ISSN: 2997-2795

Review Article

Annals of Computational Physics and Material Science

Dark Matter-Dark Energy as Massive Bosons a Quantum Multi-Entanglement System with Dark Matter or Dark Energy as Some Massive Bosons-String Particles of Spin 2

Robert Dosseh Kpotogbey*

Department of Physics, University of Texas at Austin, USA

***Corresponding Author**
Robert Dosseh Kpotogbey, Department of Physics, University of Texas at Austin, USA.

Submitted: 2024, Aug 22; **Accepted:** 2024, Sep 13; **Published:** 2024, Sep 18

Citation: Kpotogbey, R. D. (2024). Dark Matter-Dark Energy as Massive Bosons a Quantum Multi-Entanglement System with Dark Matter or Dark Energy as Some Massive Bosons-String Particles of Spin 2. *Ann Comp Phy Material Sci, 1*(3), 01-119.

Abstract

A quantum multi-entanglement system with Dark matter or dark energy as some massive bosons-string particles of spin 2. Dark matter or dark energy are some particles of spin 2 instead of near spin 0 ones.

Example 1: dE = 4.Sqrt{(gM^2)/sqrt(2)} ~ 125.5673374143954990316092809188 GeV ((gM^2) = 1393.6328858707575005182839547884 GeV²) the particle at exactly some 125 GeV and spin 1+1+1-1=2 (As a frequent Space-Time reflect.) or 1-1+1-1=0. The Source of massive bosons that are dark matter-dark energy simulators is probably the extreme side of black hole's core where the magnetic field stopping matter is getting strongest. Magnetic Field [B] = kg/(s².A), ([E] = m².kg/s²), [E].(1/(m².A)) = [B], A↗, d↘ → E↘ ([d] = m), M↗ → q.A↗,d↘ → E↗ ([M] = kg)

$$\frac{1}{m^2_{massiveboson}} \propto q$$

We strive to prevent the emission of massive gravitons into the central black hole, considering that a Modified Newtonian dynamics would require measurement of its effects starting from the sun and its (4.152±0.014).10^6 M_{sun} → L ~ 2.5 kpc=7.7142.10^{19} m new gravitational field:
$1M_{sun}$ → 0.602119460500963.10^{-6} kpc (1.857947976878612.10^{13} m=124.1960537090558 UA > 40 UA: And have, from certain constant effects, effects on the closest star to the Sun, α Cen C (Proxima Centauri), is 1.316 parsec (4.28 light years) away [1]. Or (4.152±0.014).10^6 M_{sun} → (L ~ 2.5 kpc=7.7142.10^{19} m)²

$1M_{sun}$ → 1.505298651252408.10^{-6} kpc² → 0.001226906129763972 kpc (3.785839706490093.10^{16} m)

= 253067.5548359765 UA To affect the closest star to the Sun, α Cen C (Proxima Centauri), is 1.316 parsec (4.28 light years) away. Dark Matter-Effects evolve with the quantity of black holes and black holes-cores like another row of other hypothetical phenomena [2].

1. Introduction

1.1. The Galaxy Cluster 1E 0657-56, also known as the "Bullet Cluster"

The bullet cluster is marked by a unique combination of massive spin +2 and spin -2 boson effects [3]. These massive bosons are generated from the MAD of the matter rotating around each black hole. The spin +2 bosons are then generated in these cases in the opposite direction of the massive spin -2 bosons and somehow jump over a black hole and decelerate matter towards the global collision point.

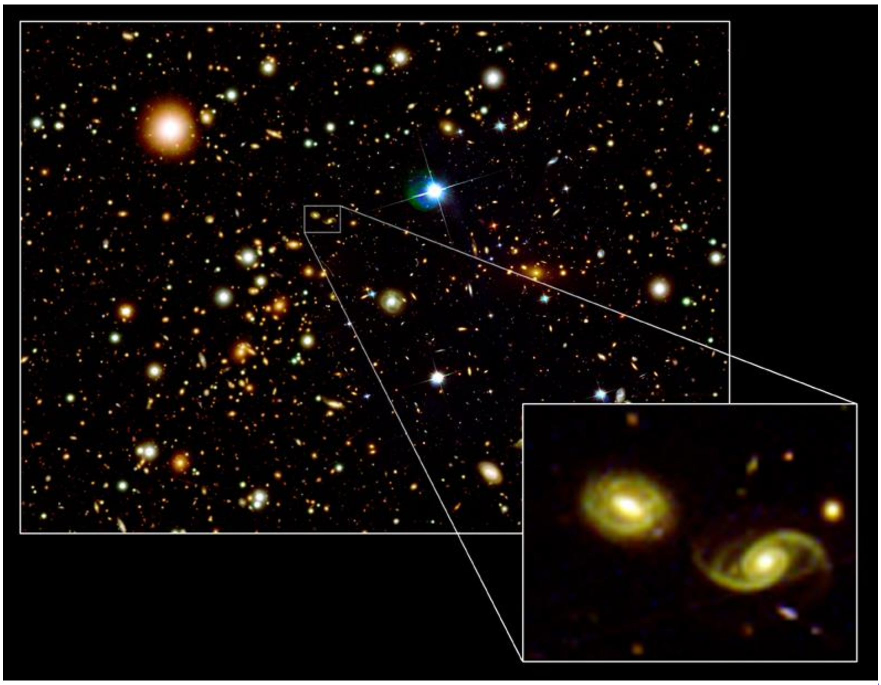

Figure 1: Galaxy Cluster in Perspective [4]

This optical image from Hubble and Magellan shows a close-up (inset) of one of the galaxies, a spiral galaxy approximately the same size as the Milky Way, within the galaxy cluster known as 1E 0657-56. The full-field view shows over a thousand galaxies in this cluster. These immense objects are among the largest structures in the Universe.

View Motion Graphic
Scale: Full-field image is 7.5 x 5.4 arcmin
(Credit: NASA/STScI; Magellan/U. Arizona/D. Clowe et al.)

The structure of the surrounding matter of each black hole undergoes changes that are directly proportional to changes in each black hole's MAD system. The effect of massive bosons with spin +2 rather than -2 or the effect of dark energy appears from 2000000 light years around a central black hole, i.e. 40 units further than the radius of a galaxy [5].

1.2. Asymmetric Entanglement Processes Observed from Graphs

$$We\ have\ (1)\ dE_{k3} = 4.\sqrt{}(gM^2.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{(2+2.k3)}+2^2}))\ (4dprocess);$$

$$(1-1)\ dE_{k3} = 4.\sqrt{}(1393.632885870757500518283954788 4.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2}))$$

1.2.1. Case#1

{gM: sqrt(1393.6328858707575005182839547884)
{if k3x! = 0 and k3x! = -1 and k3x! = -2 then (dEd(2^(7/4)*gM*sqrt(abs(k3x+1)))/(sqrt (abs (k3x))* abs(2*k3x+2)^(1/4))

f1Ek3x=-((2^(11/4)*gM*k3x^2+2^(19/4)*gM*k3x+2^(15/4)*gM)*sqrt(abs(k3x))*abs(2*k3x+2)^(3/4))/((k3x^3+4*k3x-^2+4*k3x)*sqrt(abs(k3x+1)))

f2Ek3x=((5*2^(3/4)*gM*k3x^3+25*2^(3/4)*gM*k3x^2+2^(23/4)*gM*k3x+3*2^(11/4)*gM)*abs(k3x)^(3/2)*abs(2*k3x+2)^(3/4))/((k3x^4+4*k3x^3+4*k3x^2)*abs(k3x+1)^(3/2))}}

→ Plots

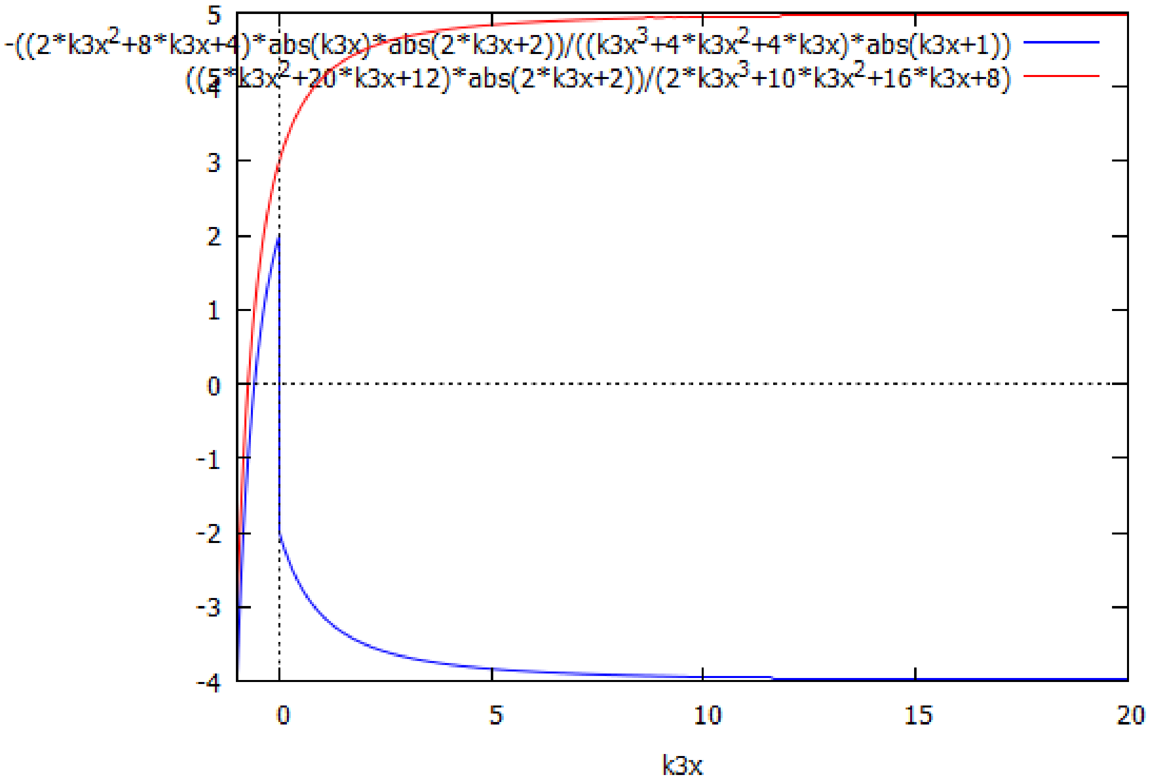

Figure 2: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(k3x)*abs(2*k3x+2))/((k3x^3+4*k3x^2+4*k3x)* abs(k3x+1)), ((5*k3x^2+20*k3x-+12)*abs(2*k3x+2))/(2*k3x^3+10*k3x^2+16*k3x+8)], [k3x, -1, 20])$

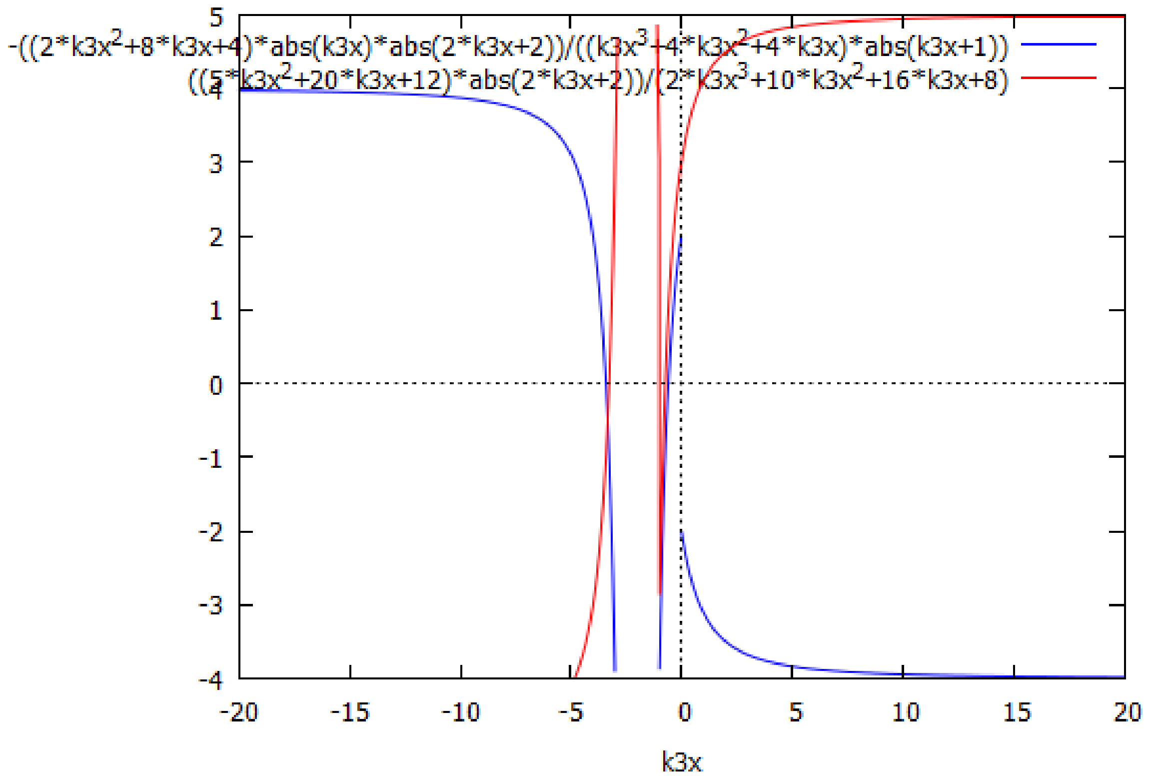

Figure 3: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(k3x)*abs(2*k3x+2))/((k3x^3+4*k3x^2+4*k3x)* abs(k3x+1)), ((5*k3x^2+20*k3x-+12)*abs(2*k3x+2))/(2*k3x^3+10*k3x^2+16*k3x+8)], [k3x,-20, 20],[y,-4,5])$

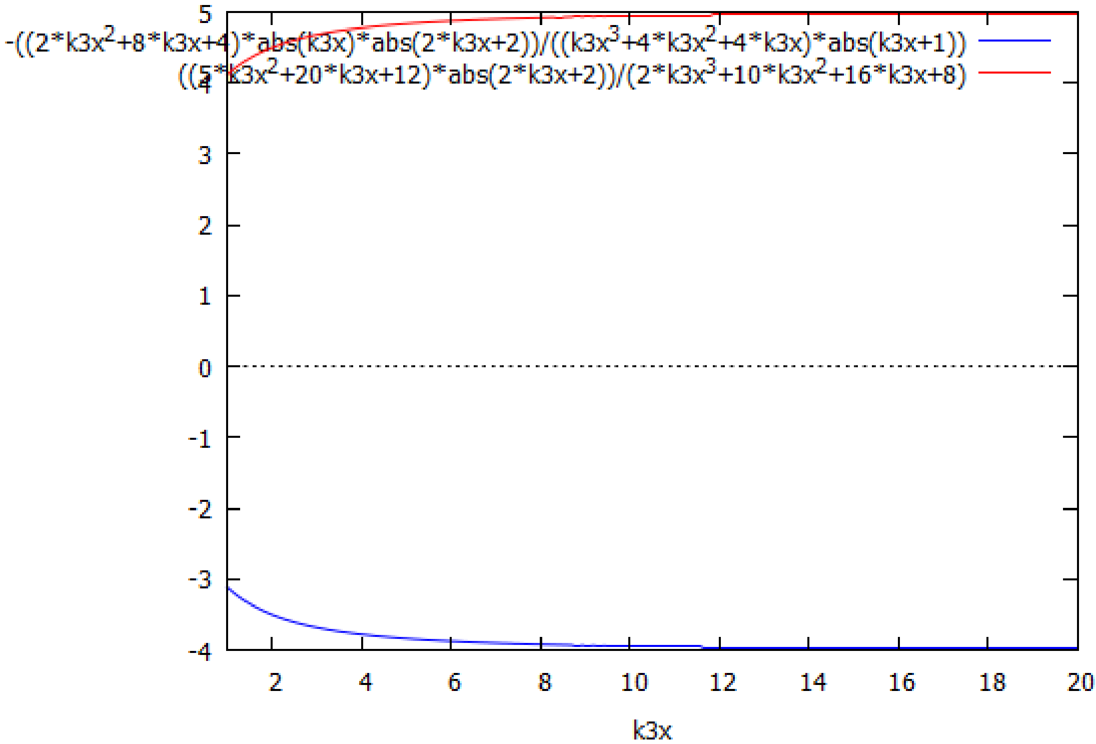

Figure 4: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(k3x)*abs(2*k3x+2))/((k3x^3+4*k3x^2+4*k3x)* abs(k3x+1)), ((5*k3x^2+20*k3x+12)*abs(2*k3x+2))/(2*k3x^3+10*k3x^2+16*k3x+8)], [k3x, 1, 20], [y,-4,5])$

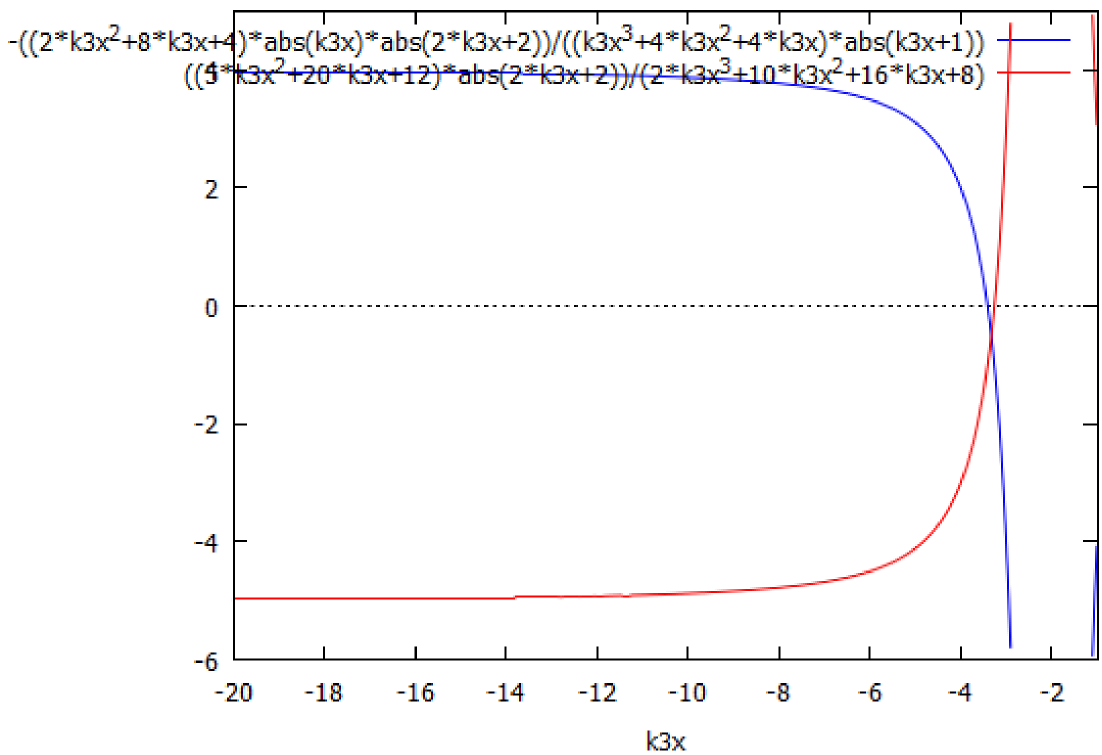

Figure 5: Plot2d ([-((2*k3x^2+8*k3x+4)* abs(k3x)*abs(2*k3x+2))/((k3x^3+4*k3x^2+4*k3x)* abs(k3x+1)), ((5*k3x^2+20*k3x+12)*abs(2*k3x+2))/(2*k3x^3+10*k3x^2+16*k3x+8)], [k3x,-20, -1], [y,-6,5])$

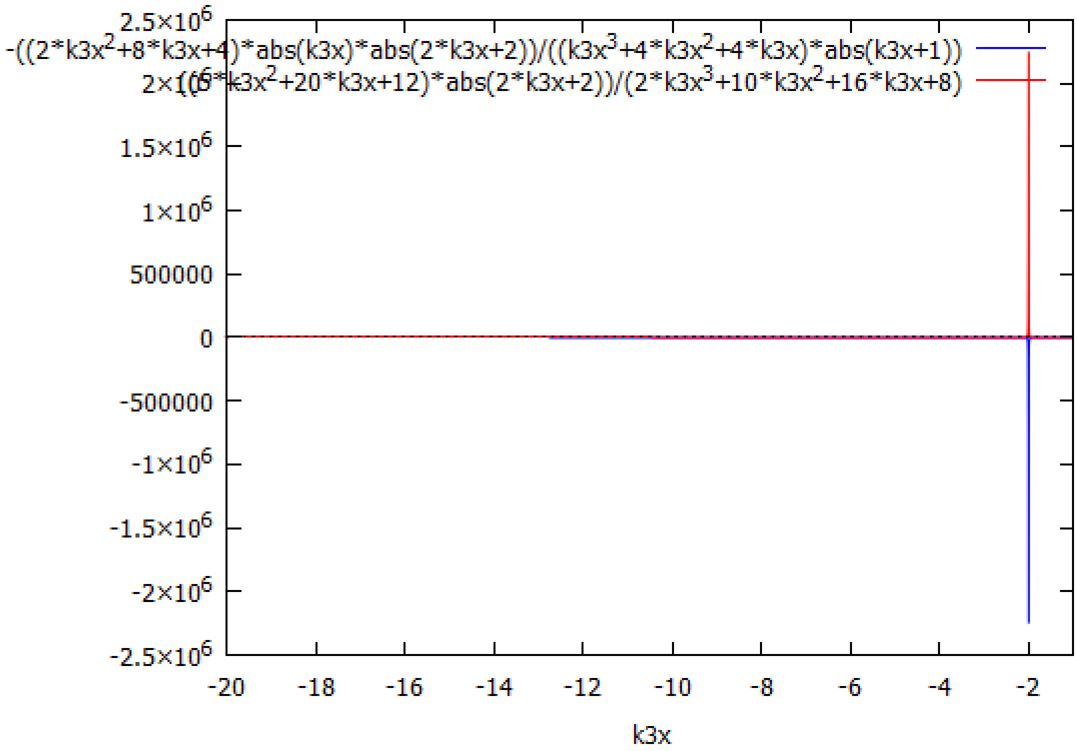

Figure 6: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(k3x)*abs(2*k3x+2))/((k3x^3+4*k3x^2+4*k3x)* abs(k3x+1)), ((5*k3x^2+20*k3x+12)*abs(2*k3x+2))/(2*k3x^3+10*k3x^2+16*k3x+8)], [k3x, -20, -1])$

1.2.2. Case#2

{gM = sqrt(1393.6328858707575005182839547884),
{if k3x! = 0 and k3x! = -1 and k3x! = -2 then (dEd = (2^(7/4)*gM*sqrt(abs(k3x+1)))/(sqrt(abs (k3x))*abs(2*k3x+2)^(1/4))

f1Ek3x
= -((2^(11/4)*gM*k3x^3+5*2^(11/4)*gM*k3x^2+3*2^(15/4)*gM*k3x+2^(15/4)*gM)*abs(k3x)^(3/2)*abs(2*k3x+2)^(3/4))/((k3x^4+4*k3x^3+4*k3x^2)*abs(k3x+1)^(3/2))

f2Ek3x
=((2^(27/4)*gM*k3x+2^(27/4)*gM)*sqrt(abs(k3x))*abs(2*k3x+2)^(3/4))/((k3x^4+8*k3x^3+24*k3x^2+32*k3x+16)*sqrt(abs(k3x+1)))

f3Ek3x
= -((5*2^(11/4)*gM*k3x^5+45*2^(11/4)*gM*k3x^4+71*2^(15/4)*gM*k3x^3+95*2^(15/4)*gM*k3x^2+7*2^(27/4)*gM*k3x+3*2^(23/4)*gM)*abs(k3x)^(3/2)*abs(2*k3x+2)^(3/4))/((k3x^6+8*k3x^5+24*k3x^4+32*k3x^3+16*k3x^2)*abs(k3x+1)^(3/2))

f4Ek3x
= -((5*2^(11/4)*gM*k3x^5+45*2^(11/4)*gM*k3x^4+71*2^(15/4)*gM*k3x^3+95*2^(15/4)*gM*k3x^2+7*2^(27/4)*gM*k3x+3*2^(23/4)*gM)*abs(k3x)^(3/2)*abs(2*k3x+2)^(3/4))/((k3x^6+8*k3x^5+24*k3x^4+32*k3x^3+16*k3x^2)*abs(k3x+1)^(3/2))

f5Ek3x
= (k3x^2*(45*gM*k3x^4+315*gM*k3x^3+594*gM*k3x^2+444*gM*k3x+120*gM)*sqrt(abs(k3x))*abs(2*k3x+2)^(7/4))/((k3x+1)^2*(2^(1/4)*k3x^6+3*2^(5/4)*k3x^5+3*2^(9/4)*k3x^4+2^(13/4)*k3x^3)*sqrt(abs (k3x+1))))}}

→ {(-((2^(11/4)*gM*k3x^3+5*2^(11/4)*gM*k3x^2+3*2^(15/4)*gM*k3x+2^(15/4)*gM)*abs(k3x)^(3/2)*abs(2*k3x+2)^(3/4))/((k3x^4+4*k3x^3+4*k3x^2)*abs(k3x+1)^(3/2)))/((2^(7/4)*gM*sqrt(abs(k3x+1)))/(sqrt(abs(k3x))*abs(2*k3x+2)^(1/4))),

(((2^(27/4)*gM*k3x+2^(27/4)*gM)*sqrt(abs(k3x))*abs(2*k3x+2)^(3/4))/((k3x^4+8*k3x^3+24*k3x^2+32*k3x+16)*sqrt(abs(k3x+1))))/((2^(7/4)*gM*sqrt(abs(k3x+1)))/(sqrt(abs(k3x))*abs(2*k3x+2)^(1/4))),

(-((5*2^(11/4)*gM*k3x^5+45*2^(11/4)*gM*k3x^4+71*2^(15/4)*gM*k3x^3+95*2^(15/4)*gM*k3x^2+7*2^(27/4)*gM*k3x-+3*2^(23/4)*gM)*abs(k3x)^(3/2)*abs(2*k3x+2)^(3/4))/((k3x^6+8*k3x^5+24*k3x^4+32*k3x^3+16*k3x^2)*abs(k3x+1)^(3/2)))/((2^(7/4)*gM*sqrt(abs(k3x+1)))/(sqrt(abs(k3x))*abs(2*k3x+2)^(1/4))),
...
/((2^(7/4)*gM*sqrt(abs(k3x+1)))/(sqrt(abs(k3x))*abs(2*k3x+2)^(1/4)))

((k3x^2*(45*gM*k3x^4+315*gM*k3x^3+594*gM*k3x^2+444*gM*k3x+120*gM)*sqrt(abs(k3x))*abs(2*k3x+2)^(7/4))/((k3x+1)^2*(2^(1/4)*k3x^6+3*2^(5/4)*k3x^5+3*2^(9/4)*k3x^4+2^(13/4)*k3x^3)*sqrt(abs(k3x+1))))/((2^(7/4)*gM*sqrt(abs(k3x+1)))/(sqrt(abs(k3x))*abs(2*k3x+2)^(1/4)))}

→ {-((2*k3x^2+8*k3x+4)*abs(2*k3x+2))/(k3x^3+5*k3x^2+8*k3x+4), ((32*k3x+32)*abs(k3x)* abs(2*k3x+2))/((k3x^4+8*k3x-^3+24*k3x^2+32*k3x+16)*abs(k3x+1)),
-((10*k3x^4+80*k3x^3+204*k3x^2+176*k3x+48)*abs(2*k3x+2))/(k3x^5+9*k3x^4+32*k3x^3 + 56*k3x^2+ 48*k3x+16),

Idem,
((45*k3x^4+315*k3x^3+594*k3x^2+444*k3x+120)*abs(k3x))/((k3x^4+6*k3x^3+12*k3x^2+8*k3x)*abs(k 3x+1))}

Plots

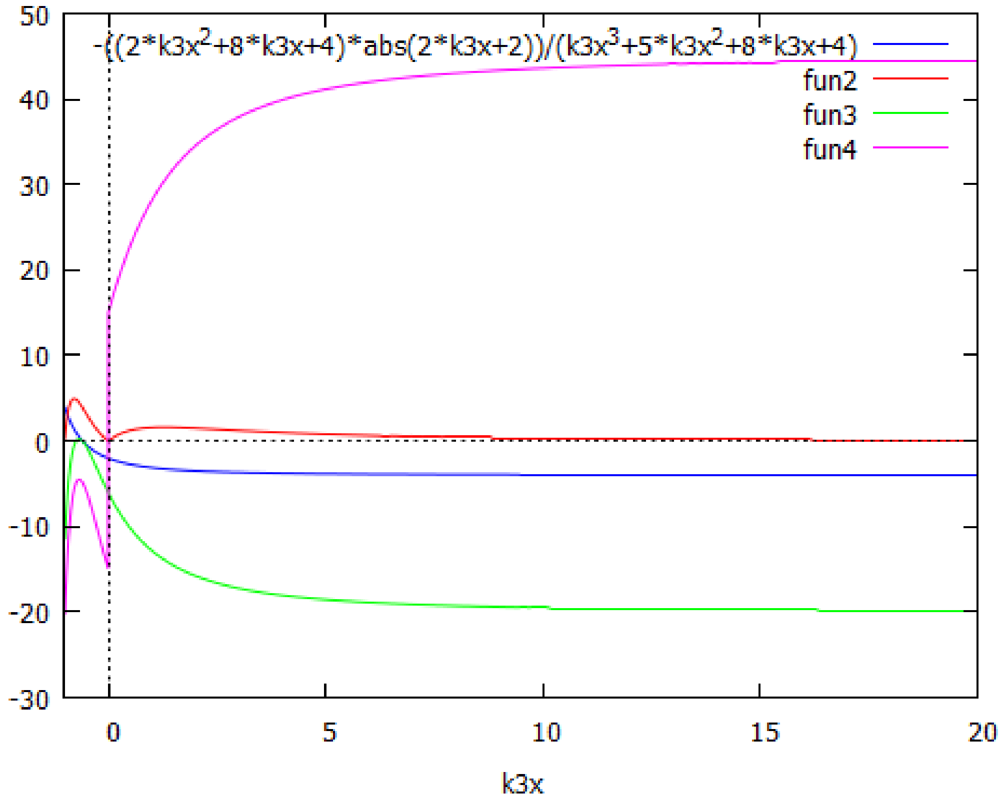

Figure 7: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(2*k3x+2))/(k3x^3+5*k3x^2+8*k3x+4), ((32*k3x + 32)*abs(k3x)*abs(2*k3x+2))/((k3x^4+8*k3x^3+24*k3x^2+32*k3x+16)*abs(k3x+1)),-((10*k3x^4+80*k3x^3+204*k3x^2+176*k3x+48)*abs(2*k3x+2))/(k3x^5+9*k3x^4+32*k3x^3+56*k3x^2+48*k3x+16),((45*k3x^4+315*k3x^3+594*k3x^2+444*k3x+120)*abs(k3x))/((k3x-^4+6*k3x^3 +12*k3x^2+8*k3x)*abs(k3x+1))], [k3x, -1, 20])$

or

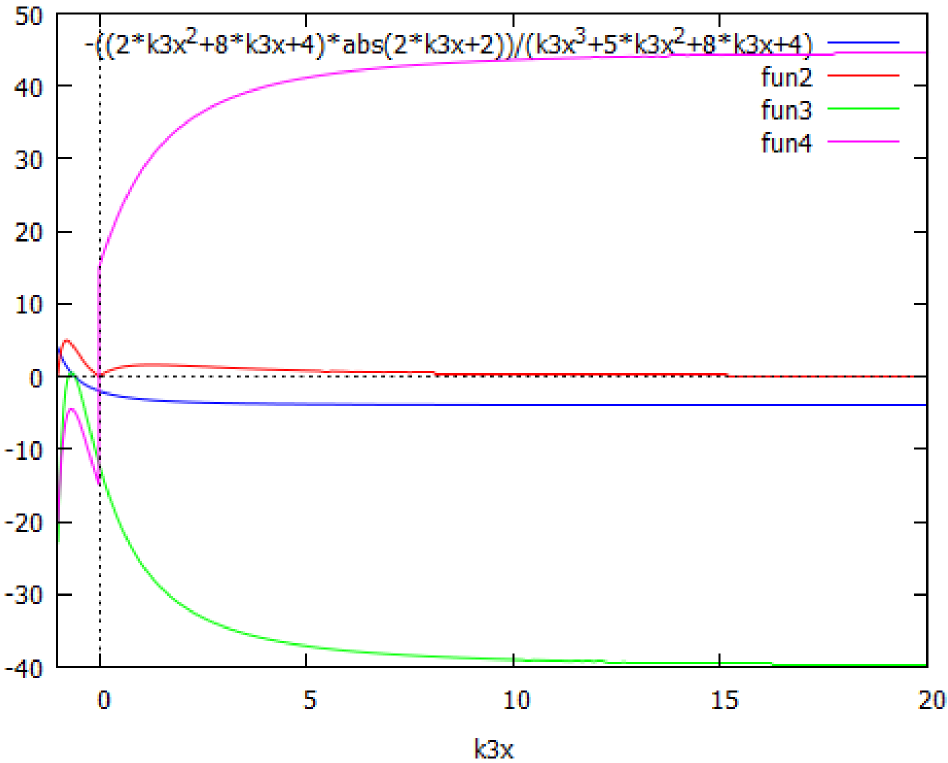

Figure 8: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(2*k3x+2))/(k3x^3+5*k3x^2+8*k3x+4), ((32*k3x +32)*abs(k3x)*abs(2*k3x+2))/((k3x^4+8*k3x^3+24*k3x^2+32*k3x+16)*abs(k3x+1)),2*(-((10*k3x^4+80*k3x^3+204*k3x^2+176*k3x+48)*abs(2*k3x+2))/(k3x^5+9*k3x^4+32*k3x^3+56*k3x^2+48*k3x+16)),((45*k3x^4+315*k3x^3+594*k3x^2+444*k3x+120)*abs(k3x))/((k3x^4+6*k3x^3+12*k3x^2+8*k3x)*abs(k3x+1))], [k3x, -1, 20])$

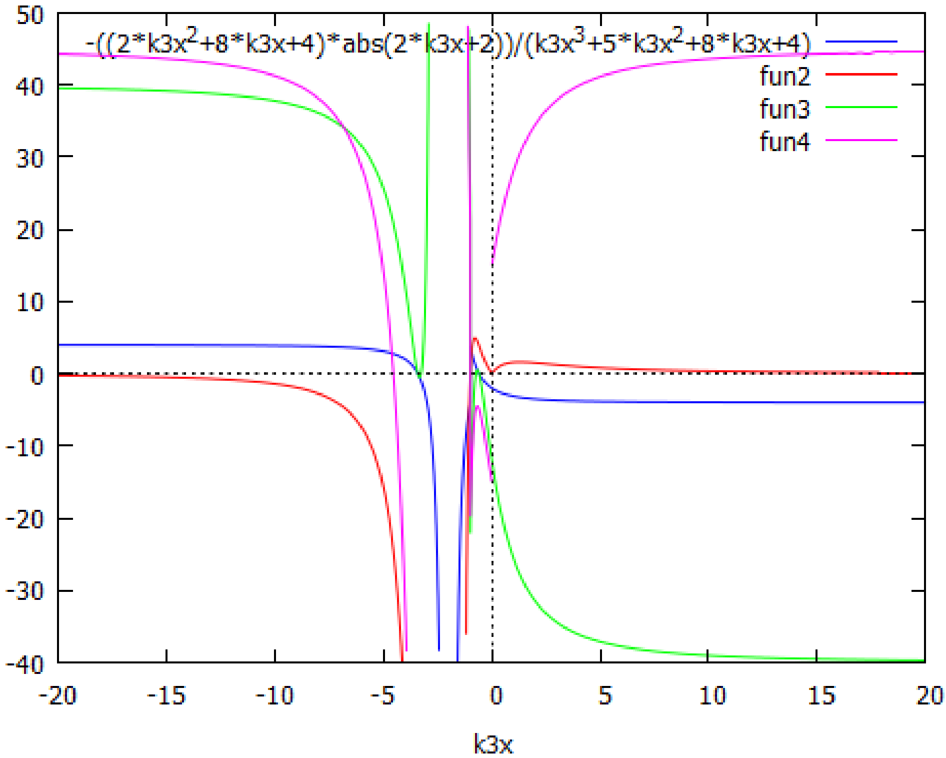

Figure 9: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(2*k3x+2))/(k3x^3+5*k3x^2+8*k3x+4), ((32*k3x +32)*abs(k3x)*abs(2*k3x+2))/((k3x^4+8*k3x^3+24*k3x^2+32*k3x+16)*abs(k3x+1)), 2*(-((10 *k3x^4+80*k3x^3+204*k3x^2+176*k3x+48)*abs(2*k3x+2))/(k3x^5+9*k3x^4+32*k3x^3+56*k3x^2+48*k3x+16)),((45*k3x^4+315*k3x^3+594*k3x^2+444*k3x+120)*abs(k3x))/((k3x^4+6* k3x^3+12*k3x^2+8*k3x)*abs(k3x+1))], [k3x, -20, 20],[y,-40,50])$

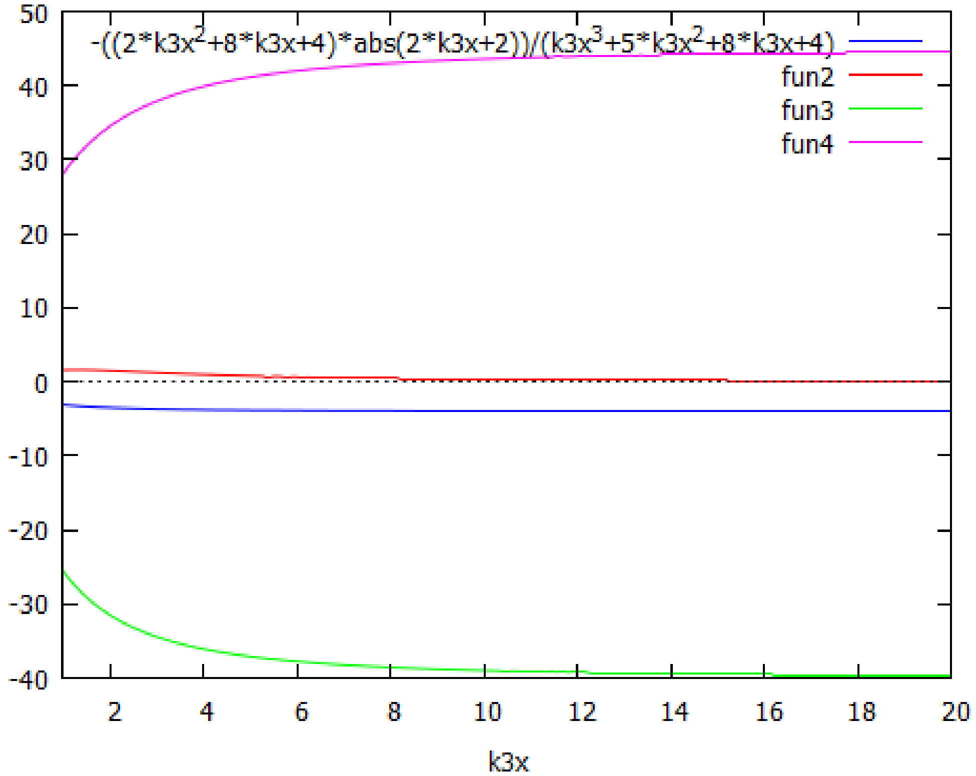

Figure 10: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(2*k3x+2))/(k3x^3+5*k3x^2+8*k3x+4), ((32*k3x +32)*abs(k3x)*abs(2*k3x+2))/((k3x^4+8*k3x^3+24*k3x^2+32*k3x+16)*abs(k3x+1)), 2*(- ((10*k3x^4+80*k3x^3+204*k3x^2+176*k3x+48)*abs(2*k3x+2))/(k3x^5+9*k3x^4+32*k3x^3+56*k3x^2+48*k3x+16)), ((45*k3x^4+315*k3x^3+594*k3x^2+444*k3x+120)*abs(k3x))/((k3x^4 + 6*k3x^3+12*k3x^2+8*k3x)*abs(k3x+1))], [k3x, 1, 20],[y,- 40,50])$

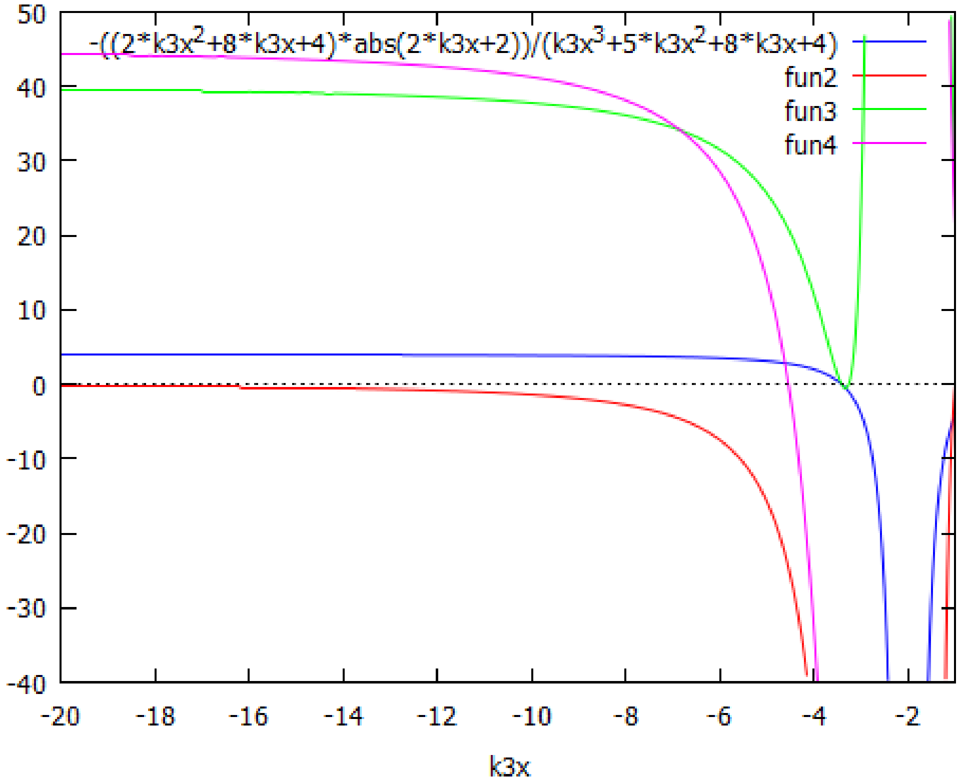

Figure 11: Plot2d ([-((2*k3x^2+8*k3x+4)*abs(2*k3x+2))/(k3x^3+5*k3x^2+8*k3x+4), ((32*k3x +32)*abs(k3x)*abs(2*k3x+2))/((k3x^4+8*k3x^3+24*k3x^2+32*k3x+16)*abs(k3x+1)),2*(-((10*k3x^4+80*k3x^3+204*k3x^2+176*k3x+48)*abs(2*k 3x+2))/(k3x^5+9*k3x^4+32*k3x^3+56*k3x^2+48*k3x+16)),((45*k3x^4+315*k3x^3+594*k3x^2+444*k3x+120)*abs(k3x))/((k3x^4+6*k 3x^3+12*k3x^2+8*k3x)*abs(k3x+1))], [k3x, -20, - 1],[y,-40,50])$

Obtaining of an algorithm for the more massive dark energy's massive boson and the distance at which the effects of dark energy begin. A quantum multi-entanglement system due to {Δk3 = dk3d or Δk3 = dk3n} ≠ 0 allowing both a tendency towards the idea of a single time and the fact of a multi dilation of time.

Constants Multi-Parts Effective of Things

$$(1bis)\ dE_{k3} = 4.\sqrt{}(gM^2.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{(2+2.k3)}+2^2}))\ (4dprocess)$$

→f0.d(dE[k3])/dk3=dE[k3]→f0=2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))

then

f1.d(f0.d(dE[k3])/dk3)/dk3 = dE[k3] → f1 = (4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))))/(d(f0*d(dE[k3])/dk3)/dk3)

f1=(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))))/(diff((2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2))))*diff((4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2)))),k3,1),k3,1))

f1 = -(4*k3^2+4*k3)/(k3+2), and dE[k3] = f1*(d(f0)/dk3)*d(dE[k3])/dk3 + f1*f0* d²(dE[k3])/d k3²,

Parts

f1*(d(f0)/dk3)*d(dE[k3])/dk3 + f1*f0*d²(dE[k3])/dk3² = -(sqrt(2*k3+2)* (2^(11/4)*gM*k3^2+ 2^ (19/4)*gM*k3+2^(15/4)*gM))/((k3^3+4*k3^2+4*k3)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))) + (5*2^(3/4)*gM*k3^3+25*2^(3/4)*gM*k3^2+2^(23/4)*gM*k3+3*2^(11/4)*gM)/((k3^4+4*k3^3+4*k3^2)*((k3+1)/(k3*sqrt(2*k3+2)))^(3/2)),

∫dE[k3]→=∫-(sqrt(2*k3+2)*(2^(11/4)*gM*k3^2+2^(19/4)*gM*k3+2^(15/4)*gM))/((k3^3+4*k3^2+4*k3)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))),k3)+∫(5*2^(3/4)*gM*k3^3+25*2^(3/4)*gM*k3^2+2^(23/4)*gM*k3+3*2^(11/4)*gM)/((k3^4+4*k3^3+4*k3^2)*((k3+1)/(k3*sqrt(2*k3+2)))^(3/2)).

f2.d(f1.d(f0.d(dE[k3])/dk3)/dk3)/dk3 = dE[k3] → f2 =(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3)) ^ (1/2) /(-2^3/(2+2*k3)+2^2)))) / (d(f1*d(f0*d(dE[k3])/dk3)/dk3)/dk3)

f2=(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))))/((((-4*k3^2)-4*k3)*((3*2^(3/4)*abs(gM)*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2))^2)/(sqrt(2*k3+2)*(4-8/(2*k3+2))*(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))^(5/2))+(3*2^(11/4)*abs(g M))/((2*k3+2)^(5/2)*(4-8/(2*k3+2))*sqrt(1/(sqrt(2*k3+2)*(48/(2*k3+2)))))+(3*2^(31/4)*abs(gM))/((2*k3+2)^(7/2)*(4-8/ (2*k3+ 2))^2*sqrt(1/(sqrt(2*k3+2)*(4-8/(2*k3+2)))))+(2^(47/4)*abs(gM))/((2*k3+2)^(9/2)*(4-8/(2*k3+2))^3*sqrt(1/(sqrt(2*k3+2)*(4-8/(2*k3+2)))))+(2^(11/4)*abs(gM)*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))/((2*k3+2)^(3/2)*(4-8/(2*k3+2))*(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))^(3/2))+(2^(27/4)*abs(gM)*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2*(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))^(3/2))-(2^(7/4)*abs(gM)*(3/((2*k3+2)^(5/2)*(4-8/(2*k3+2)))+96/((2*k3+2)^(7/2)*(4-8/(2*k3+2))^2)+512/((2*k3+2)^(9/2)*(4-8/(2*k3+2))^3)))/(sqrt(2*k3+2)*(48/(2*k3+2))*(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))^(3/2))))/(k3+2)+(((-8*k3)-4)*((-(2^(11/4)*abs(gM))/((2*k3+2)^(3/2)*(4-8/(2*k3+2))*sqrt(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))))-(2^(27/4)*abs(gM))/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2*sqrt(1/(sqrt(2*k3+2)*(4-8/(2*k3+2)))))-(2^(7/4)*abs(gM)*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))/(sqrt(2*k3+2)*(4-8/(2*k3+2))*(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))^(3/2))))/(k3+2)-((((-4*k3^2)-4*k3)*((-(2^(11/4)*abs(gM))/((2*k3+2)^(3/2)*(4-8/(2*k3+2))*sqrt(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))))-(2^(27/4)*abs(gM))/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2*sqrt(1/(sqrt(2*k3+2)*(4-8/(2*k3+2)))))-(2^(7/4)*abs(gM)*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))/(sqrt(2*k3+2)*(4-8/(2*k3+2))*(1/(sqrt(2*k3+2)*(4-8/(2*k3+2))))^(3/2))))/(k3+2)^2)

f2 = -(4*k3^2+4*k3)/(k3+2)= f1.

1.3. Parts

f2.d(f1.d(f0.d(dE[k3])/dk3)/dk3)/dk3=f2.d(f1.{(d(f0)/dk3)*d(dE[k3])/dk3 + f0*d²(dE[k3])/ dk3²}) /dk3 f2.(d(f1.{(d(f0)/dk3)*d(dE[k3])/dk3 + f0*d^2(dE[k3])/dk3^2})/dk3) = f2.((d(f1)/dk3)*{(d(f0)/dk3)* d(dE[k3])/dk3 + f0*d^2(dE[k3])/dk3^2}+ f1*{{d^2(f0)/dk3^2*d(dE[k3])/dk3 + (d^2(dE[k3])/dk3^2)* (d(f0)/dk3)} + d(f0)/dk3*d^2(dE[k3])/dk3^2 + (d^3(dE[k3])/dk3^3)*f0}) = {f2.(d(f1)/dk3)*(d(f0)/dk3)* d(dE[k3])/dk3+ f2.(d(f1)/dk3)*f0*d^2(dE[k3])/dk3^2+ f2.f1*d^2(f0)/dk3^2*d(dE[k3])/dk3 + f2.f1*(d^2(dE[k3])/dk3^2)*(d(f0)/dk3) + f2.f1*d(f0)/dk3*d^2(dE[k3])/dk3^2 + f2.f1*(d^3(dE[k3])/dk3^3)*f0};

{(-(4*k3^2+4*k3)/(k3+2))*(diff((-(4*k3^2+4*k3)/(k3+2)),k3,1))*(diff((2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))),k3,1))*(diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,1))

+(-(4*k3^2+4*k3)/(k3+2))*(diff((-(4*k3^2+4*k3)/(k3+2)),k3,1))*(2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2))))*(diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,2))

+(-(4*k3^2+4*k3)/(k3+2))*(-(4*k3^2+4*k3)/(k3+2))*diff((2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))),k3,2)*(diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,1))

+(-(4*k3^2+4*k3)/(k3+2))*(-(4*k3^2+4*k3)/(k3+2))*((diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,2)))*(diff((2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))),k3,1))

+(-(4*k3^2+4*k3)/(k3+2))*(-(4*k3^2+4*k3)/(k3+2))*diff((2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2)))),k3,1)*(diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,2))

+(-(4*k3^2+4*k3)/(k3+2))*(-(4*k3^2+4*k3)/(k3+2))*((diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,3)))*(2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4-8/(2*k3+2))^2))))};

(2) $$\Big\{-\frac{(2^{(11/4)}.gM.k3^3+5.2^{(11/4)}.gM.k3^2+3.2^{(15/4)}.gM.k3+2^{(15/4)}.gM)}{((k3^4+4.k3^3+4.k3^2).((k3+1)/(k3.\surd(2.k3+2)))^{(3/2)})}$$

$$+\frac{(\surd(2.k3+2).(2^{(27/4)}.gM.k3+2^{(27/4)}.gM))}{((k3^4+8.k3^3+24.k3^2+32.k3+16).\surd((k3+1)/(k3.\surd(2.k3+2))))}$$

$$-\frac{(5.2^{(11/4)}.gM.k3^5+45.2^{(11/4)}.gM.k3^4+71.2^{(15/4)}.gM.k3^3+95.2^{(15/4)}.gM.k3^2+7.2^{(27/4)}.gM.k3+3.2^{(23/4)}.gM)}{((k3^6+8.k3^5+24.k3^4+32.k3^3+16.k3^2).((k3+1)/(k3.\surd(2.k3+2)))^{(3/2)})}$$

$$-\frac{(5.2^{(11/4)}.gM.k3^5+45.2^{(11/4)}.gM.k3^4+71.2^{(15/4)}.gM.k3^3+95.2^{(15/4)}.gM.k3^2+7.2^{(27/4)}.gM.k3+3.2^{(23/4)}.gM)}{((k3^6+8.k3^5+24.k3^4+32.k3^3+16.k3^2).((k3+1)/(k3.\surd(2.k3+2)))^{(3/2)})}$$

$$+\frac{(\surd(2.k3+2).(45.gM.k3^4+315.gM.k3^3+594.gM.k3^2+444.gM.k3+120.gM))}{((2^{(1/4)}.k3^6+3.2^{(5/4)}.k3^5+3.2^{(9/4)}.k3^4+2^{(13/4)}.k3^3).((k3+1)/(k3.\surd(2.k3+2)))^{(5/2)})}$$

},

(2-1) $$\int dE_{k3} \longrightarrow= \int -\frac{(2^{(11/4)}.gM.k3^3+5.2^{(11/4)}.gM.k3^2+3.2^{(15/4)}.gM.k3+2^{(15/4)}.gM)}{((k3^4+4.k3^3+4.k3^2).((k3+1)/(k3.\surd(2.k3+2)))^{(3/2)})}$$

$$+\int\frac{(\surd(2.k3+2).(2^{(27/4)}.gM.k3+2^{(27/4)}.gM))}{((k3^4+8.k3^3+24.k3^2+32.k3+16).\surd((k3+1)/(k3.\surd(2.k3+2))))}$$

$$+\int-\frac{(5.2^{(11/4)}.gM.k3^5+45.2^{(11/4)}.gM.k3^4+71.2^{(15/4)}.gM.k3^3+95.2^{(15/4)}.gM.k3^2+7.2^{(27/4)}.gM.k3+3.2^{(23/4)}.gM)}{((k3^6+8.k3^5+24.k3^4+32.k3^3+16.k3^2).((k3+1)/(k3.\surd(2.k3+2)))^{(3/2)})}$$

$$+\int-\frac{(5.2^{(11/4)}.gM.k3^5+45.2^{(11/4)}.gM.k3^4+71.2^{(15/4)}.gM.k3^3+95.2^{(15/4)}.gM.k3^2+7.2^{(27/4)}.gM.k3+3.2^{(23/4)}.gM)}{((k3^6+8.k3^5+24.k3^4+32.k3^3+16.k3^2).((k3+1)/(k3.\surd(2.k3+2)))^{(3/2)})}$$

$$+\int\frac{(\surd(2.k3+2).(45.gM.k3^4+315.gM.k3^3+594.gM.k3^2+444.gM.k3+120.gM))}{((2^{(1/4)}.k3^6+3.2^{(5/4)}.k3^5+3.2^{(9/4)}.k3^4+2^{(13/4)}.k3^3).((k3+1)/(k3.\surd(2.k3+2)))^{(5/2)})}$$

With
f0
= (2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2) *(4-8/(2*k3+2))^2))))

f1 = -(4*k3^2+4*k3)/(k3+2) = f2 d(f0)/dk3 = (diff(2/(sqrt(2*k3+2)*(4-8/(2*k3+2))*((-1/((2*k3+2)^(3/2)*(4-8/(2*k3+2))))-16/((2*k3+2)^(5/2)*(4- 8/(2*k3+2))^2))),k3,1))

d(dE[k3])/dk3=(diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,1))

d²(dE[k3])/dk3² = (diff(4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2))),k3,2))

1.4. Next Algorithm

(3) It makes it possible to obtain a permanent distribution of dark matter effects from the injunction ΣE_darkenergy/ΣE_darkmatter = 3 [1].

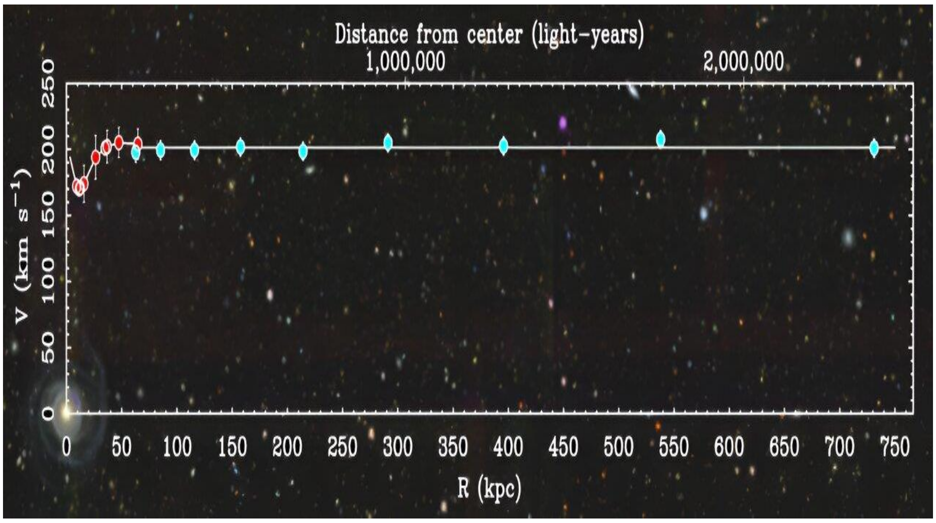

Figure 12: The primary technique Mistele used in his research, gravitational lensing, is a phenomenon predicted by Einstein's theory of general relativity. As part of the research, Mistele plotted out what's called Tully–Fisher relation on a chart to highlight the empirical relationship between the visible mass of a galaxy and its rotation speed [1].

```
from math import*
#An algorithm giving sets of massive bosons-dark energy: {dE[k3] = 2^(7/4)*gM*sqrt((k3d+1)/ (k3d*sqrt(2*k3d+2))), k3d = 1+n.dk3d}

#and sets of massive bosons-dark matter: {dE[k3] = 2^(7/4)*gM*sqrt((k3n+1)/(k3n*sqrt(2*k3n+ 2))), k3n = 1+n.dk3n}

gM = sqrt(1393.6328858707575005182839547884)
rangeX = 100
dk3n = 1.255314934818507*10**89
dk3d = 1.862840706286786*10**147

#[a+b = -1.862840706286786*10^147-rangeX*dk3n, a*rangeX+b=1.862840706286786*10^147 +rangeX*dk3n], [a,b]

a1 = (2*dk3n*rangeX+37256814125735720537987322319883924239243665409930790835462011309401573861968420889597095558741105801291734075132510755282376484359760268673406730 24)/(rangeX-1)

b1= -(dk3n*rangeX**2+(dk3n+18628407062867860268993661159941962119621832704965395417731005654700786930984210444798547779370552900645867037566255377641188242179880134336703365 12)*rangeX+18628407062867860268993661159941962119621832704965395417731005654700786930984210444798547779370552900645867037566255377641188242179880134336703365 12)/(rangeX-1)
```

```
#[a+b = -1.862840706286786*10^147-rangeX*dk3d, a*rangeX+b=1.862840706286786*10 ^147 +rangeX*dk3d],[a,b]

a2=(2*dk3d*rangeX+372568141257357205379873223198839242392436654099307908354620113094015738619684208895970955587
4110580129173407513251075528237648435976026867340673024)/(rangeX-1)

b2=-(dk3d*rangeX**2+(dk3d+1862840706286786026899366115994196211962183270496539541773100565470078693098421044479
854777937055290064586703756625537764118824217988013433670336512)*rangeX+186284070628678602689936611599419621196
2183270496539541773100565470078693098421044479854777937055290064586703756625537764118824217988013433670336512)/
(rangeX-1)

for q in range (1,300 + 1):
Sidn = 0
Sidd = 0
tmax = 10*10**q

d0 =1.862840706286786*10**147
for t0 in range(1,10000 + 1) :
if d0 > 10**(-3)*1.862840706286786*10**147 :
Sidn = 0
Sidd = 0
k3n = 1.862840706286786*10**147
 k3d = 1.862840706286786*10**147
for t in range(1, tmax + 1):
for k3n1i in range(1, rangeX + 1):
k3n1 = a1*k3n1i+b1
 k3n2 = k3n-k3n1
for k3d1i in range(1, rangeX + 1):
k3d1 = a2*k3d1i+b2
 k3d2 = k3d-k3d1
#[k3!= -1, k3!= -2, k3!= 0]

if k3n1*k3n2*k3d1*k3d2!= 0 and (k3n1!= -1 or k3n2!= -1 or k3d1!= -1 or k3d2!= -1) and (k3n1!= -2 or k3n2!=-2 or k3d1!= -2 or
k3d2!= -2):

fiEk3n1 = -((2**(11/4)*gM*k3n1**2+2**(19/4)*gM*k3n1+2**(15/4)*gM)*sqrt(abs(k3n1))*abs(2*k3n1+2)**(3/4))/((k3n1**
3+4*k3n1**2+4*k3n1)*sqrt(abs(k3n1+1)))

#imfiEk3n1 = 0
fiEk3d1= -((2**(11/4)*gM*k3d1**2+2**(19/4)*gM*k3d1+2**(15/4)*gM)*sqrt(abs(k3d1))*abs(2*k3d1+2)**(3/4))/((k3d1**
3+4*k3d1**2+4*k3d1)*sqrt(abs(k3d1+1)))

#imfiEk3d1 = 0
fiEk3n2=((5*2**(3/4)*gM*k3n2**3+25*2**(3/4)*gM*k3n2**2+2**(23/4)*gM*k3n2+3*2**(11/4)*gM)*abs(k3n2)**(3/2)*abs(
2*k3n2+2)**(3/4))/((k3n2**4+4*k3n2**3+4*k3n2**2)*abs(k3n2+1)**(3/2))

#imfiEk3n2 = 0
fiEk3d2=((5*2**(3/4)*gM*k3d2**3+25*2**(3/4)*gM*k3d2**2+2**(23/4)*gM*k3d2+3*2**(11/4)*gM)*abs(k3d2)**(3/2)*abs(
2*k3d2+2)**(3/4))/((k3d2**4+4*k3d2**3+4*k3d2**2)*abs(k3d2+1)**(3/2))

#imfiEk3d2 = 0
#fiEk3n = 2**(7/4)*sqrt((k3n+1)/(k3n*sqrt(2*k3n+2)))
#fiEk3d = 2**(7/4)*sqrt((k3d+1)/(k3d*sqrt(2*k3d+2)))
 Sidn = fiEk3n1*dk3n + fiEk3n2*dk3n + Sidn
Sidd = fiEk3d1*dk3d + fiEk3d2*dk3d + Sidd
k3n = k3n+dk3n
k3d = k3d+dk3d
if t = = tmax or (Sidd/Sidn >= 3 and Sidd/Sidn < 4):
if (Sidd/Sidn >= 3 and Sidd/Sidn < 4):
```

```
k3n = k3n-dk3n
 k3d = k3d-dk3d
#imdEd = 0
dEd = (2**(7/4)*gM*sqrt(abs(dk3d+1)))/(sqrt(abs(dk3d))*abs(2*dk3d+2)**(1/4))
print("dk3n = ",dk3n, ", {dk3d = ",dk3d, ", dEd = ",dEd," GeV, Sidd = ",Sidd," GeV, Sidn = ",Sidn," GeV}")
dk3d = dk3d-d0
d0 = d0/10
elif Sidd/Sidn < 3:
k3n = k3n-dk3n
k3d = k3d-dk3d0
dk3d = dk3d+d0
elif Sidd/Sidn >= 4:
k3n = k3n-dk3n
k3d = k3d-dk3d
dk3d : dk3d-d0/10
```

→ 10 random results obtained from this algorithm:

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.79830744151244e-74 GeV, Sidd = 1.9655608964861628 e-36 GeV, Sidn = 1.720661362331221e-35 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 1.8997311639864714e-73 GeV, Sidd = 3.9470038906840624e-36 GeV, Sidn = 3.327536832849558e-35 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 1.292657988800866e-72 GeV, Sidd = 8.362666736674076e-36 GeV, Sidn = 6.946773655328476e-34 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 8.797703726238509e-72 GeV, Sidd = 1.7458341055671412e-35 GeV, Sidn = 1.4268487449721526e-33 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 5.978410274963025e-71 GeV, Sidd = 3.438907353997907e-35 GeV, Sidn = 2.7692856524524027e-32 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 4.069281829039938e-70 GeV, Sidd = 6.825821750343634e-35 GeV, Sidn = 5.4426496841847744e-31 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.7694444026077187e-69 GeV, Sidd = 1.3386847680207826e-34 GeV, Sidn = 2.6347047724784477e-30 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 1.8854892370910166e-68 GeV, Sidd = 2.59334318002548e-34 GeV, Sidn = 1.275580526733872e-29 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 1.2820697992200913e-67 GeV, Sidd = 4.9859340816602605e-34 GeV, Sidn = 2.452573181646847e-28 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 8.721950193075937e-67 GeV, Sidd = 9.639921928740116e-34 GeV, Sidn = 4.682117092280432e-27 GeV}

Then in C++ :

```
#include <cmath>
#include <boost/multiprecision/cpp_int.hpp>
 #include <iostream>
using namespace std;
int main() {
double gM = sqrt(1393.6328858707575005182839547884);
int rangeX = 100;
double dk3n = 1.255314934818507*pow(10,89);
double dk3d = 1.862840706286786*pow(10,147);
```

```
double a1= (2*dk3n*rangeX+3725681412573572053798732231988392423924366540993079083546201130940157386196842088959709555874110580129173407513251075528237648435976026867340673024)/(ran geX-1);

doubleb 1 =-(dk3n*rangeX*rangeX+(dk3n+18628407062867860268993661159941962119621832704965395417731005654700786930984210444798547779370552 90064586703756625537764118824217988013433670336512)*rangeX+1862840706286786026899366115994196211962183270496539541773100565470078693098 42104447985477793705529006458670375662553776411882421798801343367033 6512)/(ran geX-1);

double a2= (2*dk3d*rangeX+3725681412573572053798732231988392423924366540993079083546201130940157386196842088959709555874110580129173407513251075528237648435976026867340673024)/(ran geX-1);

double b2=-(dk3d*rangeX*rangeX+(dk3d+18628407062867860268993661159941962119621832704965395417731005654700786930984 21044479854777937055290064586703756625537764118824217988013433670336512)*rangeX+1862840706286786026899366115994196 21196218327049653954177310056547007869309842104447985477793705529006458670375662553776411882421798801343367033651 2)/(ran geX-1);

for (int q=1; q<=300; q++) {
 double Sidn = 0;
double Sidd = 0;
double tmax = 10*pow(10,q);
double d0 = 1.862840706286786*pow(10,147);
for (int t0=1; t0<=10000; t0++) {
if (d0 > 10e-3*1.862840706286786*pow(10,147)) {
Sidn = 0;
Sidd = 0;
}

double k3n = 1.862840706286786*pow(10,147);
double k3d = 1.862840706286786*pow(10,147);
 for (int t=1; t<=tmax; t++) {
for (int k3n1i=1; k3n1i<=rangeX; k3n1i++) {
double k3n1 = a1*k3n1i+b1;
double k3n2 = k3n-k3n1;
for (int k3d1i=1; k3d1i<=rangeX; k3d1i++) {
double k3d1 = a2*k3d1i+b2;
double k3d2 = k3d-k3d1;
if (k3n1*k3n2*k3d1*k3d2!= 0 && (k3n1!= -1 or k3n2!= -1 or k3d1!= -1 or k3d2!= -1) && (k3n1!= -2 or k3n2!= -2 or k3d1!= -2 or k3d2!= -2)) {

double fiEk3n1=-((pow(2,11/4)*gM*pow(k3n1,2)+pow(2,19/4)*gM*k3n1+pow(2,15/4)*gM)*sqrt(abs(k3n1))*pow(abs(2*k3n1+2),3/4))/((pow(k3n1,3)+4*pow(k3n1,2)+4*k3n1)*sqrt(abs(k3n1+1)));

double fiEk3d1=-((pow(2,11/4)*gM*pow(k3d1,2)+pow(2,19/4)*gM*k3d1+pow(2,15/4)*gM)*sqrt(abs(k3d1))*pow(abs(2*k3d1+2),3/4))/((pow(k3d1,3)+4*pow(k3d1,2)+4*k3d1)*sqrt(abs(k3d1+1)));

double fiEk3n2=((5*pow(2,3/4)*gM*pow(k3n2,3)+25*pow(2,3/4)*gM*pow(k3n2,2)+pow(2, 23/4)*gM*k3n2+3*pow(2,11/4)*gM)*pow(abs(k3n2),3/2)*pow(abs(2*k3n2+2),3/4))/((pow(k3n2,4)+4*pow(k3n2,3)+4*pow(k3n2,2))*pow(abs(k3n2+1),3/2));

double fiEk3d2=((5*pow(2,3/4)*gM*pow(k3d2,3)+25*pow(2,3/4)*gM*pow(k3d2,2)+pow(2,23 /4)*gM*k3d2+3*pow(2,11/4)*gM)*pow(abs(k3d2),3/2)*pow(abs(2*k3d2+2),3/4))/((pow(k3d2,4)+4*pow(k3d2,3)+4*pow(k3d2,2))*pow(abs(k3d2+1),3/2));

Sidn = fiEk3n1*dk3n + fiEk3n2*dk3n + Sidn;
 Sidd = fiEk3d1*dk3d + fiEk3d2*dk3d + Sidd;
k3n = k3n+dk3n;
k3d = k3d+dk3d;
if (t = = tmax or (Sidd/Sidn >= 3 and Sidd/Sidn < 4)) {
if (Sidd/Sidn >= 3 and Sidd/Sidn < 4) {
k3n = k3n-dk3n;
 k3d = k3d-dk3d;
```

```
double dEd = (pow(2,7/4)*gM*sqrt(abs(dk3d+1)))/(sqrt(abs(dk3d))*pow(abs(2*dk3d+2),1/4));
cout << "dk3n = " << dk3n << ", dk3d = " << dk3d << ", dEd = " << dEd << " GeV, Sidd
= " << Sidd << " GeV, Sidn = " << Sidn << " GeV" << endl;
dk3d = dk3d-d0/10;
}
else if (Sidd/Sidn < 3) {
 k3n = k3n-dk3n;
k3d = k3d-dk3d;
dk3d = dk3d+d0/10;
}
else if (Sidd/Sidn >= 4) {
 k3n = k3n-dk3n;
k3d = k3d-dk3d;
dk3d = dk3d+d0/10;
 }
}
else {
k3n = k3n-dk3n;
k3d = k3d-dk3d;
dk3d = dk3d+d0;
 }
          }
        }
      }
    }
  }
}

return 0;
}
```

→ ... .

Or

(3-1) Algorithm in Python

```
from math import*

#An algorithm giving sets of massive bosons-dark energy: {dE[k3] = 2^(7/4)*gM*sqrt((k3d+1)/(k 3d*sqrt(2*k3d+2))), k3d = 1+n.dk3d}

#and sets of massive bosons-dark matter: {dE[k3] = 2^(7/4)*gM*sqrt((k3n+1)/(k3n*sqrt(2*k3n + 2))), k3n = 1+n.dk3n}
gM = sqrt(1393.6328858707575005182839547884)
rangeX = 100
dk3n = 1.255314934818507*10**89
dk3d = 1.862840706286786*10**147
#[a+b = -1.862840706286786*10^147-rangeX*dk3n, a*rangeX+b = 1.862840706286786*10^14 7+rangeX*dk3n], [a,b]

a1=(2*dk3n*rangeX+372568141257357205379873223198839242392436654099307908354620113094015738619684208895970955587411058012917340751325107552823764843597602686734067302 4)/(rangeX-1)

b1=-(dk3n*rangeX**2+(dk3n+186284070628678602689936611599419621196218327049653954177310056547007869309842104447985477793705529006458670375662553776411882421798801343367033 6512)* rangeX+186284070628678602689936611599419621196218327049653954177310056547007869309842104447985477793705529 00 6458670375662553776411882421798801343367033 6512)/(rangeX-1)

#[a+b= -1.862840706286786*10^147-rangeX*dk3d, a*rangeX+b = 1.862840706286786*10^147 +rangeX*dk3d], [a,b]

a2=(2*dk3d*rangeX+3725681412573572053798732231988392423924366540993079083546201113094015738619684208895970955587411058012917340751325107552823764843597602686 7 340673024)/(rangeX-1)
```

```
b2=-(dk3d*rangeX**2+(dk3d+1862840706286786026899366115994196211962183270496539541773100565470078693098421044479854777937055290064586703756625537764118824217988013433670336512)*rangeX+1862840706286786026899366115994196211962183270496539541773100565470078693098421044479854777937055290064586703756625537764118824217988013433670336512)/(rangeX-1)

for q in range (1,300 + 1):
Sidn = 0
Sidd = 0
tmax = 10*10**q
d0 =1.862840706286786*10**147
for t0 in range(1,10000 + 1):

if d0 > 10**(-3)*1.862840706286786*10**147:
Sidn = 0
Sidd = 0
k3n = 1.862840706286786*10**147
 k3d = 1.862840706286786*10**147
for t in range(1, tmax + 1):
 for k3n1i in range(1, rangeX + 1):
 k3n1 = a1*k3n1i+b1
for k3n2i in range(1, rangeX + 1):
 k3n2 = a1*k3n2i+b1
for k3n3i in range(1, rangeX + 1):
 k3n3 = a1*k3n3i+b1
for k3n4i in range(1, rangeX + 1):
 k3n4 = a1*k3n4i+b1
k3n5 = k3n-(k3n1+k3n2+k3n3+k3n4)
for k3d1i in range(1, rangeX + 1):
 k3d1 = a2*k3d1i+b2
for k3d2i in range(1, rangeX + 1):
 k3d2 = a2*k3d2i+b2
for k3d3i in range(1, rangeX + 1):
 k3d3 = a2*k3d3i+b2
for k3d4i in range(1, rangeX + 1):
 k3d4 = a2*k3d4i+b2
k3d5 = k3d-(k3d1+k3d2+k3d3+k3d4)
#[k3!= -1, k3!= -2, k3!= 0]

if k3n1*k3n2*k3n3*k3n4*k3n5*k3d1*k3d2*k3d3*k3d4*k3d5!= 0 and (k3n1!= -1 or k3n2!= -1 or k3n3!= -1 or k3n4!= -1 or k3n5!= -1 or k3d1!= -1 or k3d2!= -1 or k3d3!= -1 or k3d4!= -1 or k3d5!= -1) and (k3n1!= -2 or k3n2!= -2 or k3n3!= -2 or k3n4!= -2 or k3n5!= -2 or k3d1!= -2 or k3d2!= -2 or k3d3!= -2 or k3d4!= -2 or k3d5!= -2):

fiEk3n1= -((2**(11/4)*gM*k3n1**3+5*2**(11/4)*gM*k3n1**2+3*2**(15/4)*gM*k3n1+2**(1 5/4)*gM)*abs(k3n1)**(3/2)*abs(2*k3n1+2)**(3/4))/((k3n1**4+4*k3n1**3+4*k3n1**2)*abs(k 3n1+1)**(3/2))

#imfiEk3n1 = 0
fiEk3n2= ((2**(27/4)*gM*k3n2+2**(27/4)*gM)*sqrt(abs(k3n2))*abs(2*k3n2+2)**(3/4))/((k3n2 **4+8*k3n2**3+24*k3n2**2+32*k3n2+ 16)*sqrt(abs(k3n2+1)))

#imfiEk3n2 = 0
fiEk3n3=-((5*2**(11/4)*gM*k3n3**5+45*2**(11/4)*gM*k3n3**4+71*2**(15/4)*gM*k3n3**3+95*2**(15/4)*gM*k3n3**2+7*2**(27/4)*gM*k3n3+3*2**(23/4)*gM)*abs(k3n3)**(3/2)*abs(2*k3n3+2)**(3/4))/((k3n3**6+8*k3n3**5+24*k3n3**4+32*k3n3**3+16*k3n3**2)*abs(k3n3+1)**(3/2))

 #imfiEk3n3 = 0
fiEk3n4=-((5*2**(11/4)*gM*k3n4**5+45*2**(11/4)*gM*k3n4**4+71*2**(15/4)*gM*k3n4**3+95*2**(15/4)*gM*k3n4**2+7*2**(27/4)*gM*k3n4+3*2**(23/4)*gM)*abs(k3n4)**(3/2)*abs(2*k3n4+2)**(3/4))/((k3n4**6+8*k3n4**5+24*k3n4**4+32*k3n4**3+16*k3n4**2)*abs(k3n4+1)**(3/2))
```

```
 #imfiEk3n4 = 0
fiEk3n5= (k3n5**2*(45*gM*k3n5**4+315*gM*k3n5**3+594*gM*k3n5**2+444*gM*k3n5+ 120*gM)*sqrt(abs(k3n5))*abs(2*k3n5+2)**(7/4))/((k3n5+1)**2*(2**(1/4)*k3n5**6+3*2**(5/4)*k3n5**5+3*2**(9/4)*k3n5**4+2**(13/4)*k3n5**3)*sqrt(abs(k3n5+1)))

#imfiEk3n5] = 0
fiEk3d1 = -((2**(11/4)*gM*k3d1**3+5*2**(11/4)*gM*k3d1**2+3*2**(15/4)*gM*k3d1+2**(1 5/4)*gM)*abs(k3d1)**(3/2)*abs(2*k3d1+2)**(3/4))/((k3d1**4+4*k3d1**3+4*k3d1**2)*abs(k3d1+1)**(3/2))

#imfiEk3d1 = 0
fiEk3d2 = ((2**(27/4)*gM*k3d2+2**(27/4)*gM)*sqrt(abs(k3d2))*abs(2*k3d2+2)**(3/4))/((k3d 2**4+8*k3d2**3+24*k3d2**2+32*k3d2+ 16)*sqrt(abs(k3d2+1)))
#imfiEk3d2 = 0
fiEk3d3= -((5*2**(11/4)*gM*k3d3**5+45*2**(11/4)*gM*k3d3**4+71*2**(15/4)*gM*k3d3** 3+95*2**(15/4)*gM*k-3d3**2+7*2**(27/4)*gM*k3d3+3*2**(23/4)*gM)*abs(k3d3)**(3/2)*abs(2*k3d3+2)**(3/4))/((k3d3**6+8*k3d3**5+24*k-3d3**4+32*k3d3**3+16*k3d3**2)*abs(k3d3+1)**(3/2))

 #imfiEk3d3 = 0
fiEk3d4= -((5*2**(11/4)*gM*k3d4**5+45*2**(11/4)*gM*k3d4**4+71*2**(15/4)*gM*k3d4** 3+95*2**(15/4)*gM*k-3d4**2+7*2**(27/4)*gM*k3d4+3*2**(23/4)*gM)*abs(k3d4)**(3/2)*abs(2*k3d4+2)**(3/4))/((k3d4**6+8*k3d4**5+24*k-3d4**4+32*k3d4**3+16*k3d4**2)*abs(k3d4+1)**(3/2))

#imfiEk3d4 = 0
fiEk3d5 = (k3d5**2*(45*gM*k3d5**4+315*gM*k3d5**3+594*gM*k3d5**2+444*gM*k3d5+1 20*gM)*sqrt(abs(k-3d5))*abs(2*k3d5+2)**(7/4))/((k3d5+1)**2*(2**(1/4)*k3d5**6+3*2**(5/4)*k3d5**5+3*2**(9/4)*k3d5**4+2**(13/4)*k-3d5**3)*sqrt(abs(k3d5+1)))

#imfiEk3d5] = 0

#fiEk3n = 2**(7/4)*sqrt((k3n+1)/(k3n*sqrt(2*k3n+2)))
#fiEk3d = 2**(7/4)*sqrt((k3d+1)/(k3d*sqrt(2*k3d+2)))
Sidn = fiEk3n1*dk3n + fiEk3n2*dk3n + fiEk3n3*dk3n + fiEk3n4*dk3n + fiEk3n5*dk3n + Sidn
Sidd = fiEk3d1*dk3d + fiEk3d2*dk3d + fiEk3d3*dk3d + fiEk3d4*dk3d + fiEk3d5*dk3d + Sidd
k3n = k3n+dk3n
k3d = k3d+dk3d
if t = = tmax or (Sidd/Sidn >= 3 and Sidd/Sidn < 4) :
if (Sidd/Sidn >= 3 and Sidd/Sidn < 4) :
 k3n = k3n-dk3n
k3d = k3d-dk3d
#imdEd = 0
dEd = (2**(7/4)*gM*sqrt(abs(dk3d+1)))/(sqrt(abs(dk3d))*abs(2*dk3d+2)**(1/4))
print("dk3n = ",dk3n, ", {dk3d = ",dk3d, ", dEd = ",dEd," GeV, Sidd = ",Sidd," GeV, Sidn = ",Sidn," GeV}")
dk3d = dk3d-d0
d0 = d0/10
elif Sidd/Sidn < 3 :
 k3n = k3n-dk3n
k3d = k3d-dk3d
dk3d = dk3d+d0
elif Sidd/Sidn >= 4 :
 k3n = k3n-dk3n
k3d = k3d-dk3d
dk3d : dk3d-d0/10
```

→ 10 random results obtained from this algorithm:

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 34.00167505713088 GeV, Sidn = 91947574.94277224 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd =

33.230939375949144 GeV, Sidn = 90728907.96697694 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 33.678777500551045 GeV, Sidn = 91325416.35519207 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 34.849266349418905 GeV, Sidn = 93330984.8136157 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 33.89461233274664 GeV, Sidn = 91824290.75378305 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 33.244760246712135 GeV, Sidn = 90750340.43227911 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 34.00659639485112 GeV, Sidn = 91956656.87543966 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 34.846724496494086 GeV, Sidn = 93324629.89333412 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 34.93276333543492 GeV, Sidn = 93587080.2677182 GeV}

dk3n = 1.255314934818507e+89, {dk3d = 1.862840706286786e+147, dEd = 2.206294095250117 e-70 GeV, Sidd = 33.881464159185175 GeV, Sidn = 91736474.225333 GeV}

1.4.1. From Algorithm

Example 1:

C++ :

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
 int N = 1;
 int t = 1000;
 double dP1[1000 + 1] = {0};
 double dP2[1000 + 1] = {0};
 double dE[1000 + 1] = {0};
 double S[1000 + 1] = {0};
 double gM = sqrt(1393.6328858707575005182839547884);

for (int k3 = -t; k3 <= t; k3++) {
 if (k3!= 0 && k3!= -1 && k3!= -2) {
 dP1[k3] = -((pow(2, (11.0 / 4.0)) * gM * pow(k3, 2) + pow(2, (19.0 / 4.0)) * gM * k3 + pow(2, (15.0/ 4.0)) * gM) * sqrt(abs(k3)) * pow(abs(2 * k3 + 2), (3.0 / 4.0))) / ((pow(k3, 3) + 4 * pow(k3, 2) + 4 * k3) * sqrt(abs(k3 + 1)));

 dP2[k3] = ((5 * pow(2, (3.0 / 4.0)) * gM * pow(k3, 3) + 25 * pow(2, (3.0 / 4.0)) * gM * pow(k3, 2) +pow(2, (23.0 / 4.0)) * gM * k3 + 3 * pow(2, (11.0 / 4.0)) * gM) * pow(abs(k3), (3.0 / 2.0)) * pow(abs(2 * k3 + 2), (3.0 / 4.0))) / ((pow(k3, 4) + 4 * pow(k3, 3) + 4 * pow(k3, 2)) * pow(abs(k3 + 1), (3.0 / 2.0)));

 dE[k3] = (pow(2, (7.0 / 4.0)) * gM * sqrt(abs(k3 + 1))) / (sqrt(abs(k3)) * pow(abs(2 * k3 + 2), (1.0 /4.0)));
}
if (k3 == t) {
 N = floor(dE[1] / dE[t]);
 }
}
```

```
for (int l1 = 1; l1 <= t - 1; l1++) {
 cout << l1 << " > (" << dP1[l1] << " and " << dP2[l1] << ") GeV" << endl;
}

int l1 = t - 1 + 1;
for (int l2 = N; l2 >= l1 - 1; l2--) {
 for (int q1 = 1; q1 <= N; q1++) {
 for (int q2 = 1; q2 <= N; q2++) {
 if ((dP1[l1] + dP2[l1]) - (q1 * dP1[l2] + q2 * dP2[l2]) > 0 && (dP1[l1] + dP2[l1]) - (q1 * dP1[l2] + q2 * dP2[l2]) < (dP1[l1] + dP2[l1]) - (dP1[l1 + 1] + dP2[l1 + 1])) {
 cout << l1 << " > ~ {" << q1 << "+" << q2 << "}." << l2 << " >" << endl;
 S[l1] = S[l1] + q1 + q2;
 }
else if (l2 == l1 - 1 && q1 == N && q2 == N) {
 cout << S[l1] << endl;
}
  }
 }
}
return 0;
}
```

→ 1 > (-390.654 and 516.221) GeV
2 > (-343.916 and 442.178) GeV
3 > (-317.264 and 403.478) GeV
4 > (-298.242 and 377.188) GeV
5 > (-283.553 and 357.457) GeV
6 > (-271.7 and 341.816) GeV
7 > (-261.845 and 328.964) GeV
8 > (-253.467 and 318.127) GeV
9 > (-246.218 and 308.807) GeV
10 > (-239.857 and 300.666) GeV
11 > (-234.211 and 293.465) GeV
12 > (-229.15 and 287.028) GeV
13 > (-224.575 and 281.223) GeV
14 > (-220.41 and 275.947) GeV
15 > (-216.595 and 271.121) GeV
16 > (-213.08 and 266.681) GeV
17 > (-209.826 and 262.575) GeV
18 > (-206.802 and 258.762) GeV
19 > (-203.979 and 255.206) GeV
20 > (-201.336 and 251.878) GeV
21 > (-198.852 and 248.754) GeV
22 > (-196.512 and 245.811) GeV
23 > (-194.301 and 243.033) GeV
24 > (-192.208 and 240.403) GeV
25 > (-190.221 and 237.908) GeV
26 > (-188.332 and 235.535) GeV
27 > (-186.532 and 233.276) GeV
28 > (-184.813 and 231.12) GeV
29 > (-183.171 and 229.059) GeV
30 > (-181.598 and 227.086) GeV
31 > (-180.089 and 225.194) GeV
32 > (-178.641 and 223.378) GeV
33 > (-177.248 and 221.633) GeV
34 > (-175.908 and 219.953) GeV
35 > (-174.617 and 218.335) GeV
36 > (-173.371 and 216.774) GeV

37 > (-172.168 and 215.267) GeV
38 > (-171.006 and 213.811) GeV
39 > (-169.881 and 212.402) GeV
40 > (-168.792 and 211.038) GeV
41 > (-167.737 and 209.717) GeV
42 > (-166.714 and 208.436) GeV
43 > (-165.722 and 207.193) GeV
44 > (-164.758 and 205.986) GeV
45 > (-163.821 and 204.813) GeV
46 > (-162.91 and 203.673) GeV
47 > (-162.024 and 202.563) GeV
48 > (-161.161 and 201.484) GeV
49 > (-160.321 and 200.432) GeV
50 > (-159.502 and 199.407) GeV
51 > (-158.703 and 198.407) GeV
52 > (-157.924 and 197.432) GeV
53 > (-157.164 and 196.481) GeV
54 > (-156.421 and 195.552) GeV
55 > (-155.696 and 194.644) GeV
56 > (-154.987 and 193.757) GeV
57 > (-154.294 and 192.89) GeV
58 > (-153.616 and 192.042) GeV
59 > (-152.953 and 191.212) GeV
60 > (-152.304 and 190.399) GeV
61 > (-151.668 and 189.604) GeV
62 > (-151.045 and 188.825) GeV
63 > (-150.435 and 188.061) GeV
64 > (-149.837 and 187.313) GeV
65 > (-149.25 and 186.579) GeV
66 > (-148.675 and 185.86) GeV
67 > (-148.111 and 185.154) GeV
68 > (-147.557 and 184.461) GeV
69 > (-147.013 and 183.781) GeV
70 > (-146.48 and 183.114) GeV
71 > (-145.955 and 182.458) GeV
72 > (-145.44 and 181.814) GeV
73 > (-144.934 and 181.181) GeV
74 > (-144.437 and 180.559) GeV
75 > (-143.948 and 179.947) GeV
76 > (-143.467 and 179.346) GeV
77 > (-142.994 and 178.754) GeV
78 > (-142.529 and 178.172) GeV
79 > (-142.071 and 177.6) GeV
80 > (-141.621 and 177.036) GeV
81 > (-141.177 and 176.482) GeV
82 > (-140.74 and 175.935) GeV
83 > (-140.31 and 175.398) GeV
84 > (-139.887 and 174.868) GeV
85 > (-139.47 and 174.346) GeV
86 > (-139.059 and 173.832) GeV
87 > (-138.653 and 173.326) GeV
88 > (-138.254 and 172.826) GeV
89 > (-137.861 and 172.334) GeV
90 > (-137.472 and 171.849) GeV
91 > (-137.09 and 171.37) GeV
92 > (-136.712 and 170.898) GeV
93 > (-136.34 and 170.433) GeV
94 > (-135.973 and 169.973) GeV

95 > (-135.61 and 169.52) GeV
96 > (-135.253 and 169.073) GeV
97 > (-134.9 and 168.632) GeV
98 > (-134.551 and 168.196) GeV
99 > (-134.207 and 167.766) GeV
100 > (-133.868 and 167.341) GeV
101 > (-133.532 and 166.922) GeV
102 > (-133.201 and 166.508) GeV
103 > (-132.874 and 166.099) GeV
104 > (-132.551 and 165.694) GeV
105 > (-132.232 and 165.295) GeV
106 > (-131.916 and 164.901) GeV
107 > (-131.604 and 164.511) GeV
108 > (-131.296 and 164.126) GeV
109 > (-130.992 and 163.745) GeV
110 > (-130.691 and 163.368) GeV
111 > (-130.393 and 162.996) GeV
112 > (-130.099 and 162.628) GeV
113 > (-129.808 and 162.265) GeV
114 > (-129.52 and 161.905) GeV
115 > (-129.235 and 161.549) GeV
116 > (-128.954 and 161.197) GeV
117 > (-128.675 and 160.849) GeV
118 > (-128.4 and 160.504) GeV
119 > (-128.127 and 160.164) GeV
120 > (-127.858 and 159.826) GeV
121 > (-127.591 and 159.493) GeV
122 > (-127.327 and 159.162) GeV
123 > (-127.065 and 158.836) GeV
124 > (-126.806 and 158.512) GeV
125 > (-126.55 and 158.192) GeV
126 > (-126.297 and 157.875) GeV
127 > (-126.046 and 157.561) GeV
128 > (-125.797 and 157.25) GeV
129 > (-125.551 and 156.942) GeV
130 > (-125.307 and 156.637) GeV
131 > (-125.066 and 156.336) GeV
132 > (-124.826 and 156.037) GeV
133 > (-124.59 and 155.74) GeV
134 > (-124.355 and 155.447) GeV
135 > (-124.123 and 155.157) GeV
136 > (-123.892 and 154.869) GeV
137 > (-123.664 and 154.583) GeV
138 > (-123.438 and 154.301) GeV
139 > (-123.214 and 154.021) GeV
140 > (-122.992 and 153.743) GeV
141 > (-122.772 and 153.468) GeV
142 > (-122.554 and 153.195) GeV
143 > (-122.338 and 152.925) GeV
144 > (-122.123 and 152.657) GeV
145 > (-121.911 and 152.392) GeV
146 > (-121.701 and 152.128) GeV
147 > (-121.492 and 151.867) GeV
148 > (-121.285 and 151.609) GeV
149 > (-121.08 and 151.352) GeV
150 > (-120.876 and 151.098) GeV
151 > (-120.674 and 150.845) GeV
152 > (-120.474 and 150.595) GeV

153 > (-120.276 and 150.347) GeV
154 > (-120.079 and 150.101) GeV
155 > (-119.883 and 149.857) GeV
156 > (-119.69 and 149.615) GeV
157 > (-119.498 and 149.374) GeV
158 > (-119.307 and 149.136) GeV
159 > (-119.118 and 148.9) GeV
160 > (-118.93 and 148.665) GeV
161 > (-118.744 and 148.432) GeV
162 > (-118.559 and 148.202) GeV
163 > (-118.376 and 147.972) GeV
164 > (-118.194 and 147.745) GeV
165 > (-118.014 and 147.519) GeV
166 > (-117.835 and 147.296) GeV
167 > (-117.657 and 147.073) GeV
168 > (-117.481 and 146.853) GeV
169 > (-117.305 and 146.634) GeV
170 > (-117.132 and 146.417) GeV
171 > (-116.959 and 146.201) GeV
172 > (-116.788 and 145.987) GeV
173 > (-116.618 and 145.774) GeV
174 > (-116.449 and 145.563) GeV
175 > (-116.281 and 145.354) GeV
176 > (-116.115 and 145.146) GeV
177 > (-115.95 and 144.939) GeV
178 > (-115.786 and 144.734) GeV
179 > (-115.623 and 144.531) GeV
180 > (-115.461 and 144.328) GeV
181 > (-115.301 and 144.128) GeV
182 > (-115.141 and 143.928) GeV
183 > (-114.983 and 143.73) GeV
184 > (-114.825 and 143.533) GeV
185 > (-114.669 and 143.338) GeV
186 > (-114.514 and 143.144) GeV
187 > (-114.36 and 142.951) GeV
188 > (-114.207 and 142.76) GeV
189 > (-114.055 and 142.57) GeV
190 > (-113.904 and 142.381) GeV
191 > (-113.753 and 142.193) GeV
192 > (-113.604 and 142.007) GeV
193 > (-113.456 and 141.822) GeV
194 > (-113.309 and 141.638) GeV
195 > (-113.163 and 141.455) GeV
196 > (-113.017 and 141.273) GeV
197 > (-112.873 and 141.093) GeV
198 > (-112.73 and 140.913) GeV
199 > (-112.587 and 140.735) GeV
200 > (-112.445 and 140.558) GeV
201 > (-112.305 and 140.382) GeV
202 > (-112.165 and 140.207) GeV
203 > (-112.026 and 140.034) GeV
204 > (-111.888 and 139.861) GeV
205 > (-111.75 and 139.689) GeV
206 > (-111.614 and 139.519) GeV
207 > (-111.478 and 139.349) GeV
208 > (-111.343 and 139.18) GeV
209 > (-111.209 and 139.013) GeV
210 > (-111.076 and 138.846) GeV

211 > (-110.944 and 138.681) GeV
212 > (-110.812 and 138.516) GeV
213 > (-110.681 and 138.353) GeV
214 > (-110.551 and 138.19) GeV
215 > (-110.422 and 138.028) GeV
216 > (-110.293 and 137.868) GeV
217 > (-110.165 and 137.708) GeV
218 > (-110.038 and 137.549) GeV
219 > (-109.912 and 137.391) GeV
220 > (-109.786 and 137.234) GeV
221 > (-109.661 and 137.078) GeV
222 > (-109.537 and 136.923) GeV
223 > (-109.414 and 136.768) GeV
224 > (-109.291 and 136.615) GeV
225 > (-109.169 and 136.462) GeV
226 > (-109.047 and 136.31) GeV
227 > (-108.926 and 136.159) GeV
228 > (-108.806 and 136.009) GeV
229 > (-108.687 and 135.86) GeV
230 > (-108.568 and 135.711) GeV
231 > (-108.45 and 135.563) GeV
232 > (-108.332 and 135.416) GeV
233 > (-108.215 and 135.27) GeV
234 > (-108.099 and 135.125) GeV
235 > (-107.984 and 134.98) GeV
236 > (-107.869 and 134.837) GeV
237 > (-107.754 and 134.694) GeV
238 > (-107.64 and 134.551) GeV
239 > (-107.527 and 134.41) GeV
240 > (-107.415 and 134.269) GeV
241 > (-107.302 and 134.129) GeV
242 > (-107.191 and 133.99) GeV
243 > (-107.08 and 133.851) GeV
244 > (-106.97 and 133.713) GeV
245 > (-106.86 and 133.576) GeV
246 > (-106.751 and 133.44) GeV
247 > (-106.642 and 133.304) GeV
248 > (-106.534 and 133.169) GeV
249 > (-106.427 and 133.034) GeV
250 > (-106.32 and 132.901) GeV
251 > (-106.213 and 132.767) GeV
252 > (-106.107 and 132.635) GeV
253 > (-106.002 and 132.503) GeV
254 > (-105.897 and 132.372) GeV
255 > (-105.793 and 132.242) GeV
256 > (-105.689 and 132.112) GeV
257 > (-105.586 and 131.983) GeV
258 > (-105.483 and 131.854) GeV
259 > (-105.38 and 131.726) GeV
260 > (-105.279 and 131.599) GeV
261 > (-105.177 and 131.472) GeV
262 > (-105.076 and 131.346) GeV
263 > (-104.976 and 131.221) GeV
264 > (-104.876 and 131.096) GeV
265 > (-104.777 and 130.972) GeV
266 > (-104.678 and 130.848) GeV
267 > (-104.579 and 130.725) GeV
268 > (-104.481 and 130.602) GeV

269 > (-104.384 and 130.48) GeV
270 > (-104.286 and 130.359) GeV
271 > (-104.19 and 130.238) GeV
272 > (-104.094 and 130.118) GeV
273 > (-103.998 and 129.998) GeV
274 > (-103.902 and 129.879) GeV
275 > (-103.808 and 129.76) GeV
276 > (-103.713 and 129.642) GeV
277 > (-103.619 and 129.524) GeV
278 > (-103.525 and 129.407) GeV
279 > (-103.432 and 129.291) GeV
280 > (-103.339 and 129.175) GeV
281 > (-103.247 and 129.059) GeV
282 > (-103.155 and 128.944) GeV
283 > (-103.063 and 128.83) GeV
284 > (-102.972 and 128.716) GeV
285 > (-102.882 and 128.603) GeV
286 > (-102.791 and 128.49) GeV
287 > (-102.701 and 128.377) GeV
288 > (-102.612 and 128.265) GeV
289 > (-102.523 and 128.154) GeV
290 > (-102.434 and 128.043) GeV
291 > (-102.345 and 127.932) GeV
292 > (-102.257 and 127.822) GeV
293 > (-102.17 and 127.713) GeV
294 > (-102.082 and 127.604) GeV
295 > (-101.996 and 127.495) GeV
296 > (-101.909 and 127.387) GeV
297 > (-101.823 and 127.279) GeV
298 > (-101.737 and 127.172) GeV
299 > (-101.652 and 127.065) GeV
300 > (-101.567 and 126.959) GeV
301 > (-101.482 and 126.853) GeV
302 > (-101.397 and 126.747) GeV
303 > (-101.313 and 126.642) GeV
304 > (-101.23 and 126.538) GeV
305 > (-101.146 and 126.434) GeV
306 > (-101.063 and 126.33) GeV
307 > (-100.981 and 126.226) GeV
308 > (-100.898 and 126.124) GeV
309 > (-100.816 and 126.021) GeV
310 > (-100.735 and 125.919) GeV
311 > (-100.653 and 125.817) GeV
312 > (-100.572 and 125.716) GeV
313 > (-100.492 and 125.615) GeV
314 > (-100.411 and 125.515) GeV
315 > (-100.331 and 125.415) GeV
316 > (-100.252 and 125.315) GeV
317 > (-100.172 and 125.216) GeV
318 > (-100.093 and 125.117) GeV
319 > (-100.014 and 125.019) GeV
320 > (-99.936 and 124.921) GeV
321 > (-99.8579 and 124.823) GeV
322 > (-99.78 and 124.726) GeV
323 > (-99.7025 and 124.629) GeV
324 > (-99.6252 and 124.532) GeV
325 > (-99.5483 and 124.436) GeV
326 > (-99.4716 and 124.34) GeV

327 > (-99.3953 and 124.245) GeV
328 > (-99.3192 and 124.149) GeV
329 > (-99.2434 and 124.055) GeV
330 > (-99.1679 and 123.96) GeV
331 > (-99.0927 and 123.866) GeV
332 > (-99.0178 and 123.773) GeV
333 > (-98.9432 and 123.679) GeV
334 > (-98.8688 and 123.586) GeV
335 > (-98.7948 and 123.494) GeV
336 > (-98.721 and 123.402) GeV
337 > (-98.6474 and 123.31) GeV
338 > (-98.5742 and 123.218) GeV
339 > (-98.5012 and 123.127) GeV
340 > (-98.4285 and 123.036) GeV
341 > (-98.3561 and 122.945) GeV
342 > (-98.2839 and 122.855) GeV
343 > (-98.212 and 122.765) GeV
344 > (-98.1403 and 122.676) GeV
345 > (-98.0689 and 122.587) GeV
346 > (-97.9978 and 122.498) GeV
347 > (-97.9269 and 122.409) GeV
348 > (-97.8563 and 122.321) GeV
349 > (-97.7859 and 122.233) GeV
350 > (-97.7158 and 122.145) GeV
351 > (-97.646 and 122.058) GeV
352 > (-97.5764 and 121.971) GeV
353 > (-97.507 and 121.884) GeV
354 > (-97.4379 and 121.798) GeV
355 > (-97.369 and 121.712) GeV
356 > (-97.3004 and 121.626) GeV
357 > (-97.232 and 121.54) GeV
358 > (-97.1638 and 121.455) GeV
359 > (-97.0959 and 121.37) GeV
360 > (-97.0282 and 121.286) GeV
361 > (-96.9608 and 121.201) GeV
362 > (-96.8936 and 121.117) GeV
363 > (-96.8266 and 121.034) GeV
364 > (-96.7599 and 120.95) GeV
365 > (-96.6933 and 120.867) GeV
366 > (-96.6271 and 120.784) GeV
367 > (-96.561 and 120.702) GeV
368 > (-96.4952 and 120.619) GeV
369 > (-96.4295 and 120.537) GeV
370 > (-96.3642 and 120.456) GeV
371 > (-96.299 and 120.374) GeV
372 > (-96.234 and 120.293) GeV
373 > (-96.1693 and 120.212) GeV
374 > (-96.1048 and 120.131) GeV
375 > (-96.0405 and 120.051) GeV
376 > (-95.9764 and 119.971) GeV
377 > (-95.9125 and 119.891) GeV
378 > (-95.8489 and 119.811) GeV
379 > (-95.7854 and 119.732) GeV
380 > (-95.7222 and 119.653) GeV
381 > (-95.6592 and 119.574) GeV
382 > (-95.5963 and 119.496) GeV
383 > (-95.5337 and 119.417) GeV
384 > (-95.4713 and 119.339) GeV

385 > (-95.4091 and 119.262) GeV
386 > (-95.3471 and 119.184) GeV
387 > (-95.2853 and 119.107) GeV
388 > (-95.2237 and 119.03) GeV
389 > (-95.1623 and 118.953) GeV
390 > (-95.1011 and 118.877) GeV
391 > (-95.0401 and 118.8) GeV
392 > (-94.9792 and 118.724) GeV
393 > (-94.9186 and 118.649) GeV
394 > (-94.8582 and 118.573) GeV
395 > (-94.7979 and 118.498) GeV
396 > (-94.7379 and 118.423) GeV
397 > (-94.678 and 118.348) GeV
398 > (-94.6184 and 118.273) GeV
399 > (-94.5589 and 118.199) GeV
400 > (-94.4996 and 118.125) GeV
401 > (-94.4405 and 118.051) GeV
402 > (-94.3815 and 117.977) GeV
403 > (-94.3228 and 117.904) GeV
404 > (-94.2642 and 117.831) GeV
405 > (-94.2059 and 117.758) GeV
406 > (-94.1477 and 117.685) GeV
407 > (-94.0896 and 117.612) GeV
408 > (-94.0318 and 117.54) GeV
409 > (-93.9741 and 117.468) GeV
410 > (-93.9166 and 117.396) GeV
411 > (-93.8593 and 117.324) GeV
412 > (-93.8022 and 117.253) GeV
413 > (-93.7452 and 117.182) GeV
414 > (-93.6884 and 117.111) GeV
415 > (-93.6318 and 117.04) GeV
416 > (-93.5754 and 116.969) GeV
417 > (-93.5191 and 116.899) GeV
418 > (-93.463 and 116.829) GeV
419 > (-93.407 and 116.759) GeV
420 > (-93.3513 and 116.689) GeV
421 > (-93.2956 and 116.62) GeV
422 > (-93.2402 and 116.551) GeV
423 > (-93.1849 and 116.481) GeV
424 > (-93.1298 and 116.413) GeV
425 > (-93.0749 and 116.344) GeV
426 > (-93.0201 and 116.275) GeV
427 > (-92.9654 and 116.207) GeV
428 > (-92.911 and 116.139) GeV
429 > (-92.8566 and 116.071) GeV
430 > (-92.8025 and 116.003) GeV
431 > (-92.7485 and 115.936) GeV
432 > (-92.6946 and 115.869) GeV
433 > (-92.641 and 115.801) GeV
434 > (-92.5874 and 115.735) GeV
435 > (-92.5341 and 115.668) GeV
436 > (-92.4808 and 115.601) GeV
437 > (-92.4278 and 115.535) GeV
438 > (-92.3749 and 115.469) GeV
439 > (-92.3221 and 115.403) GeV
440 > (-92.2695 and 115.337) GeV
441 > (-92.217 and 115.271) GeV
442 > (-92.1647 and 115.206) GeV

443 > (-92.1125 and 115.141) GeV
444 > (-92.0605 and 115.076) GeV
445 > (-92.0086 and 115.011) GeV
446 > (-91.9569 and 114.946) GeV
447 > (-91.9053 and 114.882) GeV
448 > (-91.8539 and 114.818) GeV
449 > (-91.8026 and 114.753) GeV
450 > (-91.7514 and 114.69) GeV
451 > (-91.7004 and 114.626) GeV
452 > (-91.6495 and 114.562) GeV
453 > (-91.5988 and 114.499) GeV
454 > (-91.5482 and 114.436) GeV
455 > (-91.4978 and 114.372) GeV
456 > (-91.4475 and 114.31) GeV
457 > (-91.3973 and 114.247) GeV
458 > (-91.3473 and 114.184) GeV
459 > (-91.2974 and 114.122) GeV
460 > (-91.2476 and 114.06) GeV
461 > (-91.198 and 113.998) GeV
462 > (-91.1485 and 113.936) GeV
463 > (-91.0991 and 113.874) GeV
464 > (-91.0499 and 113.813) GeV
465 > (-91.0008 and 113.751) GeV
466 > (-90.9518 and 113.69) GeV
467 > (-90.903 and 113.629) GeV
468 > (-90.8543 and 113.568) GeV
469 > (-90.8057 and 113.507) GeV
470 > (-90.7573 and 113.447) GeV
471 > (-90.709 and 113.386) GeV
472 > (-90.6608 and 113.326) GeV
473 > (-90.6128 and 113.266) GeV
474 > (-90.5648 and 113.206) GeV
475 > (-90.517 and 113.146) GeV
476 > (-90.4694 and 113.087) GeV
477 > (-90.4218 and 113.027) GeV
478 > (-90.3744 and 112.968) GeV
479 > (-90.3271 and 112.909) GeV
480 > (-90.2799 and 112.85) GeV
481 > (-90.2329 and 112.791) GeV
482 > (-90.1859 and 112.733) GeV
483 > (-90.1391 and 112.674) GeV
484 > (-90.0924 and 112.616) GeV
485 > (-90.0459 and 112.558) GeV
486 > (-89.9994 and 112.499) GeV
487 > (-89.9531 and 112.442) GeV
488 > (-89.9069 and 112.384) GeV
489 > (-89.8608 and 112.326) GeV
490 > (-89.8148 and 112.269) GeV
491 > (-89.7689 and 112.211) GeV
492 > (-89.7232 and 112.154) GeV
493 > (-89.6776 and 112.097) GeV
494 > (-89.6321 and 112.04) GeV
495 > (-89.5867 and 111.984) GeV
496 > (-89.5414 and 111.927) GeV
497 > (-89.4962 and 111.87) GeV
498 > (-89.4512 and 111.814) GeV
499 > (-89.4063 and 111.758) GeV
500 > (-89.3614 and 111.702) GeV

501 > (-89.3167 and 111.646) GeV
502 > (-89.2721 and 111.59) GeV
503 > (-89.2276 and 111.535) GeV
504 > (-89.1833 and 111.479) GeV
505 > (-89.139 and 111.424) GeV
506 > (-89.0948 and 111.369) GeV
507 > (-89.0508 and 111.314) GeV
508 > (-89.0068 and 111.259) GeV
509 > (-88.963 and 111.204) GeV
510 > (-88.9193 and 111.149) GeV
511 > (-88.8757 and 111.095) GeV
512 > (-88.8322 and 111.04) GeV
513 > (-88.7888 and 110.986) GeV
514 > (-88.7455 and 110.932) GeV
515 > (-88.7023 and 110.878) GeV
516 > (-88.6592 and 110.824) GeV
517 > (-88.6162 and 110.77) GeV
518 > (-88.5733 and 110.717) GeV
519 > (-88.5305 and 110.663) GeV
520 > (-88.4879 and 110.61) GeV
521 > (-88.4453 and 110.557) GeV
522 > (-88.4028 and 110.504) GeV
523 > (-88.3605 and 110.451) GeV
524 > (-88.3182 and 110.398) GeV
525 > (-88.276 and 110.345) GeV
526 > (-88.234 and 110.293) GeV
527 > (-88.192 and 110.24) GeV
528 > (-88.1501 and 110.188) GeV
529 > (-88.1084 and 110.136) GeV
530 > (-88.0667 and 110.084) GeV
531 > (-88.0251 and 110.032) GeV
532 > (-87.9837 and 109.98) GeV
533 > (-87.9423 and 109.928) GeV
534 > (-87.901 and 109.876) GeV
535 > (-87.8599 and 109.825) GeV
536 > (-87.8188 and 109.774) GeV
537 > (-87.7778 and 109.722) GeV
538 > (-87.7369 and 109.671) GeV
539 > (-87.6961 and 109.62) GeV
540 > (-87.6554 and 109.569) GeV
541 > (-87.6148 and 109.519) GeV
542 > (-87.5743 and 109.468) GeV
543 > (-87.5339 and 109.417) GeV
544 > (-87.4935 and 109.367) GeV
545 > (-87.4533 and 109.317) GeV
546 > (-87.4132 and 109.267) GeV
547 > (-87.3731 and 109.217) GeV
548 > (-87.3331 and 109.167) GeV
549 > (-87.2933 and 109.117) GeV
550 > (-87.2535 and 109.067) GeV
551 > (-87.2138 and 109.017) GeV
552 > (-87.1742 and 108.968) GeV
553 > (-87.1347 and 108.919) GeV
554 > (-87.0953 and 108.869) GeV
555 > (-87.056 and 108.82) GeV
556 > (-87.0167 and 108.771) GeV
557 > (-86.9776 and 108.722) GeV
558 > (-86.9385 and 108.673) GeV

559 > (-86.8995 and 108.625) GeV
560 > (-86.8607 and 108.576) GeV
561 > (-86.8219 and 108.527) GeV
562 > (-86.7831 and 108.479) GeV
563 > (-86.7445 and 108.431) GeV
564 > (-86.706 and 108.383) GeV
565 > (-86.6675 and 108.335) GeV
566 > (-86.6291 and 108.287) GeV
567 > (-86.5909 and 108.239) GeV
568 > (-86.5527 and 108.191) GeV
569 > (-86.5145 and 108.143) GeV
570 > (-86.4765 and 108.096) GeV
571 > (-86.4386 and 108.048) GeV
572 > (-86.4007 and 108.001) GeV
573 > (-86.3629 and 107.954) GeV
574 > (-86.3252 and 107.907) GeV
575 > (-86.2876 and 107.86) GeV
576 > (-86.25 and 107.813) GeV
577 > (-86.2126 and 107.766) GeV
578 > (-86.1752 and 107.719) GeV
579 > (-86.1379 and 107.673) GeV
580 > (-86.1007 and 107.626) GeV
581 > (-86.0636 and 107.58) GeV
582 > (-86.0265 and 107.533) GeV
583 > (-85.9895 and 107.487) GeV
584 > (-85.9526 and 107.441) GeV
585 > (-85.9158 and 107.395) GeV
586 > (-85.8791 and 107.349) GeV
587 > (-85.8424 and 107.303) GeV
588 > (-85.8058 and 107.257) GeV
589 > (-85.7693 and 107.212) GeV
590 > (-85.7329 and 107.166) GeV
591 > (-85.6966 and 107.121) GeV
592 > (-85.6603 and 107.075) GeV
593 > (-85.6241 and 107.03) GeV
594 > (-85.588 and 106.985) GeV
595 > (-85.5519 and 106.94) GeV
596 > (-85.516 and 106.895) GeV
597 > (-85.4801 and 106.85) GeV
598 > (-85.4443 and 106.805) GeV
599 > (-85.4085 and 106.761) GeV
600 > (-85.3728 and 106.716) GeV
601 > (-85.3373 and 106.672) GeV
602 > (-85.3017 and 106.627) GeV
603 > (-85.2663 and 106.583) GeV
604 > (-85.2309 and 106.539) GeV
605 > (-85.1956 and 106.495) GeV
606 > (-85.1604 and 106.451) GeV
607 > (-85.1252 and 106.407) GeV
608 > (-85.0902 and 106.363) GeV
609 > (-85.0552 and 106.319) GeV
610 > (-85.0202 and 106.275) GeV
611 > (-84.9854 and 106.232) GeV
612 > (-84.9506 and 106.188) GeV
613 > (-84.9158 and 106.145) GeV
614 > (-84.8812 and 106.102) GeV
615 > (-84.8466 and 106.058) GeV
616 > (-84.8121 and 106.015) GeV

617 > (-84.7777 and 105.972) GeV
618 > (-84.7433 and 105.929) GeV
619 > (-84.709 and 105.886) GeV
620 > (-84.6748 and 105.844) GeV
621 > (-84.6406 and 105.801) GeV
622 > (-84.6065 and 105.758) GeV
623 > (-84.5725 and 105.716) GeV
624 > (-84.5385 and 105.673) GeV
625 > (-84.5046 and 105.631) GeV
626 > (-84.4708 and 105.589) GeV
627 > (-84.4371 and 105.546) GeV
628 > (-84.4034 and 105.504) GeV
629 > (-84.3698 and 105.462) GeV
630 > (-84.3362 and 105.42) GeV
631 > (-84.3027 and 105.379) GeV
632 > (-84.2693 and 105.337) GeV
633 > (-84.236 and 105.295) GeV
634 > (-84.2027 and 105.253) GeV
635 > (-84.1694 and 105.212) GeV
636 > (-84.1363 and 105.17) GeV
637 > (-84.1032 and 105.129) GeV
638 > (-84.0702 and 105.088) GeV
639 > (-84.0372 and 105.047) GeV
640 > (-84.0043 and 105.005) GeV
641 > (-83.9715 and 104.964) GeV
642 > (-83.9387 and 104.923) GeV
643 > (-83.906 and 104.883) GeV
644 > (-83.8734 and 104.842) GeV
645 > (-83.8408 and 104.801) GeV
646 > (-83.8083 and 104.76) GeV
647 > (-83.7758 and 104.72) GeV
648 > (-83.7434 and 104.679) GeV
649 > (-83.7111 and 104.639) GeV
650 > (-83.6789 and 104.599) GeV
651 > (-83.6466 and 104.558) GeV
652 > (-83.6145 and 104.518) GeV
653 > (-83.5824 and 104.478) GeV
654 > (-83.5504 and 104.438) GeV
655 > (-83.5185 and 104.398) GeV
656 > (-83.4866 and 104.358) GeV
657 > (-83.4547 and 104.319) GeV
658 > (-83.423 and 104.279) GeV
659 > (-83.3912 and 104.239) GeV
660 > (-83.3596 and 104.2) GeV
661 > (-83.328 and 104.16) GeV
662 > (-83.2965 and 104.121) GeV
663 > (-83.265 and 104.081) GeV
664 > (-83.2336 and 104.042) GeV
665 > (-83.2022 and 104.003) GeV
666 > (-83.1709 and 103.964) GeV
667 > (-83.1397 and 103.925) GeV
668 > (-83.1085 and 103.886) GeV
669 > (-83.0774 and 103.847) GeV
670 > (-83.0463 and 103.808) GeV
671 > (-83.0153 and 103.769) GeV
672 > (-82.9844 and 103.731) GeV
673 > (-82.9535 and 103.692) GeV
674 > (-82.9227 and 103.653) GeV

675 > (-82.8919 and 103.615) GeV
676 > (-82.8612 and 103.577) GeV
677 > (-82.8305 and 103.538) GeV
678 > (-82.7999 and 103.5) GeV
679 > (-82.7694 and 103.462) GeV
680 > (-82.7389 and 103.424) GeV
681 > (-82.7084 and 103.386) GeV
682 > (-82.6781 and 103.348) GeV
683 > (-82.6477 and 103.31) GeV
684 > (-82.6175 and 103.272) GeV
685 > (-82.5873 and 103.234) GeV
686 > (-82.5571 and 103.196) GeV
687 > (-82.527 and 103.159) GeV
688 > (-82.497 and 103.121) GeV
689 > (-82.467 and 103.084) GeV
690 > (-82.437 and 103.046) GeV
691 > (-82.4071 and 103.009) GeV
692 > (-82.3773 and 102.972) GeV
693 > (-82.3475 and 102.935) GeV
694 > (-82.3178 and 102.897) GeV
695 > (-82.2881 and 102.86) GeV
696 > (-82.2585 and 102.823) GeV
697 > (-82.229 and 102.786) GeV
698 > (-82.1995 and 102.749) GeV
699 > (-82.17 and 102.713) GeV
700 > (-82.1406 and 102.676) GeV
701 > (-82.1113 and 102.639) GeV
702 > (-82.082 and 102.603) GeV
703 > (-82.0527 and 102.566) GeV
704 > (-82.0235 and 102.529) GeV
705 > (-81.9944 and 102.493) GeV
706 > (-81.9653 and 102.457) GeV
707 > (-81.9362 and 102.42) GeV
708 > (-81.9073 and 102.384) GeV
709 > (-81.8783 and 102.348) GeV
710 > (-81.8494 and 102.312) GeV
711 > (-81.8206 and 102.276) GeV
712 > (-81.7918 and 102.24) GeV
713 > (-81.7631 and 102.204) GeV
714 > (-81.7344 and 102.168) GeV
715 > (-81.7058 and 102.132) GeV
716 > (-81.6772 and 102.097) GeV
717 > (-81.6487 and 102.061) GeV
718 > (-81.6202 and 102.025) GeV
719 > (-81.5917 and 101.99) GeV
720 > (-81.5634 and 101.954) GeV
721 > (-81.535 and 101.919) GeV
722 > (-81.5067 and 101.883) GeV
723 > (-81.4785 and 101.848) GeV
724 > (-81.4503 and 101.813) GeV
725 > (-81.4222 and 101.778) GeV
726 > (-81.3941 and 101.743) GeV
727 > (-81.366 and 101.708) GeV
728 > (-81.338 and 101.673) GeV
729 > (-81.3101 and 101.638) GeV
730 > (-81.2822 and 101.603) GeV
731 > (-81.2544 and 101.568) GeV
732 > (-81.2266 and 101.533) GeV

733 > (-81.1988 and 101.499) GeV
734 > (-81.1711 and 101.464) GeV
735 > (-81.1434 and 101.429) GeV
736 > (-81.1158 and 101.395) GeV
737 > (-81.0883 and 101.36) GeV
738 > (-81.0607 and 101.326) GeV
739 > (-81.0333 and 101.292) GeV
740 > (-81.0058 and 101.257) GeV
741 > (-80.9785 and 101.223) GeV
742 > (-80.9511 and 101.189) GeV
743 > (-80.9238 and 101.155) GeV
744 > (-80.8966 and 101.121) GeV
745 > (-80.8694 and 101.087) GeV
746 > (-80.8422 and 101.053) GeV
747 > (-80.8151 and 101.019) GeV
748 > (-80.7881 and 100.985) GeV
749 > (-80.7611 and 100.951) GeV
750 > (-80.7341 and 100.918) GeV
751 > (-80.7072 and 100.884) GeV
752 > (-80.6803 and 100.85) GeV
753 > (-80.6535 and 100.817) GeV
754 > (-80.6267 and 100.783) GeV
755 > (-80.5999 and 100.75) GeV
756 > (-80.5732 and 100.717) GeV
757 > (-80.5466 and 100.683) GeV
758 > (-80.52 and 100.65) GeV
759 > (-80.4934 and 100.617) GeV
760 > (-80.4669 and 100.584) GeV
761 > (-80.4404 and 100.551) GeV
762 > (-80.4139 and 100.517) GeV
763 > (-80.3875 and 100.484) GeV
764 > (-80.3612 and 100.452) GeV
765 > (-80.3349 and 100.419) GeV
766 > (-80.3086 and 100.386) GeV
767 > (-80.2824 and 100.353) GeV
768 > (-80.2562 and 100.32) GeV
769 > (-80.2301 and 100.288) GeV
770 > (-80.204 and 100.255) GeV
771 > (-80.1779 and 100.222) GeV
772 > (-80.1519 and 100.19) GeV
773 > (-80.126 and 100.158) GeV
774 > (-80.1 and 100.125) GeV
775 > (-80.0741 and 100.093) GeV
776 > (-80.0483 and 100.06) GeV
777 > (-80.0225 and 100.028) GeV
778 > (-79.9967 and 99.996) GeV
779 > (-79.971 and 99.9638) GeV
780 > (-79.9454 and 99.9318) GeV
781 > (-79.9197 and 99.8997) GeV
782 > (-79.8941 and 99.8677) GeV
783 > (-79.8686 and 99.8358) GeV
784 > (-79.8431 and 99.8039) GeV
785 > (-79.8176 and 99.772) GeV
786 > (-79.7922 and 99.7403) GeV
787 > (-79.7668 and 99.7085) GeV
788 > (-79.7414 and 99.6768) GeV
789 > (-79.7161 and 99.6452) GeV
790 > (-79.6908 and 99.6136) GeV

791 > (-79.6656 and 99.5821) GeV
792 > (-79.6404 and 99.5506) GeV
793 > (-79.6153 and 99.5191) GeV
794 > (-79.5902 and 99.4878) GeV
795 > (-79.5651 and 99.4564) GeV
796 > (-79.5401 and 99.4251) GeV
797 > (-79.5151 and 99.3939) GeV
798 > (-79.4901 and 99.3627) GeV
799 > (-79.4652 and 99.3316) GeV
800 > (-79.4403 and 99.3005) GeV
801 > (-79.4155 and 99.2694) GeV
802 > (-79.3907 and 99.2384) GeV
803 > (-79.3659 and 99.2075) GeV
804 > (-79.3412 and 99.1766) GeV
805 > (-79.3165 and 99.1457) GeV
806 > (-79.2919 and 99.1149) GeV
807 > (-79.2673 and 99.0842) GeV
808 > (-79.2427 and 99.0534) GeV
809 > (-79.2182 and 99.0228) GeV
810 > (-79.1937 and 98.9922) GeV
811 > (-79.1692 and 98.9616) GeV
812 > (-79.1448 and 98.9311) GeV
813 > (-79.1204 and 98.9006) GeV
814 > (-79.0961 and 98.8702) GeV
815 > (-79.0718 and 98.8398) GeV
816 > (-79.0475 and 98.8095) GeV
817 > (-79.0233 and 98.7792) GeV
818 > (-78.9991 and 98.7489) GeV
819 > (-78.975 and 98.7188) GeV
820 > (-78.9508 and 98.6886) GeV
821 > (-78.9268 and 98.6585) GeV
822 > (-78.9027 and 98.6285) GeV
823 > (-78.8787 and 98.5984) GeV
824 > (-78.8547 and 98.5685) GeV
825 > (-78.8308 and 98.5386) GeV
826 > (-78.8069 and 98.5087) GeV
827 > (-78.783 and 98.4789) GeV
828 > (-78.7592 and 98.4491) GeV
829 > (-78.7354 and 98.4193) GeV
830 > (-78.7117 and 98.3896) GeV
831 > (-78.688 and 98.36) GeV
832 > (-78.6643 and 98.3304) GeV
833 > (-78.6406 and 98.3008) GeV
834 > (-78.617 and 98.2713) GeV
835 > (-78.5934 and 98.2418) GeV
836 > (-78.5699 and 98.2124) GeV
837 > (-78.5464 and 98.183) GeV
838 > (-78.5229 and 98.1537) GeV
839 > (-78.4995 and 98.1244) GeV
840 > (-78.4761 and 98.0952) GeV
841 > (-78.4527 and 98.0659) GeV
842 > (-78.4294 and 98.0368) GeV
843 > (-78.4061 and 98.0077) GeV
844 > (-78.3828 and 97.9786) GeV
845 > (-78.3596 and 97.9496) GeV
846 > (-78.3364 and 97.9206) GeV
847 > (-78.3132 and 97.8916) GeV
848 > (-78.2901 and 97.8627) GeV

849 > (-78.267 and 97.8338) GeV
850 > (-78.244 and 97.805) GeV
851 > (-78.221 and 97.7762) GeV
852 > (-78.198 and 97.7475) GeV
853 > (-78.175 and 97.7188) GeV
854 > (-78.1521 and 97.6902) GeV
855 > (-78.1292 and 97.6616) GeV
856 > (-78.1063 and 97.633) GeV
857 > (-78.0835 and 97.6045) GeV
858 > (-78.0607 and 97.576) GeV
859 > (-78.038 and 97.5475) GeV
860 > (-78.0153 and 97.5191) GeV
861 > (-77.9926 and 97.4908) GeV
862 > (-77.9699 and 97.4625) GeV
863 > (-77.9473 and 97.4342) GeV
864 > (-77.9247 and 97.4059) GeV
865 > (-77.9021 and 97.3777) GeV
866 > (-77.8796 and 97.3496) GeV
867 > (-77.8571 and 97.3215) GeV
868 > (-77.8347 and 97.2934) GeV
869 > (-77.8122 and 97.2654) GeV
870 > (-77.7899 and 97.2374) GeV
871 > (-77.7675 and 97.2094) GeV
872 > (-77.7452 and 97.1815) GeV
873 > (-77.7229 and 97.1536) GeV
874 > (-77.7006 and 97.1258) GeV
875 > (-77.6784 and 97.098) GeV
876 > (-77.6562 and 97.0702) GeV
877 > (-77.634 and 97.0425) GeV
878 > (-77.6118 and 97.0149) GeV
879 > (-77.5897 and 96.9872) GeV
880 > (-77.5677 and 96.9596) GeV
881 > (-77.5456 and 96.9321) GeV
882 > (-77.5236 and 96.9046) GeV
883 > (-77.5016 and 96.8771) GeV
884 > (-77.4797 and 96.8496) GeV
885 > (-77.4577 and 96.8222) GeV
886 > (-77.4359 and 96.7949) GeV
887 > (-77.414 and 96.7675) GeV
888 > (-77.3922 and 96.7403) GeV
889 > (-77.3704 and 96.713) GeV
890 > (-77.3486 and 96.6858) GeV
891 > (-77.3269 and 96.6586) GeV
892 > (-77.3052 and 96.6315) GeV
893 > (-77.2835 and 96.6044) GeV
894 > (-77.2618 and 96.5774) GeV
895 > (-77.2402 and 96.5503) GeV
896 > (-77.2187 and 96.5234) GeV
897 > (-77.1971 and 96.4964) GeV
898 > (-77.1756 and 96.4695) GeV
899 > (-77.1541 and 96.4426) GeV
900 > (-77.1326 and 96.4158) GeV
901 > (-77.1112 and 96.389) GeV
902 > (-77.0898 and 96.3623) GeV
903 > (-77.0684 and 96.3355) GeV
904 > (-77.0471 and 96.3089) GeV
905 > (-77.0257 and 96.2822) GeV
906 > (-77.0045 and 96.2556) GeV

907 > (-76.9832 and 96.229) GeV
908 > (-76.962 and 96.2025) GeV
909 > (-76.9408 and 96.176) GeV
910 > (-76.9196 and 96.1495) GeV
911 > (-76.8985 and 96.1231) GeV
912 > (-76.8774 and 96.0967) GeV
913 > (-76.8563 and 96.0704) GeV
914 > (-76.8352 and 96.0441) GeV
915 > (-76.8142 and 96.0178) GeV
916 > (-76.7932 and 95.9915) GeV
917 > (-76.7722 and 95.9653) GeV
918 > (-76.7513 and 95.9392) GeV
919 > (-76.7304 and 95.913) GeV
920 > (-76.7095 and 95.8869) GeV
921 > (-76.6886 and 95.8609) GeV
922 > (-76.6678 and 95.8348) GeV
923 > (-76.647 and 95.8088) GeV
924 > (-76.6263 and 95.7829) GeV
925 > (-76.6055 and 95.7569) GeV
926 > (-76.5848 and 95.7311) GeV
927 > (-76.5641 and 95.7052) GeV
928 > (-76.5435 and 95.6794) GeV
929 > (-76.5228 and 95.6536) GeV
930 > (-76.5022 and 95.6278) GeV
931 > (-76.4817 and 95.6021) GeV
932 > (-76.4611 and 95.5764) GeV
933 > (-76.4406 and 95.5508) GeV
934 > (-76.4201 and 95.5252) GeV
935 > (-76.3996 and 95.4996) GeV
936 > (-76.3792 and 95.4741) GeV
937 > (-76.3588 and 95.4485) GeV
938 > (-76.3384 and 95.4231) GeV
939 > (-76.3181 and 95.3976) GeV
940 > (-76.2977 and 95.3722) GeV
941 > (-76.2774 and 95.3468) GeV
942 > (-76.2572 and 95.3215) GeV
943 > (-76.2369 and 95.2962) GeV
944 > (-76.2167 and 95.2709) GeV
945 > (-76.1965 and 95.2457) GeV
946 > (-76.1763 and 95.2205) GeV
947 > (-76.1562 and 95.1953) GeV
948 > (-76.1361 and 95.1702) GeV
949 > (-76.116 and 95.1451) GeV
950 > (-76.096 and 95.12) GeV
951 > (-76.0759 and 95.0949) GeV
952 > (-76.0559 and 95.0699) GeV
953 > (-76.0359 and 95.045) GeV
954 > (-76.016 and 95.02) GeV
955 > (-75.996 and 94.9951) GeV
956 > (-75.9761 and 94.9702) GeV
957 > (-75.9563 and 94.9454) GeV
958 > (-75.9364 and 94.9206) GeV
959 > (-75.9166 and 94.8958) GeV
960 > (-75.8968 and 94.871) GeV
961 > (-75.877 and 94.8463) GeV
962 > (-75.8573 and 94.8216) GeV
963 > (-75.8376 and 94.797) GeV
964 > (-75.8179 and 94.7724) GeV

```
965 > (-75.7982 and 94.7478) GeV
966 > (-75.7786 and 94.7232) GeV
967 > (-75.7589 and 94.6987) GeV
968 > (-75.7393 and 94.6742) GeV
969 > (-75.7198 and 94.6498) GeV
970 > (-75.7002 and 94.6253) GeV
971 > (-75.6807 and 94.6009) GeV
972 > (-75.6612 and 94.5766) GeV
973 > (-75.6418 and 94.5522) GeV
974 > (-75.6223 and 94.5279) GeV
975 > (-75.6029 and 94.5037) GeV
976 > (-75.5835 and 94.4794) GeV
977 > (-75.5641 and 94.4552) GeV
978 > (-75.5448 and 94.431) GeV
979 > (-75.5255 and 94.4069) GeV
980 > (-75.5062 and 94.3828) GeV
981 > (-75.4869 and 94.3587) GeV
982 > (-75.4677 and 94.3346) GeV
983 > (-75.4484 and 94.3106) GeV
984 > (-75.4293 and 94.2866) GeV
985 > (-75.4101 and 94.2626) GeV
986 > (-75.3909 and 94.2387) GeV
987 > (-75.3718 and 94.2148) GeV
988 > (-75.3527 and 94.1909) GeV
989 > (-75.3336 and 94.1671) GeV
990 > (-75.3146 and 94.1433) GeV
991 > (-75.2956 and 94.1195) GeV
992 > (-75.2766 and 94.0957) GeV
993 > (-75.2576 and 94.072) GeV
994 > (-75.2386 and 94.0483) GeV
995 > (-75.2197 and 94.0247) GeV
996 > (-75.2008 and 94.001) GeV
997 > (-75.1819 and 93.9774) GeV
998 > (-75.1631 and 93.9539) GeV
999 > (-75.1442 and 93.9303) GeV

[Program finished]
```

Or

```
from math import sqrt, floor
t = 1000
dP1 = [0]*(t+1)
dP2 = [0]*(t+1)
dE = [0]*(t+1)
S = [0]*(t+1)
gM = sqrt(1393.6328858707575005182839547884)

for k3 in range(-t, t + 1):
if k3!= 0 and k3!= -1 and k3!= -2:

dP1[k3]=-((2**(11/4)*gM*k3**2+2**(19/4)*gM*k3+2**(15/4)*gM)*sqrt(abs(k3))*abs(2*k3+2)**(3/4))/((k3**3+4*k3**2+4*k3)*sqrt(abs(k3+1)))

dP2[k3]=((5*2**(3/4)*gM*k3**3+25*2**(3/4)*gM*k3**2+2**(23/4)*gM*k3+3*2**(11/4)*gM)*abs(k3)**(3/2)*abs(2*k3+2)**(3/4))/((k3**4+4*k3**3+4*k3**2)*abs(k3+1)**(3/2))

dE[k3] = (2**(7/4)*gM*sqrt(abs(k3+1)))/(sqrt(abs(k3))*abs(2*k3+2)**(1/4))
```

```
if k3 = = t:
 N = floor(dE[1]/dE[t])

for l1 in range(1, t - 1 + 1):
 print(l1, '> (', dP1[l1], ' and ', dP2[l1], ') GeV')

for l2 in range(N, l1 - 1 + 1, -1):
 for q1 in range(1, N + 1):
 for q2 in range(1, N + 1):

 if (dP1[l1] + dP2[l1]) - (q1 * dP1[l2] + q2 * dP2[l2]) > 0 and (dP1[l1] + dP2[l1]) - (q1 * dP1[l2] + q2 * dP2[l2]) < (dP1[l1] + dP2[l1]) - (dP1[l1+1] + dP2[l1+1]):
 print(l1, '> ~ {', q1, '+', q2, '}.', l2, '>')
 S[l1] = S[l1]+q1+q2
 elif l2 = = l1-1 and q1 = = N and q2 = = N:
 print(S[l1])
```

→ 1 > (-390.6539386225637 and 516.2212760369592) GeV
2 > (-343.91613120922744 and 442.17788298329236) GeV
3 > (-317.26442383699344 and 403.4775824883504) GeV
4 > (-298.2415168443712 and 377.18780071494) GeV
5 > (-283.55268636279027 and 357.45737589351734) GeV
6 > (-271.7000873961953 and 341.8162389823102) GeV
7 > (-261.84526624588773 and 328.9638313279034) GeV
8 > (-253.46668643498933 and 318.126555423507) GeV
9 > (-246.21783835474818 and 308.8068266760182) GeV
10 > (-239.85744399004764 and 300.66637345231317) GeV
11 > (-234.21133699219496 and 293.4654027881245) GeV
12 > (-229.15010016387234 and 287.0282182465) GeV
13 > (-224.57523121688487 and 281.22257092966413) GeV
14 > (-220.4102189703796 and 275.9466520967744) GeV
15 > (-216.5945865514654 and 271.120575674613) GeV
16 > (-213.07979892902634 and 266.6806179142783) GeV
17 > (-209.8263809454959 and 262.5752134812368) GeV
18 > (-206.80184366062977 and 258.7621058869186) GeV
19 > (-203.9791637597128 and 255.20627720961787) GeV
20 > (-201.3356495824613 and 251.87841638635706) GeV
21 > (-198.85208262036286 and 248.7537674904634) GeV
22 > (-196.5120586669301 and 245.81125108511458) GeV
23 > (-194.3014759183959 and 243.0327850873355) GeV
24 > (-192.20813275710222 and 240.40275358195424) GeV
25 > (-190.22140845177665 and 237.9075868428857) GeV
26 > (-188.3320072770243 and 235.53542598072866) GeV
27 > (-186.53175166310086 and 233.27585272051078) GeV
28 > (-184.81341362942356 and 231.11966982833255) GeV
29 > (-183.17057638646466 and 229.0587213049716) GeV
30 > (-181.5975199145539 and 227.08574408101748) GeV
31 > (-180.08912575073177 and 225.19424487275262) GeV
32 > (-178.6407972778022 and 223.37839728931408) GeV
33 > (-177.2483926118093 and 221.63295535699862) GeV
34 > (-175.9081677949133 and 219.9531804421713) GeV
35 > (-174.61672846968085 and 218.33477917834816) GeV
36 > (-173.37098857393224 and 216.77385048460457) GeV
37 > (-172.168134878535 and 215.26684013697005) GeV
38 > (-171.005596413066 and 213.81050164787598) GeV
39 > (-169.88101800026317 and 212.4018624401741) GeV
40 > (-168.7922372602816 and 211.03819448603875) GeV
41 > (-167.73726455795207 and 209.71698872791055) GeV

42 > (-166.71426545658895 and 208.43593271666595) GeV
43 > (-165.7215453150557 and 207.19289099758015) GeV
44 > (-164.7575357243483 and 205.9858878521535) GeV
45 > (-163.82078252866498 and 204.81309206715846) GeV
46 > (-162.90993521596522 and 203.67280345419104) GeV
47 > (-162.02373749606568 and 202.563440885817) GeV
48 > (-161.16101891171968 and 201.4835316498401) GeV
49 > (-160.32068735093415 and 200.4317019526828) GeV
50 > (-159.50172234783844 and 199.40666842746043) GeV
51 > (-158.70316907541124 and 198.40723052293794) GeV
52 > (-157.92413294683206 and 197.4322636668871) GeV
53 > (-157.16377475359766 and 196.4807131119861) GeV
54 > (-156.4213062781863 and 195.55158838479062) GeV
55 > (-155.69598632724694 and 194.64395826882574) GeV
56 > (-154.9871171382882 and 193.75694626181598) GeV
57 > (-154.29404111881922 and 192.88972645473538) GeV
58 > (-153.61613788203184 and 192.04151978693136) GeV
59 > (-152.95282154752664 and 191.21159063722453) GeV
60 > (-152.3035382793957 and 190.39924371575674) GeV
61 > (-151.66776403726902 and 189.6038212255676) GeV
62 > (-151.0450025187881 and 188.82470026652598) GeV
63 > (-150.43478327445217 and 188.06129045741213) GeV
64 > (-149.83665997794483 and 187.31303175469765) GeV
65 > (-149.25020883693637 and 186.5793924489827) GeV
66 > (-148.67502713100927 and 185.85986732214872) GeV
67 > (-148.11073186480164 and 185.15397595013044) GeV
68 > (-147.55695852573666 and 184.46126113782879) GeV
69 > (-147.01335993682545 and 183.78128747411156) GeV
70 > (-146.4796051960183 and 183.11363999610313) GeV
71 > (-145.9553786944528 and 182.45792295307572) GeV
72 > (-145.44037920672045 and 181.81375866123273) GeV
73 > (-144.93431904695726 and 181.18078644154795) GeV
74 > (-144.4369232851736 and 180.558661633596) GeV
75 > (-143.94792901878006 and 179.9470546789938) GeV
76 > (-143.4670846947484 and 179.3456502686899) GeV
77 > (-142.99414947827856 and 178.7541465488803) GeV
78 > (-142.52889266422613 and 178.17225438081903) GeV
79 > (-142.0710931278913 and 177.59969665023058) GeV
80 > (-141.620538812079 and 177.03620762241883) GeV
81 > (-141.17702624761685 and 176.48153233952303) GeV
82 > (-140.7403601047686 and 175.93542605668404) GeV
83 > (-140.31035277320356 and 175.39765371416965) GeV
84 > (-139.88682396838757 and 174.86798944276526) GeV
85 > (-139.46960036244215 and 174.34621609996438) GeV
86 > (-139.0585152376834 and 173.83212483470675) GeV
87 > (-138.6534081612047 and 173.32551467859645) GeV
88 > (-138.25412467899852 and 172.82619216170679) GeV
89 > (-137.8605160282396 and 172.33397095123306) GeV
90 > (-137.47243886646046 and 171.84867151139431) GeV
91 > (-137.08975501645344 and 171.37012078311542) GeV
92 > (-136.71233122582447 and 170.89815188213487) GeV
93 > (-136.34003894020944 and 170.4326038142921) GeV
94 > (-135.9727540892387 and 169.97332120684294) GeV
95 > (-135.61035688440737 and 169.52015405474296) GeV
96 > (-135.25273162807156 and 169.0729574809147) GeV
97 > (-134.89976653284984 and 168.63159150959393) GeV
98 > (-134.55135355076263 and 168.19592085191354) GeV
99 > (-134.20738821149135 and 167.76581470294693) GeV

100 > (-133.86776946918437 and 167.34114654949082) GeV
101 > (-133.53239955727736 and 166.9217939879164) GeV
102 > (-133.20118385083526 and 166.5076385514695) GeV
103 > (-132.87403073595598 and 166.0985655464412) GeV
104 > (-132.55085148580895 and 165.69446389667215) GeV
105 > (-132.2315601429129 and 165.29522599589208) GeV
106 > (-131.9160734072808 and 164.9007475674275) GeV
107 > (-131.6043105300892 and 164.51092753084663) GeV
108 > (-131.29619321254913 and 164.12566787513472) GeV
109 > (-130.9916455096796 and 163.7448735380253) GeV
110 > (-130.69059373870297 and 163.3684522911326) GeV
111 > (-130.39296639180088 and 162.9963146305591) GeV
112 > (-130.09869405298565 and 162.62837367266783) GeV
113 > (-129.80770931885866 and 162.2645450547337) GeV
114 > (-129.51994672304116 and 161.90474684020404) GeV
115 > (-129.23534266407694 and 161.54889942831568) GeV
116 > (-128.95383533661862 and 161.19692546783332) GeV
117 > (-128.6753646657216 and 160.84874977468547) GeV
118 > (-128.3998722440786 and 160.504299253294) GeV
119 > (-128.12730127204287 and 160.16350282139697) GeV
120 > (-127.8575965002894 and 159.82629133818597) GeV
121 > (-127.59070417498144 and 159.49259753558323) GeV
122 > (-127.32657198531022 and 159.16235595249725) GeV
123 > (-127.06514901328862 and 158.8355028719046) GeV
124 > (-126.80638568568263 and 158.5119762606148) GeV
125 > (-126.5502337279757 and 158.19171571158265) GeV
126 > (-126.29664612026146 and 157.87466238864081) GeV
127 > (-126.04557705497233 and 157.56075897353406) GeV
128 > (-125.79698189635184 and 157.24994961514025) GeV
129 > (-125.55081714158648 and 156.94217988077233) GeV
130 > (-125.30704038351699 and 156.63739670946123) GeV
131 > (-125.0656102748536 and 156.33554836712307) GeV
132 > (-124.82648649382185 and 156.03658440352254) GeV
133 > (-124.5896297111742 and 155.74045561094576) GeV
134 > (-124.35500155850099 and 155.44711398450445) GeV
135 > (-124.12256459778126 and 155.15651268399318) GeV
136 > (-123.89228229211669 and 154.86860599723) GeV
137 > (-123.6641189775926 and 154.58334930481016) GeV
138 > (-123.4380398362172 and 154.3006990462113) GeV
139 > (-123.21401086988783 and 154.0206126871855) GeV
140 > (-122.99199887533993 and 153.74304868838368) GeV
141 > (-122.77197142003416 and 153.46796647515566) GeV
142 > (-122.55389681894002 and 153.19532640847336) GeV
143 > (-122.33774411217712 and 152.92508975692954) GeV
144 > (-122.12348304347624 and 152.65721866976338) GeV
145 > (-121.91108403942476 and 152.39167615086933) GeV
146 > (-121.70051818946303 and 152.1284260337466) GeV
147 > (-121.4917572265987 and 151.86743295734917) GeV
148 > (-121.28477350880999 and 151.60866234279771) GeV
149 > (-121.07954000110709 and 151.35208037091763) GeV
150 > (-120.87603025822565 and 151.0976539605679) GeV
151 > (-120.67421840792518 and 150.84535074772865) GeV
152 > (-120.47407913486757 and 150.59513906531495) GeV
153 > (-120.27558766505227 and 150.34698792368857) GeV
154 > (-120.07871975078454 and 150.10086699183708) GeV
155 > (-119.88345165615618 and 149.85674657919543) GeV
156 > (-119.68976014301752 and 149.6145976180825) GeV
157 > (-119.49762245742116 and 149.37439164672907) GeV

158 > (-119.30701631651833 and 149.13610079287258) GeV
159 > (-119.11791989589105 and 148.89969775789763) GeV
160 > (-118.93031181730156 and 148.66515580149928) GeV
161 > (-118.74417113684386 and 148.43244872685028) GeV
162 > (-118.55947733348104 and 148.2015508662514) GeV
163 > (-118.3762102979543 and 147.9724370672469) GeV
164 > (-118.19435032204842 and 147.74508267918685) GeV
165 > (-118.01387808820115 and 147.51946354022036) GeV
166 > (-117.83477465944247 and 147.2955559647017) GeV
167 > (-117.65702146965246 and 147.07333673099512) GeV
168 > (-117.4806003141249 and 146.85278306966342) GeV
169 > (-117.30549334042544 and 146.6338726520245) GeV
170 > (-117.13168303953417 and 146.4165835790647) GeV
171 > (-116.95915223726091 and 146.20089437069376) GeV
172 > (-116.78788408592455 and 145.98678395532937) GeV
173 > (-116.61786205628582 and 145.7742316598008) GeV
174 > (-116.44906992972491 and 145.56321719955727) GeV
175 > (-116.28149179065502 and 145.35372066917378) GeV
176 > (-116.11511201916348 and 145.14572253313966) GeV
177 > (-115.94991528387204 and 144.93920361692346) GeV
178 > (-115.78588653500907 and 144.73414509830226) GeV
179 > (-115.62301099768595 and 144.53052849894658) GeV
180 > (-115.46127416537071 and 144.32833567625272) GeV
181 > (-115.30066179355153 and 144.12754881541258) GeV
182 > (-115.1411598935846 and 143.92815042171426) GeV
183 > (-114.98275472671885 and 143.73012331306396) GeV
184 > (-114.82543279829227 and 143.5334506127234) GeV
185 > (-114.66918085209392 and 143.3381157422537) GeV
186 > (-114.51398586488536 and 143.1441024146598) GeV
187 > (-114.35983504107762 and 142.95139462772855) GeV
188 > (-114.20671580755638 and 142.75997665755347) GeV
189 > (-114.05461580865261 and 142.5698330522411) GeV
190 > (-113.90352290125253 and 142.38094862579115) GeV
191 > (-113.7534251500424 and 142.19330845214674) GeV
192 > (-113.60431082288436 and 142.00689785940767) GeV
193 > (-113.45616838631857 and 141.821702424202) GeV
194 > (-113.3089865011877 and 141.63770796621102) GeV
195 > (-113.16275401837959 and 141.45490054284227) GeV
196 > (-113.01745997468483 and 141.2732664440457) GeV
197 > (-112.87309358876507 and 141.09279218726934) GeV
198 > (-112.72964425722894 and 140.91346451254896) GeV
199 > (-112.58710155081178 and 140.73527037772794) GeV
200 > (-112.44545521065604 and 140.55819695380347) GeV
201 > (-112.30469514469006 and 140.3822316203952) GeV
202 > (-112.1648114241003 and 140.20736196133151) GeV
203 > (-112.02579427989622 and 140.03357576035123) GeV
204 > (-111.88763409956346 and 139.86086099691622) GeV
205 > (-111.75032142380304 and 139.68920584213188) GeV
206 > (-111.61384694335489 and 139.518598654772) GeV
207 > (-111.4782014959011 and 139.3490279774051) GeV
208 > (-111.34337606304865 and 139.18048253261898) GeV
209 > (-111.20936176738815 and 139.01295121934064) GeV
210 > (-111.07614986962584 and 138.84642310924838) GeV
211 > (-110.94373176578826 and 138.68088744327434) GeV
212 > (-110.81209898449553 and 138.51633362819314) GeV
213 > (-110.6812431843028 and 138.35275123329572) GeV
214 > (-110.55115615110704 and 138.1901299871449) GeV
215 > (-110.42182979561719 and 138.02845977441078) GeV

216 > (-110.29325615088653 and 137.86773063278298) GeV
217 > (-110.16542736990472 and 137.7079327499585) GeV
218 > (-110.03833572324818 and 137.54905646070227) GeV
219 > (-109.91197359678706 and 137.3910922439782) GeV
220 > (-109.78633348944717 and 137.2340307201496) GeV
221 > (-109.66140801102537 and 137.077862648246) GeV
222 > (-109.53718988005677 and 136.92257892329496) GeV
223 > (-109.41367192173254 and 136.76817057371733) GeV
224 > (-109.29084706586637 and 136.61462875878345) GeV
225 > (-109.16870834490913 and 136.4619447661292) GeV
226 > (-109.04724889200945 and 136.31011000932995) GeV
227 > (-108.9264619391194 and 136.15911602553106) GeV
228 > (-108.80634081514425 and 136.00895447313292) GeV
229 > (-108.6868789441345 and 135.85961712952945) GeV
230 > (-108.56806984351957 and 135.71109588889877) GeV
231 > (-108.44990712238175 and 135.5633827600435) GeV
232 > (-108.33238447976942 and 135.41646986428128) GeV
233 > (-108.21549570304843 and 135.27034943338205) GeV
234 > (-108.09923466629056 and 135.12501380755228) GeV
235 > (-107.98359532869816 and 134.9804554334645) GeV
236 > (-107.86857173306416 and 134.8366668623308) GeV
237 > (-107.75415800426612 and 134.693640748019) GeV
238 > (-107.6403483477937 and 134.55136984521073) GeV
239 > (-107.5271370483088 and 134.40984700760052) GeV
240 > (-107.41451846823695 and 134.2690651861335) GeV
241 > (-107.30248704639003 and 134.1290174272827) GeV
242 > (-107.1910372966184 and 133.98969687136338) GeV
243 > (-107.08016380649292 and 133.85109675088395) GeV
244 > (-106.96986123601486 and 133.71321038893265) GeV
245 > (-106.86012431635397 and 133.57603119759892) GeV
246 > (-106.75094784861352 and 133.43955267642855) GeV
247 > (-106.64232670262142 and 133.30376841091194) GeV
248 > (-106.53425581574747 and 133.1686720710045) GeV
249 > (-106.42673019174538 and 133.034257409678) GeV
250 > (-106.31974489961927 and 132.90051826150324) GeV
251 > (-106.21329507251394 and 132.76744854126167) GeV
252 > (-106.10737590662853 and 132.6350422425868) GeV
253 > (-106.00198266015246 and 132.50329343663316) GeV
254 > (-105.89711065222396 and 132.37219627077363) GeV
255 > (-105.79275526190956 and 132.24174496732337) GeV
256 > (-105.68891192720518 and 132.11193382228976) GeV
257 > (-105.58557614405716 and 131.98275720414858) GeV
258 > (-105.48274346540367 and 131.85420955264473) GeV
259 > (-105.38040950023557 and 131.72628537761756) GeV
260 > (-105.27856991267606 and 131.59897925785003) GeV
261 > (-105.1772204210793 and 131.47228583994104) GeV
262 > (-105.07635679714694 and 131.3461998372004) GeV
263 > (-104.97597486506218 and 131.22071602856627) GeV
264 > (-104.87607050064172 and 131.09582925754373) GeV
265 > (-104.77663963050372 and 130.97153443116503) GeV
266 > (-104.67767823125315 and 130.84782651896975) GeV
267 > (-104.5791823286826 and 130.7247005520059) GeV
268 > (-104.48114799698891 and 130.60215162184997) GeV
269 > (-104.38357135800536 and 130.48017487964663) GeV
270 > (-104.28644858044865 and 130.35876553516712) GeV
271 > (-104.1897758791809 and 130.2379188558855) GeV
272 > (-104.093549514485 and 130.11763016607367) GeV
273 > (-103.99776579135535 and 129.99789484591295) GeV

274 > (-103.90242105880098 and 129.87870833062345) GeV
275 > (-103.80751170916284 and 129.76006610960965) GeV
276 > (-103.71303417744349 and 129.64196372562273) GeV
277 > (-103.61898494065044 and 129.52439677393826) GeV
278 > (-103.52536051715069 and 129.4073609015502) GeV
279 > (-103.43215746603883 and 129.2908518063797) GeV
280 > (-103.33937238651585 and 129.1748652364985) GeV
281 > (-103.24700191728039 and 129.0593969893678) GeV
282 > (-103.15504273593095 and 128.94444291109045) GeV
283 > (-103.06349155837955 and 128.82999889567736) GeV
284 > (-102.97234513827577 and 128.71606088432802) GeV
285 > (-102.8816002664421 and 128.6026248647235) GeV
286 > (-102.79125377031927 and 128.48968687033303) GeV
287 > (-102.70130251342167 and 128.3772429797332) GeV
288 > (-102.61174339480307 and 128.26528931593955) GeV
289 > (-102.52257334853198 and 128.1538220457504) GeV
290 > (-102.43378934317633 and 128.04283737910256) GeV
291 > (-102.34538838129771 and 127.93233156843885) GeV
292 > (-102.25736749895478 and 127.82230090808675) GeV
293 > (-102.1697237652154 and 127.71274173364863) GeV
294 > (-102.08245428167754 and 127.6036504214025) GeV
295 > (-101.99555618199891 and 127.49502338771369) GeV
296 > (-101.90902663143493 and 127.38685708845709) GeV
297 > (-101.82286282638465 and 127.27914801844949) GeV
298 > (-101.73706199394506 and 127.1718927108917) GeV
299 > (-101.65162139147301 and 127.06508773682131) GeV
300 > (-101.56653830615487 and 126.95872970457417) GeV
301 > (-101.48181005458376 and 126.8528152592557) GeV
302 > (-101.39743398234418 and 126.7473410822217) GeV
303 > (-101.31340746360387 and 126.64230389056742) GeV
304 > (-101.22972790071245 and 126.53770043662647) GeV
305 > (-101.14639272380758 and 126.43352750747749) GeV
306 > (-101.06339939042742 and 126.32978192445974) GeV
307 > (-100.98074538512981 and 126.22646054269737) GeV
308 > (-100.89842821911813 and 126.12356025063092) GeV
309 > (-100.81644542987354 and 126.0210779695575) GeV
310 > (-100.73479458079304 and 125.9190106531788) GeV
311 > (-100.65347326083446 and 125.81735528715635) GeV
312 > (-100.5724790841668 and 125.7161088886744) GeV
313 > (-100.49180968982637 and 125.61526850601064) GeV
314 > (-100.41146274137945 and 125.51483121811344) GeV
315 > (-100.33143592658986 and 125.41479413418658) GeV
316 > (-100.25172695709217 and 125.3151543932806) GeV
317 > (-100.17233356807094 and 125.2159091638916) GeV
318 > (-100.09325351794442 and 125.11705564356575) GeV
319 > (-100.01448458805405 and 125.01859105851103) GeV
320 > (-99.93602458235904 and 124.92051266321518) GeV
321 > (-99.85787132713551 and 124.82281774006965) GeV
322 > (-99.78002267068125 and 124.72550359900023) GeV
323 > (-99.70247648302463 and 124.62856757710324) GeV
324 > (-99.6252306556386 and 124.53200703828783) GeV
325 > (-99.54828310115947 and 124.435819372924) GeV
326 > (-99.47163175310966 and 124.34000199749659) GeV
327 > (-99.39527456562558 and 124.24455235426433) GeV
328 > (-99.31920951318952 and 124.14946791092487) GeV
329 > (-99.24343459036568 and 124.05474616028474) GeV
330 > (-99.16794781154093 and 123.96038461993508) GeV
331 > (-99.09274721066932 and 123.86638083193218) GeV

332 > (-99.01783084102043 and 123.77273236248283) GeV
333 > (-98.94319677493245 and 123.6794368016355) GeV
334 > (-98.86884310356832 and 123.58649176297546) GeV
335 > (-98.79476793667621 and 123.49389488332527) GeV
336 > (-98.72096940235375 and 123.40164382244991) GeV
337 > (-98.64744564681558 and 123.30973626276615) GeV
338 > (-98.57419483416493 and 123.21816990905691) GeV
339 > (-98.50121514616843 and 123.1269424881898) GeV
340 > (-98.42850478203482 and 123.0360517488401) GeV
341 > (-98.35606195819649 and 122.94549546121794) GeV
342 > (-98.28388490809496 and 122.85527141679987) GeV
343 > (-98.2119718819693 and 122.76537742806416) GeV
344 > (-98.14032114664792 and 122.67581132823085) GeV
345 > (-98.06893098534336 and 122.58657097100489) GeV
346 > (-97.99779969745073 and 122.49765423032399) GeV
347 > (-97.92692559834848 and 122.40905900010964) GeV
348 > (-97.8563070192027 and 122.3207831940225) GeV
349 > (-97.78594230677419 and 122.2328247452211) GeV
350 > (-97.71582982322842 and 122.14518160612391) GeV
351 > (-97.64596794594844 and 122.05785174817605) GeV
352 > (-97.5763550673506 and 121.97083316161802) GeV
353 > (-97.50698959470272 and 121.88412385525913) GeV
354 > (-97.43786994994544 and 121.79772185625366) GeV
355 > (-97.36899456951578 and 121.71162520988041) GeV
356 > (-97.30036190417367 and 121.62583197932565) GeV
357 > (-97.23197041883084 and 121.54034024546927) GeV
358 > (-97.16381859238211 and 121.45514810667382) GeV
359 > (-97.0959049175395 and 121.37025367857714) GeV
360 > (-97.02822790066855 and 121.2856550938875) GeV
361 > (-96.96078606162692 and 121.201350502182) GeV
362 > (-96.89357793360554 and 121.11733806970791) GeV
363 > (-96.82660206297211 and 121.03361597918675) GeV
364 > (-96.75985700911656 and 120.9501824296214) GeV
365 > (-96.69334134429914 and 120.8670356361059) GeV
366 > (-96.62705365350028 and 120.78417382963785) GeV
367 > (-96.56099253427283 and 120.7015952569337) GeV
368 > (-96.49515659659659 and 120.61929818024662) GeV
369 > (-96.42954446273457 and 120.53728087718692) GeV
370 > (-96.36415476709139 and 120.45554164054525) GeV
371 > (-96.29898615607394 and 120.37407877811786) GeV
372 > (-96.23403728795364 and 120.29289061253468) GeV
373 > (-96.16930683273092 and 120.21197548108998) GeV
374 > (-96.10479347200145 and 120.13133173557479) GeV
375 > (-96.04049589882437 and 120.05095774211246) GeV
376 > (-95.97641281759238 and 119.97085188099584) GeV
377 > (-95.91254294390323 and 119.89101254652716) GeV
378 > (-95.84888500443382 and 119.81143814686001) GeV
379 > (-95.7854377368152 and 119.73212710384348) GeV
380 > (-95.72219988950988 and 119.65307785286852) GeV
381 > (-95.65917022169063 and 119.5742888427163) GeV
382 > (-95.59634750312065 and 119.49575853540892) GeV
383 > (-95.53373051403605 and 119.41748540606193) GeV
384 > (-95.47131804502952 and 119.33946794273886) GeV
385 > (-95.40910889693545 and 119.26170464630805) GeV
386 > (-95.34710188071698 and 119.18419403030094) GeV
387 > (-95.28529581735447 and 119.10693462077283) GeV
388 > (-95.22368953773511 and 119.02992495616517) GeV
389 > (-95.1622818825449 and 118.95316358716968) GeV

390 > (-95.10107170216098 and 118.87664907659462) GeV
391 > (-95.04005785654641 and 118.80037999923246) GeV
392 > (-94.97923921514558 and 118.7243549417299) GeV
393 > (-94.91861465678168 and 118.64857250245889) GeV
394 > (-94.85818306955491 and 118.57303129138992) GeV
395 > (-94.79794335074267 and 118.49772992996692) GeV
396 > (-94.73789440670039 and 118.42266705098358) GeV
397 > (-94.6780351527646 and 118.34784129846159) GeV
398 > (-94.61836451315614 and 118.27325132753035) GeV
399 > (-94.55888142088567 and 118.19889580430818) GeV
400 > (-94.49958481765981 and 118.12477340578542) GeV
401 > (-94.44047365378873 and 118.05088281970855) GeV
402 > (-94.38154688809495 and 117.97722274446649) GeV
403 > (-94.32280348782328 and 117.90379188897771) GeV
404 > (-94.26424242855191 and 117.83058897257925) GeV
405 > (-94.20586269410488 and 117.7576127249171) GeV
406 > (-94.1476632764653 and 117.68486188583783) GeV
407 > (-94.08964317568983 and 117.61233520528198) GeV
408 > (-94.03180139982473 and 117.5400314431782) GeV
409 > (-93.97413696482221 and 117.46794936933966) GeV
410 > (-93.91664889445842 and 117.39608776336081) GeV
411 > (-93.85933622025222 and 117.32444541451633) GeV
412 > (-93.80219798138509 and 117.25302112166054) GeV
413 > (-93.74523322462215 and 117.1818136931289) GeV
414 > (-93.688441004234 and 117.11082194664031) GeV
415 > (-93.63182038191958 and 117.04004470920052) GeV
416 > (-93.57537042673013 and 116.96948081700724) GeV
417 > (-93.51909021499407 and 116.89912911535602) GeV
418 > (-93.46297883024272 and 116.82898845854749) GeV
419 > (-93.40703536313711 and 116.7590577097958) GeV
420 > (-93.35125891139548 and 116.68933574113817) GeV
421 > (-93.29564857972204 and 116.61982143334544) GeV
422 > (-93.24020347973628 and 116.55051367583397) GeV
423 > (-93.18492272990323 and 116.4814113665785) GeV
424 > (-93.1298054554649 and 116.41251341202612) GeV
425 > (-93.07485078837203 and 116.3438187270112) GeV
426 > (-93.02005786721708 and 116.27532623467172) GeV
427 > (-92.96542583716798 and 116.20703486636593) GeV
428 > (-92.91095384990271 and 116.13894356159103) GeV
429 > (-92.8566410635444 and 116.07105126790204) GeV
430 > (-92.80248664259761 and 116.00335694083165) GeV
431 > (-92.7484897578853 and 115.935859543812) GeV
432 > (-92.69464958648628 and 115.86855804809605) GeV
433 > (-92.64096531167382 and 115.80145143268116) GeV
434 > (-92.58743612285468 and 115.73453868423243) GeV
435 > (-92.53406121550918 and 115.66781879700831) GeV
436 > (-92.4808397911317 and 115.60129077278573) GeV
437 > (-92.42777105717208 and 115.53495362078701) GeV
438 > (-92.37485422697762 and 115.46880635760765) GeV
439 > (-92.32208851973596 and 115.4028480071443) GeV
440 > (-92.26947316041829 and 115.33707760052425) GeV
441 > (-92.21700737972373 and 115.27149417603576) GeV
442 > (-92.16469041402391 and 115.2060967790586) GeV
443 > (-92.11252150530841 and 115.14088446199614) GeV
444 > (-92.06049990113092 and 115.07585628420777) GeV
445 > (-92.00862485455606 and 115.0110113119424) GeV
446 > (-91.95689562410627 and 114.94634861827254) GeV
447 > (-91.90531147371033 and 114.88186728302914) GeV

448 > (-91.85387167265161 and 114.81756639273767) GeV
449 > (-91.80257549551733 and 114.7534450405541) GeV
450 > (-91.75142222214835 and 114.68950232620239) GeV
451 > (-91.70041113758951 and 114.62573735591248) GeV
452 > (-91.64954153204052 and 114.56214924235863) GeV
453 > (-91.5988127008075 and 114.49873710459923) GeV
454 > (-91.54822394425517 and 114.43550006801637) GeV
455 > (-91.49777456775928 and 114.37243726425714) GeV
456 > (-91.4474638816599 and 114.30954783117471) GeV
457 > (-91.39729120121505 and 114.24683091277056) GeV
458 > (-91.34725584655507 and 114.18428565913726) GeV
459 > (-91.2973571426372 and 114.12191122640193) GeV
460 > (-91.24759441920106 and 114.0597067766705) GeV
461 > (-91.19796701072428 and 113.99767147797198) GeV
462 > (-91.14847425637898 and 113.93580450420443) GeV
463 > (-91.09911549998827 and 113.87410503508048) GeV
464 > (-91.04989008998395 and 113.8125722560741) GeV
465 > (-91.00079737936389 and 113.75120535836777) GeV
466 > (-90.9518367256505 and 113.6900035388004) GeV
467 > (-90.90300749084936 and 113.62896599981552) GeV
468 > (-90.85430904140851 and 113.56809194941029) GeV
469 > (-90.8057407481778 and 113.50738060108523) GeV
470 > (-90.75730198636946 and 113.4468311737941) GeV
471 > (-90.7089921355181 and 113.3864428918945) GeV
472 > (-90.66081057944213 and 113.32621498509957) GeV
473 > (-90.61275670620489 and 113.26614668842906) GeV
474 > (-90.56482990807662 and 113.20623724216215) GeV
475 > (-90.5170295814968 and 113.14648589179004) GeV
476 > (-90.46935512703645 and 113.08689188796926) GeV
477 > (-90.42180594936181 and 113.02745448647569) GeV
478 > (-90.37438145719726 and 112.9681729481588) GeV
479 > (-90.32708106328967 and 112.90904653889653) GeV
480 > (-90.27990418437236 and 112.85007452955071) GeV
481 > (-90.23285024113001 and 112.79125619592281) GeV
482 > (-90.18591865816366 and 112.73259081871029) GeV
483 > (-90.13910886395612 and 112.67407768346364) GeV
484 > (-90.09242029083782 and 112.61571608054324) GeV
485 > (-90.04585237495304 and 112.55750530507744) GeV
486 > (-89.99940455622647 and 112.49944465692056) GeV
487 > (-89.95307627833009 and 112.44153344061178) GeV
488 > (-89.90686698865044 and 112.38377096533387) GeV
489 > (-89.8607761382564 and 112.32615654487307) GeV
490 > (-89.81480318186695 and 112.26868949757888) GeV
491 > (-89.76894757781973 and 112.21136914632449) GeV
492 > (-89.72320878803951 and 112.15419481846762) GeV
493 > (-89.67758627800724 and 112.09716584581182) GeV
494 > (-89.6320795167295 and 112.04028156456793) GeV
495 > (-89.58668797670798 and 111.98354131531642) GeV
496 > (-89.5414111339095 and 111.9269444429696) GeV
497 > (-89.49624846773654 and 111.87049029673467) GeV
498 > (-89.45119946099737 and 111.81417823007683) GeV
499 > (-89.40626359987746 and 111.75800760068307) GeV
500 > (-89.3614403739104 and 111.701977770426) GeV
501 > (-89.31672927594953 and 111.64608810532845) GeV
502 > (-89.27212980213976 and 111.59033797552816) GeV
503 > (-89.22764145188972 and 111.53472675524287) GeV
504 > (-89.18326372784405 and 111.47925382273598) GeV
505 > (-89.1389961358563 and 111.42391856028232) GeV

506 > (-89.0948381849617 and 111.36872035413415) GeV
507 > (-89.05078938735056 and 111.31365859448829) GeV
508 > (-89.00684925834182 and 111.25873267545259) GeV
509 > (-88.96301731635674 and 111.20394199501317) GeV
510 > (-88.91929308289308 and 111.14928595500238) GeV
511 > (-88.87567608249945 and 111.09476396106638) GeV
512 > (-88.83216584274992 and 111.04037542263357) GeV
513 > (-88.78876189421882 and 110.98611975288321) GeV
514 > (-88.74546377045601 and 110.93199636871417) GeV
515 > (-88.70227100796212 and 110.87800469071433) GeV
516 > (-88.65918314616431 and 110.82414414313004) GeV
517 > (-88.61619972739204 and 110.77041415383597) GeV
518 > (-88.5733202968534 and 110.71681415430528) GeV
519 > (-88.53054440261111 and 110.66334357958007) GeV
520 > (-88.48787159555958 and 110.61000186824222) GeV
521 > (-88.44530142940148 and 110.55678846238425) GeV
522 > (-88.40283346062482 and 110.50370280758091) GeV
523 > (-88.36046724848039 and 110.45074435286065) GeV
524 > (-88.31820235495925 and 110.39791255067756) GeV
525 > (-88.27603834477041 and 110.34520685688365) GeV
526 > (-88.23397478531896 and 110.29262673070131) GeV
527 > (-88.19201124668417 and 110.24017163469591) GeV
528 > (-88.15014730159794 and 110.18784103474914) GeV
529 > (-88.10838252542345 and 110.13563440003189) GeV
530 > (-88.06671649613403 and 110.0835512029782) GeV
531 > (-88.02514879429218 and 110.03159091925885) GeV
532 > (-87.98367900302884 and 109.97975302775555) GeV
533 > (-87.94230670802293 and 109.92803701053519) GeV
534 > (-87.90103149748091 and 109.87644235282444) GeV
535 > (-87.85985296211673 and 109.8249685429847) GeV
536 > (-87.81877069513195 and 109.7736150724871) GeV
537 > (-87.77778429219586 and 109.72238143588784) GeV
538 > (-87.73689335142605 and 109.67126713080368) GeV
539 > (-87.69609747336908 and 109.62027165788788) GeV
540 > (-87.65539626098119 and 109.56939452080621) GeV
541 > (-87.61478931960956 and 109.51863522621323) GeV
542 > (-87.57427625697316 and 109.46799328372879) GeV
543 > (-87.53385668314455 and 109.41746820591466) GeV
544 > (-87.49353021053118 and 109.36705950825171) GeV
545 > (-87.45329645385715 and 109.31676670911695) GeV
546 > (-87.41315503014528 and 109.26658932976092) GeV
547 > (-87.37310555869911 and 109.21652689428534) GeV
548 > (-87.3331476610852 and 109.16657892962107) GeV
549 > (-87.29328096111558 and 109.11674496550596) GeV
550 > (-87.2535050848303 and 109.06702453446313) GeV
551 > (-87.21381966048028 and 109.0174171717797) GeV
552 > (-87.17422431851017 and 108.9679224154851) GeV
553 > (-87.1347186915416 and 108.91853980633017) GeV
554 > (-87.09530241435624 and 108.86926888776615) GeV
555 > (-87.0559751238794 and 108.82010920592404) GeV
556 > (-87.01673645916348 and 108.77106030959398) GeV
557 > (-86.9775860613718 and 108.72212175020485) GeV
558 > (-86.93852357376237 and 108.67329308180436) GeV
559 > (-86.89954864167207 and 108.62457386103891) GeV
560 > (-86.86066091250076 and 108.57596364713378) GeV
561 > (-86.82186003569562 and 108.52746200187377) GeV
562 > (-86.7831456627356 and 108.47906848958358) GeV
563 > (-86.74451744711618 and 108.43078267710871) GeV

564 > (-86.70597504433391 and 108.38260413379639) GeV
565 > (-86.66751811187162 and 108.33453243147663) GeV
566 > (-86.62914630918316 and 108.28656714444382) GeV
567 > (-86.59085929767883 and 108.23870784943783) GeV
568 > (-86.55265674071067 and 108.19095412562602) GeV
569 > (-86.51453830355781 and 108.14330555458471) GeV
570 > (-86.47650365341215 and 108.09576172028163) GeV
571 > (-86.4385524593641 and 108.04832220905762) GeV
572 > (-86.40068439238843 and 108.00098660960931) GeV
573 > (-86.36289912533026 and 107.95375451297137) GeV
574 > (-86.32519633289114 and 107.90662551249929) GeV
575 > (-86.28757569161537 and 107.85959920385213) GeV
576 > (-86.2500368798764 and 107.81267518497562) GeV
577 > (-86.21257957786322 and 107.76585305608502) GeV
578 > (-86.17520346756697 and 107.71913241964869) GeV
579 > (-86.13790823276787 and 107.67251288037119) GeV
580 > (-86.10069355902185 and 107.62599404517722) GeV
581 > (-86.06355913364771 and 107.57957552319509) GeV
582 > (-86.02650464571413 and 107.53325692574063) GeV
583 > (-85.98952978602689 and 107.48703786630135) GeV
584 > (-85.95263424711625 and 107.44091796052031) GeV
585 > (-85.91581772322431 and 107.3948968261808) GeV
586 > (-85.87907991029273 and 107.34897408319057) GeV
587 > (-85.84242050595012 and 107.30314935356631) GeV
588 > (-85.8058392095001 and 107.25742226141863) GeV
589 > (-85.76933572190913 and 107.21179243293679) GeV
590 > (-85.73290974579432 and 107.16625949637356) GeV
591 > (-85.69656098541178 and 107.12082308203071) GeV
592 > (-85.66028914664462 and 107.07548282224384) GeV
593 > (-85.62409393699149 and 107.03023835136808) GeV
594 > (-85.58797506555473 and 106.9850893057635) GeV
595 > (-85.55193224302913 and 106.9400353237807) GeV
596 > (-85.5159651816904 and 106.8950760457468) GeV
597 > (-85.48007359538394 and 106.85021111395096) GeV
598 > (-85.44425719951374 and 106.80544017263087) GeV
599 > (-85.40851571103111 and 106.76076286795852) GeV
600 > (-85.37284884842389 and 106.71617884802664) GeV
601 > (-85.3372563317054 and 106.67168776283506) GeV
602 > (-85.30173788240388 and 106.62728926427727) GeV
603 > (-85.26629322355143 and 106.58298300612685) GeV
604 > (-85.23092207967379 and 106.53876864402454) GeV
605 > (-85.19562417677957 and 106.49464583546472) GeV
606 > (-85.16039924234991 and 106.45061423978271) GeV
607 > (-85.12524700532819 and 106.40667351814174) GeV
608 > (-85.09016719610969 and 106.36282333351997) GeV
609 > (-85.0551595465315 and 106.31906335069806) GeV
610 > (-85.02022378986248 and 106.27539323624649) GeV
611 > (-84.98535966079321 and 106.23181265851295) GeV
612 > (-84.95056689542618 and 106.18832128761018) GeV
613 > (-84.91584523126593 and 106.1449187954037) GeV
614 > (-84.88119440720922 and 106.1016048554994) GeV
615 > (-84.84661416353569 and 106.05837914323173) GeV
616 > (-84.81210424189791 and 106.01524133565172) GeV
617 > (-84.77766438531225 and 105.9721911115151) GeV
618 > (-84.74329433814924 and 105.92922815127037) GeV
619 > (-84.70899384612441 and 105.88635213704755) GeV
620 > (-84.67476265628895 and 105.84356275264628) GeV
621 > (-84.64060051702059 and 105.80085968352446) GeV

622 > (-84.60650717801457 and 105.75824261678716) GeV
623 > (-84.57248239027447 and 105.71571124117477) GeV
624 > (-84.53852590610352 and 105.67326524705251) GeV
625 > (-84.50463747909556 and 105.63090432639893) GeV
626 > (-84.47081686412626 and 105.588628172795) GeV
627 > (-84.43706381734461 and 105.54643648141334) GeV
628 > (-84.40337809616396 and 105.50432894900743) GeV
629 > (-84.36975945925384 and 105.46230527390068) GeV
630 > (-84.3362076665311 and 105.42036515597619) GeV
631 > (-84.30272247915183 and 105.3785082966576) GeV
632 > (-84.26930365950271 and 105.33673439893994) GeV
633 > (-84.23595097119293 and 105.29504316729732) GeV
634 > (-84.20266417904591 and 105.25343430775445) GeV
635 > (-84.16944304909117 and 105.21190752783552) GeV
636 > (-84.13628734855625 and 105.17046253656244) GeV
637 > (-84.10319684585868 and 105.12909904444473) GeV
638 > (-84.07017131059806 and 105.0878167634695) GeV
639 > (-84.03721051354819 and 105.04661540709176) GeV
640 > (-84.00431422664927 and 105.00549469022465) GeV
641 > (-83.97148222300005 and 104.96445432922955) GeV
642 > (-83.93871427685029 and 104.92349404190678) GeV
643 > (-83.90601016359307 and 104.88261354748579) GeV
644 > (-83.87336965975717 and 104.84181256661576) GeV
645 > (-83.84079254299974 and 104.80109082135662) GeV
646 > (-83.80827859209863 and 104.76044803516898) GeV
647 > (-83.7758275869452 and 104.71988393290573) GeV
648 > (-83.74343930853694 and 104.67939824080221) GeV
649 > (-83.71111353897024 and 104.63899068646768) GeV
650 > (-83.6788500614331 and 104.59866099887599) GeV
651 > (-83.64664866019807 and 104.55840890835674) GeV
652 > (-83.61450912061521 and 104.51823414658644) GeV
653 > (-83.58243122910494 and 104.47813644657971) GeV
654 > (-83.55041477315115 and 104.43811554268058) GeV
655 > (-83.51845954129426 and 104.39817117055387) GeV
656 > (-83.48656532312441 and 104.35830306717652) GeV
657 > (-83.4547319092746 and 104.31851097082932) GeV
658 > (-83.42295909141393 and 104.27879462108802) GeV
659 > (-83.391246662241 and 104.23915375881549) GeV
660 > (-83.35959441547715 and 104.19958812615317) GeV
661 > (-83.3280021458599 and 104.16009746651262) GeV
662 > (-83.29646964913654 and 104.12068152456786) GeV
663 > (-83.26499672205748 and 104.08134004624675) GeV
664 > (-83.23358316236981 and 104.04207277872321) GeV
665 > (-83.20222876881118 and 104.00287947040933) GeV
666 > (-83.17093334110305 and 103.9637598709471) GeV
667 > (-83.13969667994479 and 103.92471373120102) GeV
668 > (-83.10851858700725 and 103.8857408032499) GeV
669 > (-83.07739886492661 and 103.8468408403793) GeV
670 > (-83.04633731729828 and 103.80801359707404) GeV
671 > (-83.01533374867073 and 103.76925882901017) GeV
672 > (-82.98438796453954 and 103.73057629304783) GeV
673 > (-82.95349977134136 and 103.69196574722352) GeV
674 > (-82.92266897644804 and 103.65342695074266) GeV
675 > (-82.8918953881606 and 103.61495966397253) GeV
676 > (-82.86117881570347 and 103.57656364843446) GeV
677 > (-82.83051906921868 and 103.53823866679697) GeV
678 > (-82.79991595976 and 103.49998448286834) GeV
679 > (-82.76936929928746 and 103.46180086158968) GeV

680 > (-82.73887890066136 and 103.42368756902748) GeV
681 > (-82.70844457763683 and 103.38564437236701) GeV
682 > (-82.67806614485828 and 103.34767103990502) GeV
683 > (-82.64774341785366 and 103.30976734104286) GeV
684 > (-82.61747621302929 and 103.27193304627977) GeV
685 > (-82.58726434766396 and 103.23416792720586) GeV
686 > (-82.5571076399039 and 103.19647175649543) GeV
687 > (-82.5270059087572 and 103.15884430790022) GeV
688 > (-82.49695897408854 and 103.12128535624282) GeV
689 > (-82.46696665661393 and 103.08379467740995) GeV
690 > (-82.43702877789535 and 103.0463720483459) GeV
691 > (-82.40714516033559 and 103.00901724704616) GeV
692 > (-82.37731562717316 and 102.97173005255068) GeV
693 > (-82.34754000247706 and 102.93451024493781) GeV
694 > (-82.31781811114165 and 102.89735760531745) GeV
695 > (-82.28814977888173 and 102.86027191582538) GeV
696 > (-82.25853483222741 and 102.82325295961627) GeV
697 > (-82.22897309851915 and 102.78630052085803) GeV
698 > (-82.1994644059028 and 102.74941438472533) GeV
699 > (-82.17000858332484 and 102.71259433739348) GeV
700 > (-82.14060546052723 and 102.67584016603249) GeV
701 > (-82.11125486804289 and 102.63915165880091) GeV
702 > (-82.08195663719064 and 102.60252860483982) GeV
703 > (-82.05271060007067 and 102.56597079426696) GeV
704 > (-82.02351658955959 and 102.52947801817082) GeV
705 > (-81.99437443930587 and 102.4930500686047) GeV
706 > (-81.96528398372523 and 102.45668673858098) GeV
707 > (-81.9362450579959 and 102.42038782206521) GeV
708 > (-81.9072574980541 and 102.38415311397058) GeV
709 > (-81.87832114058949 and 102.34798241015197) GeV
710 > (-81.84943582304062 and 102.31187550740069) GeV
711 > (-81.82060138359047 and 102.27583220343836) GeV
712 > (-81.79181766116199 and 102.23985229691176) GeV
713 > (-81.76308449541372 and 102.2039355873872) GeV
714 > (-81.73440172673531 and 102.16808187534487) GeV
715 > (-81.70576919624327 and 102.13229096217367) GeV
716 > (-81.67718674577657 and 102.09656265016557) GeV
717 > (-81.64865421789243 and 102.06089674251035) GeV
718 > (-81.62017145586195 and 102.02529304329035) GeV
719 > (-81.59173830366599 and 101.98975135747496) GeV
720 > (-81.56335460599095 and 101.95427149091569) GeV
721 > (-81.5350202082245 and 101.9188532503406) GeV
722 > (-81.50673495645161 and 101.88349644334953) GeV
723 > (-81.47849869745032 and 101.84820087840858) GeV
724 > (-81.45031127868772 and 101.81296636484522) GeV
725 > (-81.42217254831588 and 101.77779271284332) GeV
726 > (-81.39408235516787 and 101.7426797334379) GeV
727 > (-81.3660405487537 and 101.70762723851024) GeV
728 > (-81.33804697925645 and 101.67263504078313) GeV
729 > (-81.31010149752827 and 101.63770295381545) GeV
730 > (-81.28220395508646 and 101.6028307919979) GeV
731 > (-81.25435420410977 and 101.56801837054778) GeV
732 > (-81.22655209743432 and 101.53326550550425) GeV
733 > (-81.19879748854987 and 101.49857201372363) GeV
734 > (-81.17109023159607 and 101.46393771287451) GeV
735 > (-81.14343018135868 and 101.42936242143331) GeV
736 > (-81.11581719326585 and 101.39484595867921) GeV
737 > (-81.0882511233843 and 101.36038814468998) GeV

738 > (-81.06073182841581 and 101.32598880033702) GeV
739 > (-81.03325916569337 and 101.29164774728095) GeV
740 > (-81.00583299317773 and 101.25736480796691) GeV
741 > (-80.97845316945363 and 101.22313980562032) GeV
742 > (-80.95111955372637 and 101.1889725642422) GeV
743 > (-80.92383200581821 and 101.15486290860474) GeV
744 > (-80.89659038616463 and 101.12081066424693) GeV
745 > (-80.86939455581125 and 101.08681565747018) GeV
746 > (-80.84224437640995 and 101.05287771533405) GeV
747 > (-80.81513971021563 and 101.01899666565164) GeV
748 > (-80.78808042008268 and 100.98517233698573) GeV
749 > (-80.76106636946167 and 100.95140455864404) GeV
750 > (-80.73409742239588 and 100.9176931606755 4) GeV
751 > (-80.70717344351795 and 100.88403797386573) GeV
752 > (-80.68029429804653 and 100.85043882973274) GeV
753 > (-80.65345985178313 and 100.81689556052324) GeV
754 > (-80.62666997110856 and 100.78340799920812) GeV
755 > (-80.59992452297986 and 100.74997597947863) GeV
756 > (-80.573223374927 and 100.71659933574213) GeV
757 > (-80.54656639504962 and 100.68327790311818) GeV
758 > (-80.51995345201392 and 100.6500115174345) GeV
759 > (-80.49338441504938 and 100.61680001522299) GeV
760 > (-80.46685915394563 and 100.58364323371579) GeV
761 > (-80.44037753904946 and 100.55054101084143) GeV
762 > (-80.41393944126145 and 100.51749318522083) GeV
763 > (-80.38754473203302 and 100.48449959616346) GeV
764 > (-80.3611932833635 and 100.45156008366365) GeV
765 > (-80.33488496779675 and 100.41867448839648) GeV
766 > (-80.30861965841846 and 100.38584265171436) GeV
767 > (-80.28239722885296 and 100.35306441564293) GeV
768 > (-80.25621755326024 and 100.3203396228776) GeV
769 > (-80.23008050633308 and 100.28766811677971) GeV
770 > (-80.20398596329397 and 100.25504974137276) GeV
771 > (-80.17793379989239 and 100.22248434133898) GeV
772 > (-80.15192389240161 and 100.18997176201556) GeV
773 > (-80.12595611761614 and 100.15751184939093) GeV
774 > (-80.10003035284856 and 100.12510445010155) GeV
775 > (-80.07414647592692 and 100.09274941142789) GeV
776 > (-80.04830436519167 and 100.06044658129122) GeV
777 > (-80.02250389949312 and 100.02819580824993) GeV
778 > (-79.99674495818843 and 99.99599694149612) GeV
779 > (-79.97102742113894 and 99.96384983085223) GeV
780 > (-79.9453511687074 and 99.93175432676736) GeV
781 > (-79.91971608175527 and 99.89971028031404) GeV
782 > (-79.89412204163997 and 99.8677175431848) GeV
783 > (-79.8685689302121 and 99.83577596768878) GeV
784 > (-79.84305662981302 and 99.80388540674832) GeV
785 > (-79.81758502327182 and 99.7720457138958) GeV
786 > (-79.79215399390306 and 99.74025674327018) GeV
787 > (-79.76676342550383 and 99.70851834961378) GeV
788 > (-79.74141320235132 and 99.67683038826902) GeV
789 > (-79.71610320920024 and 99.64519271517517) GeV
790 > (-79.69083333128012 and 99.61360518686517) GeV
791 > (-79.66560345429279 and 99.58206766046239) GeV
792 > (-79.64041346441 and 99.55057999367745) GeV
793 > (-79.6152632482707 and 99.5191420448051) GeV
794 > (-79.5901526929786 and 99.48775367272106) GeV
795 > (-79.5650816860997 and 99.45641473687908) GeV

796 > (-79.54005011565987 and 99.42512509730747) GeV
797 > (-79.51505787014221 and 99.39388461460634) GeV
798 > (-79.4901048384849 and 99.36269314994455) GeV
799 > (-79.46519091007842 and 99.33155056505652) GeV
800 > (-79.44031597476351 and 99.30045672223933) GeV
801 > (-79.41547992282847 and 99.26941148434959) GeV
802 > (-79.39068264500698 and 99.23841471480077) GeV
803 > (-79.36592403247566 and 99.20746627755993) GeV
804 > (-79.34120397685172 and 99.1765660371449) GeV
805 > (-79.3165223701907 and 99.14571385862148) GeV
806 > (-79.29187910498405 and 99.11490960760031) GeV
807 > (-79.2672740741569 and 99.08415315023427) GeV
808 > (-79.24270717106575 and 99.05344435321541) GeV
809 > (-79.21817828949624 and 99.02278308377225) GeV
810 > (-79.19368732366084 and 98.99216920966683) GeV
811 > (-79.16923416819664 and 98.96160259919192) GeV
812 > (-79.14481871816314 and 98.9310831211686) GeV
813 > (-79.12044086903992 and 98.90061064494274) GeV
814 > (-79.0961005167247 and 98.87018504038312) GeV
815 > (-79.07179755753089 and 98.83980617787809) GeV
816 > (-79.04753188818555 and 98.80947392833302) GeV
817 > (-79.02330340582715 and 98.77918816316772) GeV
818 > (-78.99911200800365 and 98.74894875431372) GeV
819 > (-78.97495759267005 and 98.71875557421149) GeV
820 > (-78.95084005818649 and 98.68860849580788) GeV
821 > (-78.92675930331616 and 98.65850739255362) GeV
822 > (-78.90271522722308 and 98.62845213840046) GeV
823 > (-78.87870772947014 and 98.59844260779873) GeV
824 > (-78.85473671001697 and 98.56847867569478) GeV
825 > (-78.83080206921792 and 98.53856021752841) GeV
826 > (-78.80690370782008 and 98.50868710923017) GeV
827 > (-78.78304152696116 and 98.47885922721912) GeV
828 > (-78.75921542816752 and 98.44907644840004) GeV
829 > (-78.73542531335224 and 98.41933865016118) GeV
830 > (-78.71167108481296 and 98.38964571037152) GeV
831 > (-78.68795264523021 and 98.35999750737854) GeV
832 > (-78.6642698976651 and 98.33039392000566) GeV
833 > (-78.6406227455576 and 98.30083482754978) GeV
834 > (-78.61701109272461 and 98.27132010977898) GeV
835 > (-78.59343484335793 and 98.24184964693002) GeV
836 > (-78.56989390202239 and 98.2124233197059) GeV
837 > (-78.546388173654 and 98.18304100927378) GeV
838 > (-78.522917563558 and 98.15370259726218) GeV
839 > (-78.49948197740703 and 98.12440796575898) GeV
840 > (-78.47608132123926 and 98.095156997309) GeV
841 > (-78.45271550145654 and 98.06594957491164) GeV
842 > (-78.42938442482253 and 98.03678558201857) GeV
843 > (-78.4060879984609 and 98.00766490253156) GeV
844 > (-78.38282612985354 and 97.97858742080011) GeV
845 > (-78.35959872683873 and 97.9495530216193) GeV
846 > (-78.3364056976093 and 97.92056159022731) GeV
847 > (-78.31324695071095 and 97.89161301230347) GeV
848 > (-78.29012239504036 and 97.86270717396592) GeV
849 > (-78.26703193984359 and 97.83384396176939) GeV
850 > (-78.24397549471414 and 97.80502326270313) GeV
851 > (-78.22095296959132 and 97.77624496418844) GeV
852 > (-78.19796427475856 and 97.74750895407702) GeV
853 > (-78.17500932084164 and 97.71881512064827) GeV

854 > (-78.15208801880686 and 97.69016335260756) GeV
855 > (-78.12920027995965 and 97.66155353908393) GeV
856 > (-78.1063460159426 and 97.6329855696279) GeV
857 > (-78.0835251387339 and 97.60445933420972) GeV
858 > (-78.06073756064566 and 97.57597472321693) GeV
859 > (-78.03798319432227 and 97.54753162745232) GeV
860 > (-78.01526195273878 and 97.5191299381322) GeV
861 > (-77.99257374919915 and 97.49076954688395) GeV
862 > (-77.96991849733479 and 97.46245034574426) GeV
863 > (-77.94729611110276 and 97.43417222715702) GeV
864 > (-77.92470650478445 and 97.40593508397131) GeV
865 > (-77.90214959298355 and 97.37773880943942) GeV
866 > (-77.87962529062492 and 97.34958329721476) GeV
867 > (-77.85713351295271 and 97.32146844135018) GeV
868 > (-77.83467417552893 and 97.29339413629572) GeV
869 > (-77.81224719423187 and 97.26536027689674) GeV
870 > (-77.78985248525447 and 97.2373667583921) GeV
871 > (-77.76748996510298 and 97.20941347641208) GeV
872 > (-77.74515955059516 and 97.18150032697659) GeV
873 > (-77.72286115885899 and 97.15362720649318) GeV
874 > (-77.70059470733104 and 97.12579401175522) GeV
875 > (-77.67836011375503 and 97.09800063994003) GeV
876 > (-77.65615729618034 and 97.07024698860688) GeV
877 > (-77.63398617296039 and 97.04253295569534) GeV
878 > (-77.61184666275135 and 97.01485843952331) GeV
879 > (-77.58973868451064 and 96.9872233387852) GeV
880 > (-77.56766215749528 and 96.9596275525501) GeV
881 > (-77.54561700126078 and 96.93207098026004) GeV
882 > (-77.52360313565936 and 96.90455352172808) GeV
883 > (-77.50162048083877 and 96.87707507713664) GeV
884 > (-77.47966895724069 and 96.84963554703566) GeV
885 > (-77.4577484855995 and 96.82223483234083) GeV
886 > (-77.43585898694063 and 96.7948728343318) GeV
887 > (-77.41400038257946 and 96.76754945465052) GeV
888 > (-77.39217259411964 and 96.74026459529942) GeV
889 > (-77.37037554345189 and 96.7130181586398) GeV
890 > (-77.34860915275263 and 96.68581004738995) GeV
891 > (-77.32687334448248 and 96.65864016462358) GeV
892 > (-77.30516804138506 and 96.63150841376796) GeV
893 > (-77.2834931664855 and 96.60441469860257) GeV
894 > (-77.26184864308928 and 96.577358923257) GeV
895 > (-77.24023439478067 and 96.55034099220957) GeV
896 > (-77.21865034542158 and 96.52336081028565) GeV
897 > (-77.19709641915018 and 96.4964182826558) GeV
898 > (-77.17557254037959 and 96.4695133148344) GeV
899 > (-77.15407863379662 and 96.44264581267791) GeV
900 > (-77.13261462436036 and 96.41581568238318) GeV
901 > (-77.11118043730106 and 96.38902283048603) GeV
902 > (-77.0897759981187 and 96.3622671638594) GeV
903 > (-77.06840123258183 and 96.33554858971199) GeV
904 > (-77.04705606672626 and 96.30886701558657) GeV
905 > (-77.02574042685374 and 96.28222234935839) GeV
906 > (-77.00445423953083 and 96.25561449923367) GeV
907 > (-76.98319743158757 and 96.22904337374806) GeV
908 > (-76.96196993011635 and 96.20250888176497) GeV
909 > (-76.94077166247048 and 96.17601093247427) GeV
910 > (-76.91960255626319 and 96.14954943539051) GeV
911 > (-76.89846253936628 and 96.12312430035159) GeV

912 > (-76.87735153990901 and 96.09673543751713) GeV
913 > (-76.85626948627676 and 96.07038275736699) GeV
914 > (-76.83521630710996 and 96.04406617069988) GeV
915 > (-76.81419193130289 and 96.01778558863181) GeV
916 > (-76.79319628800252 and 95.99154092259451) GeV
917 > (-76.77222930660713 and 95.96533208433415) GeV
918 > (-76.7512909167655 and 95.93915898590976) GeV
919 > (-76.73038104837542 and 95.91302153969183) GeV
920 > (-76.70949963158276 and 95.88691965836085) GeV
921 > (-76.68864659678019 and 95.86085325490592) GeV
922 > (-76.6678218746061 and 95.83482224262322) GeV
923 > (-76.64702539594354 and 95.80882653511468) GeV
924 > (-76.62625709191886 and 95.7828660462866) GeV
925 > (-76.60551689390086 and 95.7569406903482) GeV
926 > (-76.58480473349948 and 95.73105038181014) GeV
927 > (-76.56412054256488 and 95.70519503548341) GeV
928 > (-76.54346425318609 and 95.67937456647752) GeV
929 > (-76.52283579769016 and 95.65358889019961) GeV
930 > (-76.50223510864093 and 95.62783792235278) GeV
931 > (-76.48166211883796 and 95.60212157893473) GeV
932 > (-76.46111676131551 and 95.57643977623667) GeV
933 > (-76.44059896934144 and 95.55079243084168) GeV
934 > (-76.42010867641612 and 95.52517945962356) GeV
935 > (-76.39964581627136 and 95.49960077974544) GeV
936 > (-76.37921032286945 and 95.4740563086586) GeV
937 > (-76.35880213040205 and 95.44854596410083) GeV
938 > (-76.33842117328915 and 95.42306966409555) GeV
939 > (-76.31806738617797 and 95.39762732695021) GeV
940 > (-76.2977407039421 and 95.37221887125509) GeV
941 > (-76.27744106168028 and 95.34684421588211) GeV
942 > (-76.25716839471559 and 95.32150327998347) GeV
943 > (-76.23692263859422 and 95.29619598299031) GeV
944 > (-76.2167037290846 and 95.27092224461164) GeV
945 > (-76.19651160217643 and 95.24568198483294) GeV
946 > (-76.17634619407954 and 95.22047512391504) GeV
947 > (-76.15620744122306 and 95.19530158239267) GeV
948 > (-76.13609528025427 and 95.17016128107352) GeV
949 > (-76.1160096480378 and 95.14505414103674) GeV
950 > (-76.09595048165451 and 95.11998008363193) GeV
951 > (-76.07591771840063 and 95.09493903047783) GeV
952 > (-76.05591129578669 and 95.06993090346116) GeV
953 > (-76.03593115153666 and 95.04495562473527) GeV
954 > (-76.01597722358694 and 95.02001311671928) GeV
955 > (-75.99604945008545 and 94.99510330209655) GeV
956 > (-75.97614776939066 and 94.97022610381369) GeV
957 > (-75.95627212007066 and 94.94538144507936) GeV
958 > (-75.93642244090223 and 94.92056924936308) GeV
959 > (-75.91659867086997 and 94.89578944039407) GeV
960 > (-75.89680074916522 and 94.87104194216013) GeV
961 > (-75.87702861518532 and 94.84632667890654) GeV
962 > (-75.85728220853264 and 94.82164357513473) GeV
963 > (-75.83756146901362 and 94.79699255560142) GeV
964 > (-75.81786633663796 and 94.77237354531725) GeV
965 > (-75.7981967516176 and 94.7477864695458) GeV
966 > (-75.778552654366 and 94.72323125380247) GeV
967 > (-75.75893398549705 and 94.69870782385323) GeV
968 > (-75.7393406858244 and 94.67421610571374) GeV
969 > (-75.71977269636041 and 94.64975602564813) GeV

```
970 > (-75.70022995831538 and 94.62532751016782) GeV
971 > (-75.68071241309664 and 94.60093048603068) GeV
972 > (-75.6612200023077 and 94.57656488023977) GeV
973 > (-75.64175266774738 and 94.55223062004217) GeV
974 > (-75.62231035140894 and 94.52792763292823) GeV
975 > (-75.60289299547932 and 94.50365584663015) GeV
976 > (-75.58350054233819 and 94.47941518912135) GeV
977 > (-75.5641329345572 and 94.45520558861494) GeV
978 > (-75.54479011489896 and 94.43102697356301) GeV
979 > (-75.52547202631652 and 94.40687927265546) GeV
980 > (-75.5061786119523 and 94.38276241481903) GeV
981 > (-75.4869098151373 and 94.35867632921621) GeV
982 > (-75.46766557939038 and 94.33462094524431) GeV
983 > (-75.44844584841732 and 94.31059619253425) GeV
984 > (-75.42925056611021 and 94.28660200094988) GeV
985 > (-75.41007967654635 and 94.26263830058656) GeV
986 > (-75.39093312398775 and 94.23870502177051) GeV
987 > (-75.37181085288009 and 94.21480209505765) GeV
988 > (-75.35271280785213 and 94.19092945123262) GeV
989 > (-75.33363893371474 and 94.16708702130788) GeV
990 > (-75.31458917546036 and 94.1432747365226) GeV
991 > (-75.29556347826184 and 94.11949252834175) GeV
992 > (-75.27656178747213 and 94.09574032845522) GeV
993 > (-75.25758404862309 and 94.07201806877669) GeV
994 > (-75.23863020742502 and 94.04832568144276) GeV
995 > (-75.21970020976572 and 94.02466309881203) GeV
996 > (-75.20079400170984 and 94.0010302534641) GeV
997 > (-75.18191152949801 and 93.9774270781986) GeV
998 > (-75.16305273954622 and 93.95385350603433) GeV
999 > (-75.14421757844504 and 93.93030947020827) GeV

[Program finished]
```

Or

Instead of :

```
from math import*
t = 1000
dP1 = [0]*(t+1)
dP2 = [0]*(t+1)
dP3 = [0]*(t+1)
dP4 = [0]*(t+1)
dP5 = [0]*(t+1)
dE = [0]*(t+1)
S = [0]*(t+1)
gM = sqrt(1393.63288587075750051828395 47884)
for k3 in range(-t,t + 1):

#[k3!= -1, k3!= -2, k3!= 0]
if k3!= 0 and k3!= -1 and k3!= -2:

dP1[k3]=-((2**(11/4)*gM*k3**3+5*2**(11/4)*gM*k3**2+3*2**(15/4)*gM*k3+2**(15/4)*gM*abs(k3)**(3/2)*abs(2*k3+2)**(3/4))/((k 3**4+4*k3**3+4*k3**2)*abs(k3+1)**(3/2))

#imdP1[k3] = 0
dP2[k3]=((2**(27/4)*gM*k3+2**(27/4)*gM)*sqrt(abs(k3))*abs(2*k3+2)**(3/4))/((k3**4+8*k3**3+24*k3**2+32*k3+16)*sqrt(abs(k3+1)))
```

```
#imdP2[k3] = 0
dP3[k3]=-((5*2**(11/4)*gM*k3**5+45*2**(11/4)*gM*k3**4+71*2**(15/4)*gM*k3**3+95*2**(15/4)*gM*k3**2+7*2**(27/4)*gM*k3+3*2**(23/4)*gM)*abs(k3)**(3/2)*abs(2*k3+2)**(3/4))/((k3**6+8*k3**5+24*k3**4+32*k3**3+16*k3**2)*abs(k3+1)**(3/2))

#imdP3[k3] = 0
dP4[k3]=-((5*2**(11/4)*gM*k3**5+45*2**(11/4)*gM*k3**4+71*2**(15/4)*gM*k3**3+95*2**(15/4)*gM*k3**2+7*2**(27/4)*gM*k3+3*2**(23/4)*gM)*abs(k3)**(3/2)*abs(2*k3+2)**(3/4))/((k3**6+8*k3**5+24*k3**4+32*k3**3+16*k3**2)*abs(k3+1)**(3/2))


#imdP4[k3] = 0
dP5[k3]= (k3**2*(45*gM*k3**4+315*gM*k3**3+594*gM*k3**2+444*gM*k3+120*gM)* sqrt(abs(k3))*abs(2*k3+2)**(7/4))/((k3+1)**2*(2**(1/4)*k3**6+3*2**(5/4)*k3**5+3*2**(9/4)*k3**4+2**(13/4)*k3**3)*sqrt(abs(k3+1)))

#imdP5[k3] = 0
#imdE[k3] = 0
dE[k3] = (2**(7/4)*gM*sqrt(abs(k3+1)))/(sqrt(abs(k3))*abs(2*k3+2)**(1/4))
 if k3 = = t:
     N = floor(dE[1]/dE[t])
for l1 in range(1,t-1 + 1):
    print(l1,'> (',dP1[l1],' and ',dP2[l1],') GeV')
for l2 in range(N,l1-1 + 1,-1):
  for q1 in range(1,N + 1):
    for q2 in range(1,N + 1):
        for q3 in range(1,N + 1):
           for q4 in range(1,N + 1):
              for q5 in range(1,N + 1):
if(dP1[l1]+dP2[l1]+dP3[l1]+dP4[l1]+dP5[l1])-(q1*dP1[l2]+q2*dP2[l2]+q3*dP3[l2]+q4*dP4[l2]+ q5*dP5[l2])>0 and (dP1[l1]+dP2[l1]+dP3[l1]+dP4[l1]+dP5[l1])-(q1*dP1[l2]+q2*dP2[l2]+q3*d  P3[l2]+q4*dP4[l2]+q5*dP5[l2])<(dP1[l1]+dP2[l1]+dP3[l1]+dP4[l1]+dP5[l1])-(dP1[l1+1]+dP2[l 1+1]+dP3[l1+1]+dP4[l1+1]+dP5[l1+1]) :
          print(l1,'> ~ {',q1,'+',q2,'+',q3,'+',q4,'+',q5,'}.',l2,'>')

S[l1] = S[l1]+q1+q2+q3+q4+q5
elif l2==l1-1 and q1==N and q2==N and q3==N and q4==N and q5==N:
print(S[l1])
→ ...
```

Then in C++ :

```
#include <iostream>
 #include <cmath>
using namespace std;

int main()
{
  const int t = 1000;
  double dP1[t+1]={}, dP2[t+1]={}, dP3[t+1]={}, dP4[t+1]={}, dP5[t+1]={}, dE[t+1]={}, S[t+1]={};
  double gM = sqrt(1393.632885870757500518283954788);
  int k3 = 1;
  int l1 = 1;
  int q1 = 1;
  int q2 = 1;
  int q3 = 1;
  int q4 = 1;
  int q5 = 1;

for(int k3 = -t; k3 <= t; k3++)
{
  if(k3!= 0 && k3!= -1 && k3!= -2)
```

```
   {
dP1[k3]= -((pow(2,11.0/4)*gM*pow(k3,3)+5*pow(2,11.0/4)*gM*pow(k3,2)+3*pow(2,15.0 /4)*gM*k3+pow(2,15.0/4)*gM)*pow
(abs(k3),3.0/2)*pow(abs(2*k3+2),3.0/4))/((pow(k3,4)+4*pow(k3,3)+4*pow(k3,2))*pow(abs(k3+1),3.0/2));

dP2[k3] = ((pow(2,27.0/4)*gM*k3+pow(2,27.0/4)*gM)*sqrt(abs(k3))*pow(abs(2*k3+2),3.0/ 4))/((pow(k3,4)+8*pow(k3,3)+24*p
ow(k3,2)+32*k3+16)*sqrt(abs(k3+1)));

dP3[k3]=-((5*pow(2,11.0/4)*gM*pow(k3,5)+45*pow(2,11.0/4)*gM*pow(k3,4)+71*pow(2, 15.0/4)*gM*pow(k3,3)+95*pow
(2,15.0/4)*gM*po w(k3,2)+7*pow(2,27.0/4)*gM*k3+3*pow(2, 23.0/4)*gM)*pow(abs(k3),3.0/2)*pow(abs(2*k3+2),3.0/4))/((po
w(k3,6)+8*pow(k3,5)+24*pow(k 3,4)+32*pow(k3,3)+16*pow(k3,2))*pow(abs(k3+1),3.0/2));

dP4[k3] = -((5*pow(2,11.0/4)*gM*pow(k3,5)+45*pow(2,11.0/4)*gM*pow(k3,4)+71*pow(2, 15.0/4)*gM*pow(k3,3)+95*pow
(2,15.0/4)*gM*pow (k3,2)+7*pow(2,27.0/4)*gM*k3+3*pow(2, 23.0/4)*gM)*pow(abs(k3),3.0/2)*pow(abs(2*k3+2),3.0/4))/((po
w(k3,6)+8*pow(k3,5)+24*pow(k3,4)+32*pow(k3,3)+16*pow(k3,2))*pow(abs(k3+1),3.0/2));

dE[k3] = (pow(2,7.0/4)*gM*sqrt(abs(k3+1)))/(sqrt(abs(k3))*pow(abs(2*k3+2),1.0/4));
    }
  }
int N = 0;
for(int l1 = 1; l1 <= t-1; l1++)
 {
   cout<< l1 << " > ("<< dP1[l1] <<" and "<< dP2[l1] <<") GeV"<< endl;
 }
if(k3 = = t)
{
 N = floor(dE[1]/dE[t]);
 }
for(int l2 = N; l2 >= l1-1; l2--)
{
 for(int q1 = 1; q1 <= N; q1++)
  {
   for(int q2 = 1; q2 <= N; q2++)
    {
     for(int q3 = 1; q3 <= N; q3++)
      {
       for(int q4 = 1; q4 <= N; q4++)
        {
         for(int q5 = 1; q5 <= N; q5++)
          {
            if((dP1[l1]+dP2[l1]+dP3[l1]+dP4[l1]+dP5[l1])-(q1*dP1[l2]+q2*dP2[l2]+q3*dP3[l2]+q 4*dP4[l2]+q5*dP5[l2]) > 0
&& (dP1[l1]+dP2[l1]+dP3[l1]+dP4[l1]+dP5[l1])-(q1*dP1[l2]+q2*d P2[l2]+q3*dP3[l2]+q4*dP4[l2]+q5*dP5[l2])<(dP1[l1]+d-
P2[l1]+dP3[l1]+dP4[l1]+dP5[l1])-(dP1[l 1+1]+dP2[l1+1]+dP3[l1+1]+dP4[l1+1]+dP5[l1+1]))
            {
    cout<< l1 << " > ~ {"<< q1 << "+"<< q2 << "+"<< q3 << "+"<< q4 << "+"<< q5 << "}. "<< l2
<<">"<< endl;
      S[l1] = S[l1] + q1 + q2 + q3 + q4 + q5;
      }
     }
    }
   }
  }
 }
}
 if(l2 == l1-1 && q1 == N && q2 == N && q3 == N && q4 == N && q5 == N)
{
 cout<< S[l1] << endl;
 }
}
return 0;
}
```

→ 1 > (-390.654 and 198.427) GeV
2 > (-343.916 and 147.393) GeV
3 > (-317.264 and 105.939) GeV
4 > (-298.242 and 77.9716) GeV
5 > (-283.553 and 59.0991) GeV
6 > (-271.7 and 46.0137) GeV
7 > (-261.845 and 36.6641) GeV
8 > (-253.467 and 29.7953) GeV
9 > (-246.218 and 24.6235) GeV
10 > (-239.857 and 20.645) GeV
11 > (-234.211 and 17.5266) GeV
12 > (-229.15 and 15.042) GeV
13 > (-224.575 and 13.0336) GeV
14 > (-220.41 and 11.3893) GeV
15 > (-216.595 and 10.0276) GeV
16 > (-213.08 and 8.88854) GeV
17 > (-209.826 and 7.92684) GeV
18 > (-206.802 and 7.10816) GeV
19 > (-203.979 and 6.40599) GeV
20 > (-201.336 and 5.79959) GeV
21 > (-198.852 and 5.27261) GeV
22 > (-196.512 and 4.812) GeV
23 > (-194.301 and 4.40724) GeV
24 > (-192.208 and 4.04983) GeV
25 > (-190.221 and 3.73277) GeV
26 > (-188.332 and 3.45031) GeV
27 > (-186.532 and 3.19769) GeV
28 > (-184.813 and 2.97092) GeV
29 > (-183.171 and 2.76664) GeV
30 > (-181.598 and 2.58203) GeV
31 > (-180.089 and 2.41469) GeV
32 > (-178.641 and 2.26256) GeV
33 > (-177.248 and 2.12389) GeV
34 > (-175.908 and 1.99716) GeV
35 > (-174.617 and 1.88107) GeV
36 > (-173.371 and 1.77447) GeV
37 > (-172.168 and 1.67637) GeV
38 > (-171.006 and 1.58592) GeV
39 > (-169.881 and 1.50235) GeV
40 > (-168.792 and 1.42499) GeV
41 > (-167.737 and 1.35325) GeV
42 > (-166.714 and 1.28662) GeV
43 > (-165.722 and 1.22461) GeV
44 > (-164.758 and 1.16684) GeV
45 > (-163.821 and 1.11291) GeV
46 > (-162.91 and 1.06252) GeV
47 > (-162.024 and 1.01535) GeV
48 > (-161.161 and 0.971147) GeV
49 > (-160.321 and 0.92967) GeV
50 > (-159.502 and 0.890704) GeV
51 > (-158.703 and 0.854053) GeV
52 > (-157.924 and 0.819541) GeV
53 > (-157.164 and 0.787007) GeV
54 > (-156.421 and 0.756306) GeV
55 > (-155.696 and 0.727304) GeV
56 > (-154.987 and 0.699882) GeV
57 > (-154.294 and 0.673928) GeV
58 > (-153.616 and 0.649342) GeV

59 > (-152.953 and 0.62603) GeV
60 > (-152.304 and 0.603907) GeV
61 > (-151.668 and 0.582897) GeV
62 > (-151.045 and 0.562925) GeV
63 > (-150.435 and 0.543927) GeV
64 > (-149.837 and 0.525842) GeV
65 > (-149.25 and 0.508612) GeV
66 > (-148.675 and 0.492186) GeV
67 > (-148.111 and 0.476515) GeV
68 > (-147.557 and 0.461554) GeV
69 > (-147.013 and 0.447263) GeV
70 > (-146.48 and 0.433602) GeV
71 > (-145.955 and 0.420535) GeV
72 > (-145.44 and 0.40803) GeV
73 > (-144.934 and 0.396055) GeV
74 > (-144.437 and 0.38458) GeV
75 > (-143.948 and 0.37358) GeV
76 > (-143.467 and 0.363029) GeV
77 > (-142.994 and 0.352902) GeV
78 > (-142.529 and 0.343179) GeV
79 > (-142.071 and 0.333837) GeV
80 > (-141.621 and 0.324859) GeV
81 > (-141.177 and 0.316225) GeV
82 > (-140.74 and 0.307919) GeV
83 > (-140.31 and 0.299924) GeV
84 > (-139.887 and 0.292226) GeV
85 > (-139.47 and 0.28481) GeV
86 > (-139.059 and 0.277662) GeV
87 > (-138.653 and 0.270771) GeV
88 > (-138.254 and 0.264124) GeV
89 > (-137.861 and 0.25771) GeV
90 > (-137.472 and 0.251519) GeV
91 > (-137.09 and 0.24554) GeV
92 > (-136.712 and 0.239765) GeV
93 > (-136.34 and 0.234183) GeV
94 > (-135.973 and 0.228788) GeV
95 > (-135.61 and 0.22357) GeV
96 > (-135.253 and 0.218522) GeV
97 > (-134.9 and 0.213637) GeV
98 > (-134.551 and 0.208909) GeV
99 > (-134.207 and 0.20433) GeV
100 > (-133.868 and 0.199894) GeV
101 > (-133.532 and 0.195597) GeV
102 > (-133.201 and 0.191431) GeV
103 > (-132.874 and 0.187393) GeV
104 > (-132.551 and 0.183476) GeV
105 > (-132.232 and 0.179676) GeV
106 > (-131.916 and 0.17599) GeV
107 > (-131.604 and 0.172411) GeV
108 > (-131.296 and 0.168936) GeV
109 > (-130.992 and 0.165562) GeV
110 > (-130.691 and 0.162285) GeV
111 > (-130.393 and 0.1591) GeV
112 > (-130.099 and 0.156004) GeV
113 > (-129.808 and 0.152995) GeV
114 > (-129.52 and 0.150069) GeV
115 > (-129.235 and 0.147223) GeV
116 > (-128.954 and 0.144455) GeV

117 > (-128.675 and 0.141761) GeV
118 > (-128.4 and 0.139139) GeV
119 > (-128.127 and 0.136586) GeV
120 > (-127.858 and 0.134101) GeV
121 > (-127.591 and 0.13168) GeV
122 > (-127.327 and 0.129322) GeV
123 > (-127.065 and 0.127025) GeV
124 > (-126.806 and 0.124786) GeV
125 > (-126.55 and 0.122603) GeV
126 > (-126.297 and 0.120476) GeV
127 > (-126.046 and 0.118401) GeV
128 > (-125.797 and 0.116377) GeV
129 > (-125.551 and 0.114403) GeV
130 > (-125.307 and 0.112477) GeV
131 > (-125.066 and 0.110598) GeV
132 > (-124.826 and 0.108763) GeV
133 > (-124.59 and 0.106972) GeV
134 > (-124.355 and 0.105224) GeV
135 > (-124.123 and 0.103516) GeV
136 > (-123.892 and 0.101848) GeV
137 > (-123.664 and 0.100219) GeV
138 > (-123.438 and 0.0986268) GeV
139 > (-123.214 and 0.0970711) GeV
140 > (-122.992 and 0.0955507) GeV
141 > (-122.772 and 0.0940643) GeV
142 > (-122.554 and 0.0926112) GeV
143 > (-122.338 and 0.0911902) GeV
144 > (-122.123 and 0.0898005) GeV
145 > (-121.911 and 0.0884413) GeV
146 > (-121.701 and 0.0871115) GeV
147 > (-121.492 and 0.0858104) GeV
148 > (-121.285 and 0.0845372) GeV
149 > (-121.08 and 0.0832911) GeV
150 > (-120.876 and 0.0820714) GeV
151 > (-120.674 and 0.0808773) GeV
152 > (-120.474 and 0.0797081) GeV
153 > (-120.276 and 0.0785632) GeV
154 > (-120.079 and 0.0774419) GeV
155 > (-119.883 and 0.0763435) GeV
156 > (-119.69 and 0.0752675) GeV
157 > (-119.498 and 0.0742133) GeV
158 > (-119.307 and 0.0731803) GeV
159 > (-119.118 and 0.0721679) GeV
160 > (-118.93 and 0.0711756) GeV
161 > (-118.744 and 0.070203) GeV
162 > (-118.559 and 0.0692494) GeV
163 > (-118.376 and 0.0683144) GeV
164 > (-118.194 and 0.0673975) GeV
165 > (-118.014 and 0.0664983) GeV
166 > (-117.835 and 0.0656163) GeV
167 > (-117.657 and 0.0647511) GeV
168 > (-117.481 and 0.0639023) GeV
169 > (-117.305 and 0.0630695) GeV
170 > (-117.132 and 0.0622522) GeV
171 > (-116.959 and 0.0614502) GeV
172 > (-116.788 and 0.060663) GeV
173 > (-116.618 and 0.0598903) GeV
174 > (-116.449 and 0.0591317) GeV

175 > (-116.281 and 0.058387) GeV
176 > (-116.115 and 0.0576557) GeV
177 > (-115.95 and 0.0569375) GeV
178 > (-115.786 and 0.0562322) GeV
179 > (-115.623 and 0.0555395) GeV
180 > (-115.461 and 0.054859) GeV
181 > (-115.301 and 0.0541905) GeV
182 > (-115.141 and 0.0535337) GeV
183 > (-114.983 and 0.0528884) GeV
184 > (-114.825 and 0.0522542) GeV
185 > (-114.669 and 0.0516309) GeV
186 > (-114.514 and 0.0510183) GeV
187 > (-114.36 and 0.0504162) GeV
188 > (-114.207 and 0.0498243) GeV
189 > (-114.055 and 0.0492423) GeV
190 > (-113.904 and 0.0486702) GeV
191 > (-113.753 and 0.0481075) GeV
192 > (-113.604 and 0.0475543) GeV
193 > (-113.456 and 0.0470101) GeV
194 > (-113.309 and 0.046475) GeV
195 > (-113.163 and 0.0459485) GeV
196 > (-113.017 and 0.0454307) GeV
197 > (-112.873 and 0.0449212) GeV
198 > (-112.73 and 0.04442) GeV
199 > (-112.587 and 0.0439267) GeV
200 > (-112.445 and 0.0434414) GeV
201 > (-112.305 and 0.0429638) GeV
202 > (-112.165 and 0.0424937) GeV
203 > (-112.026 and 0.042031) GeV
204 > (-111.888 and 0.0415755) GeV
205 > (-111.75 and 0.0411272) GeV
206 > (-111.614 and 0.0406858) GeV
207 > (-111.478 and 0.0402512) GeV
208 > (-111.343 and 0.0398232) GeV
209 > (-111.209 and 0.0394018) GeV
210 > (-111.076 and 0.0389869) GeV
211 > (-110.944 and 0.0385782) GeV
212 > (-110.812 and 0.0381757) GeV
213 > (-110.681 and 0.0377792) GeV
214 > (-110.551 and 0.0373886) GeV
215 > (-110.422 and 0.0370038) GeV
216 > (-110.293 and 0.0366248) GeV
217 > (-110.165 and 0.0362513) GeV
218 > (-110.038 and 0.0358833) GeV
219 > (-109.912 and 0.0355206) GeV
220 > (-109.786 and 0.0351632) GeV
221 > (-109.661 and 0.034811) GeV
222 > (-109.537 and 0.0344639) GeV
223 > (-109.414 and 0.0341218) GeV
224 > (-109.291 and 0.0337845) GeV
225 > (-109.169 and 0.033452) GeV
226 > (-109.047 and 0.0331243) GeV
227 > (-108.926 and 0.0328011) GeV
228 > (-108.806 and 0.0324825) GeV
229 > (-108.687 and 0.0321683) GeV
230 > (-108.568 and 0.0318586) GeV
231 > (-108.45 and 0.0315531) GeV
232 > (-108.332 and 0.0312518) GeV

233 > (-108.215 and 0.0309546) GeV
234 > (-108.099 and 0.0306615) GeV
235 > (-107.984 and 0.0303724) GeV
236 > (-107.869 and 0.0300873) GeV
237 > (-107.754 and 0.0298059) GeV
238 > (-107.64 and 0.0295284) GeV
239 > (-107.527 and 0.0292546) GeV
240 > (-107.415 and 0.0289844) GeV
241 > (-107.302 and 0.0287178) GeV
242 > (-107.191 and 0.0284548) GeV
243 > (-107.08 and 0.0281952) GeV
244 > (-106.97 and 0.027939) GeV
245 > (-106.86 and 0.0276862) GeV
246 > (-106.751 and 0.0274367) GeV
247 > (-106.642 and 0.0271904) GeV
248 > (-106.534 and 0.0269472) GeV
249 > (-106.427 and 0.0267072) GeV
250 > (-106.32 and 0.0264703) GeV
251 > (-106.213 and 0.0262364) GeV
252 > (-106.107 and 0.0260055) GeV
253 > (-106.002 and 0.0257775) GeV
254 > (-105.897 and 0.0255524) GeV
255 > (-105.793 and 0.0253301) GeV
256 > (-105.689 and 0.0251106) GeV
257 > (-105.586 and 0.0248938) GeV
258 > (-105.483 and 0.0246797) GeV
259 > (-105.38 and 0.0244683) GeV
260 > (-105.279 and 0.0242595) GeV
261 > (-105.177 and 0.0240532) GeV
262 > (-105.076 and 0.0238495) GeV
263 > (-104.976 and 0.0236482) GeV
264 > (-104.876 and 0.0234494) GeV
265 > (-104.777 and 0.023253) GeV
266 > (-104.678 and 0.023059) GeV
267 > (-104.579 and 0.0228673) GeV
268 > (-104.481 and 0.0226778) GeV
269 > (-104.384 and 0.0224907) GeV
270 > (-104.286 and 0.0223058) GeV
271 > (-104.19 and 0.022123) GeV
272 > (-104.094 and 0.0219424) GeV
273 > (-103.998 and 0.021764) GeV
274 > (-103.902 and 0.0215876) GeV
275 > (-103.808 and 0.0214133) GeV
276 > (-103.713 and 0.021241) GeV
277 > (-103.619 and 0.0210707) GeV
278 > (-103.525 and 0.0209023) GeV
279 > (-103.432 and 0.0207359) GeV
280 > (-103.339 and 0.0205714) GeV
281 > (-103.247 and 0.0204088) GeV
282 > (-103.155 and 0.0202481) GeV
283 > (-103.063 and 0.0200891) GeV
284 > (-102.972 and 0.019932) GeV
285 > (-102.882 and 0.0197766) GeV
286 > (-102.791 and 0.019623) GeV
287 > (-102.701 and 0.019471) GeV
288 > (-102.612 and 0.0193208) GeV
289 > (-102.523 and 0.0191723) GeV
290 > (-102.434 and 0.0190254) GeV

291 > (-102.345 and 0.0188801) GeV
292 > (-102.257 and 0.0187364) GeV
293 > (-102.17 and 0.0185943) GeV
294 > (-102.082 and 0.0184537) GeV
295 > (-101.996 and 0.0183147) GeV
296 > (-101.909 and 0.0181771) GeV
297 > (-101.823 and 0.0180411) GeV
298 > (-101.737 and 0.0179065) GeV
299 > (-101.652 and 0.0177734) GeV
300 > (-101.567 and 0.0176417) GeV
301 > (-101.482 and 0.0175114) GeV
302 > (-101.397 and 0.0173825) GeV
303 > (-101.313 and 0.0172549) GeV
304 > (-101.23 and 0.0171287) GeV
305 > (-101.146 and 0.0170039) GeV
306 > (-101.063 and 0.0168803) GeV
307 > (-100.981 and 0.016758) GeV
308 > (-100.898 and 0.016637) GeV
309 > (-100.816 and 0.0165173) GeV
310 > (-100.735 and 0.0163988) GeV
311 > (-100.653 and 0.0162815) GeV
312 > (-100.572 and 0.0161655) GeV
313 > (-100.492 and 0.0160506) GeV
314 > (-100.411 and 0.0159369) GeV
315 > (-100.331 and 0.0158244) GeV
316 > (-100.252 and 0.015713) GeV
317 > (-100.172 and 0.0156027) GeV
318 > (-100.093 and 0.0154936) GeV
319 > (-100.014 and 0.0153855) GeV
320 > (-99.936 and 0.0152785) GeV
321 > (-99.8579 and 0.0151726) GeV
322 > (-99.78 and 0.0150678) GeV
323 > (-99.7025 and 0.014964) GeV
324 > (-99.6252 and 0.0148613) GeV
325 > (-99.5483 and 0.0147595) GeV
326 > (-99.4716 and 0.0146588) GeV
327 > (-99.3953 and 0.014559) GeV
328 > (-99.3192 and 0.0144602) GeV
329 > (-99.2434 and 0.0143624) GeV
330 > (-99.1679 and 0.0142656) GeV
331 > (-99.0927 and 0.0141696) GeV
332 > (-99.0178 and 0.0140747) GeV
333 > (-98.9432 and 0.0139806) GeV
334 > (-98.8688 and 0.0138874) GeV
335 > (-98.7948 and 0.0137951) GeV
336 > (-98.721 and 0.0137037) GeV
337 > (-98.6474 and 0.0136132) GeV
338 > (-98.5742 and 0.0135236) GeV
339 > (-98.5012 and 0.0134348) GeV
340 > (-98.4285 and 0.0133468) GeV
341 > (-98.3561 and 0.0132597) GeV
342 > (-98.2839 and 0.0131733) GeV
343 > (-98.212 and 0.0130878) GeV
344 > (-98.1403 and 0.0130031) GeV
345 > (-98.0689 and 0.0129192) GeV
346 > (-97.9978 and 0.0128361) GeV
347 > (-97.9269 and 0.0127537) GeV
348 > (-97.8563 and 0.0126721) GeV

349 > (-97.7859 and 0.0125912) GeV
350 > (-97.7158 and 0.0125111) GeV
351 > (-97.646 and 0.0124317) GeV
352 > (-97.5764 and 0.0123531) GeV
353 > (-97.507 and 0.0122752) GeV
354 > (-97.4379 and 0.0121979) GeV
355 > (-97.369 and 0.0121214) GeV
356 > (-97.3004 and 0.0120456) GeV
357 > (-97.232 and 0.0119704) GeV
358 > (-97.1638 and 0.0118959) GeV
359 > (-97.0959 and 0.0118221) GeV
360 > (-97.0282 and 0.011749) GeV
361 > (-96.9608 and 0.0116765) GeV
362 > (-96.8936 and 0.0116046) GeV
363 > (-96.8266 and 0.0115334) GeV
364 > (-96.7599 and 0.0114628) GeV
365 > (-96.6933 and 0.0113929) GeV
366 > (-96.6271 and 0.0113235) GeV
367 > (-96.561 and 0.0112548) GeV
368 > (-96.4952 and 0.0111866) GeV
369 > (-96.4295 and 0.0111191) GeV
370 > (-96.3642 and 0.0110521) GeV
371 > (-96.299 and 0.0109857) GeV
372 > (-96.234 and 0.0109199) GeV
373 > (-96.1693 and 0.0108547) GeV
374 > (-96.1048 and 0.01079) GeV
375 > (-96.0405 and 0.0107259) GeV
376 > (-95.9764 and 0.0106623) GeV
377 > (-95.9125 and 0.0105993) GeV
378 > (-95.8489 and 0.0105368) GeV
379 > (-95.7854 and 0.0104749) GeV
380 > (-95.7222 and 0.0104134) GeV
381 > (-95.6592 and 0.0103525) GeV
382 > (-95.5963 and 0.0102921) GeV
383 > (-95.5337 and 0.0102322) GeV
384 > (-95.4713 and 0.0101728) GeV
385 > (-95.4091 and 0.0101139) GeV
386 > (-95.3471 and 0.0100555) GeV
387 > (-95.2853 and 0.0099976) GeV
388 > (-95.2237 and 0.00994017) GeV
389 > (-95.1623 and 0.0098832) GeV
390 > (-95.1011 and 0.00982671) GeV
391 > (-95.0401 and 0.00977068) GeV
392 > (-94.9792 and 0.00971512) GeV
393 > (-94.9186 and 0.00966) GeV
394 > (-94.8582 and 0.00960534) GeV
395 > (-94.7979 and 0.00955113) GeV
396 > (-94.7379 and 0.00949735) GeV
397 > (-94.678 and 0.00944402) GeV
398 > (-94.6184 and 0.00939111) GeV
399 > (-94.5589 and 0.00933863) GeV
400 > (-94.4996 and 0.00928657) GeV
401 > (-94.4405 and 0.00923493) GeV
402 > (-94.3815 and 0.00918371) GeV
403 > (-94.3228 and 0.00913289) GeV
404 > (-94.2642 and 0.00908248) GeV
405 > (-94.2059 and 0.00903247) GeV
406 > (-94.1477 and 0.00898286) GeV

407 > (-94.0896 and 0.00893364) GeV
408 > (-94.0318 and 0.00888481) GeV
409 > (-93.9741 and 0.00883636) GeV
410 > (-93.9166 and 0.00878829) GeV
411 > (-93.8593 and 0.0087406) GeV
412 > (-93.8022 and 0.00869329) GeV
413 > (-93.7452 and 0.00864634) GeV
414 > (-93.6884 and 0.00859976) GeV
415 > (-93.6318 and 0.00855354) GeV
416 > (-93.5754 and 0.00850767) GeV
417 > (-93.5191 and 0.00846216) GeV
418 > (-93.463 and 0.00841701) GeV
419 > (-93.407 and 0.0083722) GeV
420 > (-93.3513 and 0.00832773) GeV
421 > (-93.2956 and 0.0082836) GeV
422 > (-93.2402 and 0.00823981) GeV
423 > (-93.1849 and 0.00819636) GeV
424 > (-93.1298 and 0.00815323) GeV
425 > (-93.0749 and 0.00811043) GeV
426 > (-93.0201 and 0.00806796) GeV
427 > (-92.9654 and 0.00802581) GeV
428 > (-92.911 and 0.00798397) GeV
429 > (-92.8566 and 0.00794245) GeV
430 > (-92.8025 and 0.00790124) GeV
431 > (-92.7485 and 0.00786033) GeV
432 > (-92.6946 and 0.00781973) GeV
433 > (-92.641 and 0.00777944) GeV
434 > (-92.5874 and 0.00773944) GeV
435 > (-92.5341 and 0.00769974) GeV
436 > (-92.4808 and 0.00766033) GeV
437 > (-92.4278 and 0.00762122) GeV
438 > (-92.3749 and 0.00758239) GeV
439 > (-92.3221 and 0.00754385) GeV
440 > (-92.2695 and 0.00750559) GeV
441 > (-92.217 and 0.00746761) GeV
442 > (-92.1647 and 0.0074299) GeV
443 > (-92.1125 and 0.00739247) GeV
444 > (-92.0605 and 0.00735531) GeV
445 > (-92.0086 and 0.00731843) GeV
446 > (-91.9569 and 0.0072818) GeV
447 > (-91.9053 and 0.00724545) GeV
448 > (-91.8539 and 0.00720935) GeV
449 > (-91.8026 and 0.00717351) GeV
450 > (-91.7514 and 0.00713793) GeV
451 > (-91.7004 and 0.00710261) GeV
452 > (-91.6495 and 0.00706753) GeV
453 > (-91.5988 and 0.00703271) GeV
454 > (-91.5482 and 0.00699813) GeV
455 > (-91.4978 and 0.0069638) GeV
456 > (-91.4475 and 0.00692971) GeV
457 > (-91.3973 and 0.00689586) GeV
458 > (-91.3473 and 0.00686225) GeV
459 > (-91.2974 and 0.00682887) GeV
460 > (-91.2476 and 0.00679573) GeV
461 > (-91.198 and 0.00676282) GeV
462 > (-91.1485 and 0.00673015) GeV
463 > (-91.0991 and 0.00669769) GeV
464 > (-91.0499 and 0.00666547) GeV

465 > (-91.0008 and 0.00663347) GeV
466 > (-90.9518 and 0.00660169) GeV
467 > (-90.903 and 0.00657012) GeV
468 > (-90.8543 and 0.00653878) GeV
469 > (-90.8057 and 0.00650765) GeV
470 > (-90.7573 and 0.00647674) GeV
471 > (-90.709 and 0.00644603) GeV
472 > (-90.6608 and 0.00641554) GeV
473 > (-90.6128 and 0.00638525) GeV
474 > (-90.5648 and 0.00635518) GeV
475 > (-90.517 and 0.0063253) GeV
476 > (-90.4694 and 0.00629563) GeV
477 > (-90.4218 and 0.00626616) GeV
478 > (-90.3744 and 0.00623688) GeV
479 > (-90.3271 and 0.00620781) GeV
480 > (-90.2799 and 0.00617893) GeV
481 > (-90.2329 and 0.00615024) GeV
482 > (-90.1859 and 0.00612174) GeV
483 > (-90.1391 and 0.00609344) GeV
484 > (-90.0924 and 0.00606532) GeV
485 > (-90.0459 and 0.00603739) GeV
486 > (-89.9994 and 0.00600965) GeV
487 > (-89.9531 and 0.00598209) GeV
488 > (-89.9069 and 0.00595471) GeV
489 > (-89.8608 and 0.00592752) GeV
490 > (-89.8148 and 0.0059005) GeV
491 > (-89.7689 and 0.00587366) GeV
492 > (-89.7232 and 0.00584699) GeV
493 > (-89.6776 and 0.0058205) GeV
494 > (-89.6321 and 0.00579419) GeV
495 > (-89.5867 and 0.00576804) GeV
496 > (-89.5414 and 0.00574207) GeV
497 > (-89.4962 and 0.00571626) GeV
498 > (-89.4512 and 0.00569062) GeV
499 > (-89.4063 and 0.00566515) GeV
500 > (-89.3614 and 0.00563984) GeV
501 > (-89.3167 and 0.00561469) GeV
502 > (-89.2721 and 0.0055897) GeV
503 > (-89.2276 and 0.00556488) GeV
504 > (-89.1833 and 0.00554021) GeV
505 > (-89.139 and 0.00551571) GeV
506 > (-89.0948 and 0.00549135) GeV
507 > (-89.0508 and 0.00546716) GeV
508 > (-89.0068 and 0.00544312) GeV
509 > (-88.963 and 0.00541922) GeV
510 > (-88.9193 and 0.00539549) GeV
511 > (-88.8757 and 0.0053719) GeV
512 > (-88.8322 and 0.00534846) GeV
513 > (-88.7888 and 0.00532516) GeV
514 > (-88.7455 and 0.00530201) GeV
515 > (-88.7023 and 0.00527901) GeV
516 > (-88.6592 and 0.00525615) GeV
517 > (-88.6162 and 0.00523344) GeV
518 > (-88.5733 and 0.00521087) GeV
519 > (-88.5305 and 0.00518843) GeV
520 > (-88.4879 and 0.00516614) GeV
521 > (-88.4453 and 0.00514398) GeV
522 > (-88.4028 and 0.00512196) GeV

523 > (-88.3605 and 0.00510008) GeV
524 > (-88.3182 and 0.00507833) GeV
525 > (-88.276 and 0.00505672) GeV
526 > (-88.234 and 0.00503523) GeV
527 > (-88.192 and 0.00501388) GeV
528 > (-88.1501 and 0.00499266) GeV
529 > (-88.1084 and 0.00497157) GeV
530 > (-88.0667 and 0.00495061) GeV
531 > (-88.0251 and 0.00492978) GeV
532 > (-87.9837 and 0.00490907) GeV
533 > (-87.9423 and 0.00488848) GeV
534 > (-87.901 and 0.00486803) GeV
535 > (-87.8599 and 0.00484769) GeV
536 > (-87.8188 and 0.00482748) GeV
537 > (-87.7778 and 0.00480739) GeV
538 > (-87.7369 and 0.00478742) GeV
539 > (-87.6961 and 0.00476757) GeV
540 > (-87.6554 and 0.00474783) GeV
541 > (-87.6148 and 0.00472822) GeV
542 > (-87.5743 and 0.00470872) GeV
543 > (-87.5339 and 0.00468934) GeV
544 > (-87.4935 and 0.00467007) GeV
545 > (-87.4533 and 0.00465092) GeV
546 > (-87.4132 and 0.00463188) GeV
547 > (-87.3731 and 0.00461295) GeV
548 > (-87.3331 and 0.00459414) GeV
549 > (-87.2933 and 0.00457543) GeV
550 > (-87.2535 and 0.00455684) GeV
551 > (-87.2138 and 0.00453835) GeV
552 > (-87.1742 and 0.00451998) GeV
553 > (-87.1347 and 0.0045017) GeV
554 > (-87.0953 and 0.00448354) GeV
555 > (-87.056 and 0.00446548) GeV
556 > (-87.0167 and 0.00444753) GeV
557 > (-86.9776 and 0.00442968) GeV
558 > (-86.9385 and 0.00441193) GeV
559 > (-86.8995 and 0.00439429) GeV
560 > (-86.8607 and 0.00437675) GeV
561 > (-86.8219 and 0.00435931) GeV
562 > (-86.7831 and 0.00434197) GeV
563 > (-86.7445 and 0.00432472) GeV
564 > (-86.706 and 0.00430758) GeV
565 > (-86.6675 and 0.00429054) GeV
566 > (-86.6291 and 0.00427359) GeV
567 > (-86.5909 and 0.00425674) GeV
568 > (-86.5527 and 0.00423998) GeV
569 > (-86.5145 and 0.00422332) GeV
570 > (-86.4765 and 0.00420676) GeV
571 > (-86.4386 and 0.00419029) GeV
572 > (-86.4007 and 0.00417391) GeV
573 > (-86.3629 and 0.00415762) GeV
574 > (-86.3252 and 0.00414143) GeV
575 > (-86.2876 and 0.00412532) GeV
576 > (-86.25 and 0.00410931) GeV
577 > (-86.2126 and 0.00409339) GeV
578 > (-86.1752 and 0.00407755) GeV
579 > (-86.1379 and 0.0040618) GeV
580 > (-86.1007 and 0.00404614) GeV

581 > (-86.0636 and 0.00403057) GeV
582 > (-86.0265 and 0.00401509) GeV
583 > (-85.9895 and 0.00399969) GeV
584 > (-85.9526 and 0.00398437) GeV
585 > (-85.9158 and 0.00396914) GeV
586 > (-85.8791 and 0.003954) GeV
587 > (-85.8424 and 0.00393893) GeV
588 > (-85.8058 and 0.00392395) GeV
589 > (-85.7693 and 0.00390906) GeV
590 > (-85.7329 and 0.00389424) GeV
591 > (-85.6966 and 0.0038795) GeV
592 > (-85.6603 and 0.00386485) GeV
593 > (-85.6241 and 0.00385027) GeV
594 > (-85.588 and 0.00383578) GeV
595 > (-85.5519 and 0.00382136) GeV
596 > (-85.516 and 0.00380702) GeV
597 > (-85.4801 and 0.00379276) GeV
598 > (-85.4443 and 0.00377858) GeV
599 > (-85.4085 and 0.00376447) GeV
600 > (-85.3728 and 0.00375044) GeV
601 > (-85.3373 and 0.00373648) GeV
602 > (-85.3017 and 0.0037226) GeV
603 > (-85.2663 and 0.00370879) GeV
604 > (-85.2309 and 0.00369506) GeV
605 > (-85.1956 and 0.0036814) GeV
606 > (-85.1604 and 0.00366781) GeV
607 > (-85.1252 and 0.0036543) GeV
608 > (-85.0902 and 0.00364085) GeV
609 > (-85.0552 and 0.00362748) GeV
610 > (-85.0202 and 0.00361418) GeV
611 > (-84.9854 and 0.00360095) GeV
612 > (-84.9506 and 0.00358779) GeV
613 > (-84.9158 and 0.0035747) GeV
614 > (-84.8812 and 0.00356168) GeV
615 > (-84.8466 and 0.00354872) GeV
616 > (-84.8121 and 0.00353584) GeV
617 > (-84.7777 and 0.00352302) GeV
618 > (-84.7433 and 0.00351027) GeV
619 > (-84.709 and 0.00349758) GeV
620 > (-84.6748 and 0.00348496) GeV
621 > (-84.6406 and 0.00347241) GeV
622 > (-84.6065 and 0.00345992) GeV
623 > (-84.5725 and 0.0034475) GeV
624 > (-84.5385 and 0.00343514) GeV
625 > (-84.5046 and 0.00342284) GeV
626 > (-84.4708 and 0.00341061) GeV
627 > (-84.4371 and 0.00339844) GeV
628 > (-84.4034 and 0.00338634) GeV
629 > (-84.3698 and 0.00337429) GeV
630 > (-84.3362 and 0.00336231) GeV
631 > (-84.3027 and 0.00335039) GeV
632 > (-84.2693 and 0.00333853) GeV
633 > (-84.236 and 0.00332673) GeV
634 > (-84.2027 and 0.00331499) GeV
635 > (-84.1694 and 0.00330331) GeV
636 > (-84.1363 and 0.00329169) GeV
637 > (-84.1032 and 0.00328013) GeV
638 > (-84.0702 and 0.00326863) GeV

639 > (-84.0372 and 0.00325719) GeV
640 > (-84.0043 and 0.0032458) GeV
641 > (-83.9715 and 0.00323447) GeV
642 > (-83.9387 and 0.0032232) GeV
643 > (-83.906 and 0.00321198) GeV
644 > (-83.8734 and 0.00320082) GeV
645 > (-83.8408 and 0.00318972) GeV
646 > (-83.8083 and 0.00317867) GeV
647 > (-83.7758 and 0.00316768) GeV
648 > (-83.7434 and 0.00315674) GeV
649 > (-83.7111 and 0.00314586) GeV
650 > (-83.6789 and 0.00313503) GeV
651 > (-83.6466 and 0.00312425) GeV
652 > (-83.6145 and 0.00311353) GeV
653 > (-83.5824 and 0.00310286) GeV
654 > (-83.5504 and 0.00309225) GeV
655 > (-83.5185 and 0.00308168) GeV
656 > (-83.4866 and 0.00307117) GeV
657 > (-83.4547 and 0.00306071) GeV
658 > (-83.423 and 0.0030503) GeV
659 > (-83.3912 and 0.00303995) GeV
660 > (-83.3596 and 0.00302964) GeV
661 > (-83.328 and 0.00301938) GeV
662 > (-83.2965 and 0.00300918) GeV
663 > (-83.265 and 0.00299902) GeV
664 > (-83.2336 and 0.00298891) GeV
665 > (-83.2022 and 0.00297886) GeV
666 > (-83.1709 and 0.00296885) GeV
667 > (-83.1397 and 0.00295889) GeV
668 > (-83.1085 and 0.00294897) GeV
669 > (-83.0774 and 0.00293911) GeV
670 > (-83.0463 and 0.00292929) GeV
671 > (-83.0153 and 0.00291952) GeV
672 > (-82.9844 and 0.0029098) GeV
673 > (-82.9535 and 0.00290012) GeV
674 > (-82.9227 and 0.00289049) GeV
675 > (-82.8919 and 0.00288091) GeV
676 > (-82.8612 and 0.00287137) GeV
677 > (-82.8305 and 0.00286188) GeV
678 > (-82.7999 and 0.00285243) GeV
679 > (-82.7694 and 0.00284303) GeV
680 > (-82.7389 and 0.00283367) GeV
681 > (-82.7084 and 0.00282436) GeV
682 > (-82.6781 and 0.00281509) GeV
683 > (-82.6477 and 0.00280587) GeV
684 > (-82.6175 and 0.00279669) GeV
685 > (-82.5873 and 0.00278755) GeV
686 > (-82.5571 and 0.00277846) GeV
687 > (-82.527 and 0.0027694) GeV
688 > (-82.497 and 0.00276039) GeV
689 > (-82.467 and 0.00275143) GeV
690 > (-82.437 and 0.0027425) GeV
691 > (-82.4071 and 0.00273362) GeV
692 > (-82.3773 and 0.00272478) GeV
693 > (-82.3475 and 0.00271598) GeV
694 > (-82.3178 and 0.00270722) GeV
695 > (-82.2881 and 0.0026985) GeV
696 > (-82.2585 and 0.00268982) GeV

697 > (-82.229 and 0.00268118) GeV
698 > (-82.1995 and 0.00267258) GeV
699 > (-82.17 and 0.00266402) GeV
700 > (-82.1406 and 0.00265551) GeV
701 > (-82.1113 and 0.00264703) GeV
702 > (-82.082 and 0.00263859) GeV
703 > (-82.0527 and 0.00263018) GeV
704 > (-82.0235 and 0.00262182) GeV
705 > (-81.9944 and 0.0026135) GeV
706 > (-81.9653 and 0.00260521) GeV
707 > (-81.9362 and 0.00259696) GeV
708 > (-81.9073 and 0.00258875) GeV
709 > (-81.8783 and 0.00258058) GeV
710 > (-81.8494 and 0.00257244) GeV
711 > (-81.8206 and 0.00256434) GeV
712 > (-81.7918 and 0.00255628) GeV
713 > (-81.7631 and 0.00254825) GeV
714 > (-81.7344 and 0.00254026) GeV
715 > (-81.7058 and 0.00253231) GeV
716 > (-81.6772 and 0.00252439) GeV
717 > (-81.6487 and 0.00251651) GeV
718 > (-81.6202 and 0.00250866) GeV
719 > (-81.5917 and 0.00250085) GeV
720 > (-81.5634 and 0.00249308) GeV
721 > (-81.535 and 0.00248534) GeV
722 > (-81.5067 and 0.00247763) GeV
723 > (-81.4785 and 0.00246996) GeV
724 > (-81.4503 and 0.00246232) GeV
725 > (-81.4222 and 0.00245472) GeV
726 > (-81.3941 and 0.00244715) GeV
727 > (-81.366 and 0.00243961) GeV
728 > (-81.338 and 0.00243211) GeV
729 > (-81.3101 and 0.00242464) GeV
730 > (-81.2822 and 0.0024172) GeV
731 > (-81.2544 and 0.0024098) GeV
732 > (-81.2266 and 0.00240243) GeV
733 > (-81.1988 and 0.00239509) GeV
734 > (-81.1711 and 0.00238778) GeV
735 > (-81.1434 and 0.00238051) GeV
736 > (-81.1158 and 0.00237327) GeV
737 > (-81.0883 and 0.00236606) GeV
738 > (-81.0607 and 0.00235888) GeV
739 > (-81.0333 and 0.00235173) GeV
740 > (-81.0058 and 0.00234462) GeV
741 > (-80.9785 and 0.00233753) GeV
742 > (-80.9511 and 0.00233048) GeV
743 > (-80.9238 and 0.00232346) GeV
744 > (-80.8966 and 0.00231646) GeV
745 > (-80.8694 and 0.0023095) GeV
746 > (-80.8422 and 0.00230257) GeV
747 > (-80.8151 and 0.00229567) GeV
748 > (-80.7881 and 0.0022888) GeV
749 > (-80.7611 and 0.00228195) GeV
750 > (-80.7341 and 0.00227514) GeV
751 > (-80.7072 and 0.00226836) GeV
752 > (-80.6803 and 0.0022616) GeV
753 > (-80.6535 and 0.00225488) GeV
754 > (-80.6267 and 0.00224818) GeV

755 > (-80.5999 and 0.00224151) GeV
756 > (-80.5732 and 0.00223487) GeV
757 > (-80.5466 and 0.00222826) GeV
758 > (-80.52 and 0.00222168) GeV
759 > (-80.4934 and 0.00221513) GeV
760 > (-80.4669 and 0.0022086) GeV
761 > (-80.4404 and 0.0022021) GeV
762 > (-80.4139 and 0.00219563) GeV
763 > (-80.3875 and 0.00218918) GeV
764 > (-80.3612 and 0.00218277) GeV
765 > (-80.3349 and 0.00217638) GeV
766 > (-80.3086 and 0.00217002) GeV
767 > (-80.2824 and 0.00216368) GeV
768 > (-80.2562 and 0.00215737) GeV
769 > (-80.2301 and 0.00215109) GeV
770 > (-80.204 and 0.00214483) GeV
771 > (-80.1779 and 0.0021386) GeV
772 > (-80.1519 and 0.0021324) GeV
773 > (-80.126 and 0.00212622) GeV
774 > (-80.1 and 0.00212007) GeV
775 > (-80.0741 and 0.00211394) GeV
776 > (-80.0483 and 0.00210784) GeV
777 > (-80.0225 and 0.00210177) GeV
778 > (-79.9967 and 0.00209572) GeV
779 > (-79.971 and 0.00208969) GeV
780 > (-79.9454 and 0.00208369) GeV
781 > (-79.9197 and 0.00207772) GeV
782 > (-79.8941 and 0.00207176) GeV
783 > (-79.8686 and 0.00206584) GeV
784 > (-79.8431 and 0.00205994) GeV
785 > (-79.8176 and 0.00205406) GeV
786 > (-79.7922 and 0.00204821) GeV
787 > (-79.7668 and 0.00204238) GeV
788 > (-79.7414 and 0.00203657) GeV
789 > (-79.7161 and 0.00203079) GeV
790 > (-79.6908 and 0.00202504) GeV
791 > (-79.6656 and 0.0020193) GeV
792 > (-79.6404 and 0.00201359) GeV
793 > (-79.6153 and 0.0020079) GeV
794 > (-79.5902 and 0.00200224) GeV
795 > (-79.5651 and 0.0019966) GeV
796 > (-79.5401 and 0.00199098) GeV
797 > (-79.5151 and 0.00198539) GeV
798 > (-79.4901 and 0.00197981) GeV
799 > (-79.4652 and 0.00197426) GeV
800 > (-79.4403 and 0.00196874) GeV
801 > (-79.4155 and 0.00196323) GeV
802 > (-79.3907 and 0.00195775) GeV
803 > (-79.3659 and 0.00195229) GeV
804 > (-79.3412 and 0.00194685) GeV
805 > (-79.3165 and 0.00194143) GeV
806 > (-79.2919 and 0.00193603) GeV
807 > (-79.2673 and 0.00193066) GeV
808 > (-79.2427 and 0.00192531) GeV
809 > (-79.2182 and 0.00191998) GeV
810 > (-79.1937 and 0.00191467) GeV
811 > (-79.1692 and 0.00190938) GeV
812 > (-79.1448 and 0.00190411) GeV

813 > (-79.1204 and 0.00189887) GeV
814 > (-79.0961 and 0.00189364) GeV
815 > (-79.0718 and 0.00188844) GeV
816 > (-79.0475 and 0.00188325) GeV
817 > (-79.0233 and 0.00187809) GeV
818 > (-78.9991 and 0.00187295) GeV
819 > (-78.975 and 0.00186782) GeV
820 > (-78.9508 and 0.00186272) GeV
821 > (-78.9268 and 0.00185764) GeV
822 > (-78.9027 and 0.00185258) GeV
823 > (-78.8787 and 0.00184753) GeV
824 > (-78.8547 and 0.00184251) GeV
825 > (-78.8308 and 0.00183751) GeV
826 > (-78.8069 and 0.00183252) GeV
827 > (-78.783 and 0.00182756) GeV
828 > (-78.7592 and 0.00182262) GeV
829 > (-78.7354 and 0.00181769) GeV
830 > (-78.7117 and 0.00181278) GeV
831 > (-78.688 and 0.0018079) GeV
832 > (-78.6643 and 0.00180303) GeV
833 > (-78.6406 and 0.00179818) GeV
834 > (-78.617 and 0.00179335) GeV
835 > (-78.5934 and 0.00178854) GeV
836 > (-78.5699 and 0.00178375) GeV
837 > (-78.5464 and 0.00177897) GeV
838 > (-78.5229 and 0.00177422) GeV
839 > (-78.4995 and 0.00176948) GeV
840 > (-78.4761 and 0.00176476) GeV
841 > (-78.4527 and 0.00176006) GeV
842 > (-78.4294 and 0.00175537) GeV
843 > (-78.4061 and 0.00175071) GeV
844 > (-78.3828 and 0.00174606) GeV
845 > (-78.3596 and 0.00174143) GeV
846 > (-78.3364 and 0.00173682) GeV
847 > (-78.3132 and 0.00173223) GeV
848 > (-78.2901 and 0.00172765) GeV
849 > (-78.267 and 0.00172309) GeV
850 > (-78.244 and 0.00171855) GeV
851 > (-78.221 and 0.00171403) GeV
852 > (-78.198 and 0.00170952) GeV
853 > (-78.175 and 0.00170503) GeV
854 > (-78.1521 and 0.00170055) GeV
855 > (-78.1292 and 0.0016961) GeV
856 > (-78.1063 and 0.00169166) GeV
857 > (-78.0835 and 0.00168724) GeV
858 > (-78.0607 and 0.00168283) GeV
859 > (-78.038 and 0.00167844) GeV
860 > (-78.0153 and 0.00167407) GeV
861 > (-77.9926 and 0.00166971) GeV
862 > (-77.9699 and 0.00166537) GeV
863 > (-77.9473 and 0.00166105) GeV
864 > (-77.9247 and 0.00165674) GeV
865 > (-77.9021 and 0.00165245) GeV
866 > (-77.8796 and 0.00164817) GeV
867 > (-77.8571 and 0.00164391) GeV
868 > (-77.8347 and 0.00163967) GeV
869 > (-77.8122 and 0.00163544) GeV
870 > (-77.7899 and 0.00163123) GeV

871 > (-77.7675 and 0.00162703) GeV
872 > (-77.7452 and 0.00162285) GeV
873 > (-77.7229 and 0.00161869) GeV
874 > (-77.7006 and 0.00161454) GeV
875 > (-77.6784 and 0.0016104) GeV
876 > (-77.6562 and 0.00160629) GeV
877 > (-77.634 and 0.00160218) GeV
878 > (-77.6118 and 0.00159809) GeV
879 > (-77.5897 and 0.00159402) GeV
880 > (-77.5677 and 0.00158996) GeV
881 > (-77.5456 and 0.00158592) GeV
882 > (-77.5236 and 0.00158189) GeV
883 > (-77.5016 and 0.00157787) GeV
884 > (-77.4797 and 0.00157387) GeV
885 > (-77.4577 and 0.00156989) GeV
886 > (-77.4359 and 0.00156592) GeV
887 > (-77.414 and 0.00156196) GeV
888 > (-77.3922 and 0.00155802) GeV
889 > (-77.3704 and 0.00155409) GeV
890 > (-77.3486 and 0.00155018) GeV
891 > (-77.3269 and 0.00154628) GeV
892 > (-77.3052 and 0.0015424) GeV
893 > (-77.2835 and 0.00153853) GeV
894 > (-77.2618 and 0.00153467) GeV
895 > (-77.2402 and 0.00153083) GeV
896 > (-77.2187 and 0.001527) GeV
897 > (-77.1971 and 0.00152318) GeV
898 > (-77.1756 and 0.00151938) GeV
899 > (-77.1541 and 0.00151559) GeV
900 > (-77.1326 and 0.00151182) GeV
901 > (-77.1112 and 0.00150806) GeV
902 > (-77.0898 and 0.00150431) GeV
903 > (-77.0684 and 0.00150058) GeV
904 > (-77.0471 and 0.00149686) GeV
905 > (-77.0257 and 0.00149315) GeV
906 > (-77.0045 and 0.00148946) GeV
907 > (-76.9832 and 0.00148578) GeV
908 > (-76.962 and 0.00148211) GeV
909 > (-76.9408 and 0.00147846) GeV
910 > (-76.9196 and 0.00147482) GeV
911 > (-76.8985 and 0.00147119) GeV
912 > (-76.8774 and 0.00146757) GeV
913 > (-76.8563 and 0.00146397) GeV
914 > (-76.8352 and 0.00146038) GeV
915 > (-76.8142 and 0.00145681) GeV
916 > (-76.7932 and 0.00145324) GeV
917 > (-76.7722 and 0.00144969) GeV
918 > (-76.7513 and 0.00144615) GeV
919 > (-76.7304 and 0.00144262) GeV
920 > (-76.7095 and 0.00143911) GeV
921 > (-76.6886 and 0.00143561) GeV
922 > (-76.6678 and 0.00143212) GeV
923 > (-76.647 and 0.00142864) GeV
924 > (-76.6263 and 0.00142518) GeV
925 > (-76.6055 and 0.00142172) GeV
926 > (-76.5848 and 0.00141828) GeV
927 > (-76.5641 and 0.00141485) GeV
928 > (-76.5435 and 0.00141144) GeV

929 > (-76.5228 and 0.00140803) GeV
930 > (-76.5022 and 0.00140464) GeV
931 > (-76.4817 and 0.00140126) GeV
932 > (-76.4611 and 0.00139789) GeV
933 > (-76.4406 and 0.00139453) GeV
934 > (-76.4201 and 0.00139118) GeV
935 > (-76.3996 and 0.00138785) GeV
936 > (-76.3792 and 0.00138452) GeV
937 > (-76.3588 and 0.00138121) GeV
938 > (-76.3384 and 0.00137791) GeV
939 > (-76.3181 and 0.00137462) GeV
940 > (-76.2977 and 0.00137134) GeV
941 > (-76.2774 and 0.00136808) GeV
942 > (-76.2572 and 0.00136482) GeV
943 > (-76.2369 and 0.00136158) GeV
944 > (-76.2167 and 0.00135835) GeV
945 > (-76.1965 and 0.00135512) GeV
946 > (-76.1763 and 0.00135191) GeV
947 > (-76.1562 and 0.00134871) GeV
948 > (-76.1361 and 0.00134552) GeV
949 > (-76.116 and 0.00134235) GeV
950 > (-76.096 and 0.00133918) GeV
951 > (-76.0759 and 0.00133602) GeV
952 > (-76.0559 and 0.00133288) GeV
953 > (-76.0359 and 0.00132974) GeV
954 > (-76.016 and 0.00132662) GeV
955 > (-75.996 and 0.0013235) GeV
956 > (-75.9761 and 0.0013204) GeV
957 > (-75.9563 and 0.00131731) GeV
958 > (-75.9364 and 0.00131423) GeV
959 > (-75.9166 and 0.00131115) GeV
960 > (-75.8968 and 0.00130809) GeV
961 > (-75.877 and 0.00130504) GeV
962 > (-75.8573 and 0.001302) GeV
963 > (-75.8376 and 0.00129897) GeV
964 > (-75.8179 and 0.00129595) GeV
965 > (-75.7982 and 0.00129294) GeV
966 > (-75.7786 and 0.00128994) GeV
967 > (-75.7589 and 0.00128695) GeV
968 > (-75.7393 and 0.00128397) GeV
969 > (-75.7198 and 0.001281) GeV
970 > (-75.7002 and 0.00127804) GeV
971 > (-75.6807 and 0.00127509) GeV
972 > (-75.6612 and 0.00127215) GeV
973 > (-75.6418 and 0.00126922) GeV
974 > (-75.6223 and 0.0012663) GeV
975 > (-75.6029 and 0.00126338) GeV
976 > (-75.5835 and 0.00126048) GeV
977 > (-75.5641 and 0.00125759) GeV
978 > (-75.5448 and 0.00125471) GeV
979 > (-75.5255 and 0.00125183) GeV
980 > (-75.5062 and 0.00124897) GeV
981 > (-75.4869 and 0.00124612) GeV
982 > (-75.4677 and 0.00124327) GeV
983 > (-75.4484 and 0.00124044) GeV
984 > (-75.4293 and 0.00123761) GeV
985 > (-75.4101 and 0.00123479) GeV
986 > (-75.3909 and 0.00123199) GeV

987 > (-75.3718 and 0.00122919) GeV
988 > (-75.3527 and 0.0012264) GeV
989 > (-75.3336 and 0.00122362) GeV
990 > (-75.3146 and 0.00122085) GeV
991 > (-75.2956 and 0.00121809) GeV
992 > (-75.2766 and 0.00121534) GeV
993 > (-75.2576 and 0.00121259) GeV
994 > (-75.2386 and 0.00120986) GeV
995 > (-75.2197 and 0.00120713) GeV
996 > (-75.2008 and 0.00120441) GeV
997 > (-75.1819 and 0.00120171) GeV
998 > (-75.1631 and 0.00119901) GeV
999 > (-75.1442 and 0.00119632) GeV

[Program finished]

4d origin of dE = 4.Sqrt{(gM^2)/sqrt(2)} ~ 125.5673374143954990316092809188 GeV ((gM^2) = 1393.6328858707575005182839547884 GeV²) (conf) [6,7].

Possibility of obtaining any mass of massive boson-massive graviton and masses lower than those of neutrinos and which can come from fluctuations of energies of the quantum vacuum [8,9].

1.5. A Less and Less Spin 0 Particles

(4) $Example\,1\ :\ dE = 4.\sqrt{\frac{gM^2}{\sqrt{2}}} \sim 125.5673374143954990316092809188\,GeV\ (gM^2 = 1393.6328858707575005182839547884\,GeV^2)$ [10-14].

The particle at exactly some 125 GeV
And spin
1+1+1-1 = 2 (As a frequent Space-Time reflect)
or
1-1+1-1 = 0
Fusion of 4 identical particles that are to disappear spontaneously or after one wave lengh: A single 4d-String appearing and disappearing spontaneously (Dt <=> 0) [10-12,14,15].

It si obtain a dark energy\massive gravitons that it is emitted to long distances because of higher values for Δk3 = dk3d

Dark matter\massive gravitons that it is emitted to short distances because of lesser values for Δk3 = dk3n

Algorithm to find Δk3 = dk3d
We set dE = h*C/L
L ~ 2.5 kpc = 7.7142*10^19 m
(1 pc = 3.086*10^13 kilomètres)
h = 6.62607004*10^(-34) m²kg/s
C = 299792458 m/s
dE = 6.62607004*10^(-34)*299792458/(7.7142*10^19) = 2.575050976344609*10^(-45) joules = 1.607219735509044*10^(-35) GeV

block(dE: 1.607219735509044*10^(-35),
gM : sqrt(1393.6328858707575005182839547884),
find_root(dE = 2^(7/4)*gM*sqrt((k3+1)/(k3*sqrt(2*k3+2))),k3,10,10^150))
→ k3 = 1.862840706286786*10^{147} (Obtaining of k3min for dark matter effect\massive graviton.)

Dark matter\massive gravitons that it is emitted to short distances because of lesser values for Δk3 = dk3n = k3 = 1.862840706286786*10^{147}
→ Dark energy\massive gravitons that it is emitted to long distances because of higher values for Δk3 = dk3d

1.5.1. Algorithm

from math import*

```
#An algorithm giving sets of massive bosons-dark energy: {dE[k3] = 2^(7/4)*gM*sqrt((k3d+1)/(k 3d*sqrt(2*k3d+2))), k3d = 1+n.dk3d}

#and sets of massive bosons-dark matter: {dE[k3] = 2^(7/4)*gM*sqrt((k3n+1)/(k3n*sqrt (2*k3n+2 ))), k3n = 1+n.dk3n}
gM = sqrt(1393.6328858707575005182839547884)
for q in range (1,300 + 1):
 Sidn = 0
Sidd = 0
tmax = 10*10**q
dk3n = 1.255314934818507*10**89
dk3d = 1.862840706286786*10**147
d0 =1.862840706286786*10**147
for t0 in range(1,10000 + 1) :
if d0 > 10**(-3)*1.862840706286786*10**147 :
     Sidn = 0
Sidd = 0
k3n = 1.862840706286786*10**147
k3d = 1.862840706286786*10**147
for t in range(1, tmax + 1) :

fiEk3n = 2**(7/4)*sqrt((k3n+1)/(k3n*sqrt(2*k3n+2)))
 fiEk3d = 2**(7/4)*sqrt((k3d+1)/(k3d*sqrt(2*k3d+2)))
Sidn = fiEk3n*dk3n + Sidn
Sidd = fiEk3d*dk3d + Sidd
 k3n = k3n+dk3n
k3d = k3d+dk3d
 if t = = tmax :
       if (Sidd/Sidn >= 3 and Sidd/Sidn < 4):
    k3n = k3n-dk3n
k3d = k3d-dk3d
dEd = 2**(7/4)*gM*sqrt((dk3d+1)/(dk3d*sqrt(2*dk3d+2)))
print("dk3n = ",dk3n, ", {dk3d = ",dk3d, ", dEd = ",dEd," GeV}")
dk3d = dk3d-d0
d0 = d0/10
elif Sidd/Sidn < 3:
 k3n = k3n-dk3n
k3d = k3d-dk3d
dk3d = dk3d+d0
elif Sidd/Sidn >= 4:
k3n = k3n-dk3n
k3d = k3d-dk3d
dk3d : dk3d-d0/10
→ ... ....
```

((q = 1, dk3n = 1.255314934818507*10**89, dk3d = 1.255314934818507*10**89, d0 = 1.255314934818507*10**89 and d0 > 10**(-3)*1.255314934818507*10**89)

→ dk3n=1.255314934818507*10^89, {dk3d=4.895728245792177*10^89, dEd=1.6072197 35509 044*10^(-35) GeV}
dk3n=1.255314934818507*10^89, {dk3d=3.891476297937371*10^89, dEd=1.60721973550904 4*10^(- 35) GeV}
dk3n=1.255314934818507*10^89, {dk3d=3.778497953803705*10^89, dEd=1.60721973550904 4*10^(- 35) GeV}
[Program finished])

Example with: {for dk in range (1,50 + 1):
 tmax = 1000
for t0 in range(1,1000 + 1) :}

→ dk3n=1.0 dk3d=5.0
dk3n=1.0 dk3d=float(22/5)=4.4

dk3n=1.0 dk3d=float(433/100)=4.33
dk3n=2.0 dk3d=9.0
dk3n=2.0 dk3d=float(87/10)=8.7
dk3n=2.0 dk3d=float(863/100)=8.63
dk3n=3.0 dk3d=13.0
dk3n=3.0 dk3d=13.0
dk3n=3.0 dk3d=float(1293/100)=12.93
dk3n=4.0 dk3d=18.0
dk3n=4.0 dk3d=float(173/10)=17.3
dk3n=4.0 dk3d=float(1723/100)=17.23
dk3n=5.0 dk3d=22.0
dk3n=5.0 dk3d=float(108/5)=21.6
dk3n=5.0 dk3d=float(1077/50)=21.54
dk3n=6.0 dk3d=26.0
dk3n=6.0 dk3d=float(259/10)=25.9
dk3n=6.0 dk3d=float(646/25)=25.84
dk3n=7.0 dk3d=31.0
dk3n=7.0 dk3d=float(151/5)=30.2
dk3n=7.0 dk3d=float(1507/50)=30.14
dk3n=8.0 dk3d=35.0
dk3n=8.0 dk3d=float(69/2)=34.5
dk3n=8.0 dk3d=float(861/25)=34.44
dk3n=9.0 dk3d=39.0
dk3n=9.0 dk3d=float(194/5)=38.8
dk3n=9.0 dk3d=float(1937/50)=38.74
dk3n=10.0 dk3d=44.0
dk3n=10.0 dk3d=float(431/10)=43.1
dk3n=10.0 dk3d=float(1076/25)=43.04
dk3n=11.0 dk3d=48.0
dk3n=11.0 dk3d=float(237/5)=47.4
dk3n=11.0 dk3d=float(2367/50)=47.34
dk3n=12.0 dk3d=float(52)=52.0
dk3n=12.0 dk3d=float(517/10)=51.7
dk3n=12.0 dk3d=float(1291/25)=51.64
dk3n=13.0 dk3d=float(56)=56.0
dk3n=13.0 dk3d=float(56)=56.0
dk3n=13.0 dk3d=float(1119/20)=55.95
dk3n=14.0 dk3d=float(61)=61.0
dk3n=14.0 dk3d=float(603/10)=60.3
dk3n=14.0 dk3d=float(241/4)=60.25
dk3n=15.0 dk3d=float(65)=65.0
dk3n=15.0 dk3d=float(323/5)=64.6
dk3n=15.0 dk3d=float(1291/20)=64.55
dk3n=16.0 dk3d=float(69)=69.0
dk3n=16.0 dk3d=float(689/10)=68.9
dk3n=16.0 dk3d=float(1377/20)=68.85
dk3n=17.0 dk3d=float(74)=74.0
dk3n=17.0 dk3d=float(366/5)=73.2
dk3n=17.0 dk3d=float(1463/20)=73.15
dk3n=18.0 dk3d=float(78)=78.0
dk3n=18.0 dk3d=float(155/2)=77.5
dk3n=18.0 dk3d=float(1549/20)=77.45
dk3n=19.0 dk3d=float(82)=82.0
dk3n=19.0 dk3d=float(409/5)=81.8
dk3n=19.0 dk3d=float(327/4)=81.75
dk3n=20.0 dk3d=float(87)=87.0
dk3n=20.0 dk3d=float(861/10)=86.1
dk3n=20.0 dk3d=float(4303/50)=86.06

dk3n=21.0 dk3d=float(91)=91.0
dk3n=21.0 dk3d=float(452/5)=90.4
dk3n=21.0 dk3d=float(2259/25)=90.36
dk3n=22.0 dk3d=float(95)=95.0
dk3n=22.0 dk3d=float(947/10)=94.7
dk3n=22.0 dk3d=float(4733/50)=94.66
dk3n=23.0 dk3d=float(99)=99.0
dk3n=23.0 dk3d=float(99)=99.0
dk3n=23.0 dk3d=float(2474/25)=98.96
dk3n=24.0 dk3d=float(104)=104.0
dk3n=24.0 dk3d=float(1033/10)=103.3
dk3n=24.0 dk3d=float(5163/50)=103.26
dk3n=25.0 dk3d=float(108)=108.0
dk3n=25.0 dk3d=float(538/5)=107.6
dk3n=25.0 dk3d=float(2689/25)=107.56
dk3n=26.0 dk3d=float(112)=112.0
dk3n=26.0 dk3d=float(1119/10)=111.9
dk3n=26.0 dk3d=float(5593/50)=111.86
dk3n=27.0 dk3d=float(117)=117.0
dk3n=27.0 dk3d=float(581/5)=116.2
dk3n=27.0 dk3d=float(2904/25)=116.16
dk3n=28.0 dk3d=float(121)=121.0
dk3n=28.0 dk3d=float(2412)=120.5
dk3n=28.0 dk3d=float(12047/100)=120.47
dk3n=29.0 dk3d=float(125)=125.0
dk3n=29.0 dk3d=float(624/5)=124.8
dk3n=29.0 dk3d=float(12477/100)=124.77
dk3n=30.0 dk3d=float(130)=130.0
dk3n=30.0 dk3d=float(1291/10)=129.1
dk3n=30.0 dk3d=float(12907/100)=129.07
dk3n=31.0 dk3d=float(134)=134.0
dk3n=31.0 dk3d=float(667/5)=133.4
dk3n=31.0 dk3d=float(13337/100)=133.37
dk3n=32.0 dk3d=float(138)=138.0
dk3n=32.0 dk3d=float(1377/10)=137.7
dk3n=32.0 dk3d=float(13767/100)=137.67
dk3n=33.0 dk3d=float(142)=142.0
dk3n=33.0 dk3d=float(142)=142.0
dk3n=33.0 dk3d=float(14197/100)=141.97
dk3n=34.0 dk3d=float(147)=147.0
dk3n=34.0 dk3d=float(1463/10)=146.3
dk3n=34.0 dk3d=float(14627/100)=146.27
dk3n=35.0 dk3d=float(151)=151.0
dk3n=35.0 dk3d=float(753/5)=150.6
dk3n=35.0 dk3d=float(15057/100)=150.57
dk3n=36.0 dk3d=float(155)=155.0
dk3n=36.0 dk3d=float(1549/10)=154.9
dk3n=36.0 dk3d=float(3872/25)=154.88
dk3n=37.0 dk3d=float(160)=160.0
dk3n=37.0 dk3d=float(796/5)=159.2
dk3n=37.0 dk3d=float(7959/50)=159.18
dk3n=38.0 dk3d=float(164)=164.0
dk3n=38.0 dk3d=float(327/2)=163.5
dk3n=38.0 dk3d=float(4087/25)=163.48
dk3n=39.0 dk3d=float(168)=168.0
dk3n=39.0 dk3d=float(839/5)=167.8
dk3n=39.0 dk3d=float(8389/50)=167.78
dk3n=40.0 dk3d=float(173)=173.0

```
dk3n=40.0 dk3d=float(1721/10)=172.1
dk3n=40.0 dk3d=float(4302/25)=172.08
dk3n=41.0 dk3d=float(177)=177.0
dk3n=41.0 dk3d=float(88/25)=176.4
dk3n=41.0 dk3d=float(8819/50)=176.38
dk3n=42.0 dk3d=float(181)=181.0
dk3n=42.0 dk3d=float(1807/10)=180.7
dk3n=42.0 dk3d=float(4517/25)=180.68
dk3n=43.0 dk3d=float(185)=185.0
dk3n=43.0 dk3d=float(185)=185.0
dk3n=43.0 dk3d=float(18499/100)=184.99
dk3n=44.0 dk3d=float(190)=190.0
dk3n=44.0 dk3d=float(1893/10)=189.3
dk3n=44.0 dk3d=float(18929/100)=189.29
dk3n=45.0 dk3d=float(194)=194.0
dk3n=45.0 dk3d=float(968/5)=193.6
dk3n=45.0 dk3d=float(19359/100)=193.59
dk3n=46.0 dk3d=float(198)=198.0
dk3n=46.0 dk3d=float(1979/10)=197.9
dk3n=46.0 dk3d=float(19789/100)=197.89
dk3n=47.0 dk3d=float(203)=203.0
dk3n=47.0 dk3d=float(1011/5)=202.2
dk3n=47.0 dk3d=float(20219/100)=202.19
dk3n=48.0 dk3d=float(207)=207.0
dk3n=48.0 dk3d=float(413/2)=206.5
dk3n=48.0 dk3d=float(20649/100)=206.49
dk3n=49.0 dk3d=float(211)=211.0
dk3n=49.0 dk3d=float(1054/5)=210.8
dk3n=49.0 dk3d=float(21079/100)=210.79
dk3n=50.0 dk3d=float(216)=216.0
dk3n=50.0 dk3d=float(2151/10)=215.1
dk3n=50.0 dk3d=float(21509/100)=215.09

[Program finished]
```

1.6. Trend Variations

$$dE_{k3} = 4.\sqrt(gM^2.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{(2+2.k3)}+2^2}))\ (4dprocess)$$

(1 bis) 4d superstring;

dE[k3] = 2^(7/4)*abs(gM)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))

$$idE_{k3} = 4.\frac{\sqrt{gM^2}.-i.\frac{-\frac{2^3}{2+2.k3}^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2}}{\sqrt{gM^2}} = 2^{\frac{7}{4}}.\sqrt{\frac{(k3+1)}{k3.\sqrt{(2.k3+2)}}}$$

idE[k3] = 2^(7/4)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))

or

dE2[k3] = (2^(7/2)*gM^2*k3+2^(7/2)*gM^2)/(k3*sqrt(2*k3+2)),

idE2[k3] = (2^(7/2)*k3+2^(7/2))/(k3*sqrt(2*k3+2));

$$Example\,1\ :\ dE = 4.\sqrt{\frac{gM^2}{\sqrt{2}}} \sim 125.567337414395499031609280918\,8\,GeV\ (gM^2 = 1393.6328858707575005182839547884\,GeV^2)$$

1.6.1. Algorithm

```
from math import*
#An algorithm giving sets of massive bosons-dark energy: {dE[k3] = 2^(7/4)*gM*sqrt((k3n+1)/ (k3n*sqrt(2*k3n+2))), k3n = 1+n.dk3n}

#and sets of massive bosons-dark matter: {dE[k3] = 2^(7/4)*gM*sqrt((k3d+1)/(k3d*sqrt(2*k3d+2 ))), k3d = 1+n.dk3d}

for dk in range (1,100 + 1):
Sidn = 0
Sidd = 0
tmax = 10000
dk3n = dk
dk3d = dk+1
d0 = 1
for t0 in range(1,10000 + 1):
if d0 > 10**(-3) :
   Sidn = 0
Sidd = 0
k3n = 1
k3d = 1

for t in range(1, tmax + 1) :
fiEk3n = 2**(7/4)*sqrt((k3n+1)/(k3n*sqrt(2*k3n+2)))
fiEk3d = 2**(7/4)*sqrt((k3d+1)/(k3d*sqrt(2*k3d+2)))
Sidn = fiEk3n*dk3n + Sidn
Sidd = fiEk3d*dk3d + Sidd
 k3n = k3n+dk3n
k3d = k3d+dk3d

if t = = tmax :
    if (Sidd/Sidn >= 3 and Sidd/Sidn < 4) :
        k3n = k3n-dk3n
 k3d = k3d-dk3d
print("dk3n = ",dk3n, "dk3d = ",dk3d)
 dk3d = dk3d-d0
d0 = d0/10
     elif Sidd/Sidn < 3 :
          k3n = k3n-dk3n
          k3d = k3d-dk3d
          dk3d = dk3d+d0
     elif Sidd/Sidn >= 4 :
k3n = k3n-dk3n
k3d = k3d-dk3d
dk3d : dk3d-d0/10
→ ... ....
```

Simple Variations to induce from bosons energy-bosons of approximately 10^(-47) kg (5.60958860892704*10^(-21) GeV).

$$\int dE_{k3}\, dk3 = \int 4.\sqrt{}(gM^2.-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{(2+2.k3)}+2^2})\, dk3$$

$$= 2^{\frac{11}{4}}.gM.\int \sqrt{}(\frac{1}{\sqrt{}(2.k3+2).(4-8/(2.k3+2))})\, dk3 \qquad (5)$$

We obtain

k3min > $1.255314934818507.10^{89}$

SdEmax = 5.60958860892704*10^(-21) GeV

(k3 = $1.255314934818507.10^{89}$ (From massive bosons-4d string energy generator)) [16]

dk is the number of identical massive bosons
tx is the number of bosons groups
k3i ≥ k3min (to obtain some less massive bosons but traveling larger distances)
Example of result : k3i=10^110,dk=18, tx=251 ⇔k3i=10^105,dk=1, tx=254

1.6.2. Fusion of 4 Identical Particles that are to Disappear Spontaneously or After One Wave Lengh

A single 4d-String appearing and disappearing spontaneously (Dt <=> 0) [10-12,14,15]

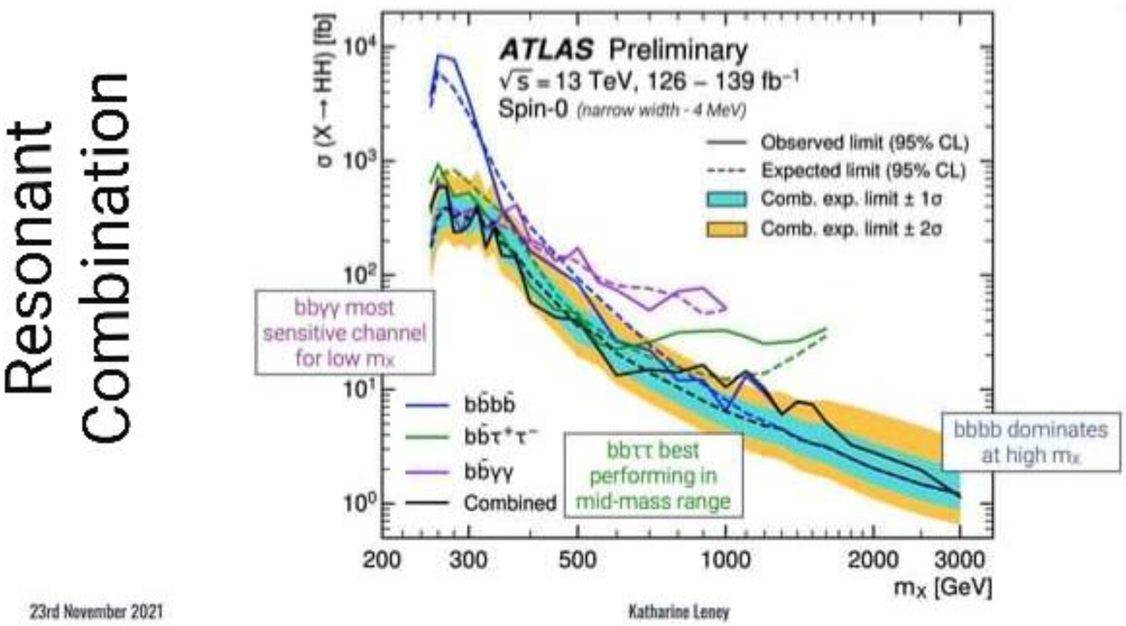

Figure 13: Observation of More than Spin 0 Events for Spin 2 Events

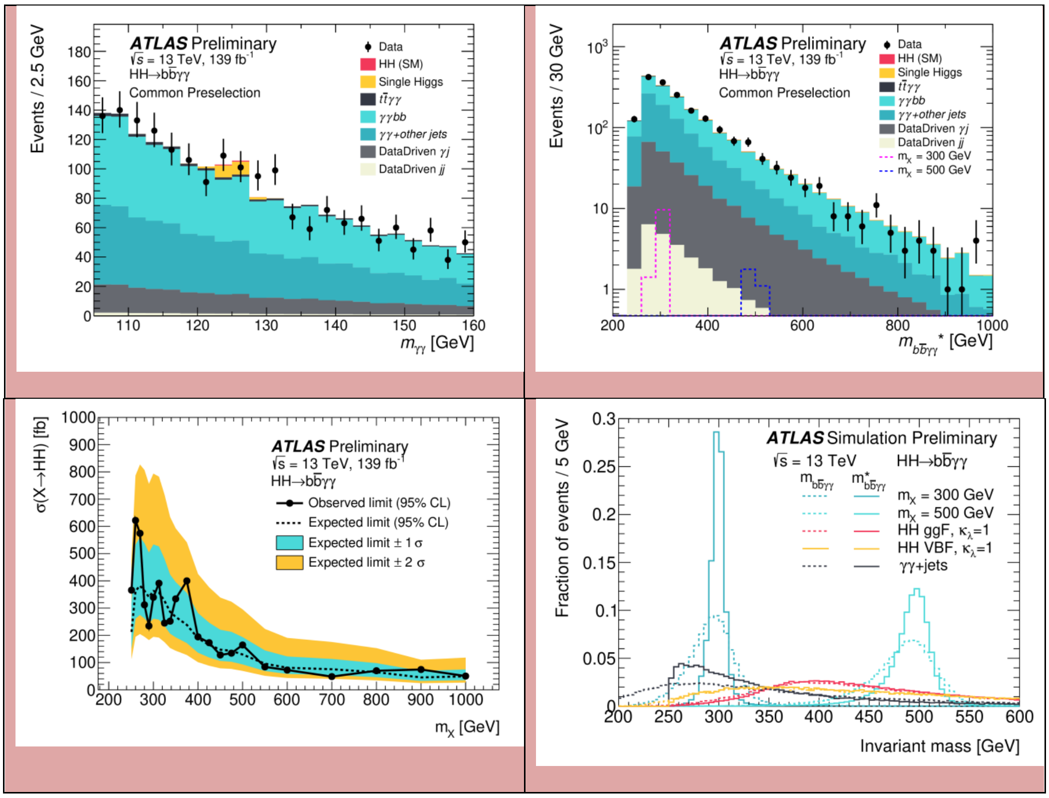

Figure 14: Decay Channel to Produce Spin 2 Events

As well as {π+,π-} → {{μ+,μ-},νμ, anti_νμ};
The π± mesons have a mass of 139.6 MeV/c2 and a mean lifetime of 2.6033*10^(-8) s. They decay due to the weak interaction. The primary decay mode of a pion, with a branching of 99.98770 ± 0,00004 %, is a leptonic decay into a muon and a muon neutrino:
π+ → μ+ + νμ
π− → μ− + anti_νμ

The second most common decay mode of a pion, with a branching of 0.0123 %, is also a leptonic decay into an electron and the corresponding electron antineutrino. This "electronic mode" was discovered at CERN in 1958:

π+ → e+ + νe
π− → e− + anti_νe

We have a {particle_spin +2, particle_spin-2} → 2.{b,anti_b}
or
{particle_spin +2, particle_spin-2} → {{b,anti_b}+.{τ,anti_τ}}

1.7. Instead of Photons Generated

The π0 has a slightly smaller mass than the charged pions (134.976 6 ± 0.000 6 MeV / c²) and a much shorter lifetime of 8.4 ± 0.6*10^(-17) s. At the end of this period, the π0 decays due to the electromagnetic interaction. The most common decay (98.798% of decays) gives two γ photons:

π0 → 2.γ

Or

In 1.198 ± 0.032% of cases, the decay products are a γ photon and an electron-positron pair:

π0 → γ +e-+e+

As

2.particle_spin 0 → {b,anti_b, 2.γ}.

1.7.1. From 4d Superstring a Particle at Some 125 GeV

$$dE_{k3} = 4.\sqrt{}(gM^2.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{(2+2.k3)}+2^2})) \ (4dprocess)$$

4d superstring,

$$idE_{k3} = 4.\frac{\sqrt{}gM^2.-i.\frac{-\frac{2^3}{2+2.k3}^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2}}{\sqrt{}gM^2} = 2^{\frac{7}{4}}.\sqrt{\frac{(k3+1)}{k3.\sqrt{}(2.k3+2)}}$$

or

$$dE2_{k3} = 16.gM^2.-i.\frac{-\frac{2^3}{2+2.k3}^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2},$$

$$idE2_{k3} = 16.\frac{gM^2.-i.\frac{-\frac{2^3}{2+2.k3}^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2}}{gM^2} = \frac{2^{\frac{7}{2}}.k3+2^{\frac{7}{2}}}{k3.\sqrt{2.k3+2}};$$

$$Example\,1\;:\;dE = 4.\sqrt{\frac{gM^2}{\sqrt{2}}} \sim 125.567337414395499031609280918\,8\,GeV\;(gM^2 = 1393.632885870757500518283954788\,4\,GeV^2)$$

[12-14]

The particle at exactly some 125 GeV
And spin
1+1+1-1 = 2 (As a frequent Space-Time reflect.)
or
1-1+1-1 = 0
With
dE = sqrt(dE2) sets:

$$dE_{k3} = 4.\sqrt{(gM^2.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{(2+2.k3)}+2^2}))}\;(4dprocess)$$

$$idE_{k3} = 4.\frac{\sqrt{gM^2.-i.\frac{-\frac{2^3}{2+2.k3}^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2}}}{\sqrt{gM^2}} = 2^{\frac{7}{4}}.\sqrt{\frac{(k3+1)}{k3.\sqrt{(2.k3+2)}}}$$

or

$$dE2_{k3} = 16.gM^2.-i.\frac{-\frac{2^3}{2+2.k3}^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2},$$

$$idE2_{k3} = 16.\frac{gM^2.-i.\frac{-\frac{2^3}{2+2.k3}^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2}}{gM^2} = \frac{2^{\frac{7}{2}}.k3+2^{\frac{7}{2}}}{k3.\sqrt{2.k3+2}};$$

$$Example\,1\;:\;dE = 4.\sqrt{\frac{gM^2}{\sqrt{2}}} \sim 125.567337414395499031609280918\,8\,GeV\;(gM^2 = 1393.632885870757500518283954788\,4\,GeV^2)$$

Examples: [11-14, 17-19]

dE = 4*sqrt((gM^2)*|Val[1,2]|), Val[1,2] = X/(X²+Y²) with k3=1 and Y=2 and [X=-sqrt(2)*%i, X=sqrt(2)*%i],
Val[2,2] = X/(X²+Y²) with k3 = 2 and Y = 2 and [X=-(2*%i)/sqrt(3),X=(2*%i)/sqrt(3)] → Val[2,2] = (sqrt(3)*%i)/4
→ dE = 98.26175177406498 GeV,
Val[3,2] = X/(X²+Y²) with k3 = 3 and Y = 2 and [X=-%i,X=%i] → Val[3,2] = %i/3
→ dE = 86.21315865135692 GeV,
....
Val[4,2] = X/(X²+Y²) with k3 = 4 and Y = 2 and [X=-(2*%i)/sqrt(5),X=(2*%i)/sqrt(5)] → Val[4,2] = (sqrt(5)*%i)/8
→ dE = 78.94628387056885 GeV,
Val[5,2] = X/(X²+Y²) with k3 = 5 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(3),X=(sqrt(2)*%i)/sqrt(3)] → Val[5,2] = (sqrt(3)*%i)/(5*sqrt(2))
→ dE = 73.90468953072723 GeV,
Val[6,2] = X/(X²+Y²) with k3 = 6 and Y = 2 and [X=-(2*%i)/sqrt(7),X=(2*%i)/sqrt(7)] → Val[6,2] = (sqrt(7)*%i)/12
→ dE = 70.11615158611494 GeV,
Val[7,2] = X/(X²+Y²) with k3 = 7 and Y = 2 and [X=-%i/sqrt(2),X=%i/sqrt(2)] → Val[7,2] = (sqrt(2)*%i)/7
→ dE = 67.11856508201555,
Val[8,2] = X/(X²+Y²) with k3 = 8 and Y = 2 and [X=-(2*%i)/3,X=(2*%i)/3] → Val[8,2] = (3*%i)/16
→ dE = 64.6598689885177,
Val[9,2] = X/(X²+Y²) with k3 = 9 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(5),X=(sqrt(2)*%i)/sqrt(5)] → Val[9,2] = (sqrt(5)*%i)/(9*sqrt(2))
→ dE = 62.58898832127002,
Val[10,2] = X/(X²+Y²) with k3 = 10 and Y = 2 and [X=-(2*%i)/sqrt(11),X=(2*%i)/sqrt(11)] → Val[10,2] = (sqrt(11)*%i)/20
→ dE = 60.8089294622656,

………....
Val[11,2] = $X/(X^2+Y^2)$ with k3 = 11 and Y = 2 and [X=-%i/sqrt(3),X=%i/sqrt(3)] → Val[11,2] = (sqrt(3)*%i)/11
→ dE = 59.25406579592956,
Val[12,2] = $X/(X^2+Y^2)$ with k3 = 12 and Y = 2 and [X=-(2*%i)/sqrt(13),X=(2*%i)/sqrt(13)] → Val[12,2] = (sqrt(13)*%i)/24
→ dE = 57.87811808262757,
Val[13,2] = $X/(X^2+Y^2)$ with k3 = 13 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(7),X=(sqrt(2)*%i)/sqrt(7)] → Val[13,2] = (sqrt(7)*%i)/(13*sqrt(2))
→ dE = 56.64733971277926,
Val[14,2] = $X/(X^2+Y^2)$ with k3 = 14 and Y = 2 and [X=-(2*%i)/sqrt(15),X=(2*%i)/sqrt(15)] → Val[14,2] = (sqrt(15)*%i)/28
→ dE = 55.53643312639486,
Val[15,2] = $X/(X^2+Y^2)$ with k3 = 15 and Y = 2 and [X=-%i/2,X=%i/2] → Val[15,2] = (2*%i)/15
→ dE = 54.52598912314765,
Val[16,2] = $X/(X^2+Y^2)$ with k3 = 16 and Y = 2 and [X=-(2*%i)/sqrt(17),X=(2*%i)/sqrt(17)] → Val[16,2] = (sqrt(17)*%i)/32
→ dE = 53.60081898525197,
Val[17,2] = $X/(X^2+Y^2)$ with k3 = 17 and Y = 2 and [X=-(sqrt(2)*%i)/3,X=(sqrt(2)*%i)/3] → Val[17,2] = (3*%i)/(17*sqrt(2))
→ dE = 52.74883253574095,
…......
Val[18,2] = $X/(X^2+Y^2)$ with k3 = 18 and Y = 2 and [X=-(2*%i)/sqrt(19),X=(2*%i)/sqrt(19)] → Val[18,2] = (sqrt(19)*%i)/36
→ dE = 51.9602622262889,
Val[19,2] = $X/(X^2+Y^2)$ with k3 = 19 and Y = 2 and [X=-%i/sqrt(5),X=%i/sqrt(5)] → Val[19,2] = (sqrt(5)*%i)/19
→ dE = 51.2271134499051,
Val[20,2] = $X/(X^2+Y^2)$ with k3 = 20 and Y = 2 and [X=-(2*%i)/sqrt(21),X=(2*%i)/sqrt(21)] → Val[20,2] = (sqrt(21)*%i)/40
→ dE = 50.54276680389589 GeV,
Val[21,2] = $X/(X^2+Y^2)$ with k3 = 21 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(11), X=(sqrt(2)*%i) /sqrt(11)] → Val[21,2] = (sqrt(11)*%i)/(21*sqrt(2))
→ dE = 49.90168487010055 GeV,
Val[22,2] = $X/(X^2+Y^2)$ with k3 = 22 and Y = 2 and [X=-(2*%i)/sqrt(23),X=(2*%i)/sqrt(23)] → Val[22,2] = (sqrt(23)*%i)/44
→ dE = dE = 49.29919241818454 GeV,
Val[23,2] = $X/(X^2+Y^2)$ with k3 = 23 and Y = 2 and [X=-%i/sqrt(6),X=%i/sqrt(6)] → Val[23,2] = (sqrt(6)*%i)/23
→ dE = 48.73130916893959 GeV,
Val[24,2] = $X/(X^2+Y^2)$ with k3 = 24 and Y = 2 and [X=-(2*%i)/5,X=(2*%i)/5] → Val[24,2] = (5*%i)/48
→ dE = 48.19462082485205 GeV,
……...

Val[25,2] = $X/(X^2+Y^2)$ with k3 = 25 and Y = 2 and [X=-(sqrt(2)*%i)/ sqrt(13),X=(sqrt(2)*%i)/ sqrt(13)] → Val[25,2] = (sqrt(13)*%i)/(25*sqrt(2))
→ dE = 47.68617839110907 GeV,
Val[26,2] = $X/(X^2+Y^2)$ with k3 = 26 and Y = 2 and [X=-(2*%i)/3^(3/2),X=(2*%i)/3^(3/2)] → Val[26,2] = (3^(3/2)*%i)/52
→ dE = 47.20341870370431 GeV,
Val[27,2] = $X/(X^2+Y^2)$ with k3 = 27 and Y = 2 and [X=-%i/sqrt(7),X=%i/sqrt(7)] → Val[27,2] = (sqrt(7)*%i)/27
→ dE = 46.74410105740996 GeV,
Val[28,2] = $X/(X^2+Y^2)$ with k3 = 28 and Y = 2 and [X=-(2*%i)/sqrt(29),X=(2*%i)/sqrt(29)] → Val[28,2] = (sqrt(29)*%i)/56
→ dE = 46.30625619890902 GeV,
Val[29,2] = $X/(X^2+Y^2)$ with k3 = 29 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(15),X=(sqrt(2)*%i)/sqrt(15)] → Val[29,2] = (sqrt(15)*%i)/(29*sqrt(2))
→ dE = 45.88814491850693 GeV,
Val[30,2] = $X/(X^2+Y^2)$ with k3 = 30 and Y = 2 and [X=-(2*%i)/sqrt(31),X=(2*%i)/sqrt(31)] → Val[30,2] = (sqrt(31)*%i)/60
→ dE = 45.4882241664636 GeV,
Val[31,2] = $X/(X^2+Y^2)$ with k3 = 31 and Y = 2 and [X=-%i/2^(3/2),X=%i/2^(3/2)] → Val[31,2] = (2^(3/2)*%i)/31
→ dE = 45.10511912202092 GeV,
....
Val[32,2] = $X/(X^2+Y^2)$ with k3 = 32 and Y = 2 and [X=-(2*%i)/sqrt(33),X=(2*%i)/sqrt(33)] → Val[32,2] = (sqrt(33)*%i)/64
→ dE = 44.73760001151199 GeV,
Val[33,2] = $X/(X^2+Y^2)$ with k3 = 33 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(17),X=(sqrt(2)*%i)/sqrt(17)] → Val[33,2] = (sqrt(17)*%i)/(33*sqrt(2))
→ dE = 44.38456274518938 GeV,
Val[34,2] = $X/(X^2+Y^2)$ with k3 = 34 and Y = 2 and [X=-(2*%i)/sqrt(35),X=(2*%i)/sqrt(35)] → Val[34,2] = (sqrt(5)*sqrt(7)*%i)/68
→ dE = 44.04501264725804 GeV,

Val[35,2] = X/(X²+Y²) with k3 = 35 and Y = 2 and [X=-%i/3,X=%i/3] → Val[35,2] = (3*%i)/35
→ dE = 43.71805070866736 GeV,
Val[36,2] = X/(X²+Y²) with k3 = 36 and Y = 2 and [X=-(2*%i)/sqrt(37),X=(2*%i)/sqrt(37)] → Val[36,2] = (sqrt(37)*%i)/72
→ dE = 43.40286191067236 GeV,
Val[37,2] = X/(X²+Y²) with k3 = 37 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(19),X=(sqrt(2)*%i)/sqrt(19)] → Val[37,2] = (sqrt(19)*%i)/(37*sqrt(2))
→ dE = 43.09870525843511 GeV,
Val[38,2] = X/(X²+Y²) with k3 = 38 and Y = 2 and [X=-(2*%i)/sqrt(39),X=(2*%i)/sqrt(39)] → Val[38,2] = (sqrt(39)*%i)/76
→ dE = 42.80490523481002 GeV,
……....
Val[39,2] = X/(X²+Y²) with k3 = 39 and Y = 2 and [X=-%i/sqrt(10),X=%i/sqrt(10)] → Val[39,2] = (sqrt(10)*%i)/39
→ dE = 42.52084443991102 GeV,
Val[40,2] = X/(X²+Y²) with k3 = 40 and Y = 2 and [X=-(2*%i)/sqrt(41),X=(2*%i)/sqrt(41)] → Val[40,2] = (sqrt(41)*%i)/80
→ dE = 42.2459572257572 GeV,
Val[41,2] = X/(X²+Y²) with k3 = 41 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(21),X=(sqrt(2)*%i)/sqrt (21)] → Val[41,2] = (sqrt(21)*%i)/(41*sqrt(2))
→ dE = 41.9797241699585 GeV,
Val[42,2] = X/(X²+Y²) with k3 = 42 and Y = 2 and [X=-(2*%i)/sqrt(43),X=(2*%i)/sqrt(43)] → Val[42,2] = (sqrt(43)*%i)/84
→ dE = 41.72166726007706 GeV,
Val[43,2] = X/(X²+Y²) with k3 = 43 and Y = 2 and [X=-%i/sqrt(11),X=%i/sqrt(11)] → Val[43,2] = (sqrt(11)*%i)/43
→ dE = 41.47134568252444 GeV,
Val[44,2] = X/(X²+Y²) with k3 = 44 and Y = 2 and [X=-(2*%i)/(3*sqrt(5)),X=(2*%i)/(3*sqrt(5))] → Val[44,2] = (3*sqrt(5)*%i)/88
→ dE = 41.22835212780523 GeV,
Val[45,2] = X/(X²+Y²) with k3 = 45 and Y = 2 and [X=-(sqrt(2)*%i)/sqrt(23),X=(sqrt(2)*%i)/sqrt(23)] → Val[45,2] = (sqrt(23)*%i)/(45*sqrt(2))
→ dE = 40.99230953849354 GeV,
……...
Val[46,2] = X/(X²+Y²) with k3 = 46 and Y = 2 and [X=-(2*%i)/sqrt(47),X=(2*%i)/sqrt(47)] → Val[46,2] = (sqrt(47)*%i)/92
→ dE = 40.76286823822587 GeV,
Val[47,2] = X/(X²+Y²) with k3 = 47 and Y = 2 and [X=-%i/(2*sqrt(3)),X=%i/(2*sqrt(3))] → Val[47,2] = (2*sqrt(3)*%i)/47
→ dE = 40.53970338975132 GeV,
Val[48,2] = X/(X²+Y²) with k3 = 48 and Y = 2 and [X=-(2*%i)/7,X=(2*%i)/7] → Val[48,2] = (7*%i)/96
→ dE = 40.32251273812042 GeV,
Val[49,2] = X/(X²+Y²) with k3 = 49 and Y = 2 and [X=-(sqrt(2)*%i)/5,X=(sqrt(2)*%i)/5] → Val[49,2] = (5*%i)/(49*sqrt(2))
→ dE = 40.11101460174871 GeV,
Val[50,2] = X/(X²+Y²) with k3 = 50 and Y = 2 and [X=-(2*%i)/sqrt(51),X=(2*%i)/sqrt(51)] → Val[50,2] = (sqrt(51)*%i)/100
→ dE = 39.90494607962204 GeV,
Val[51,2] = X/(X²+Y²) with k3 = 51 and Y = 2 and [X=-%i/sqrt(13),X=%i/sqrt(13)] → Val[51,2] = (sqrt(13)*%i)/51
→ dE = 39.70406144752673 GeV,
Val[52,2] = X/(X²+Y²) with k3 = 52 and Y = 2 and [X=-(2*%i)/sqrt(53),X=(2*%i)/sqrt(53)] → Val[52,2] = (sqrt(53)*%i)/104
→ dE = 39.5081307200551 GeV,
…........

Obtaining of bosons, each having a mass of approximately 10^(-47) kg with k3 = 1.2553149348185 07.10[89] (From massive bosons-4d string energy generator) [16,18].

$$dE_{k3} = 4.\sqrt{}(gM^2.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{(2+2.k3)}+2^2}))\ (4d\ process);$$

$$dE_{k3} = 4.\sqrt{}(1393.6328858707575005182839547884.(-i.\frac{(-\frac{2^3}{(2+2.k3)})^{\frac{1}{2}}}{-\frac{2^3}{2+2.k3}+2^2}))$$

→ import cmath
```
for k3 in range(1, 6):
  dE = 4 * cmath.sqrt(1393.6328858707575005182839547884 * (-1j * (-2 ** 3 / (2 + 2 * k3)) ** (1/2) / (-2 ** 3 / (2+ 2 * k3) + 2 ** 2)))
```

```
  print("dE = ",dE," GeV")
dE=125.5673374143955 GeV (Exactlly a Higgs boson mass)
dE=98.26175177406499 GeV
dE=86.21315865135692 GeV
dE=78.94628387056885 GeV
dE=73.90468953072724 GeV;

from math import *
for k3 in range(1,6):
   dEk3 = (2**(11/4)*(1393.6328858707575005182839547884)**(1/2))/(abs(2*k3+2)**(1/4)*(a bs(8/(2*k3+2)- 4))**(1/2))
   print('dE[', k3, '] =', dEk3)
→ ...
```

Or

[dE[k3] = 4*sqrt(gM^2*(-%i*(-2^3/(2+2*k3))^(1/2)/(-2^3/(2+2*k3)+2^2)))],[k3]; [(4*k3*sq rt(2* k3+2)*dE[k3]^2)^2 = ((2^(11/2)*gM^2)+2^(11/2)*gM^2*k3)^2],[k3];

[k3 = -1,
k3 = -(8*gM^2*sqrt(dE[k3]^4+16*gM^4)-32*gM^4)/dE[k3]^4,
k3 = (8*gM^2*sqrt(dE[k3]^4+16*gM^4)+32*gM^4)/dE[k3]^4]

--dE[k3] = 5.60958860892704*10^(-21) GeV→ {k3 = - (8*(1393.6328858707575005182839547 884)*sqrt((5.60958860892704*10^(- 21))^4+16*(1393.6328858707575005182839547884)^2)-32*(1393.6328858707575005182839547884)^2)/(5.60958860892704*10^(-21))^4,

k3 = (8*(1393.6328858707575005182839547884)*sqrt((5.60958860892704*10^(-21))^4+16*(1 393.632885870757500518283954 7884)^2)+32*(1393.6328858707575005182839547884)^2)/(5.60958860892704*10^(-21))^4};

{k3 = 0.0
k3 = 1.255314934818507*10^89}.

1.8. Introduction of Interactions with a 4d-String

1.8.1. Absorption of Massive Gravitons with an Open String

On N massive gravitons n will be instantly absorbed and (N!/((N-n)!.n!)) total absorptions (where massive gravitons are repeated) in (N!/((N-n)!.n!))/n distinct time intervals will be necessary for the absorption of N massive gravitons.

We obtain two different possible energy values for n massive gravitons in N.
In q.(N!/((N-n)!.n!)).dt0 times with q → infinite we have E = (1/2).q.(sEa + sEb). (6)
But would explain heterogeneities with dark matter [20-23].

We obtain from result : An open string absorbing a massive graviton
$H = \int d\sigma\, [X'^2(\sigma)/2 + (\partial X/\partial\sigma)^2/2] = m^2 = E\_A.E\_B/C^4 + \text{Potential Energy}^2/C^4$ (7)
For values (kmax,|kmin|) 100000, rich values of k3 and dEk3 are not obtained, but for values (kmax,|kmin|) ≥ 100000, we look for |real(values dEk3)| ≤ 4 GeV.

With dE²[k3] = {2^(7/2)*gM^2*k3/(k3*sqrt(2*k3+2)) → Potential energy
+2^(7/2)*gM^2/(k3*sqrt(2*k3+2)) →E_A.E_B Kinematic energies} (8)
We calculate E_Potential energy = gM.√(2^(7/2)*k3/(k3*sqrt(2*k3+2)))
And obtain E_Kinematic energy ≠ E_A or E_B; E_Kinematic energy = dEk3 - E_Potential energy
Interactions of two 4d string one of spin +2 and one of spin -2 with an open string. 4d string : Trefoil Knot [15].

With dE²[k3] =
{2^(7/2)*gM^2*k3/(k3*sqrt(2*k3+2)) → Potential energy
+2^(7/2)*gM^2/(k3*sqrt(2*k3+2)) →E_A.E_B Kinematic energies}
We calculate E_Potential energy = gM.√(2^(7/2)*k3/(k3*sqrt(2*k3+2))) (8)
And obtain E_Kinematic energy ≠ E_A or E_B; E_Kinematic energy = dEk3 - E_Potential energy
k3a → 4d massive graviton mass calculation.

k3b → a second 4d massive graviton mass calculation.

We obtain

ξ = (E_A.E_B/C⁴ + PEnergy²/C⁴)/(E_A + PEnergy) (9)

then

ξ = (PEnergy²+E_A.E_B)/(C⁴.PEnergy+C⁴.E_A) (10)

or

ξ = (E_A.E_B/C⁴ + PEnergy²/C⁴)/(E_B + PEnergy) (11)

then

ξ = (PEnergy²+E_A.E_B)/(C⁴.PEnergy+C⁴.E_B) (12)

with dE[k3] = 2^(7/4)*abs(gM)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))

and

E_A.E_B = dE_A[k3].dE_B[k3]

1.8.2. Algorithm in Python :

1.8.2.1. Example 1:

```
#n = 5, N = 8
import random
import math

N = 8
kmax = 100
kmin = 1
t = 0
k3 = [0]*(N+1)
dE = [0]*(N+1)

while t <= N:
   k_3 = -random.randint(0, kmax) + 2 * kmax

if k_3 > 0:
     t += 1
     k3[t-1] = k_3
    dE[t-1] = (1127186345 * math.sqrt((k_3 + 1)/(k_3 * math.sqrt(2*k_3+2))))/8976748
    print(k3[t-1],dE[t-1])

l = 0
sEa = 0
  for i1 in range(N):
    for i2 in range(i1 + 1, N):
     for i3 in range(i2 + 1, N):
       for i4 in range(i3 + 1, N):
         for i5 in range(i4 + 1, N):
          l += 1
          print(l, ">")
          print("k3[",i1,"] = ",k3[i1], ", dE[",i1,"] = ",dE[i1])
          print("k3[",i2,"] = ",k3[i2], ", dE[",i2,"] = ",dE[i2])
          print("k3[",i3,"] = ",k3[i3], ", dE[",i3,"] = ",dE[i3])
          print("k3[",i4,"] = ",k3[i4], ", dE[",i4,"] = ",dE[i4])
          print("k3[",i5,"] = ",k3[i5], ", dE[",i5,"] = ",dE[i5])
          sEa += dE[i1] + dE[i2] + dE[i3] + dE[i4] + dE[i5]
sEb = 0
for i1 in range(N):
   for i2 in range(i1 + 1, N):
     for i3 in range(i2 + 1, N):
         sEb += dE[i1] + dE[i2] + dE[i3]
print("sEa = ",sEa)
```

```
print("when")
print("sEb = ",sEb)
```

1.8.2.2. Example 2:

```
#n = 5, N = 8*4
import math
import random
import itertools

N = 8*4
kmax = 1000
kmin = 1
t = 0
k3 = [0]*(N+1)
dE = [0]*(N+1)
while t < N:
    k_3 = -random.randint(0, kmax) + 2*kmax
    if k_3 > 0:
        t += 1
        k3[t] = k_3
        dE[t] = (1127186345*math.sqrt((k_3+1)/(k_3*math.sqrt(2*k_3+2))))/8976748
        print("k3[", t, "] = ", k3[t], ", dE[", t, "] = ", dE[t], sep="")

l = 0
sEa = 0
for indices in itertools.combinations(range(1, N+1), 5):
l += 1
   print(l, ":")
for i in indices:
    print("k3[", i, "] = ", k3[i], ", dE[", i, "] = ", dE[i], sep="")
sEa += sum(dE[i] for i in indices)

sEb = 0
for indices in itertools.combinations(range(1, N+1), 27):
                sEb += sum(dE[i] for i in indices)
print("Result for sEa:", sEa)
print("when")
print("Result for sEb:", sEb)
```

1.9. Algorithm in C++

1.9.1. Example 1:

```
//n = 5, N = 8
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
      srand(time(0));
int N = 8;
int kmax = 100;
int kmin = 1;
int t = 0;
int k3[N+1];
double dE[N+1];

while (t <= N) {
```

```
        int k_3 = -rand() % kmax + 2 * kmax;
        if (k_3 > 0) {
        t++;
        k3[t] = k_3;
        dE[t] = (1127186345 * sqrt((k_3 + 1) / (k_3 * sqrt(2 * k_3 + 2)))) / 8976748;
                                    cout << "k3[" << t << "] = " << k3[t] << ", dE[" << t << "] = " << dE[t] << endl;
      }
    }

int l = 0;
double sEa = 0;

for (int i1 = 1; i1 <= N; i1++) {
  for (int i2 = 1; i2 <= N; i2++) {
    for (int i3 = 1; i3 <= N; i3++) {
      for (int i4 = 1; i4 <= N; i4++) {
        for (int i5 = 1; i5 <= N; i5++) {
             if (i1 < i2 && i2 < i3 && i3 < i4 && i4 < i5) {
l++;
cout << l << ": " << endl;
cout << "k3[" << i1 << "] = " << k3[i1] << ", dE[" << i1 << "] = " << dE[i1] << endl;
cout << "k3[" << i2 << "] = " << k3[i2] << ", dE[" << i2 << "] = " << dE[i2] << endl;
cout << "k3[" << i3 << "] = " << k3[i3] << ", dE[" << i3 << "] = " << dE[i3] << endl;
cout << "k3[" << i4 << "] = " << k3[i4] << ", dE[" << i4 << "] = " << dE[i4] << endl;
 cout << "k3[" << i5 << "] = " << k3[i5] << ", dE[" << i5 << "] = " << dE[i5] << endl;
sEa += dE[i1] + dE[i2] + dE[i3] + dE[i4] + dE[i5];
          }
        }
      }
    }
  }
}

double sEb = 0;
for (int i1 = 1; i1 <= N; i1++) {
   for (int i2 = 1; i2 <= N; i2++) {
      for (int i3 = 1; i3 <= N; i3++) {
          if (i1 < i2 && i2 < i3) {
          sEb += dE[i1] + dE[i2] + dE[i3];
        }
      }
    }
}

cout << "Result for sEa: " << sEa << endl;
 cout << "when" << endl;
cout << "Result for sEb: " << sEb << endl;

return 0;
}
```

→ k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849

k3[9] = 163, dE[9] = 29.5962
1:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
2:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
3:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[7] = 198, dE[7] = 28.1838
4:

k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[8] = 170, dE[8] = 29.2849
5:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
6:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
7:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[8] = 170, dE[8] = 29.2849
8:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
9:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
10:
k3[1] = 180, dE[1] = 28.8671

k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
11:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
12:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
13:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[8] = 170, dE[8] = 29.2849
14:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
15:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
16:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
17:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
18:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
19:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849

20:
k3[1] = 180, dE[1] = 28.8671
k3[2] = 129, dE[2] = 31.3914
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
21:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
22:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
23:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[8] = 170, dE[8] = 29.2849
24:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
25:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
26:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
27:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
28:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
29:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421

k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
30:
k3[1] = 180, dE[1] = 28.8671
k3[3] = 149, dE[3] = 30.2725
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
31:
k3[1] = 180, dE[1] = 28.8671
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
32:
k3[1] = 180, dE[1] = 28.8671
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
33:
k3[1] = 180, dE[1] = 28.8671
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
34:
k3[1] = 180, dE[1] = 28.8671
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
35:
k3[1] = 180, dE[1] = 28.8671
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
36:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
37:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
38:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[8] = 170, dE[8] = 29.2849
39:
k3[2] = 129, dE[2] = 31.3914

k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
40:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
41:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
42:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
43:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
44:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
45:
k3[2] = 129, dE[2] = 31.3914
k3[3] = 149, dE[3] = 30.2725
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
46:
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
47:
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
48:
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849

```
49:
k3[2] = 129, dE[2] = 31.3914
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
50:
k3[2] = 129, dE[2] = 31.3914
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
51:
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
52:
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[8] = 170, dE[8] = 29.2849
53:
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
54:
k3[3] = 149, dE[3] = 30.2725
k3[4] = 152, dE[4] = 30.1211
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
55:
k3[3] = 149, dE[3] = 30.2725
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
56:
k3[4] = 152, dE[4] = 30.1211
k3[5] = 162, dE[5] = 29.6421
k3[6] = 190, dE[6] = 28.4774
k3[7] = 198, dE[7] = 28.1838
k3[8] = 170, dE[8] = 29.2849
Result for sEa: 8268.41
when
Result for sEb: 4961.05
[Program finished]
```

Example 2 :

```
//n = 5, N = 8*4
#include <iostream>
 #include <cmath>
#include <cstdlib>
```

```
#include <ctime>
using namespace std;
int main() {
srand(time(0));
int N = 8*4;
int kmax = 1000;
 int kmin = 1;
int t = 0;
int k3[N+1];
double dE[N+1];

while (t <= N) {
     int k_3 = -rand() % kmax + 2 * kmax;
 if (k_3 > 0) {
        t++;
        k3[t] = k_3;
        dE[t] = (1127186345 * sqrt((k_3 + 1) / (k_3 * sqrt(2 * k_3 + 2)))) / 8976748;
                                        cout << "k3[" << t << "] = " << k3[t] << ", dE[" << t << "] = " << dE[t] << endl;
  }
}

int l = 0;
double sEa = 0;

for (int i1 = 1; i1 <= N; i1++) {
   for (int i2 = 1; i2 <= N; i2++) {
      for (int i3 = 1; i3 <= N; i3++) {
        for (int i4 = 1; i4 <= N; i4++) {
          for (int i5 = 1; i5 <= N; i5++) {
             if (i1 < i2 && i2 < i3 && i3 < i4 && i4 < i5) {
              l++;
              cout << l << ": " << endl;
              cout << "k3[" << i1 << "] = " << k3[i1] << ", dE[" << i1 << "] = " << dE[i1] << endl;
              cout << "k3[" << i2 << "] = " << k3[i2] << ", dE[" << i2 << "] = " << dE[i2] << endl;
              cout << "k3[" << i3 << "] = " << k3[i3] << ", dE[" << i3 << "] = " << dE[i3] << endl;
                                        cout << "k3[" << i4 << "] = " << k3[i4] << ", dE[" << i4 << "] = " << dE[i4] << endl;
              cout << "k3[" << i5 << "] = " << k3[i5] << ", dE[" << i5 << "] = " << dE[i5] << endl;
              sEa += dE[i1] + dE[i2] + dE[i3] + dE[i4] + dE[i5];
             }
          }
        }
      }
   }
}

double sEb = 0;
for (int i1 = 1; i1 <= N; i1++) {
   for (int i2 = 1; i2 <= N; i2++) {
      for (int i3 = 1; i3 <= N; i3++) {
      for (int i4 = 1; i4 <= N; i4++) {
      for (int i5 = 1; i5 <= N; i5++) {
      for (int i6 = 1; i6 <= N; i6++) {
      for (int i7 = 1; i7 <= N; i7++) {
      for (int i8 = 1; i8 <= N; i8++) {
for (int i9 = 1; i9 <= N; i9++) {
for (int i10 = 1; i10 <= N; i10++) {
for (int i11 = 1; i11 <= N; i11++) {
     for (int i12 = 1; i12 <= N; i12++) {
```

```
        for (int i13 = 1; i13 <= N; i13++) {
for (int i14 = 1; i14 <= N; i14++) {
for (int i15 = 1; i15 <= N; i15++) {
for (int i16 = 1; i16 <= N; i16++) {
for (int i17 = 1; i17 <= N; i17++) {
for (int i18 = 1; i18 <= N; i18++) {
for (int i19 = 1; i19 <= N; i19++) {
for (int i20 = 1; i20 <= N; i20++) {
for (int i21 = 1; i21 <= N; i21++) {
      for (int i22 = 1; i22 <= N; i22++) {
            for (int i23 = 1; i23 <= N; i23++) {
            for (int i24 = 1; i24 <= N; i24++) {
            for (int i25 = 1; i25 <= N; i25++) {
            for (int i26 = 1; i26 <= N; i26++) {
            for (int i27 = 1; i27 <= N; i27++) {
                  if (i1 < i2 && i2 < i3 && i3 < i4 && i4 < i5 && i5 < i6 && i6 < i7 && i7 < i8 && i8 < i9 && i9 < i10 &&i10 < i11
&& i11 < i12 && i12 < i13 && i13 < i14 && i14 < i15 && i15 < i16 && i16 < i17 && i17 < i18 && i18 < i19 && i19 < i20 &&
i20 < i21 && i21 < i22 && i22 < i23 && i23 < i24 && i24 < i25 && i25 < i26 && i26 < i27) {
                  sEb += dE[i1] + dE[i2] + dE[i3] + dE[i4] + dE[i5] + dE[i6] + dE[i7] + dE[i8] + dE[i9] + dE[i10] + dE[i11] + dE[i12] +
dE[i13] + dE[i14] + dE[i15] + dE[i16] + dE[i17] + dE[i18] + dE[i19] + dE[i20] + dE[i21] + dE[i22] + dE[i23] + dE[i24] + dE[i25]
+ dE[i26] + dE[i27];
                   }
               }
           }
                   }
               }
           }
                   }
               }
           }
                   }
               }
           }
                    }
                }
            }
                    }
                 }
            }
                     }
                  }
             }
                      }
                  }
              }

              }
 }
 }
}
      cout << "Result for sEa: " << sEa << endl; cout << "when" << endl;
      cout << "Result for sEb: " << sEb << endl;
       return 0;
     }
```

We have

An algorithm giving sets of massive bosons-dark energy: {dE[k3] = 2^(7/4)*gM*sqrt((k3d+1)/(k3 d*sqrt(2*k3d+2))), k3d = 1+n.dk3d} and sets of massive bosons-dark matter: {dE[k3] = 2^(7/4)*gM*sqrt((k3n+1)/(k3n*sqrt(2*k3n+2))), k3n= 1+n.dk3n}

gM = sqrt(1393.6328858707575005182839547884);
and
dk3n = 1.255314934818507.10^{89}
dk3d = 1.862840706286786.10^{147}

1.10. Search for dk3

dE^2[k3] =
{2^(7/2)*gM^2*k3/(k3*sqrt(2*k3+2)) → Potential energy
+2^(7/2)*gM^2/(k3*sqrt(2*k3+2)) →E_A.E_B Kinematic energies}

dE[k3] = 2^(7/4)*abs(gM)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))

E_A.E_B=dE_A[k3].dE_B[k3]=(2^(7/4)*abs(gM)*sqrt((k3a+1)/(k3a*sqrt(2*k3a+2))))*(2^(7/4)*abs(gM)*sqrt((k3b+1)/(k3b*sqrt(2*k3b+2))))=2^(7/2)*gM^2*sqrt((k3a+1)/(k3a*sqrt(2*k3a+2)))*sqrt((k3b+1)/(k3b*sqrt(2*k3b+2)))

[2^(7/2)*gM^2*sqrt((k3a+1)/(k3a*sqrt(2*k3a+2)))*sqrt((k3b+1)/(k3b*sqrt(2*k3b+2))) = 2^(7/2) *gM^2/(k3*sqrt(2*k3+2))], [k3]

→[(2^(7/2)*gM^2*sqrt((k3a+1)/(k3a*sqrt(2*k3a+2)))*sqrt((k3b+1)/(k3b*sqrt(2*k3b+2))))^2=(2^(7/2)*gM^2/(k3*sqrt(2*k3+2)))^2], [k3]

→[k3=-(%i*(3^(5/2)*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)^(2/3)-sqrt(3))

+9*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)^(2/3)+6*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)^(1/3)+1)/(18*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a*(2* k3b+2)+2*k3b+2))-1 /27)^(1/3)),

k3=(%i*(3^(5/2)*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)^(2/3)-sqrt(3))

-9*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)^(2/3)-6*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)^(1/3)-1)/(18*(((2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(k3a*k3b*(k3b*(k3a*(27*sqrt(2*k3a+2)*sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3^(3/2)*(k3a+1)*(k3b+1))+(k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2))/(2*(k3a* (2*k3b+2)+2*k3b+2))-1/27)^(1/3)),

k3=(9*(9*(2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(27*k3a^2*sqrt(2*k3a+2)*k3b^2*sqrt(2*k3b+2)+((-8*k3a^2)-8*k3a)*k3b^2+((-8*k3a^2)-8*k3a)*k3b)+3^(7/2)*k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2)+((-4*sqrt(3)*k3a)-4*sqrt(3))*k3b-4*sqrt(3)*k3a-4*sqrt(3))^(2/3)-3*((4*3^(7/2)*k3a+4*3^(7/2))*k3b+4*3^(7/2)*k3a+4*3^(7/2))^(1/3)*(9*(2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(27*k3a^2*sqrt(2*k3a+2)*k3b^2*sqrt(2*k3b+2)+((-8*k3a^2)-8*k3a)*k3b^2+((-8*k3a^2)-8*k3a)*k3b)+3^(7/2)*k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2)+((-4*sqrt(3)*k3a)-4*sqrt(3))*k3b-4*sqrt(3)*k3a-4*sqrt(3))^(1/3)+((4*3^(7/2)*k3a+4*3^(7/2))*k3b+4*3^(7/2)*k3a+4*3^(7/2))^(2/3))/(9*((4*3^(7/2)*k3a+4*3^(7/2))*k3b+4*3^(7/2)*k3a+4*3^(7/2))^(1/3)*(9*(2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(27*k3a^2*sqrt(2*k3a+2)*k3b^2*sqrt(2*k3b+2)+((-8*k3a^2)-8*k3a)*k3b^2+((-8*k3 a^2)-8*k3a)*k3b)+3^(7/2)*k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2)+((-4*sqrt(3)*k3a)-4*sqrt(3))* k3b- 4*sqrt(3)*k3a-4*sqrt(3))^(1/3))]

Example 1/3:
import cmath

```
kmin = -10
kmax = 10
for k3a in range(kmin, kmax+1):
  for k3b in range(kmin, kmax+1):
     if k3a*k3b*(4*3**(3/2)*(k3a+1)*(k3b+1))*(2*(k3a*(2*k3b+2)+2*k3b+2)) != 0:
if
18*(((2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(k3a*k3b*(k3b*(k3a*(27*cmath.sqrt(2*k3a+2)*cmath.sqrt(2*k-
3b+2)-8)-8)-8*k3a-8)))/(4*3**(3/2)*(k3a+1)*(k3b+1))+(k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2))/(2*(k3a*(2*k
3b+2)+2*k3b+2))-1/27)**(1/3) != 0:

                    k3 = -(cmath.sqrt(-1)*(3**(5/2)*(((2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(k3a*k3b*(k3b*(k3a*(27*c-
math.sqrt(2*k3a+2)*cmath.sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3**(3/2)*(k3a+1)*(k3b+1))+(k3a*cmath.sqrt(2*k3a+2)*k-
3b*cmath.sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)**(2/3)-cmath.sqrt(3))+9*(((2*k3a+2)**(1/4)*(2*k-
3b+2)**(1/4)*cmath.sqrt(k3a*k3b*(k3b*(k3a*(27*cmath.sqrt(2*k3a+2)*cmath.sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/
(4*3**(3/2)*(k3a+1)*(k3b+1))+(k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2))/(2*(k3a*(2*k3b+2)+2*k-
3b+2))-1/27)**(2/3)+6*(((2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(k3a*k3b*(k3b*(k3a*(27*cmath.sqrt(2*k3a+2)*c-
math.sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3**(3/2)*(k3a+1)*(k3b+1))+(k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2))/
(2*(k3a*(2*k3b+2)+2*k3b+2))-1/27)**(1/3)+1)/(18*(((2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(k3a*k3b*(k-
3b*(k3a*(27*cmath.sqrt(2*k3a+2)*cmath.sqrt(2*k3b+2)-8)-8)-8*k3a-8)))/(4*3**(3/2)*(k3a+1)*(k3b+1))+(k3a*cmath.s
qrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2))/(2*(k3a*(2*k 3b+2)+2*k3b+2))-1/27)**(1/3))
          print("k3 = ",k3, "k3a = ",k3a, "k3b = ",k3b)
→ ... ......
```

Example 3/3

```
import cmath
kmin = -10
max = 10

for k3a in range(kmin, kmax+1):
   for k3b in range(kmin, kmax+1):
if k3a*k3b != 0:
              if(4*3**(3/2)*(k3a+1)*(k3b+1))*(2*(k3a*(2*k3b+2)+2*k3b+2))*(9*((4*3**(7/2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k
3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.
sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.
sqrt(2*k3b+ 2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3)) != 0:

              k3 = (9*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k-
3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k-
3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(2/3)-3*((4*3**(7/2)*k
3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*c-
math.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*c-
math.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.
sqrt(3))**(1/3)+((4*3**(7/ 2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(2/3))/(9*((4*3**(7/2)*k3a+4*3**(7/2 ))*k3
b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt (27*k3a**2*cmath.sqrt(2*k3a+2)*k-
3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k-
3b*cmath.sqrt(2*k3b+2)+((-4 *cmath.sqrt(3)*k3a)- 4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3))
print("k3 = ",k3, "k3a = ",k3a, "k3b = ",k3b)

→ ... ....
```

And

```
import cmath
import math
kmin = -100000
kmax = 100000

for k3a in range(kmin, kmax+1):
  for k3b in range(kmin, kmax+1):
```

```
                    if k3a*k3b != 0:
                if(4*3**(3/2)*(k3a+1)*(k3b+1))*(2*(k3a*(2*k3b+2)+2*k3b+2))*(9*((4*3**(7/2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k
3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.
sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.
sqrt(2*k3b+ 2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3) )!= 0:

                  k3=(9*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k-
3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k-
3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(2/3)-3*((4*3**(7/2)*k
3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*c-
math.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*c-
math.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.
sqrt(3))**(1/3)+((4*3**(7/2 )*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(2/3))/(9*((4*3**(7/2)*k3a+4*3**(7/2) )*k3
b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k-
3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k-
3b*cmath.sqrt(2*k3b+2)+((-4* cmath.sqrt(3)*k3a)- 4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3))
          if (abs(math.floor(abs(k3))-abs(k3)) <= 0.00000001) or (abs(math.ceil(abs(k3))-abs(k3)) <= 0.00000001):
          print("k3 = ",k3, "k3a = ",k3a, "k3b = ",k3b)
→ ...
k3 = (4.000000004358974+0j) k3a = 1 k3b = 12801
k3 = (5.000000000435709+0j) k3a = 1 k3b = 45001
k3 = (5.000000003098156+0j) k3a = 2 k3b = 16876
k3 = (6.000000000462856+0j) k3a = 2 k3b = 47629
k3 = (6.000000001318018+0j) k3a = 3 k3b = 28225
k3 = (6.000000002665895+0j) k3a = 4 k3b = 19846
k3 = (7.000000000527919+0j) k3a = 4 k3b = 48021
k3 = (8.000000000332687+0j) k3a = 6 k3b = 64513
k3 = (7.000000001934029+0j) k3a = 7 k3b = 25089
k3 = (7.0000000026068765+0j) k3a = 8 k3b = 21610
k3 = (8.000000000636057+0j) k3a =  8 k3b = 46657
k3 = (8.000000000825255+0j) k3a = 9 k3b = 40961
k3 = (9.0000000002365+0j) k3a = 9 k3b = 81001
k3 = (9.000000000297902+0j) k3a = 10 k3b = 72172
k3 = (8.000000001543293+0j) k3a = 12 k3b = 29953
k3 = (9.000000000712818+0j) k3a = 15 k3b = 46657
k3 = (8.000000002852211+0j) k3a = 16 k3b = 22033
k3 = (8.00000000365744+0j) k3a = 18 k3b = 19457
k3 = (9.000000001048175+0j) k3a = 18 k3b = 38476
k3 = (10.000000000425901+0j) k3a = 20 k3b = 63526
k3 = (10.000000000519831+0j) k3a = 22 k3b = 57501
k3 = (10.000000000678323+0j) k3a = 25 k3b = 50337
k3 = (9.000000002443311+0j) k3a = 27 k3b = 25201
k3 = (9.000000003038068+0j) k3a = 30 k3b = 22600
k3 = (11.000000000435202+0j) k3a = 33 k3b = 65825
k3 = (12.000000000241666+0j) k3a = 39 k3b = 92161
k3 = (10.000000002172683+0j) k3a = 44 k3b = 28126
k3 = (11.000000000785192+0j) k3a = 44 k3b = 49006
……....
k3 = (18.00000000223377+0j) k3a = 1026 k3b = 36973
k3 = (17.999999990966415+0j) k3a = 1028 k3b = 36901
k3 = (20.0000000007486+0j) k3a = 1050 k3b = 67265
k3 = (21.000000000465235+0j) k3a = 1078 k3b = 87400
k3 = (16.00000000895825+0j) k3a = 1088 k3b = 17425
k3 = (16.000000000566697+0j) k3a = 1090 k3b = 17393
 k3 = (14.99999999213551+0j) k3a = 1092 k3b = 11880
k3 = (20.00000000097798+0j) k3a = 1200 k3b = 58850
k3 = (19.999999993160305+0j) k3a = 1202 k3b = 58752
…………
```

4d origin of dE = 4.Sqrt{(gM^2)/sqrt(2)} ~ 125.5673374143954990316092809188 GeV ((gM^2) = 1393.6328858707575005182839547884 GeV^2)

1.11. Initially

{dE : 1.607219735509044*10^(-35),
gM : sqrt(1393.6328858707575005182839547884),
root(dE = 2^(7/4)*gM*sqrt((k3+1)/(k3*sqrt(2*k3+2))),k3,10,10^150)}
→ k3 = 1.862840706286786*10^{147} (Obtaining of k3min for dark matter effect\massive graviton.)
With
dE[k3] = 2^(7/4)*abs(gM)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))

idE[k3] = 2^(7/4)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))
or
dE^2[k3] = (2^(7/2)*gM^2*k3+2^(7/2)*gM^2)/(k3*sqrt(2*k3+2)),
idE^2[k3] = (2^(7/2)*k3+2^(7/2))/(k3*sqrt(2*k3+2));
dE = 4*sqrt((gM^2)*|Val[1,2]|), Val[1,2] = X/(X^2+Y^2) with k3=1 and Y=2 and [X=-sqrt(2)*%i, X=sqrt(2)*%i]; dE = 4.Sqrt{(gM^2)/sqrt(2)} ~ 125.5673374143954990316092809188 GeV

Val[2,2] = X/(X^2+Y^2) with k3 = 2 and Y = 2 and [X=-(2*%i)/sqrt(3),X=(2*%i)/sqrt(3)] → Val[2,2] = (sqrt(3)*%i)/4
→ dE = 98.26175177406498 GeV,
Val[3,2] = X/(X^2+Y^2) with k3 = 3 and Y = 2 and [X=-%i,X=%i] → Val[3,2] = %i/3
→ dE = 86.21315865135692 GeV,
……..
Val[4,2] = X/(X^2+Y^2) with k3 = 4 and Y = 2 and [X=-(2*%i)/sqrt(5),X=(2*%i)/sqrt(5)] → Val[4,2] = (sqrt(5)*%i)/8
→ dE = 78.94628387056885 GeV,
………...

We obtain from result : An open string absorbing a massive graviton
$H = \int d\sigma\, [X'^2(\sigma)/2 + (\partial X/\partial \sigma)^2/2] = m^2 = E\_A.E\_B/C^4 + \text{Potential Energy}^2/C^4$
Thanks to the resolution of the Riemann hypothesis with the result-consequence a = ½ for n≡∞ :
{∫(1/(n+k)^a,n) = (n+k)^(1-a)/(1-a)
= -n^(1-a)/(1-a);
n+k = (-1)^(1/(1-a))*n;
{n+k = (-1)^(1/(1-a))*n,
k=-1}
→ [n = -1/((-1)^(1/(1-a))-1)]
--a = ½→ [n = -1/((-1)^(1/(1-(1/2)))-1) = +∞].}, we introduce:

k3 variations and melting sign for b values:
sn1 = s2 = sin(log(2)*b)

Eq1 :
Function: [0 =(8*s2^6-24*s2^4+24*s2^2-8)^(2/3)*(24*k3^2*s2^6-72*k3^2*s2^4-24*k3^2)+4*s2^4],
Eq2:
[0 = 64*k3^4*(k3^2*((-27*2^(2*a+2)*(8*s2^6-24*s2^4+24*s2^2-8)^(4/3))-9*2^(4*a+3)*(8*s2^6-24*s2^4+24*s2^2-8)^(2/3)+2^(6*a+4))+81*2^(6*a+6)*k3^6*(8*s2^6-24*s2^4+24*s2^2-8)^(4/3)-9*(8*s2^6-24*s2^4+24*s2^2-8)^(4/3)-9*2^(6*a+6)*k3^4*(8*s2^6-24*s2^4+24*s2^2-8)^(2/3) -3*2^(2*a+1)*(8*s2^6-24*s2^4+24*s2^2-8)^(2/3))].

Examples :
k3 = -5,
Root (0 = (8*s2^6-24*s2^4+24*s2^2-8)^(2/3)*(24*(-5)^2*s2^6-72*(-5)^2*s2^4-24*(-5)^2)+4*s2 ^4, {s2, - 1, 1})
{s2->-0.98828435414}
{s2->0.98828435413}.,

{s2->-0.98828435414}
[log(2)*b+2*k*%pi = asin(-0.98828435414)]

```
#include <iostream>
#include <cmath>
```

```
int main() {
    for (int k = -10; k <= 10; k++) {
        if (k != 0) {
            float b = (asin(-0.98828435414) - 2 * k * M_PI) / log(2);
std::cout << "b = "<< b << std::endl;
            }
        }
return 0;
}
```

→ b=88.60207635515901
b=79.53735607150462
b=70.47263578785024
b=61.40791550419585
b=52.34319522054146
b=43.27847493688707
b=34.21375465323268
b=25.1490343695783 (near ± 25.010857580145688763…)
b=16.08431408592391
b=7.019593802269521
b=-11.10984676503925
b=-20.17456704869364
b=-29.23928733234803
b=-38.30400761600242
b=-47.3687278996568
b=-56.43344818331119
b=-65.49816846696557
b=-74.56288875061996
b=-83.62760903427434
b=-92.69232931792874

Going further :
k Partie imaginaire de ρk
1 ± 14.134 725 141 734 693 790…
2 ± 21.022 039 638 771 554 993…
3 ± 25.010 857 580 145 688 763…
4 ± 30.424 876 125 859 513 210…

Interactions of two 4d string one of spin +2 and one of spin -2 with an open string. 4d string : Trefoil Knot [15]. For values (kmax-,|kmin|) ≪100000, rich values of k3 and dEk3 are not obtained, but for values (kmax,|kmin|) ≥ 100000, we look for |real(values dEk3)| ≤ 4 GeV.
With dE²[k3] =
{2^(7/2)*gM^2*k3/(k3*sqrt(2*k3+2)) ⟶ Potential energy
+2^(7/2)*gM^2/(k3*sqrt(2*k3+2)) ⟶E_A.E_B Kinematic energies}
We calculate E_Potential energy = gM.√(2^(7/2)*k3/(k3*sqrt(2*k3+2)))
And obtain E_Kinematic energy ≠ E_A or E_B; E_Kinematic energy = dEk3 - E_Potential energy

The energy of an open string is with
E_correction = real(ζ(½ + i.b)) > 0 = real1(ζ(½ + i.b))
With
ζ(s) = 1^(-s) + 2^(-s) + 3^(-s) + 4^(-s) + ...
Then
ΔE(k) = E_correction * ℏ * ω(k)
ΔE(k) = real1(ζ(½ + i.b)) * ℏ * ω(k)
With
real1(ζ(½ + i.b))/real(iEp + iEk) = image1(ζ(½ + i.b))/image(iEp + iEk)

Potential energy = 2^(7/4)*gM*sqrt(k3/(k3*sqrt(2*k3+2)))
Kinematic energy in the Way that :

ΔE(k) = real1(ζ(½ + i.b)) * ℏ * ω(k)
With
real1(ζ(½ + i.b) * ℏ * ω(k))/real(iEp + iEk) = image1(ζ(½ + i.b) * ℏ * ω(k))/image(iEp + iEk);

real1(ζ(½ + i.b) * ℏ * sqrt(k/m) * sin(kλ/2))/real(iEp + iEk) = image1(ζ(½ + i.b) * ℏ * sqrt(k/m) * sin(kλ/2))/image(iEp + iEk);

real1(ζ(½ + i.b) * ℏ * sqrt(k/m) * sin(kλ/2))/real(iEp + iEk) = image1(ζ(½ + i.b) * ℏ * sqrt(k/m) * sin(kλ/2))/image(iEp + iEk)

with
ω(k) = sqrt(k/m) * sin(kλ/2)
m = idEk3 = 2^(7/4)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))
Ek = idEk3.gM - Ep = 2^(7/4)*gM*sqrt((k3+1)/(k3*sqrt(2*k3+2))) - Ep,
real1(ζ(½ + i.b) * ℏ * sqrt(k/m) * sin(kλ/2))/real(idEk3) = image1(ζ(½ + i.b) * ℏ * sqrt(k/m) * sin(kλ/2))/image(idEk3)
Where k is the wave vector associated with the vibration and λ is the wavelength of the vibration.
If the wave vector k is close to zero, it means that the wavelength is very long and the atoms will vibrate together as if they were a single atom. The corresponding angular frequency will then be close to zero and the associated energy variation will be small.

1.11.1. Algorithm

```
import cmath
import math
import sys

def main():
    h = 6.626070e-34 # J s
    Lmin = 1
    Lmax = 1000
    k_i = 1
    k_f = 1000
   kmin = -100000
   kmax = 100000
   i0 = 0
   i1 = 0
   s2 = [0] * 101
   Ns = sys.maxsize
   for t in range(-100, -1+1):
         s2[-100] = -0.99941091795
         s2[-99] = -0.99940496942
         s2[-98] = -0.9993988995
         s2[-97] = -0.99939270444
         s2[-96] = -0.99938638046
         s2[-95] = -0.9993799232
         s2[-94] = -0.99937332878
         s2[-93] = -0.99936659252
         s2[-92] = -0.99935970985
         s2[-91] = -0.99935267598
         s2[-90] = -0.99934548584
         s2[-89] = -0.99933813417
         s2[-88] = -0.99933061549
         s2[-87] = -0.99932292402
         s2[-86] = -0.99931505375
         s2[-85] = -0.99930699836
         s2[-84] = -0.99929869643
         s2[-83] = -0.99929030539
         s2[-82] = -0.99928165377
         s2[-81] = -0.9992727885
         s2[-80] = -0.99926370171
         s2[-79] = -0.9992543849
         s2[-78] = -0.99924482941
```

s2[-77] = -0.99923502576
s2[-76] = -0.99922496418
s2[-75] = -0.99921463444
s2[-74] = -0.99920402561
s2[-73] = -0.99919312627
s2[-72] = -0.99918192428
s2[-71] = -0.99917040687
s2[-70] = -0.99915856056
s2[-69] = -0.99914635017
s2[-68] = -0.99913382302
s2[-67] = -0.99912090071
s2[-66] = -0.99910758696
s2[-65] = -0.99909386377
s2[-64] = -0.9990797119
s2[-63] = -0.999065111
s2[-62] = -0.99905003931
s2[-61] = -0.99903447372
s2[-60] = -0.99901838949
s2[-59] = -0.99900176035
s2[-58] = -0.99898455808
s2[-57] = -0.9989667525
s2[-56] = -0.99894831137
s2[-55] = -0.99892920002
s2[-54] = -0.99890938121
s2[-53] = -0.99888881494
s2[-52] = -0.99886745811
s2[-51] = -0.99884526425
s2[-50] = -0.99882218315
s2[-49] = -0.99879816055
s2[-48] = -0.99877313759
s2[-47] = -0.99874705052
s2[-46] = -0.99871982993
s2[-45] = -0.99869140034
s2[-44] = -0.99866167938
s2[-43] = -0.99863057697
s2[-42] = -0.99859799455
s2[-41] = -0.99856382388
s2[-40] = -0.99852794595
s2[-39] = -0.99849022951
s2[-38] = -0.99845048535
s2[-37] = -0.99840868533
s2[-36] = -0.99836451834
s2[-35] = -0.99831782967
s2[-34] = -0.99826839698
s2[-33] = -0.99821597105
s2[-32] = -0.99816027152
s2[-31] = -0.99810098191
s2[-30] = -0.99803774353
s2[-29] = -0.99797014832
s2[-28] = -0.99789768542
s2[-27] = -0.99781993386
s2[-26] = -0.99773619296
s2[-25] = -0.99764575523
s2[-24] = -0.99754778226
s2[-23] = -0.99744130083
s2[-22] = -0.9973251523
s2[-21] = -0.99719795755
s2[-20] = -0.99705806211

```
        s2[-19] = -0.99690346375
        s2[-18] = -0.996731716
        s2[-17] = -0.99653979777
        s2[-16] = -0.99632393409
        s2[-15] = -0.99607934533
        s2[-14] = -0.99579988915
        s2[-13] = -0.99547753758
        s2[-12] = -0.99510159327
        s2[-11] = -0.99465747946
        s2[-10] = -0.99412479292
        s2[-9] = -0.99347415065
        s2[-8] = -0.9926614346
        s2[-7] = -0.99161749897
        s2[-6] = -0.9902273088
        s2[-5] = -0.98828435414
        s2[-4] = -0.98537717072
        s2[-3] = -0.98055121855
        s2[-2] = -0.97097209774
        s2[-1] = -0.94282328841
for k_0 in range(-10, 10):
    if k_0 != 0:
        b = (math.asin(s2[t]) - 2 * k_0 * math.pi) / math.log(2)
        ReS = 1
        ImS = 0
        for l in range(2, Ns + 1):
          s0 = 1/l**(1/2+b*cmath.sqrt(-1))
          if s0.real > 0:
             ReS += s0.real
             ImS += s0.imag
              if l >= Ns:
              print("b = ", b)
        print("ReS+i.ImS = ", ReS, "+i.", ImS)
        print()

for x in range(kmin, kmax+1):
    for y in range(kmin, kmax+1):
        if x*y != 0:
           eq1 = (4*3**(3/2)*(x+1)*(y+1))
           eq2 = (2*(x*(2*y+2)+2*y+2))
eq3 = (9*((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(1/3)*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x-**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*c-math.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y- 4*cmath.sqrt(3)*x-4*cmath.sqrt(3))**(1/3))

        if eq1 * eq2 * eq3 != 0:
                    k3 = (9*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-8*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*cmath.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y-4*cmath.sqrt(3)*x-4*cmath.sqrt(3))**(2/3)-3*((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(1/3)*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-8*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*cmath.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y-4*cmath.sqrt(3)*x-4*cmath.sqrt(3))**(1/3)+((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(2/3))/(9*((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(1/3)*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-8*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*cmath.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y-4*cmath.sqrt(3)* x-4*cmath.sqrt(3))**(1/3))

          if (abs(math.floor(abs(k3)) - abs(k3)) <= 0.00000001) or (abs(math.ceil(abs(k3)) - abs(k3)) <= 0.00000001):
         i0 += 1
         gM = cmath.sqrt(1393.6328858707575005182839547884)
         dEk3 = 2**(7/4)*gM*cmath.sqrt((k3+1)/(k3*cmath.sqrt(2*k3+2)))
```

```
            E_A = 2**(7/4)*gM*cmath.sqrt((x+1)/(x*cmath.sqrt(2*x+2))) # x = k3a
            E_B = 2**(7/4)*gM*cmath.sqrt((y+1)/(y*cmath.sqrt(2*y+2))) # y = k3b
            idEk3 = 2**(7/4)*cmath.sqrt((k3+1)/(k3*cmath.sqrt(2*k3+2)))

for Lambda in range(Lmin, Lmax + 1):
    for k in range(k_i, k_f + 1):
       # m = idEk3
       Sx=(ReS+cmath.sqrt(-1)*ImS)*(h/(2*math.pi))*cmath.sqrt(k/idEk3)*math.sin(k*Lambd a/2)

        if (Sx.real/idEk3.real)/(Sx.imag/idEk3.imag) < 1.2 and
(Sx.real/idEk3.real)/(Sx.imag/idEk3.imag) > 1/1.2 or (abs(dEk3.real) <= 4 or abs(abs(dEk3.real) - E_A.real) <= 0.01 or abs(abs(dEk3.real) - E_B.real) <= 0.01):
        i1 += 1
        print("Case ", i1, "/", i0, ":")
        print("k3 = ", k3, "k3a = ", x, "k3b = ", y)
        rk3 = cmath.sqrt(k3.real**2 + k3.imag**2)
        print("rk3 = ", rk3)
        dErk3 = 2**(7/4)*gM*cmath.sqrt((rk3.real+1)/(rk3.real*cmath.sqrt(2*rk3.real+2)))
        rdEk3 = cmath.sqrt(dEk3.real**2 + dEk3.imag**2)
      E_potential = gM*cmath.sqrt(2**(7/2)*k3/(k3*cmath.sqrt(2*k3+2)))
      E_kinematic = dEk3 - E_potential
      print("dEk3 = ", dEk3, " GeV")
      print("rdEk3 = ", rdEk3)
      print("dErk3 = ", dErk3, " GeV")
      print("E_potential = ", E_potential, " GeV")
      print("E_kinematic = ", E_kinematic, " GeV")
      print()
      if abs(abs(dEk3.real) - E_A.real) <= 0.01 or abs(abs(dEk3.real) - E_B.real) <= 0.01:
         print("E_A = ", E_A, " GeV")
         print("E_B = ", E_B, " GeV")
         print("real(dEk3) = ", dEk3.real, " GeV")
         print()
       if(Sx.real/idEk3.real)/(Sx.imag/idEk3.imag)<1.2and(Sx.real/idEk3.real)/(Sx.imag/idEk3.ima g) > 1/1.2:
        print("k = ", k)
        print("λ = ", Lambda)
        print()
if_name_== "_main_":
      main()
```

A better algorithm ζ(½ + i.b) generator is :

```
import cmath
import math
import sys
def main():
    s2 = [0] * 101
    Ns = sys.maxsize
    for t in range(-100, -1+1):
        s2[-100] = -0.99941091795
        s2[-99] = -0.99940496942
        s2[-98] = -0.9993988995
        s2[-97] = -0.99939270444
        s2[-96] = -0.99938638046
        s2[-95] = -0.9993799232
        s2[-94] = -0.99937332878
        s2[-93] = -0.99936659252
        s2[-92] = -0.99935970985
        s2[-91] = -0.99935267598
        s2[-90] = -0.99934548584
```

s2[-89] = -0.99933813417
s2[-88] = -0.99933061549
s2[-87] = -0.99932292402
s2[-86] = -0.99931505375
s2[-85] = -0.99930699836
s2[-84] = -0.99929869643
s2[-83] = -0.99929030539
s2[-82] = -0.99928165377
s2[-81] = -0.9992727885
s2[-80] = -0.99926370171
s2[-79] = -0.9992543849
s2[-78] = -0.99924482941
s2[-77] = -0.99923502576
s2[-76] = -0.99922496418
s2[-75] = -0.99921463444
s2[-74] = -0.99920402561
s2[-73] = -0.99919312627
s2[-72] = -0.99918192428
s2[-71] = -0.99917040687
s2[-70] = -0.99915856056
s2[-69] = -0.99914635017
s2[-68] = -0.99913382302
s2[-67] = -0.99912090071
s2[-66] = -0.99910758696
s2[-65] = -0.99909386377
s2[-64] = -0.9990797119
s2[-63] = -0.999065111
s2[-62] = -0.99905003931
s2[-61] = -0.99903447372
s2[-60] = -0.99901838949
s2[-59] = -0.99900176035
s2[-58] = -0.99898455808
s2[-57] = -0.9989667525
s2[-56] = -0.99894831137
s2[-55] = -0.99892920002
s2[-54] = -0.99890938121
s2[-53] = -0.99888881494
s2[-52] = -0.99886745811
s2[-51] = -0.99884526425
s2[-50] = -0.99882218315
s2[-49] = -0.99879816055
s2[-48] = -0.99877313759
s2[-47] = -0.99874705052
s2[-46] = -0.99871982993
s2[-45] = -0.99869140034
s2[-44] = -0.99866167938
s2[-43] = -0.99863057697
s2[-42] = -0.99859799455
s2[-41] = -0.99856382388
s2[-40] = -0.99852794595
s2[-39] = -0.99849022951
s2[-38] = -0.99845048535
s2[-37] = -0.99840868533
s2[-36] = -0.99836451834
s2[-35] = -0.99831782967
s2[-34] = -0.99826839698
s2[-33] = -0.99821597105
s2[-32] = -0.99816027152

```
        s2[-31] = -0.99810098191
        s2[-30] = -0.99803774353
        s2[-29] = -0.99797014832
        s2[-28] = -0.99789768542
        s2[-27] = -0.99781993386
        s2[-26] = -0.99773619296
        s2[-25] = -0.99764575523
        s2[-24] = -0.99754778226
        s2[-23] = -0.99744130083
        s2[-22] = -0.9973251523
        s2[-21] = -0.99719795755
        s2[-20] = -0.99705806211
        s2[-19] = -0.99690346375
        s2[-18] = -0.996731716
        s2[-17] = -0.99653979777
        s2[-16] = -0.99632393409
        s2[-15] = -0.99607934533
        s2[-14] = -0.99579988915
        s2[-13] = -0.99547753758
        s2[-12] = -0.99510159327
        s2[-11] = -0.99465747946
        s2[-10] = -0.99412479292
        s2[-9] = -0.99347415065
        s2[-8] = -0.9926614346
        s2[-7] = -0.99161749897
        s2[-6] = -0.9902273088
        s2[-5] = -0.98828435414
        s2[-4] = -0.98537717072
        s2[-3] = -0.98055121855
        s2[-2] = -0.97097209774
        s2[-1] = -0.94282328841

for k_0 in range(-10, 10):
    if k_0 != 0:
        b = (math.asin(s2[t]) - 2 * k_0 * math.pi) / math.log(2)
        ReS = 1
        ImS = 0
        for l in range(2, Ns + 1):
            s0 = 1/l**(1/2+b*cmath.sqrt(-1))
              if s0.real > 0:
               ReS += s0.real
               ImS += s0.imag
              if l >= Ns:
                print("b = ",b)
         print("ReS+i.ImS = ", ReS, "+i.", ImS)
             print()
if_name_== "_main_":
main()
```

We obtain results similar to :

```
import cmath
import math
def main():
    s2 = [0] * 6
    Ns = 100000000
    for t in range(-5, -1+1):
        s2[-5] = -0.98828435414
        s2[-4] = -0.98537717072
```

```
            s2[-3] = -0.98055121855
            s2[-2] = -0.97097209774
            s2[-1] = -0.94282328841
        for k_0 in range(-10, 10):
            if k_0 != 0:
                b = (math.asin(s2[t]) - 2 * k_0 * math.pi) / math.log(2)
                ReS = 1
                ImS = 0
            for l in range(2, Ns + 1):
                s0 = 1/l**(1/2+b*cmath.sqrt(-1))
            if s0.real > 0:
                ReS += s0.real
                ImS += s0.imag
            if l >= Ns:
                print("b = ",b)
          print("ReS+i.ImS = ", ReS, "+i.", ImS)
          print()

if_name_== "_main_":
    main()
```

→ b = 88.60207635515901
ReS+i.ImS = 6307.844896030516 +i. -28.94938165387543
b = 79.53735607150463
ReS+i.ImS = 6434.37737066339 +i. 11.242220705893391
ReS+i.ImS = 6335.630982820043 +i. -44.06143464461285
b = 61.40791550419585
ReS+i.ImS = 6388.238302501553 +i. 108.25945207135854
ReS+i.ImS = 6382.172075003123 +i. -61.24789663387373
b = 43.27847493688707
ReS+i.ImS = 6265.140783718337 +i. 97.21067855763751
ReS+i.ImS = 6478.890842451481 +i. -93.52071152141292
ReS+i.ImS = 6181.543372422819 +i. -122.22486115007362
b = 16.08431408592391
ReS+i.ImS = 6692.738286329365 +i. 142.35970324080344
ReS+i.ImS = 6099.5141308650045 +i. -433.9284953584362
ReS+i.ImS = 6221.895473340011 +i. 280.61671644842875
b = -20.174567048693643
ReS+i.ImS = 6618.610594190466 +i. -134.43579349798716
ReS+i.ImS = 6213.541197470559 +i. 107.90232364995555
ReS+i.ImS = 6471.434206720912 +i. 84.35606321963408
b = -47.3687278996568
ReS+i.ImS = 6268.873089628677 +i. -80.16737814175987
ReS+i.ImS = 6383.614914697005 +i. 56.515709447766845
b = -65.49816846696558
ReS+i.ImS = 6382.085124104104 +i. -101.30322907646526
ReS+i.ImS = 6339.58171487993 +i. 44.51747397056369
b = -83.62760903427434
ReS+i.ImS = 6431.167438821539 +i. -15.818420310540942
b = 88.62804490801527
ReS+i.ImS = 6306.284430038629 +i. 22.223163074471593
ReS+i.ImS = 6425.97239428184 +i. -39.483157160381566
... .
From
b = 88.60207635515901
b = 79.53735607150463
b = 70.47263578785024
b = 61.40791550419585
b = 52.34319522054147

b = 43.27847493688707
b = 34.213754653232684
b = 25.149034369578295
b = 16.08431408592391
b = 7.0195938022695215
b = -11.109846765039254
b = -20.174567048693643
b = -29.239287332348027
b = -38.304007616002416
b = -47.3687278996568
b = -56.433448183311185
b = -65.49816846696558
b = -74.56288875061996
b = -83.62760903427434
b = 88.62804490801527
b = 79.56332462436087
b = 70.49860434070649
b = 61.433884057052104
b = 52.369163773397716
b = 43.30444348974333
b = 34.23972320608894
b = 25.175002922434555
b = 16.11028263878017
b = 7.045562355125782
b = -11.083878212182993
b = -20.14859849583738
b = -29.21331877949177
b = -38.27803906314616
b = -47.34275934680055
b = -56.40747963045494
b = -65.47219991410932
b = -74.5369201977637
b = -83.6016404814181
b = 88.66602097606453
b = 79.60130069241013
b = 70.53658040875575
b = 61.47186012510136
b = 52.40713984144697
b = 43.34241955779259
b = 34.2776992741382
b = 25.21297899048381
b = 16.14825870682942
b = 7.083538423175033
b = -11.045902144133743
b = -20.11062242778813
b = -29.175342711442514
b = -38.240062995096906
b = -47.30478327875129
b = -56.369503562405676
b = -65.43422384606006
b = -74.49894412971446
b = -83.56366441336884
b = 88.72948326417199
b = 79.66476298051761
b = 70.60004269686323
b = 61.53532241320883
b = 52.47060212955444
b = 43.405881845900055

b = 34.34116156224567
b = 25.27644127859128
b = 16.211720994936893
b = 7.147000711282508
b = -10.982439856026266
b = -20.047160139680656
b = -29.111880423335045
b = -38.17660070698943
b = -47.24132099064382
b = -56.30604127429821
b = -65.37076155795259
b = -74.43548184160699
b = -83.50020212526137
b = 88.87124212439919
b = 79.8065218407448
b = 70.74180155709041
b = 61.67708127343602
b = 52.612360989781635
b = 43.54764070612725
b = 34.482920422472866
b = 25.418200138818474
b = 16.353479855164085
b = 7.2887595715097
b = -10.840680995799076
b = -19.905401279453464
b = -28.970121563107853
b = -38.03484184676224
b = -47.09956213041663
b = -56.16428241407101
b = -65.2290026977254
b = -74.29372298137979
b = -83.35844326503418
.....

The function used to calculate k3 gives either false integers or real integers.
For a definition of 1.0e-5, as in the command line: if (abs(math.floor(abs(k3))-abs(k3)) <= 0.00001) or (abs(math.ceil(abs(k3))-abs(k3)) <= 0.00001)

Many of the k3s obtained are false integers.
For a definition of 1.0e-8, as in the command line: if (abs(math.floor(abs(k3))-abs(k3)) <= 0.00000001) or (abs(math.ceil(abs(k3))-abs(k3)) <= 0.00000001)
The k3 obtained are much less likely to be false integers.

With a definition increased from 1.0e-5 to 1.0e-8 and ever more increased with the calculation definition always proportionally increased, we will have more and more exact k3 values over a wider margin [kmin, kmax].

These obtained k3 will help in the calculation of massive gravitons for dark matter (dark matter effect), initially obtained with a calculated variation of dk3n = $1.255314934818507*10^{89}$
And massive gravitons for dark energy, initially obtained with a calculated variation of dk3d = $1.862840706286786*10^{147}$ (dk3 $\neq$ 1 leads to a quantum entanglement process.).
And 3 = ∑_EnergieDarkEnergy/∑_EnergieDarkMatter.

We have
k3=(9*(9*(2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(27*k3a^2*sqrt(2*k3a+2)*k3b^2*sqrt(2*k3b+2)+((-8*k3a^2)-8*k3a)*k3b^2+((-8*k3a^2)-8*k3a)*k3b)+3^(7/2)*k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2)+((-4*sqrt(3)*k3a)-4*sqrt(3))*k3b-4*sqrt(3)*k3a-4*sqrt(3))^(2/3)-3*((4*3^(7/2)*k3a+4*3^(7/2))*k3b+4*3^(7/2)*k3a+4*3^(7/2))^(1/3)*(9*(2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(27*k3a^2*sqrt(2*k3a+2)*k3b^2*sqrt(2*k3b+2)+((-8*k3a^2)-8*k3a)*k3b^2+((-8*k3a^2)-8*k3a)*k3b)+3^(7/2)*k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2)+((-4*sqrt(3)*k3a)-4*sqrt(3))*k3b-4*sqrt(3)*k3a-4*sqrt(3))^(1/3)+((4*3^(

7/2)*k3a+4*3^(7/2))*k3b+4*3^(7/2)*k3a+4*3^(7/2))^(2/3))/(9*((4*3^(7/2)*k3a+4*3^(7/2))*k3b+4*3^(7/2)*k3a+4*3^(7/2))^(1/3)*(9*(2*k3a+2)^(1/4)*(2*k3b+2)^(1/4)*sqrt(27*k3a^2*sqrt(2*k3a+2)*k3b^2*sqrt(2*k3b+2)+((-8*k3a^2)-8*k3a)*k3b^2+((-8*k3a^2)-8*k3a)*k3b)+3^(7/2)*k3a*sqrt(2*k3a+2)*k3b*sqrt(2*k3b+2)+((-4*sqrt(3)*k3a)-4*sqrt(3))*k3b-4*sqrt(3)*k3a-4*sqrt(3))^(1/3))

k3a → 4d massive graviton mass calculation.
k3b → a second 4d massive graviton mass calculation.
Algorithm :

```
import cmath
 import math
kmin = -100000
kmax = 100000
i0 = 0
i1 = 0

for x in range(kmin, kmax+1):
   for y in range(kmin, kmax+1):
     if x*y != 0:
        eq1 = (4*3**(3/2)*(x+1)*(y+1))
        eq2 = (2*(x*(2*y+2)+2*y+2))
            eq3 = (9*((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(1/3)*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-8*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*cmath.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y- 4*cmath.sqrt(3)*x-4*cmath.sqrt(3))**(1/3))

     if eq1 * eq2 * eq3 != 0:
                                        k3= (9*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-8*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*cmath.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y-4*cmath.sqrt(3)*x-4*cmath.sqrt(3))**(2/3)-3*((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(1/3)*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-8*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*cmath.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y-4*cmath.sqrt(3)*x-4*cmath.sqrt(3))**(1/3)+((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(2/3))/(9*((4*3**(7/2)*x+4*3**(7/2))*y+4*3**(7/2)*x+4*3**(7/2))**(1/3)*(9*(2*x+2)**(1/4)*(2*y+2)**(1/4)*cmath.sqrt(27*x**2*cmath.sqrt(2*x+2)*y**2*cmath.sqrt(2*y+2)+((-8*x**2)-8*x)*y**2+((-8*x**2)-8*x)*y)+3**(7/2)*x*cmath.sqrt(2*x+2)*y*cmath.sqrt(2*y+2)+((-4*cmath.sqrt(3)*x)-4*cmath.sqrt(3))*y-4*cmath.sqrt(3)*x-4 *cmath.sqrt(3))**(1/3))
        if (abs(math.floor(abs(k3)) - abs(k3)) <= 0.00000001) or (abs(math.ceil(abs(k3)) - abs(k3)) <= 0.00000001):
       i0 += 1
      gM = cmath.sqrt(1393.6328858707575005182839547884)
      dEk3 = 2**(7/4)*gM*cmath.sqrt((k3+1)/(k3*cmath.sqrt(2*k3+2)))
      E_A = 2**(7/4)*gM*cmath.sqrt((x+1)/(x*cmath.sqrt(2*x+2)))#x = k3a
      E_B = 2**(7/4)*gM*cmath.sqrt((y+1)/(y*cmath.sqrt(2*y+2)))#y = k3b

     if abs(dEk3.real) <= 4 or abs(abs(dEk3.real) - E_A.real) <= 0.01 or abs(abs(dEk3.real) - E_B.real) <= 0.01:
    i1 += 1
    print("Case ", i1, "/", i0, ":")
   print("k3 = ", k3, "k3a = ", x, "k3b = ", y)
   rk3 = cmath.sqrt(k3.real**2 + k3.imag**2)
  print("rk3 = ", rk3)

dErk3 = 2**(7/4)*gM*cmath.sqrt((rk3.real+1)/(rk3.real*cmath.sqrt(2*rk3.real+2)))
rdEk3 = cmath.sqrt(dEk3.real**2 + dEk3.imag**2)
E_potential = gM*cmath.sqrt(2**(7/2)*k3/(k3*cmath.sqrt(2*k3+2)))
E_kinematic = dEk3 - E_potential
 print("dEk3 = ", dEk3, " GeV")
print("rdEk3 = ", rdEk3)
print("dErk3 = ", dErk3, " GeV")
print("E_potential = ", E_potential, " GeV")
print("E_kinematic = ", E_kinematic, " GeV")
```

```
 print()
if abs(abs(dEk3.real) - E_A.real) <= 0.01 or abs(abs(dEk3.real) - E_B.real) <= 0.01:
     print("E_A = ",E_A," GeV")
     print("E_B = ",E_B," GeV")
     print("real(dEk3) = ",dEk3.real," GeV")
     print()
```

We obtain from result : An open string absorbing a massive graviton

$H = \int d\sigma\, [X'^2(\sigma)/2 + (\partial X/\partial\sigma)^2/2] = m^2 = E\_A.E\_B/C^4 + \text{Potential Energy}^2/C^4$

For values (kmax,|kmin|) ≪100000, rich values of k3 and dEk3 are not obtained, but for values (kmax,|kmin|) ≥ 100000, we look for |real(values dEk3)| ≤ 4 GeV.

With dE²[k3] =
{2^(7/2)*gM^2*k3/(k3*sqrt(2*k3+2)) → Potential energy
+2^(7/2)*gM^2/(k3*sqrt(2*k3+2)) →E_A.E_B Kinematic energies}
We calculate E_Potential energy = gM.√(2^(7/2)*k3/(k3*sqrt(2*k3+2)))
And obtain E_Kinematic energy ≠ E_A or E_B; E_Kinematic energy = dEk3 - E_Potential energy

We obtain from previous algorithm:

```
import cmath
import math
kmin = -10000000
kmax = 10000000
i0 = 0
i1 = 0

for k3a in range(kmin, kmax+1):
    for k3b in range(kmin, kmax+1):
       if k3a*k3b != 0:
              if(4*3**(3/2)*(k3a+1)*(k3b+1))*(2*(k3a*(2*k3b+2)+2*k3b+2))*(9*((4*3**(7/2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k
3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.
sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.
sqrt(2*k3b+ 2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3)) != 0:

             k3 = (9*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a +2)*k3b**2*cmath.sqrt(2*k-
3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k-
3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(2/3)-3*((4*3**(7/2)*k
3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*c-
math.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*c-
math.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.
sqrt(3))**(1/3)+((4*3**(7/ 2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(2/3))/(9*((4*3**(7/2)*k3a+4*3**(7/2 ))*k3
b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k-
3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k-
3b*cmath.sqrt(2*k3b+2)+((-4 *cmath.sqrt(3)*k3a)- 4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3))

           if (abs(math.floor(abs(k3))-abs(k3)) <= 0.00000001) or (abs(math.ceil(abs(k3))-abs(k3)) <= 0.00000001):
          i0 += 1
          gM = cmath.sqrt(1393.6328858707575005182839547884)
#dE[k3] = 2^(7/4)*abs(gM)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))
          dEk3 = 2**(7/4)*gM*cmath.sqrt((k3+1)/(k3*cmath.sqrt(2*k3+2)))
          if abs(dEk3.real) <= 4:
          i1 += 1
         print("Case ",i1,"/",i0,":")
print("k3 = ",k3, "k3a = ",k3a, "k3b = ",k3b)
 rk3 = cmath.sqrt(k3.real**2 + k3.imag**2)
print("rk3 = ",rk3)
dErk3 = 2**(7/4)*gM*cmath.sqrt((rk3.real+1)/(rk3.real*cmath.sqrt(2*rk3.real+2))) rdEk3 = cmath.sqrt(dEk3.real**2 + dEk3.
imag**2)
```

```
#H = ∫dσ [X'²(σ)/2 + (∂X/∂σ)²/2] = m² = E_A.E_B/C⁴ ÷ Potential Energy²/C⁴
#with dE[k3] = 2^(7/4)*abs(gM)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))
#and
#E_A.E_B = dE_A[k3].dE_B[k3]
#dE²[k3] =
#{2^(7/2)*gM^2*k3/(k3*sqrt(2*k3+2)) → Potential energy
#+2^(7/2)*gM^2/(k3*sqrt(2*k3+2)) →E_A.E_B Kinematic energies}
        E_potential = gM*cmath.sqrt(2**(7/2)*k3/(k3*cmath.sqrt(2*k3+2)))
        E_kinematic = dEk3 - E_potential
        print("dEk3 = ",dEk3," GeV")
        print("rdEk3 = ",rdEk3)
        print("dErk3 = ",dErk3," GeV")
        print("E_potential = ",E_potential," GeV")
        print("E_kinematic = ",E_kinematic," GeV")
        print()
→ ... ....
```

1.11.2. Previous Algorithm Is

```
import cmath
import math
kmin = -100000
kmax = 100000

for k3a in range(kmin, kmax+1):
    for k3b in range(kmin, kmax+1):
        if k3a*k3b != 0:
            if(4*3**(3/2)*(k3a+1)*(k3b+1))*(2*(k3a*(2*k3b+2)+2*k3b+2))*(9*((4*3**(7/2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3 b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))** (1/3)) != 0:

            k3=(9*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a +2)*k3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(2/3)3*((4*3**(7/2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3)+((4*3**(7/2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(2/3))/(9*((4*3**(7/2)*k3a+4*3**(7/2))*k3b+4*3**(7/2)*k3a+4*3**(7/2))**(1/3)*(9*(2*k3a+2)**(1/4)*(2*k3b+2)**(1/4)*cmath.sqrt(27*k3a**2*cmath.sqrt(2*k3a+2)*k3b**2*cmath.sqrt(2*k3b+2)+((-8*k3a**2)-8*k3a)*k3b**2+((-8*k3a**2)-8*k3a)*k3b)+3**(7/2)*k3a*cmath.sqrt(2*k3a+2)*k3b*cmath.sqrt(2*k3b+2)+((-4*cmath.sqrt(3)*k3a)-4*cmath.sqrt(3))*k3b-4*cmath.sqrt(3)*k3a-4*cmath.sqrt(3))**(1/3))

            if (abs(math.floor(abs(k3))-abs(k3)) <= 0.00000001) or (abs(math.ceil(abs(k3))-abs(k3)) <= 0.00000001):
            print("k3 = ",k3, "k3a = ",k3a, "k3b = ",k3b)
             rk3 = cmath.sqrt(k3.real**2 + k3.imag**2)
            print("rk3 = ",rk3)
            gM = cmath.sqrt(1393.6328858707575005182839547884)
#dE[k3] = 2^(7/4)*abs(gM)*sqrt((k3+1)/(k3*sqrt(2*k3+2)))
           dEk3 = 2**(7/4)*gM*cmath.sqrt((k3+1)/(k3*cmath.sqrt(2*k3+2)))
           dErk3 = 2**(7/4)*gM*cmath.sqrt((rk3.real+1)/(rk3.real*cmath.sqrt(2*rk3.real+2)))
           rdEk3 = cmath.sqrt(dEk3.real**2 + dEk3.imag**2)
           print("dEk3 = ",dEk3," GeV")
           print("rdEk3 = ",rdEk3)
           print("dErk3 = ",dErk3," GeV")
           print()

→ k3 = (15.252015994715551+26.988442113219644j) k3a = -99977 k3b = -9167
```

rk3 = (30.999999993575265+0j)
dEk3 = (43.289423533179345-12.027879046945008j) GeV
rdEk3 = (44.92932298847736+0j)
dErk3 = (45.10511912443094+0j) GeV

k3 = (22.751358660433908+39.979690844327074j) k3a = -99898 k3b = -96922
rk3 = (46.000000009822514+0j)
dEk3 = (39.20302478717349-10.766641702615907j) GeV
rdEk3 = (40.654615064175026+0j)
dErk3 = (40.76286823600352+0j) GeV

k3 = (15.751953006689266+27.85455037490676j) k3a = -99866 k3b = -11092
rk3 = (32.00000000176821+0j)
dEk3 = (42.945174657934906-11.919051459927925j) GeV
rdEk3 = (44.568506976394715+0j)
dErk3 = (44.737600010875255+0j) GeV

k3 = (32.826138145249125+19.142744186934145j) k3a = -99812 k3b = 31559
rk3 = (38.000000007070796+0j)
dEk3 = (42.3779587280709-5.765780019855431j) GeV
rdEk3 = (42.76839493358904+0j)
dErk3 = (42.804905232767744+0j) GeV

k3 = (38.88617135675854+22.646537862515675j) k3a = -99737 k3b = 86462
rk3 = (44.99999999939494+0j)
dEk3 = (40.59220979108459-5.49627999540387j) GeV
rdEk3 = (40.96262429961367+0j)
dErk3 = (40.99230953863432+0j) GeV

k3 = (21.751420409895772+38.24755822747227j) k3a = -99590 k3b = -74535
rk3 = (44.000000002408086+0j)
dEk3 = (39.64295002946673-10.899444991120085j) GeV
rdEk3 = (41.114004769096006+0j)
dErk3 = (41.2283521272286+0j) GeV

k3 = (36.29030566382351+21.14269886255414j) k3a = -99515 k3b = 57448
rk3 = (42.00000000436177+0j)
dEk3 = (41.311076474648466-5.603445696050985j) GeV
rdEk3 = (41.68937086551974+0j)
dErk3 = (41.72166725896865+0j) GeV

k3 = (21.251453441824104+37.38148909625858j) k3a = -99232 k3b = -65200
rk3 = (43.0000000051595+0j)
dEk3 = (39.872435550273735-10.96898397410855j) GeV
rdEk3 = (41.35371478035538+0j)
dErk3 = (41.471345681252146+0j) GeV
... .
(k3 = (15.252015994715551+26.988442113219644j) k3a = -99977 k3b = -9167
rk3 = (30.999999993575265+0j)
dEk3 = (43.289423533179345-12.027879046945008j) GeV
rdEk3 = (44.92932298847736+0j)
dErk3 = (45.10511912443094+0j) GeV
E_potential = (43.1025066447868-11.336743959217163j) GeV
E_kinematic = (0.1869168883925454-0.6911350877278455j) GeV
... .)

1.12. About Dark Matter and Dark Energy Process

Massive bosons are generated as in the case of Goldstone bosons and in a cosmological context with the Coleman-de Luccia Tun-

neling, as an impossible false vaccum. When ΔΦ is extremely large False vacuum decay is replaced by massive bosons emissions. With equations:
To nucleate, a bubble must overcome an energy barrier of height
$\Phi_c=3\gamma/4R^2-\Delta\Phi$, (E. 1)
where ΔΦ is the difference in energy between the true and false vacuums, γ is the unknown (possibly extremely large) surface tension of the domain wall, and R is the radius of the bubble. Rewriting E. 1 gives the critical radius as

$R_c=\sqrt{(3\gamma/(4\Delta\Phi))}$.(E. 2)
A bubble smaller than the critical size can overcome the potential barrier via quantum tunnelling of instantons to lower energy states. When the energy at the MAD is smaller do to conversions in massive bosons, we have the value R equivalent to becoming smaller and a Tunneling effect can occur (with particles sent outside black hole's core)

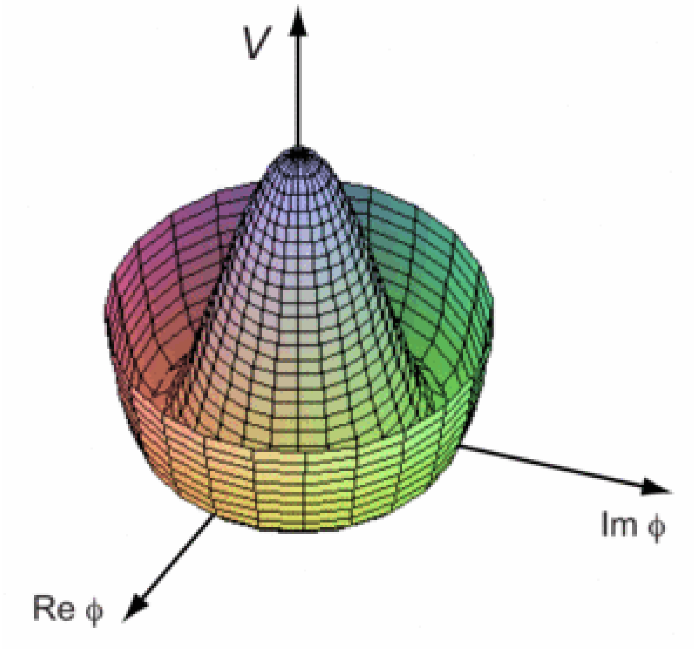

Figure 15: Goldstone Boson

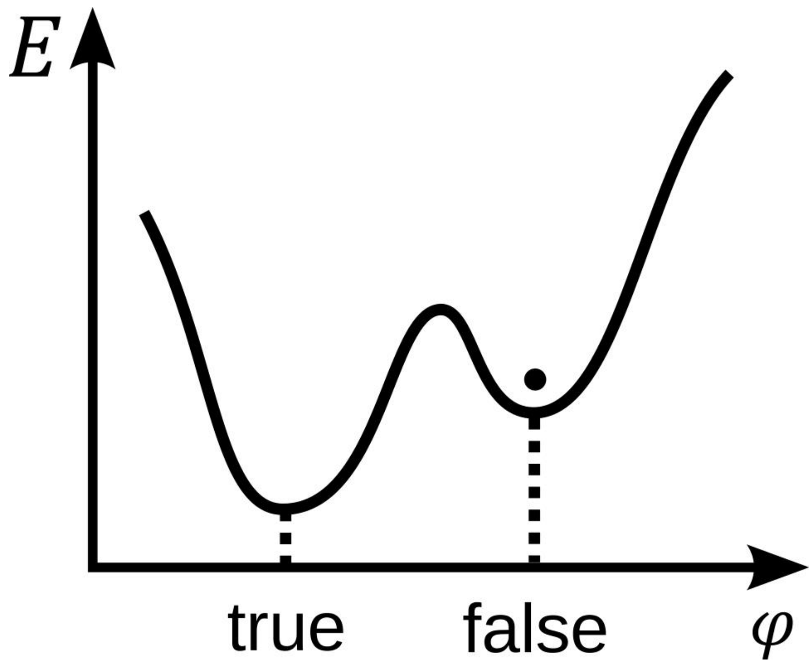

Figure 16: A scalar field φ (which represents physical position) in a false vacuum. The energy E is higher in the false vacuum than that in the true vacuum or ground state, but there is a barrier preventing the field from classically rolling down to the true vacuum. Therefore, the transition to the true vacuum must be stimulated by the creation of high-energy particles or through quantum-mechanical tunneling [10,17-22].

2. EHT Results

It is find that most simulations naturally produce a low level of circular polarization consistent with our upper limit and that Faraday conversion is likely the dominant production mechanism for circular polarization at 230 GHz in M87*. All methods find a moderate level of resolved circular polarization across the image (〈| v| 〉 < 3.7%), which is consistent with the low fraction of circular polarization measured by the Millimeter/Submillimeter Observatory of the Atacama (ALMA) (| vint| < 1%) over the entire image. This means that there is some amount of circular polarization present in the image, but it is not very intense. But tentative detection of a low value of the Stokes intensity ν is carried out with a signal-to-noise ratio (SNR) of about 5. This suggests that there is an indication of the presence of some linear polarization in the data, but this is still weak and requires further analysis to confirm. Despite this broad agreement, the methods show substantial variation in the morphology of the circularly polarized emission, indicating that our conclusions are strongly dependent on the imaging assumptions because of the limited baseline coverage, uncertain telescope gain calibration, and weakly polarized signal. Therefore the EHT results confirm an effective and continuous MAD process., (National Radio Astronomy Observatory) [23].

A supermassive black hole's strong magnetic fields are revealed in a new light Closure phases observed on the ALMA-SMT-PV triangle during M87* observations on 2017 April 5–11 in low band (left column) and in high band (right column). Top: closure phases of scan-averaged visibilities for all epochs, RR* in red, LL* in blue. Bottom: difference of closure phases between RR* and LL*. The zero level of the closure difference (i.e., no detected) is marked with a black dashed line. A light-green band shows the RR* − LL* difference, inferred by band, under the constant difference assumption [4]. For this antenna triangle, the average offset in closure phases between RR* and LL* (combining all epochs and assuming a constant residual value) is 6.7 ± 1.3 deg in the low band and 7.9 ± 1.6 deg in the high band, indicated with green bands in the bottom panels. The offset is consistent despite each band being calibrated independently. Moreover, the measured offset is difficult to explain with the conservative systematic nonclosing error budget discussed in Section 8.4 of Paper III, and hence it implies a tentative detection of a weak Stokes ν_medium at the level of S/N ∼5.

While this measurement implies the presence of a fractional CP reaching ∼3% somewhere in the visibility domain, the measurement cannot be directly translated into a quantitative image domain constraint. The ALMA-SMT-PV triangle shown is the one that produces closure phase differences with the most clear deviation from zero. In Appendix B, we show the results for all other triangles including ALMA. None of these other triangles show an unambiguous detection of a nonzero closure phase difference like that seen on the ALMA-SMT-PV triangle, suggesting that SMT-PV is the baseline most sensitive to the CP signatures in M87*. Even though the RR* and LL* closure phases are robust against antenna gains, they may be affected by instrumental polarimetric leakage (antenna D-terms, the D matrix term in Equation (3)). However, the effect of D-term uncertainties in the parallel-hand visibilities is much smaller than in the cross-hands (e.g., Smirnov 2011), which implies that Stokes is much less affected by instrumental polarization than Stokes Q and μ. To verify a negligible impact of the polarimetric leakage on the closure phase signal, we have compared the closure phase values between the data with and without D-term calibration. For all triangles related to ALMA, the effect of the D-terms on the RR* − LL* closure differences is always less than the standard deviation of the thermal noise on the closure phase difference. For the triangle shown in previous Figure, the maximum effect of the D-terms is only 0.42σ. Hence, we can conclude that the closure phase differences indicating the presence of circular polarization on EHT baselines, presented in previous Figure, are robust against both antenna gains and polarimetric leakage. In Appendix C, we discuss evidence for polarization in "closure trace" products, quantities that are insensitive to all station-based systematic factors, including D-terms.

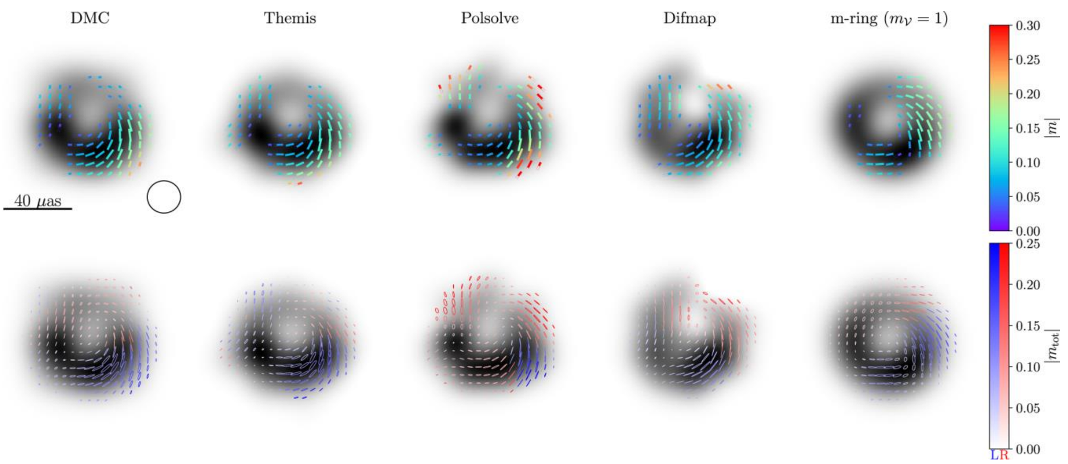

Figure 17: Reconstructions of 2017 EHT M87* data from April 11, low band. The top row shows total intensity images from all reconstruction methods in gray scale and fractional linear polarization in colored ticks as in Paper VII. The second row shows the same grayscale total intensity image overlaid with colored ellipses indicating the total polarization fraction |m_tot| = √(Q² + μ² + ν²)/x. The size of each ellipse indicates the total polarized brightness; the orientation of each ellipse indicates the linear EVPA, and axis ratio indicates the relative fraction of circular polarization. The color of each ellipse indicates the sign of circular polarization [4].

Circular polarization imaging results from 2017 EHT observations of M87* on April 5 (top two rows) and April 6 (bottom two rows). Images of circular polarization on these consecutive days are expected to be nearly identical, as is seen in total intensity and linear polarization. The top/bottom row in each pair shows results from imaging the high/low-band data. In each panel, total intensity is indicated in the colored linear-scale contours, and the Stokes brightness is indicated in the diverging color map, with red/blue indicating a positive/negative sign. The color bar ranges are fixed in both plots (and in next Figure). For posterior exploration methods (DMC, Themis, m-ring fitting), the posterior-average image is shown. All images are blurred with the same 20 μas FWHM Gaussian, shown with the black inset circle in the upper left panels [4]. The same as previously, but for 2017 EHT observations of M87* on April 10 (top two rows) and April 11 (bottom two rows) [4].

Circular polarization properties of passing Paper VIII one-zone models. Left: distribution of the Faraday conversion optical depth τρ Q in passing models. In all passing models τρ Q > 1, indicating that most circular polarization is likely produced by Faraday conversion. Middle: distribution of the ratio of the Faraday rotation to Faraday conversion optical depths. In all cases, τρ V > τρ Q , indicating that with a constant field orientation in the emission region, circular polarization will dominate over linear polarization in these models. Right: the average fractional circular polarization between 5rg and 10rg in one-zone models with a rotating magnetic field direction along the line of sight, as a function of the angular rotation frequency ω. We show three different models: a model with low Faraday conversion depth (blue), a model with median conversion depth (orange), and a model with high conversion depth (green). Dashed lines show corresponding results for one-zone models with no intrinsic emission of circular polarization, j_ν = 0 [4].

A random selection of representative snapshots from our GRMHD image library. The color scales for each snapshot are normalized individually. The first three rows are presented at native resolution in symmetric logarithmic scale with three decades in dynamic range shown to better visualize faint features. The bottom three rows plot Stokes x in contours and Stokes ν in color after blurring with a 20 μas FWHM Gaussian, both in linear scale. Models exhibit a wide variety of morphologies and almost always show sign reversals, at both perfect resolution and EHT resolution [4]. Example GRMHD snapshot (MAD a* = − 0.94 Rlow = 10 Rhigh =160) plotted with both magnetic field configurations, aligned (left) and reversed (right). The top panels show the images blurred to EHT resolution, and the bottom panels show the images at their native resolution. As shown in the left panels, this snapshot fails simultaneous polarimetric constraints with the aligned-field configuration, overproducing vnet. However, as shown in the right panels, flipping the magnetic field polarity produces some oppositely signed regions that reduce □ vnet□ . Flipping the field has no effect on the total intensity image [4].

This illustrates that it is not easy to predict which regions of a given image change upon a reversal of the magnetic field direction. We expect that regions dominated by intrinsic synchrotron emission should trivially flip sign, while regions dominated by Faraday conversion may remain unchanged unless Faraday rotation is significant along those geodesics. We further explore the effect of flipping the magnetic field polarity on linear polarization in Appendix I. While there are noticeable differences in the distribution of the ∠β2 parameter. Palumbo et al across all GRMHD models, the effect is less dramatic for linear polarization metrics than it is for circular. Imaging results from the three synthetic data sets in Table 4. The first row shows the total intensity in gray scale and the fractional linear polarization in the colored ticks. The left column is the ground-truth simulation image. From left to right, the next columns show the posterior-average images from DMC and Themis, the final CLEAN images from polsolve and DIFMAP, and the posterior average of the m-ring model fits. All images are blurred with the same 20 μas FWHM circular Gaussian beam. The second row shows the total intensity image with eight contour levels on a linear scale and the Stokes ν structure in a a diverging color map. The first and second rows show the results for model 01 (low resolved circular polarization), the third and fourth rows show the results for model 02 (moderate resolved circular polarization), and the fifth and sixth rows show the results for model 03 (high resolved circular polarization). Note that the color bar for ν has different maximum values in the second, fourth, and sixth rows as the GRMHD simulation images become more polarized [4].

Examples of three m-ring models in Stokes x and p (top panels) and Stokes x and ν (bottom panels) [4]. Throughout the panels, the Stokes x structure (heat map) is kept constant with F = 0.5 Jy, d = 40 μas, α = 10 μas, and $\beta\_x,1$ = 0.2 - 0.1i. The top middle panel shows a linear polarization structure with m_net = β_p,0 = 0.1, β_p,-1 = 0.1 + 0.2i, and β_p,1 = -0.1 + 0.1i. In the top left and right panels, nonzero β_p,2 components have been added with opposite sign. The bottom left panel shows a dipolar circular polarization structure (contours) oriented toward the north (β_ν,0 = 0.14). The net circular polarization is zero, so that the north and south half of the ring are identical with opposite sign in Stokes β_ν. In the bottom middle panel, we have rotated the circular polarization structure by −45° and introduced a nonzero net circular polarization (ν_net ≡ β_ν,0 = 0.05), so that the symmetry is broken. Finally, in the bottom right panel we have added a nonzero β_ν,2 component, increasing the complexity of the azimuthal structure in Stokes ν. The model shown in the middle panels is used for our geometric tests [32]. Test of plasma content, where we ray-trace a single snapshot of the MAD a = + 0.5 Rhigh = 80 Rlow = 10 aligned-field model with an increasing positron-to-electron ratio, denoted as f in the upper left corner of each panel. The Stokes ν structure clearly evolves as f increases, but we do not observe a clear discriminant of plasma content [4].

In next figure, we spot-check our sensitivity to our assumption of thermal electrons by ray-tracing a GRMHD snapshot with three different assumptions for the electron distribution function. A new plasma density scale is found for each image to match the total flux of the image with thermal electrons, 0.7 Jy. This snapshot corresponds to the MAD a = + 0.5 Rhigh = 80 Rlow = 10 aligned-field model. In the top row, we plot images blurred to EHT resolution, while in the bottom row we plot perfect-resolution images of Stokes in symmetric logarithmic scale. Compared to the leftmost image assuming thermal electrons, we find very little changes in the variable kappa model shown in the middle panel. This is broadly consistent with our findings in Event Horizon Telescope Collaboration et al. (2022d) for Sgr A* in total intensity. Thus, at least for this snapshot, exchanging a thermal distribution for a physically motivated nonthermal electron distribution has very little effect on Stokes . However, we report dramatic differences when switching to a constant kappa model with a value of κ = 5. As found in many other studies, nonthermal electrons make the image noticeably larger and more diffuse (e.g., Özel 2000; Mao et al. 2017; Fromm et al. 2022; Ricarte et al. 2023). Intriguingly, although the plasma density is decreased only by a factor of 4 compared to the thermal model, the Stokes ν signal almost entirely vanishes. This may be due to the fact that Faraday conversion is caused by the coldest relativistic electrons, which occur at a smaller fraction in κ models by definition. A single GRMHD snapshot (MAD a = + 0.5 Rhigh = 80 Rlow = 10 aligned field) ray-traced with three different electron distribution functions. The physically motivated variable kappa model produces both a Stokes x and Stokes ν image very similar to the thermal model. However, a model with fixed κ = 5 produces a much more diffuse Stokes x and extremely little Stokes ν [24-26].

References

1. Mistele, T., McGaugh, S., Lelli, F., Schombert, J., & Li, P. (2024). Indefinitely Flat Circular Velocities and the Baryonic Tully–Fisher Relation from Weak Lensing. *The Astrophysical Journal Letters, 969*(1), L3.
2. Li, J., & Nath, P. (2023). Big bang initial conditions and self-interacting hidden dark matter. *Physical Review D, 108*(11), 115008.
3. Chandra X-ray, NASA RELEASE 06-297, NASA Finds Direct Proof of Dark Matter.
4. Roelofs, F., Johnson, M. D., Chael, A., Janssen, M., Wielgus, M., Broderick, A. E., ... & Lowitz, A. E. (2023). Polarimetric geometric modeling for mm-VLBI observations of black holes. *The Astrophysical Journal Letters, 957*(2), L21.
5. Benisty, D., Davis, A. C., & Evans, N. W. (2023). Constraining Dark Energy from the Local Group Dynamics. *The Astrophysical Journal Letters, 953*(1), L2.
6. Patil, Y. S., Höller, J., Henry, P. A., Guria, C., Zhang, Y., Jiang, L., ... & Harris, J. G. (2022). Measuring the knot of non-Hermitian degeneracies and non-commuting braids. *Nature, 607*(7918), 271-275.
7. Relation entre structures de l'Univers et énergie sombre
8. Could massive gravitons be viable dark matter candidates?, Massive Gravitons as Feebly Interacting Dark Matter Candidates.
9. Cai, H., Cacciapaglia, G., & Lee, S. J. (2022). Massive gravitons as feebly interacting dark matter candidates. Physical review letters, 128(8), 081806.
10. Aad, G., Abbott, B., Abdallah, J., Abdel Khalek, S., Abdinov, O., Aben, R., ... & Barklow, T. (2014). Search for scalar diphoton resonances in the mass range 65–600 GeV with the ATLAS detector in pp collision data at s= 8 TeV. *Physical review letters,*

113(17), 171801.
11. ATLAS-CONF-2011-161, Search for the Standard Model Higgs boson in the diphoton decay channel with 4.9 fb-1 of ATLAS data at √s=7 TeV.
12. Bernstein polynomial.
13. Search for the Standard Model Higgs boson in the diphoton decay channel with 4.9 fb-1 of ATLAS data at √s=7 TeV, ATLAS-CONF-2011-161, 11 December 2011.
14. Search for the Standard Model Higgs boson in the diphoton decay channel with 4.9 fb-1 of ATLAS data at √s=7 TeV, ATLAS-CONF-2011-161, 11 December 2011.
15. Trefoil Knot.
16. Observations Constraining Dark Matter Clouds Around Spinning Black Holes in our Galaxy
17. Search for the Standard Model Higgs boson in the diphoton decay channel with 4.9 fb-1 of ATLAS data at √s=7 TeV, ATLAS-CONF-2011-161, 11 December 2011.
18. CMS-PAS-HIG-13-006, Search for the standard model Higgs boson in the Z boson plus a photon channel in pp collisions at sqrt-s = 7 and 8\,TeV.
19. ESA/Hubble & NASA, R. Tully.
20. ESA/Hubble & NASA, R. van der Burg.
21. Shen, Z., Danieli, S., Van Dokkum, P., Abraham, R., Brodie, J. P., Conroy, C., ... & Chowdhury, D. D. (2021). A tip of the red giant branch distance of 22.1±1.2 Mpc to the dark matter deficient galaxy NGC 1052–DF2 from 40 orbits of Hubble Space Telescope imaging. *The Astrophysical Journal Letters, 914*(1), L12.
22. Dark Matter & Dark Energy, Discover the universe's building blocks – matter, dark matter, and dark energy.
23. Alberdi, A., Cho, I., Dahale, R., Foschi, M., Fuentes, A., Gómez, J. L., ... & Event Horizon Telescope Collaboration. (2023). First M87 Event Horizon Telescope Results. IX. Detection of Near-horizon Circular Polarization.
24. Sumner, T. Strange New Phase of Matter Created in Quantum Computer Acts Like It Has Two Time Dimensions.
25. Dumitrescu, P. T., Bohnet, J. G., Gaebler, J. P., Hankin, A., Hayes, D., Kumar, A., ... & Potter, A. C. (2022). Dynamical topological phase realized in a trapped-ion quantum simulator. *Nature, 607*(7919), 463-467.
26. Asencio, E., Banik, I., Mieske, S., Venhola, A., Kroupa, P., & Zhao, H. (2022). The distribution and morphologies of Fornax Cluster dwarf galaxies suggest they lack dark matter. *Monthly Notices of the Royal Astronomical Society, 515*(2), 2981-3013.