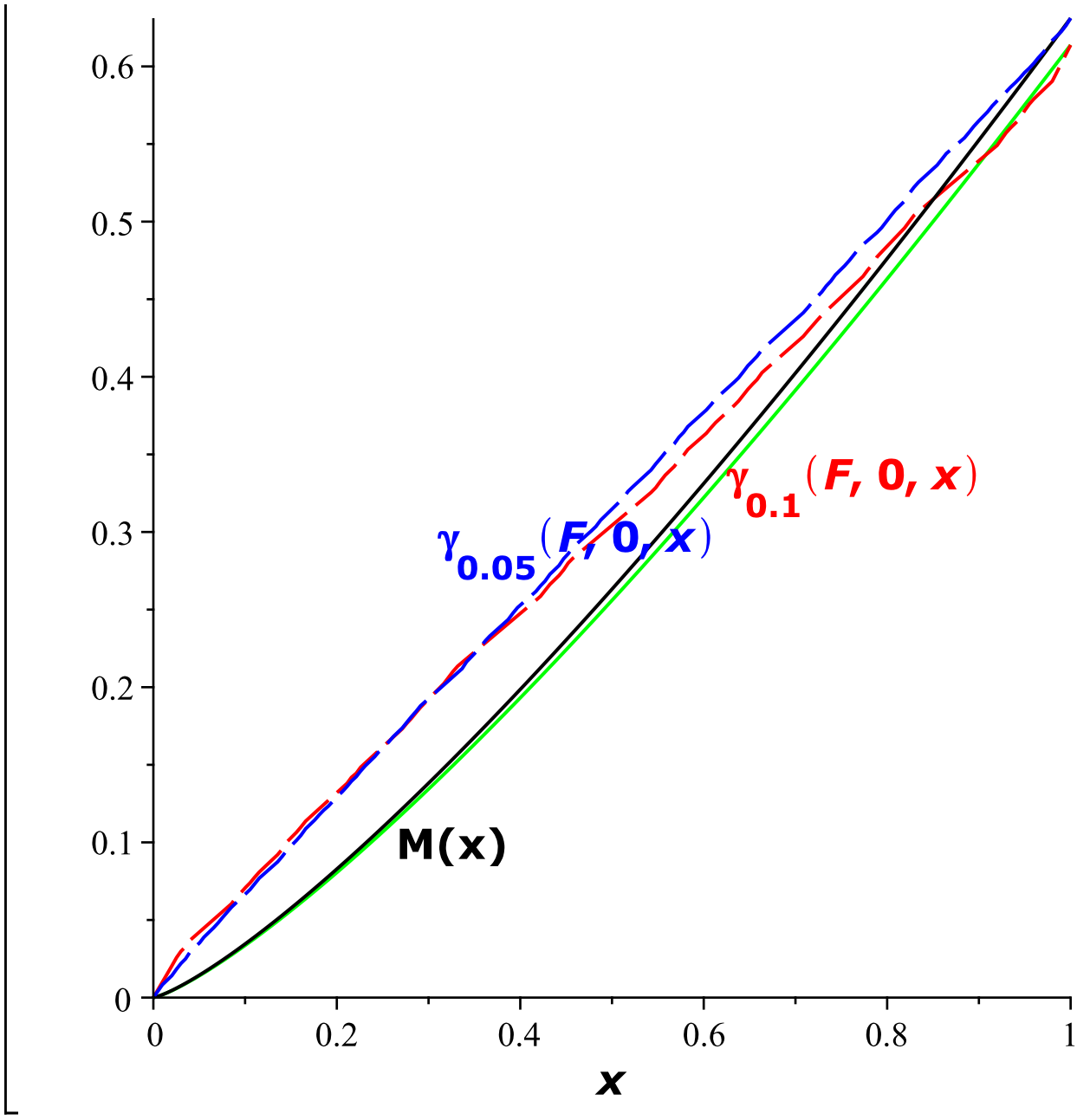
$$eq01,\ \{[\varnothing, eq01]\}$$

$$minimize\left(0.2837910731\ a^2 - 0.8842282179\ MeasureLinearKoxa001\ a + MeasureLinearKoxa001^2,\ a = 0..1,\ location\right),\ \varnothing \qquad \textbf{(2)}$$

$$0.0005685074739,\ \{[\{a = 0.4628219337\},\ 0.0005685074739]\} \qquad (3)$$

```
> FractalDerivModelText005:=textplot([0.63, 0.29, ('typeset')(gamma
  [0.05](F,0,x)),'font' = ["Verdana",bold, 14]], 'align' = 'left',
  color=blue):
  FractalDerivModelText01:=textplot([0.92, 0.33, ('typeset')(gamma
  [0.1](F,0,x)),'font' = ["Verdana",bold, 14]], 'align' = 'left',
  color=red):
  PowerLawModelText:=textplot([0.4, 0.1, ('typeset')("M(x)"),'font'
  = ["Verdana",bold, 14]], 'align' = 'left',color=black):



  MassCurva005:=plot([0.6310745085*x^(ln(4)/ln(3)),subs(xxx=x,
  LinearInterpol([[0,0],seq([SubdivizionTrue005[j],sigma005[j]],j=
  2..N005)]))],x=SubdivizionTrue005[1]..SubdivizionTrue005
  [N005],labels=[x,"  "],linestyle = [solid, longdash],legendstyle
  = [font = ["ROMAN", 13], location = bottom],labelfont = ["ROMAN",
  15],labelfont = ["Verdana",bold, 14],color=[black,blue]):
  MassCurva01:=plot([0.6137887588*x^(ln(4)/ln(3)),subs(xxx=x,
  LinearInterpol([[0,0],seq([SubdivizionTrue01[j],sigma01[j]],j=2..
  N01)]))],x=SubdivizionTrue01[1]..SubdivizionTrue01[N01],labels=
  [x,"  "],linestyle = [solid, longdash],legendstyle = [font =
  ["ROMAN", 13], location = bottom],labelfont = ["ROMAN",
  15],labelfont = ["Verdana",bold, 14],color=[green,red]):

  evalf(sigma01[N01]);

  #plot(abs(0.6137887588*x^(ln(4)/ln(3))-LinearInterpol([[0,0],seq(
  [SubdivizionTrue01[j],sigma01[j]],j=2..N01)])),x=0..1);
  #plot(abs(0.6310745085*x^(ln(4)/ln(3))-LinearInterpol([[0,0],seq(
  [SubdivizionTrue005[j],sigma005[j]],j=2..N005)])),x=0..1);

  display(FractalDerivModelText005,FractalDerivModelText01,
  PowerLawModelText,MassCurva01,MassCurva005);
```

$$0.6137887588$$




```
> #
  N:=1000:

  solverBakleyLeverett:=proc(T,D,sig, withOrNotPalka)
            local m,mu,F_,S,x,t,xS0,i,k,Urav,SmaxNotTrue,xFr,
  RaznicaIntForFront,Sfr1,Sfr2;
            global N,K,SS;
             #
            m:=0.7:
            mu[0] := 0.4;
             #   -
            F_ := s^2/(s^2+mu[0]*(1-s)^2);
```

```
            #
            xS0:=0;

            #
            for k from 0 to N do
                        S[k] := k/N;
            end do:


            Urav:=subs(t=T,(D*diff(F_,s)*t/(m*sig)))^(1/D):

            for i from 0 to N do
                        x[i]:=subs(s=S[i],Urav);
            end do;

            print("sssa");

            if withOrNotPalka=1 then
                        #

                        SmaxNotTrue:=subs(solve({diff(Urav,s)=0,
s>0,s<=1},s),s):
                        xFr:=subs(s=SmaxNotTrue,Urav):
                        RaznicaIntForFront:=1:
                        i:=0:
                        while RaznicaIntForFront>0.001 do
                                    i:=i+1:
                                    xFr:=xFr-0.002:
                                    Sfr1:=fsolve(xFr=Urav,s,0..
SmaxNotTrue);
                                    Sfr2:=fsolve(xFr=Urav,s,
SmaxNotTrue..1);
                        #
                                    RaznicaIntForFront:=abs(evalf(Int
(xFr-Urav,s=0..Sfr1,epsilon =0.0001))-evalf(Int(Urav-xFr,s=
Sfr1..Sfr2,epsilon =0.0001)));
                        end do:


                        #
                        k:=N:
                        while S[k]>Sfr2 do
                                    x[k]:=subs(s=S[k],Urav);
                                    k:=k-1:
                        end do;

                        #
                        for i from 0 to k do
                                    x[i]:=xFr:
                        end do;
            fi:

            [seq(x[kk],kk=0..N)];
end proc:
```

```
>
  #
  N:=1000:

  solverBakleyLeverettFractalDeriv:=proc(T,SubdivizionTrue,sigma,
  withOrNotPalka)
            local Nsubdivizion,m,mu,F_,s,InvSigma,S,x,t,xS0,i,k,
  xFrontFractal,kkk,Urav,SmaxNotTrue,xFr,RaznicaIntForFront,Sfr1,
  Sfr2,z,ss,xx;
            global N,SS;
            Nsubdivizion:=numelems(SubdivizionTrue):
             #
            m:=0.7:
            mu[0] := 0.4;
             #   -
            F_ := s^2/(s^2+mu[0]*(1-s)^2);

             #
            InvSigma:=evalf(LinearInterpol([[0,0],seq([sigma[j],
  SubdivizionTrue[j]],j=2..Nsubdivizion)])):

            #
            xS0:=0;

            #
            for k from 0 to N do
                      S[k] := evalf(k/N);
            end do:



            Urav:=subs(xxx=(diff(F_,s)*T/m),InvSigma);


            for i from 0 to N do
                      x[i]:=subs(s=S[i],Urav);
            end do;

            if withOrNotPalka=1 then
                       #
    (       )
                      SmaxNotTrue:=fsolve(diff(Urav,s)=0,s,0.1..1):
                      xFr:=eval(Urav,s=SmaxNotTrue): print(%);
                      RaznicaIntForFront:=1:
                      i:=0:
                      while RaznicaIntForFront>0.001 do
                                i:=i+1:
                                xFr:=xFr-0.002:
                                Sfr1:=fsolve(xFr=Urav,s,0..
  SmaxNotTrue);
                                Sfr2:=fsolve(xFr=Urav,s,
  SmaxNotTrue..1);
```

```
                                    #

                                    RaznicaIntForFront:=0:
                                    for k from 0 to N-1 do
                                              if S[k]<=Sfr1 then

  RaznicaIntForFront:=RaznicaIntForFront+eval(xFr-Urav,s=S[k])*(S
  [k+1]-S[k]):
                                              fi:
                                              if S[k]>Sfr1 and S[k]<=
  Sfr2 then

  RaznicaIntForFront:=RaznicaIntForFront-eval(Urav-xFr,s=S[k])*(S
  [k+1]-S[k]):
                                              fi:
                                    od:
                          end do:

               #
                          k:=N:

                          while S[k]>Sfr2 do
                                    x[k]:=subs(s=S[k],Urav);
                                    k:=k-1:
                          end do;

                          #
                          for i from 0 to k do
                                    x[i]:=xFr:
                          end do;
               fi:

  [[seq(x[kk],kk=0..N)],xFr];
  end proc:
> Dkocha:=ln(4)/ln(3):
  #-------------------------------------------------------------
  ---------------- 0.1

  TT:=0.05:
  SolFractal01_005:=solverBakleyLeverettFractalDeriv(TT,
  SubdivizionTrue01,sigma01):
  sig01_005:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa01,xxx=%
  [2]):
  SolPower01_005:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
  print("fuck"):


  TT:=0.1:
  SolFractal01_01:=solverBakleyLeverettFractalDeriv(TT,
  SubdivizionTrue01,sigma01):
  print("fuck"):
  sig01_01:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa01,xxx=%[2]
  ):
  SolPower01_04:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
  print("fuck"):
```

```
TT:=0.15:
SolFractal01_015:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue01,sigma01):
sig01_015:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa01,xxx=%
[2]):
SolPower01_015:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
print("fuck"):

TT:=0.2:
SolFractal01_02:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue01,sigma01):
sig01_02:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa01,xxx=%[2]
):
SolPower01_02:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
print("big fuck"):

#-------------------------------------------------------------
---------------  0.05


TT:=0.05:
SolFractal005_005:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
sig01_005:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa005,xxx=%
[2]):
SolPower005_005:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
print("fuck"):


TT:=0.1:
SolFractal005_01:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
print("fuck"):
sig005_01:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa005,xxx=%
[2]):
SolPower005_04:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
print("fuck"):

TT:=0.15:
SolFractal005_015:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
sig01_015:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa005,xxx=%
[2]):
SolPower005_015:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
print("fuck"):

TT:=0.2:
SolFractal005_02:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
sig01_02:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa005,xxx=%
[2]):
SolPower005_02:=evalf(solverBakleyLeverett(TT,Dkocha,%)):
print("very big fuck"):
```

0.2343911661

"fuck"

0.5034945209

"fuck"

"fuck"

0.7634850328

"fuck"

0.9984998775

"big fuck"

0.2374438925

"fuck"

0.4850058299

"fuck"

"fuck"

0.7343867321

"fuck"

0.9742952233

"very big fuck" **(4)**

```
> #

  Dkocha:=ln(4)/ln(3):
  TT:=0.05:
  SolFractal01_005:=solverBakleyLeverettFractalDeriv(TT,
  SubdivizionTrue01,sigma01,1):
  sig01_005:=Dkocha*%[2]^(-Dkocha)*eval(MeasureLinearKoxa01,xxx=%
  [2]);
  SolPower01_005:=evalf(solverBakleyLeverett(TT,Dkocha,%,0)):
  SolFractal01_005:=solverBakleyLeverettFractalDeriv(TT,
  SubdivizionTrue01,sigma01,0):
  print("fuck"):

  writedata(WithoutStickOutSolFractal01_005, SolFractal01_005[1],
  float):
  writedata(WithoutStickOutSolPower01_005, SolPower01_005, float):
```

0.2343911661

$$sig01\_005 := \frac{0.2086532502\,\ln(2)\,0.1523911661^{-\frac{2\ln(2)}{\ln(3)}}}{\ln(3)}$$

"sssa"

"fuck" **(5)**

```
> SolPower01_005[2]
```

0.001698581619 **(6)**

```
> writedata(OutSolFractal01_005, SolFractal01_005[1], float):
  writedata(OutSolFractal01_01, SolFractal01_01[1], float):
  writedata(OutSolFractal01_015, SolFractal01_015[1], float):
  writedata(OutSolFractal01_02, SolFractal01_02[1], float):
```

```
writedata(OutSolPower01_005, SolPower01_005, float):
writedata(OutSolPower01_01, SolPower01_04, float):
writedata(OutSolPower01_015, SolPower01_015, float):
writedata(OutSolPower01_02, SolPower01_02, float):


writedata(OutSolFractal005_005, SolFractal005_005[1], float):
writedata(OutSolFractal005_01, SolFractal005_01[1], float):
writedata(OutSolFractal005_015, SolFractal005_015[1], float):
writedata(OutSolFractal005_02, SolFractal005_02[1], float):

writedata(OutSolPower005_005, SolPower005_005, float):
writedata(OutSolPower005_01, SolPower005_04, float):
writedata(OutSolPower005_015, SolPower005_015, float):
writedata(OutSolPower005_02, SolPower005_02, float):
> # - - - - -

Dkocha:=ln(4)/ln(3):
Wrongsig005:=evalf(Dkocha*eval(MeasureLinearKoxa005,xxx=1));
#-----------------------------------------------------------------
---------------  0.05


TT:=0.05:
WrongSolFractal005_005:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
WrongSolPower005_005:=evalf(solverBakleyLeverett(TT,Dkocha,
Wrongsig005)):
print("fuck"):


TT:=0.1:
WrongSolFractal005_01:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
WrongSolPower005_01:=evalf(solverBakleyLeverett(TT,Dkocha,
Wrongsig005)):
print("fuck"):

TT:=0.15:
WrongSolFractal005_015:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
WrongSolPower005_015:=evalf(solverBakleyLeverett(TT,Dkocha,
Wrongsig005)):
print("fuck"):

TT:=0.2:
WrongSolFractal005_02:=solverBakleyLeverettFractalDeriv(TT,
SubdivizionTrue005,sigma005):
WrongSolPower005_02:=evalf(solverBakleyLeverett(TT,Dkocha,
Wrongsig005)):
print("very big fuck"):
```

Wrongsig005 := 0.7963273680

0.2374438925

"fuck"

0.4850058299

"fuck"

0.7343867321

"fuck"

0.9742952233

"very big fuck"

(7)

```
> writedata(WrongOutSolFractal005_005, WrongSolFractal005_005[1],
  float):
  writedata(WrongOutSolFractal005_01, WrongSolFractal005_01[1],
  float):
  writedata(WrongOutSolFractal005_015, WrongSolFractal005_015[1],
  float):
  writedata(WrongOutSolFractal005_02, WrongSolFractal005_02[1],
  float):

  writedata(WrongOutSolPower005_005, WrongSolPower005_005, float):
  writedata(WrongOutSolPower005_01, WrongSolPower005_01, float):
  writedata(WrongOutSolPower005_015, WrongSolPower005_015, float):
  writedata(WrongOutSolPower005_02, WrongSolPower005_02, float):
>
>
```