ATOMIC AND ASTRING FUNCTIONS INTEGRALS (Python Code)

By Sergei Yu. Eremenko, PhD, Dr.Eng., Professor, Honorary Professor

- https://www.researchgate.net/profile/Sergei_Eremenko
- https://www.amazon.com/Sergei-Eremenko/e/B082F3MQ4L
- https://www.linkedin.com/in/sergei-eremenko-3862079
- https://www.facebook.com/SergeiEremenko.Author

Atomic functions (AF) described in many books and hundreds of papers have been discovered in 1970s by Academician NAS of Ukraine Rvachev V.L. (https://ru.wikipedia.org/w/index.php?oldid=83948367) (author's teacher) and professor Rvachev V.A. and advanced by many followers, notably professor Kravchenko V.F. (https://ru.wikipedia.org/w/index.php?oldid=84521570), H. Gotovac (https://www.researchgate.net/profile/Hrvoje_Gotovac), V.M. Kolodyazhni (https://www.researchgate.net/profile/Volodymyr_Kolodyazhny), O.V. Kravchenko (https://www.researchgate.net/profile/Oleg_Kravchenko) as well as the author S.Yu. Eremenko (https://www.researchgate.net/profile/Sergei_Eremenko) [1-4] for a wide range of applications in mathematical physics, boundary value problems, statistics, radio-electronics, telecommunications, signal processing, and others.

As per historical survey (https://www.researchgate.net/publication/308749839), some elements, analogs, subsets or Fourier transformations of AFs sometimes named differently (Fabius function, hat function, compactly supported smooth function) have been probably known since 1930s and rediscovered many times by scientists from different countries, including Fabius, W.Hilberg and others. However, the most comprehensive 50+ years' theory development supported by many books, dissertations, hundreds of papers, lecture courses and multiple online resources have been performed by the schools of V.L. Rvachev, V.A. Rvachev and V.F. Kravchenko.

In 2017-2020, Sergei Yu. Eremenko, in papers "Atomic Strings and Fabric of Spacetime", "Atomic Solitons as a New Class of Solitons", "Atomic Machine Learning" and book "Soliton Nature" [1-8], has introduced AString atomic function as an integral and 'composing branch' of Atomic Function up(x): AString'(x) = AString(2x+1) - AString(2x-1) = up(x)

AString function, is a smooth solitonic kink function by joining of which on a periodic lattice it is possible to compose a straight-line resembling flat spacetime as well as to build 'solitonic atoms' composing different fields. It may lead to novel models of spacetime and quantized gravity where AString may describe Spacetime Quantum, or Spacetime Metriant. Also, representing of different fields via shift and stretches of AStrings and Atomic Functions may lead to unified theory where AString may describe some fundamental building block of quantum fields, like a string, elementary spacetime distortion or metriant.

So, apart from traditional areas of AF applications in mathematical physics, radio-electronics and signal processing, AStrings and Atomic Functions may be expanded to Spacetime Physics, String theory, General and Special Relativity, Theory of Solitons, Lattice Physics, Quantized Gravity, Cosmology, Dark matter and Multiverse theories as well as Finite Element Methods, Nonarchimedean Computers, Atomic regression analysis, Atomic Kernels, Machine Learning and Artificial Intelligence.

1. Atomic Function up(x) (introduced in 1971 by V.L.Rvachev and V.A.Rvachev)

```
import numpy as np
import pylab as pl
pl.rcParams["figure.figsize"] = 9,6

####################################################################
##This script calculates the values of Atomic Function up(x) (1971)
####################################################################
```

```python
################### One Pulse of atomic function
def up1(x: float) -> float:
    #Atomic function table
    up_y = [0.5, 0.48, 0.460000017,0.440000421,0.420003478,0.400016184, 0.380053256, 0.360139056,
            0.340308139, 0.320605107,0.301083436, 0.281802850, 0.262826445, 0.244218000, 0.226041554,
            0.208361009, 0.191239338, 0.174736305, 0.158905389, 0.143991189, 0.129427260, 0.115840866,
            0.103044024, 0.9110444278e-01, 0.798444445e-01, 0.694444445e-01, 0.598444445e-01,
            0.510444877e-01, 0.430440239e-01, 0.358409663e-01, 0.294282603e-01, 0.237911889e-01,
            0.189053889e-01, 0.147363055e-01, 0.112393379e-01, 0.836100883e-02, 0.604155412e-02,
            0.421800000e-02, 0.282644445e-02, 0.180999032e-02, 0.108343562e-02, 0.605106267e-03,
            0.308138660e-03, 0.139055523e-03, 0.532555251e-04, 0.161841328e-04, 0.347816874e-05,
            0.420576116e-05, 0.167693347e-07, 0.354008603e-10, 0]
    up_x = np.arange(0.5, 1.01, 0.01)

    res = 0.
    if ((x>=0.5) and (x<=1)):
        for i in range(len(up_x) - 1):
            if (up_x[i] >= x) and (x < up_x[i+1]):
                N1 = 1 - (x - up_x[i])/0.01
                res = N1 * up_y[i] + (1 - N1) * up_y[i+1]
                return res
    return res

############### Atomic Function Pulse with width, shift and scale #############
def upulse(t: float, a = 1., b = 0., c = 1., d = 0.) -> float:
    x = (t - b)/a
    res = 0.
    if (x >= 0.5) and (x <= 1):
        res = up1(x)
    elif (x >= 0.0) and (x < 0.5):
        res = 1 - up1(1 - x)
    elif (x >= -1 and x <= -0.5):
        res = up1(-x)
    elif (x > -0.5) and (x < 0):
        res = 1 - up1(1 + x)
    res = d + res * c
    return res

############### Atomic Function Applied to list with width, shift and scale #############
def up(x: list, a = 1., b = 0., c = 1., d = 0.) -> list:
    res = []
    for i in range(len(x)):
        res.append(upulse(x[i], a, b, c, d))
    return res

x = np.arange(-2.0, 2.0, 0.01)
pl.title('Atomic Function up(x)')
pl.plot(x, up(x), label='Atomic Function')
pl.grid(True)
pl.show()
```

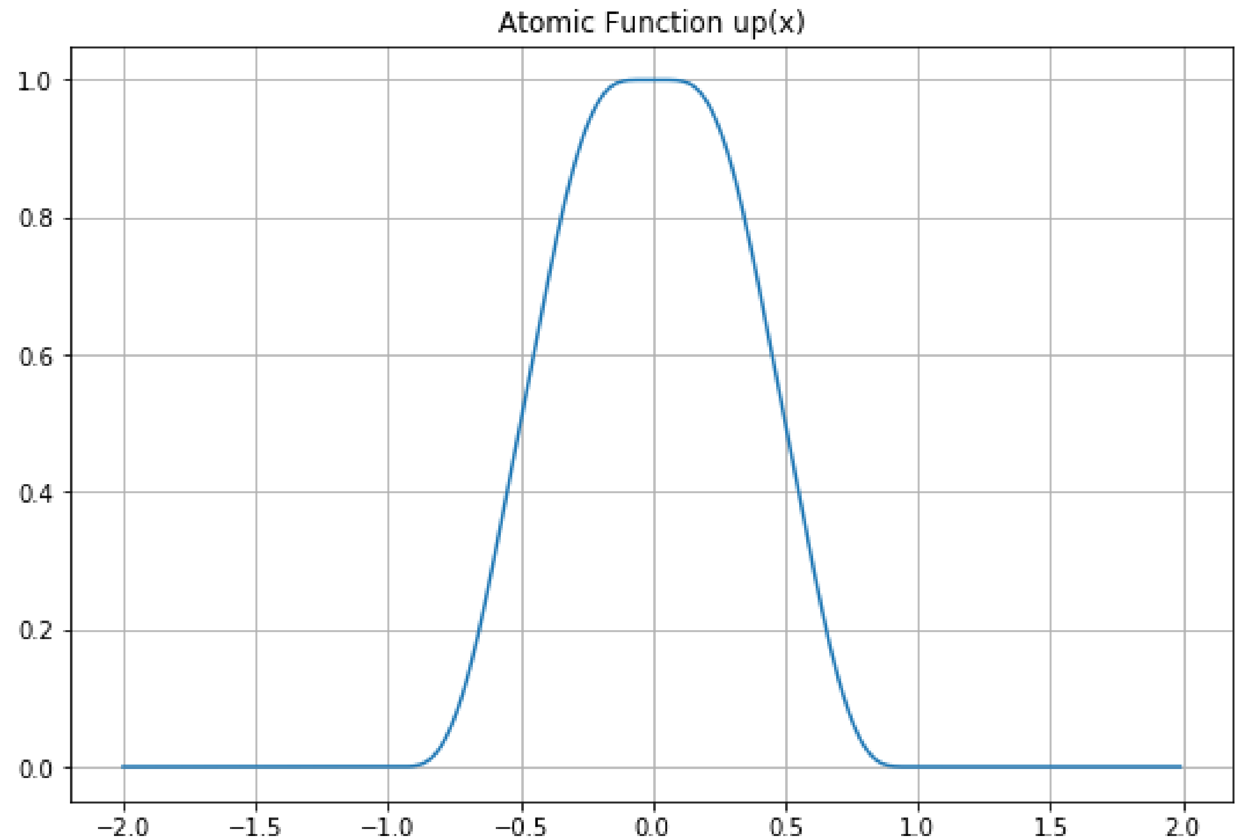

Figure 1: png

2. Atomic String Function (AString) is an Integral and Composing Branch of Atomic Function up(x) (introduced in 2017 by S. Yu. Eremenko)

AString function is solitary kink function which simultaneously is integral and composing branch of atomic function up(x)

AString'(x) = AString(2x+1) - AString(2x-1) = up(x)

```
############### Atomic String #############
def AString1(x: float) -> float:
    res = 1 * (upulse(x/2.0 - 0.5) - 0.5)
    return res

############### Atomic String Pulse with width, shift and scale #############
def AStringPulse(t: float, a = 1., b = 0., c = 1., d = 0.) -> float:
    x = (t - b)/a
    if (x < -1):
        res = -0.5
    elif (x > 1):
        res = 0.5
    else:
        res = AString1(x)
    res = d + res * c
    return res

###### Atomic String Applied to list with width, shift and scale #############
def AString(x: list, a = 1., b = 0., c = 1., d = 0.) -> list:
    res = []
    for i in range(len(x)):
        res.append(AStringPulse(x[i], a, b, c, d))
        #res[i] = AStringPulse(x[i], a, b, c)
    return res

###### Summation of two lists #############
def Sum(x1: list, x2: list) -> list:
    res = []
    for i in range(len(x1)):
       res.append(x1[i] + x2[i])
    return res
```

```
x = np.arange(-2.0, 2.0, 0.01)
pl.title('Atomic String Function')
pl.plot(x, AString(x, 1.0, 0, 1, 0), label='Atomic String')
pl.grid(True)
pl.show()
```

Atomic String, Atomic Function (AF) and AF Derivative plotted together

```
x = np.arange(-2.0, 2.0, 0.01)

#This Calculates Derivative
dx = x[1] - x[0]
dydx = np.gradient(up(x), dx)
```

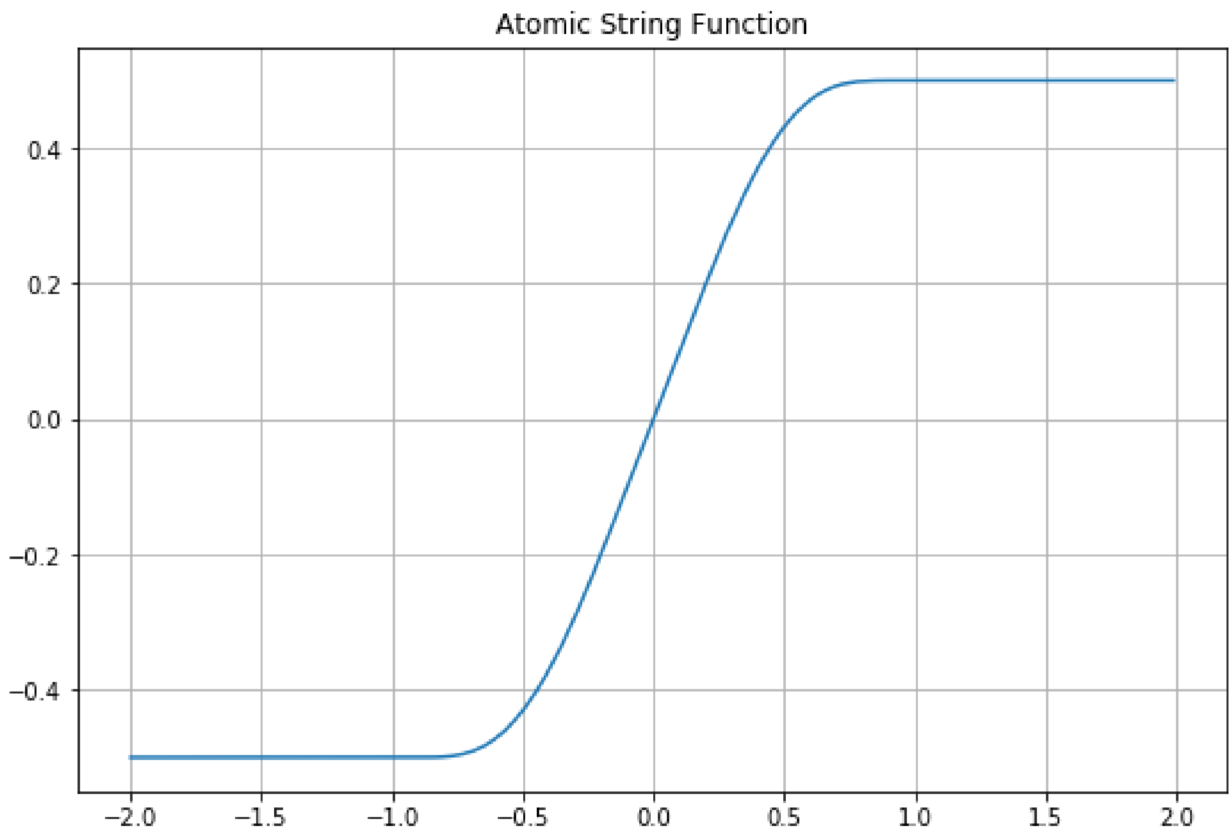

Figure 2: png

```
pl.plot(x, up(x), label='Atomic Function')
pl.plot(x, AString(x, 1.0, 0, 1, 0), linewidth=2, label='Atomic String Function')
pl.plot(x, dydx, '--', label='A-Function Derivative')

pl.title('Atomic and AString Functions')
pl.legend(loc='best', numpoints=1)
pl.grid(True)
pl.show()
```

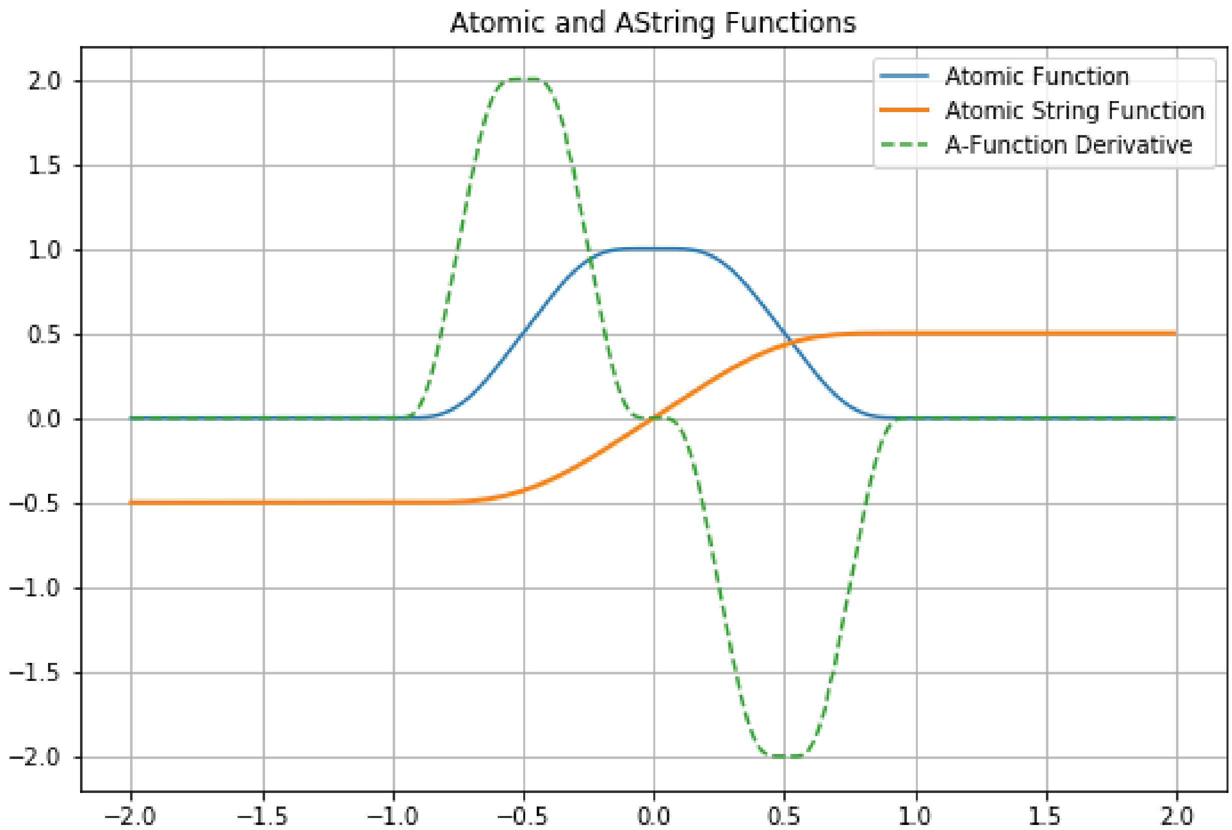

Figure 3: png

3. Atomic Function Integrals

3.1. Integral of basic Atomic Function up(x)

```
from scipy.integrate import simps
```

```
x = np.arange(-1.0, 1.0, 0.01)
I1 = simps(up(x), x)
print(I1)
```

```
1.0000000000000886
```

Summary - Integral(up(x),-1,1) = 1, as expected

3.2. Integrals of stretched Atomic Function c*up(x/a)

```
a = 0.5; c = 1
I2 = simps(up(x, a, 0., c), x)
print(I2)
```

```
0.5
```

```
a = 0.1; c = 2
I3 = simps(up(x, a, 0., c), x)
print(I3)
```

```
0.2
```

Summary - Integral(c*up(x/a),-1,1)* = c*a - Width times Height

3.3. Integral of the chain of Atomic Function pulses

The Atomic Function pulses superposition set at points -2, -1, 0, +1, +2... can exactly represent a Unity (number 1), so-called 'Partition of Unity'

1 = ... up(x-3) + up(x-2) + up(x-1) + up(x-0) + up(x+1) + up(x+2) + up(x+3) + ...

```
x = np.arange(-2.0, 2.0, 0.01)
pl.plot(x, up(x, 1, -1), '--', linewidth=1, label='Atomic Function at x=-1')
pl.plot(x, up(x, 1, +0), '--', linewidth=1, label='Atomic Function at x=0')
pl.plot(x, up(x, 1, -1), '--', linewidth=1, label='Atomic Function at x=-1')
pl.plot(x, Sum(up(x, 1, -1), Sum(up(x), up(x, 1, 1))), linewidth=2, label='Atomic Function Compounding'
pl.title('Partition of Unity - AF Compounding represent 1')
pl.legend(loc='best', numpoints=1)
pl.grid(True)
pl.show()
```

```
Continuum = Sum(up(x, 1, -1), Sum(up(x), up(x, 1, 1))) # Summation of three atomic functions
```

```
I4 = simps(Continuum, x)
print(I4)
```

```
3.000000000000089
```

Integral(Sum(up(x),n),-1,1) = n - Integral of Sum of n AF pulses equal to n. Important for quantisation

4. Integrals of Atomic String Function

4.1. Integral of basic Atomic String Function AString(x)

```
x = np.arange(-1.0, 1.0, 0.001)
I1 = simps(AString(x), x)
print(I1)
```

```
-0.00047999999999912
```

```
x = np.arange(0, 1.0, 0.001)
I1 = simps(AString(x), x)
print(I1)
```

```
0.3607624646774283
```

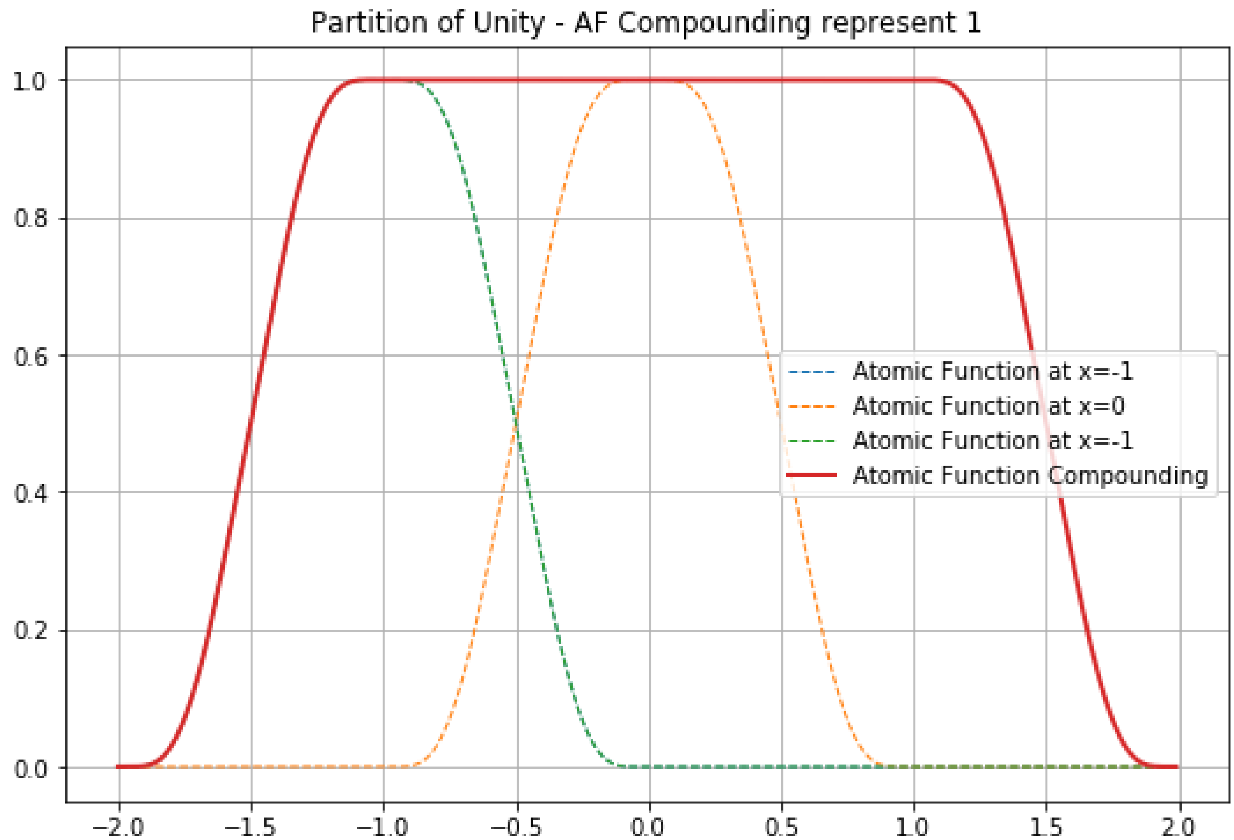
Partition of Unity - AF Compounding represent 1
1.0
0.8
0.6
0.4
0.2
0.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
Atomic Function at x=-1
Atomic Function at x=0
Atomic Function at x=-1
Atomic Function Compounding

Figure 4: png

```
x = np.arange(-1, 0.0, 0.001)
I1 = simps(AString(x), x)
print(I1)
```

```
-0.3612619646679367
```

Summary - Integral(AString(x),-1,1) = 0; Integral(AString(x),0,1) ~ 0.36;

4.2. Integrals of stretched Atomic String function c*AString(x/a)

```
a = 0.5; c = 1
x = np.arange(0, a, 0.001)

I2 = simps(AString(x, a, 0., c), x)
print(I2)
```

```
0.18012635733931628
```

```
a = 2; c = 5
x = np.arange(0, a, 0.001)

I3 = simps(AString(x, a, 0., c), x)
print(I3)
```

```
3.610174021767732
```

Summary: Integral(c*AString(x/a),0,a) = c*a*Integral(AString(x),0,1) meaning the length/energy of extended continuum can be composed from smaller parts

5. Some properties of Atomic and AString functions

5.1. AString Kinks and Solitonic Atoms

Solitonic mathematical properties of AString and Atomic Functions have been explored in author's paper [3] (Eremenko, S.Yu. Atomic solitons as a new class of solitons; 2018; https://www.researchgate.net/publication/329465767). They both satisfy differential equations with shifted arguments which introduce special kind of nonlinearity typical for all mathematical solitons.

AString belong to the class of Solitonic Kinks similar to sine-Gordon, Frenkel-Kontorova, tanh and others. Unlike other kinks, AStrings are truly solitary (compactly-supported) and also have a unique property of composing of both straight-line and solitonic atoms on lattice resembling particle-like properties of solitons.

Atomic Function up(x) is not actually a mathematical soliton, but a complex object composed from summation of two opposite AString kinks, and in solitonic terminology, is called 'solitonic atoms' (like bions).

5.2. Partition of Line from AStrings - resembles quantisation of space

Combination/summation of Atomic Strings can exactly represent a straight line: x = …Astring(x-2) + Astring(x-1) + AString(x) + Astring(x+1) + Astring(x+2)…

x = …Astring(x-2) + Astring(x-1) + AString(x) + Astring(x+1) + Astring(x+2)…

Partition based on AString function with width 1 and height 1

```
x = np.arange(-3, 3, 0.01)

pl.plot(x, AString(x, 1, -1.0, 1, 0),   '--',  linewidth=1, label='AString 1')
pl.plot(x, AString(x, 1, +0.0, 1, 0),   '--',  linewidth=1, label='AString 2')
pl.plot(x, AString(x, 1, +1.0, 1, 0),   '--',  linewidth=1, label='AString 3')
```

```
AS2 = Sum(AString(x, 1, -1.0, 1, 0), AString(x, 1, +0.0, 1, 0))
AS3 = Sum(AS2, AString(x, 1, +1.0, 1, 0))
pl.plot(x, AS3,  label='AStrings Sum', linewidth=2)

pl.title('Atomic Strings compose Line')
pl.legend(loc='best', numpoints=1)
pl.grid(True)
pl.show()
```

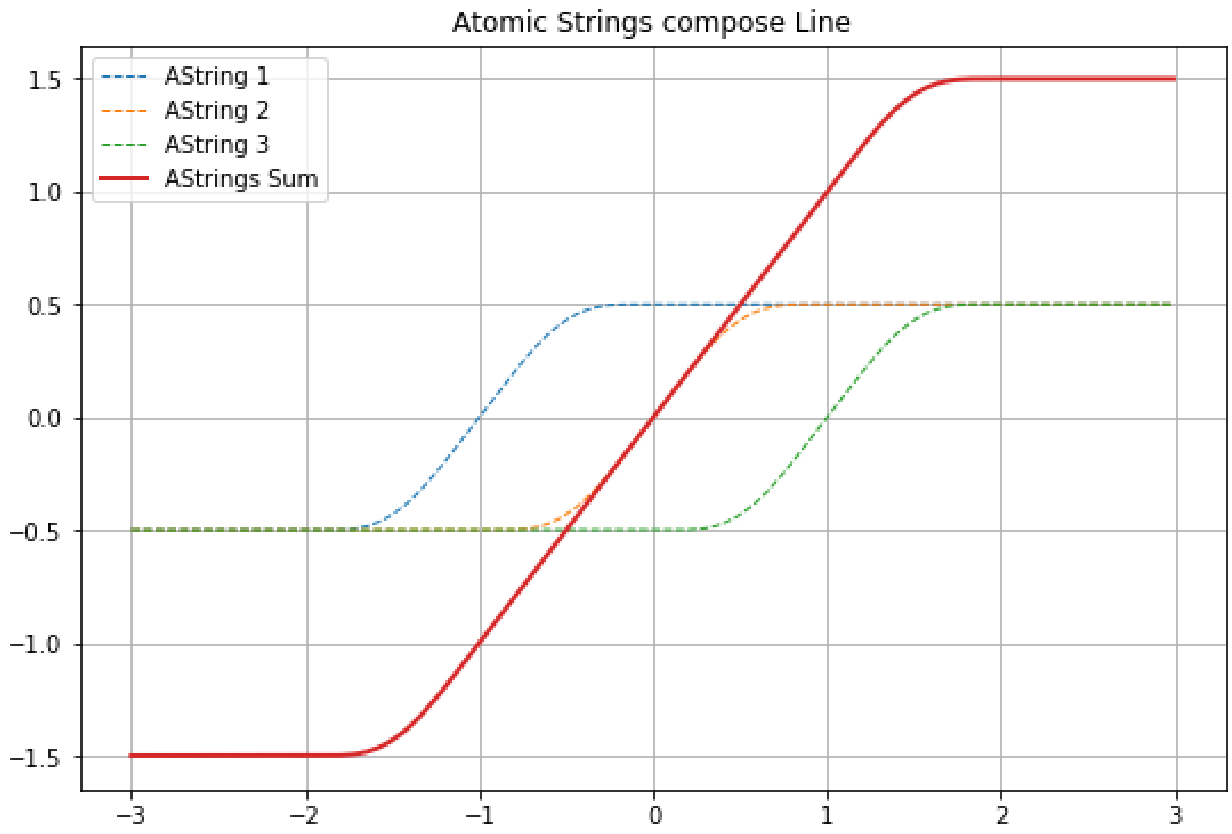

Figure 5: png

Partition based on AString with certain width and height depending on a size of 'quanta'

```
x = np.arange(-40.0, 40.0, 0.01)

width  = 10.0
height = 10.0

#pl.plot(x, ABline (x, 1, 0),             label='ABLine 1*x')
pl.plot(x, AString(x, width, -3*width/2, height, -3*width/2), '--',  linewidth=1, label='AString 1')
pl.plot(x, AString(x, width, -1*width/2, height, -1*width/2), '--',  linewidth=1, label='AString 2')
pl.plot(x, AString(x, width, +1*width/2, height, +1*width/2), '--',  linewidth=1, label='AString 3')
pl.plot(x, AString(x, width, +3*width/2, height, +3*width/2), '--',  linewidth=1, label='AString 4')

AS2 = Sum(AString(x, width, -3*width/2, height, -3*width/2), AString(x, width, -1*width/2, height, -1*wi
```

```
AS3 = Sum(AS2, AString(x, width,+1*width/2, height, +1*width/2))
AS4 = Sum(AS3, AString(x, width,+3*width/2, height, +3*width/2))
pl.plot(x, AS4,  label='AStrings Joins', linewidth=2)

pl.title('Atomic Strings Combinations')
pl.legend(loc='best', numpoints=1)
pl.grid(True)
pl.show()
```

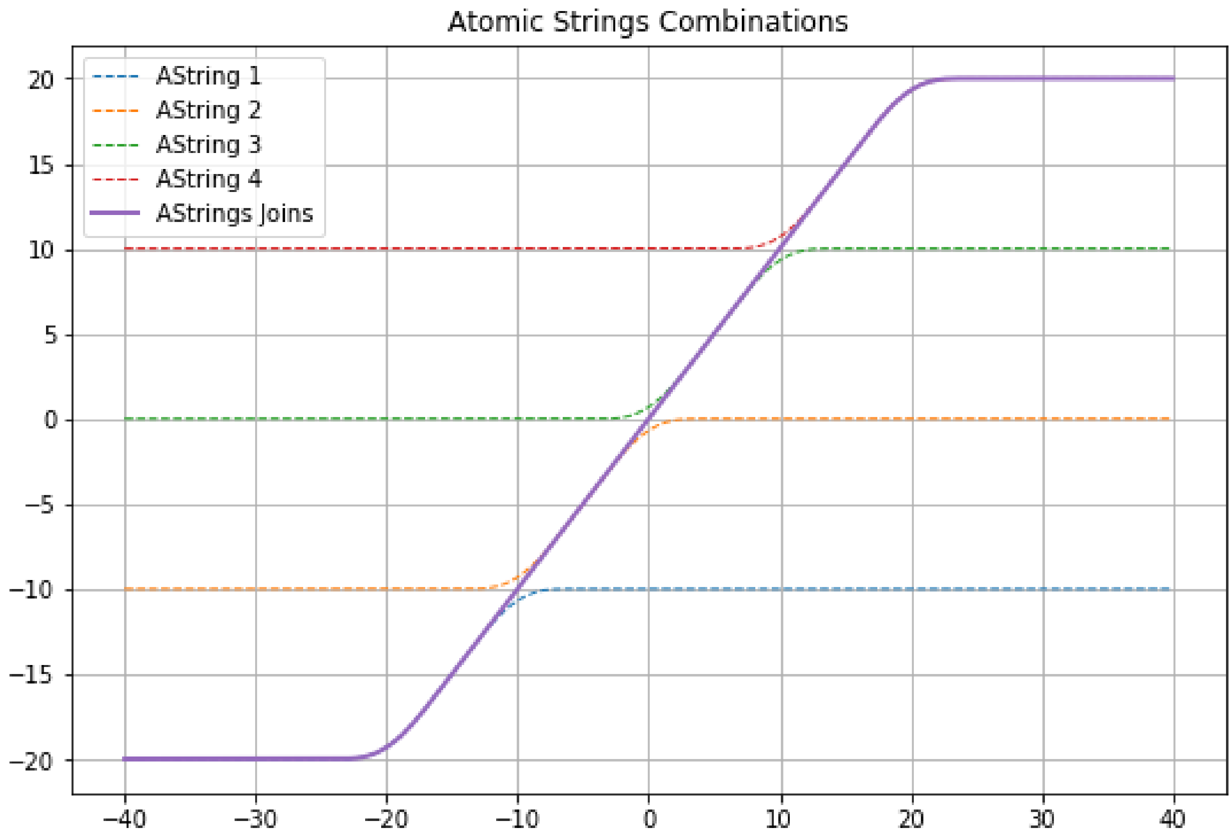

Figure 6: png

5.3. Representing curved continua via AStrings and Atomic Functions

Shifts and stretches of Atomic adn AString functions allows reproducing curved surfaces (eq curved spacetime). Details are in author's papers "Atomic Strings and Fabric of Spacetime", "Atomic Solitons as a New Class of Solitons".

```
x = np.arange(-50.0, 50.0, 0.1)
dx = x[1] - x[0]

CS6 = Sum(up(x, 5, -30, 5, 5), up(x, 15, 0, 15, 5))
CS6 = Sum(CS6, up(x, 10, +30, 10, 5))
pl.plot(x, CS6,  label='Spacetime Density distribution')
```

```
IntC6 = np.cumsum(CS6)*dx/50
pl.plot(x, IntC6, label='Spacetime Shape (Geodesics)')

DerC6 = np.gradient(CS6, dx)
pl.plot(x, DerC6, label='Spacetime Curvature')

LightTrajectory = -10 -IntC6/5
pl.plot(x, LightTrajectory, label='Light Trajectory')

pl.title('Shape of Curved Spacetime model')
pl.legend(loc='best', numpoints=1)
pl.grid(True)
pl.show()
```

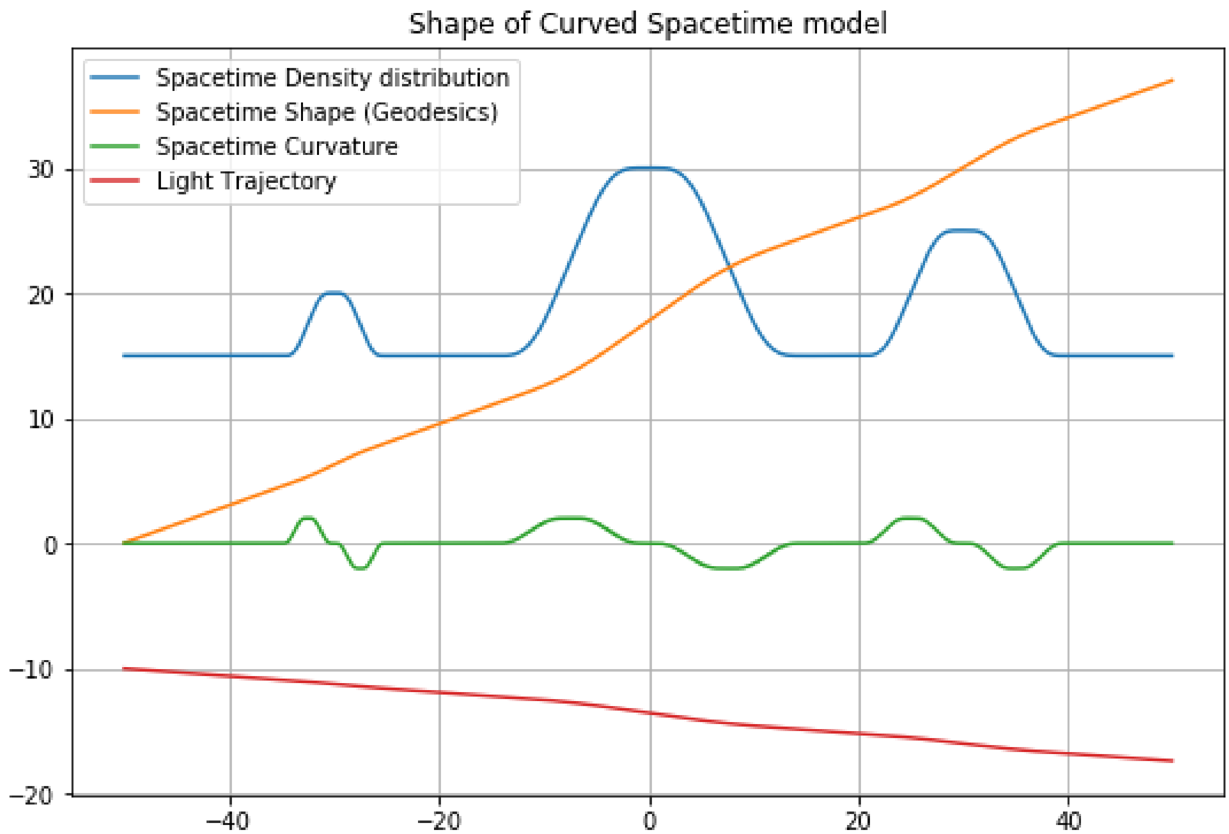

Figure 7: png

6. 'Soliton Nature' book by S.Eremenko

6.1. AStrings and Atomic functions are described in the book 'Soliton Nature'

Soliton Nature book is easy-to-read, pictorial, interactive book which uses beautiful photography, video channel, and computer scripts in R and Python to demonstrate existing and explore new solitons – the magnificent and versatile energy concentration phenomenon of nature. New class of atomic solitons can be used to describe Higgs boson ('the god particle') fields, spacetime quanta and other fundamental building blocks of nature.

```python
#pl.rcParams["figure.figsize"] = 16,12
book = pl.imread('BookSpread_small.png')
pl.imshow(book)
```

```
<matplotlib.image.AxesImage at 0x227db94b508>
```

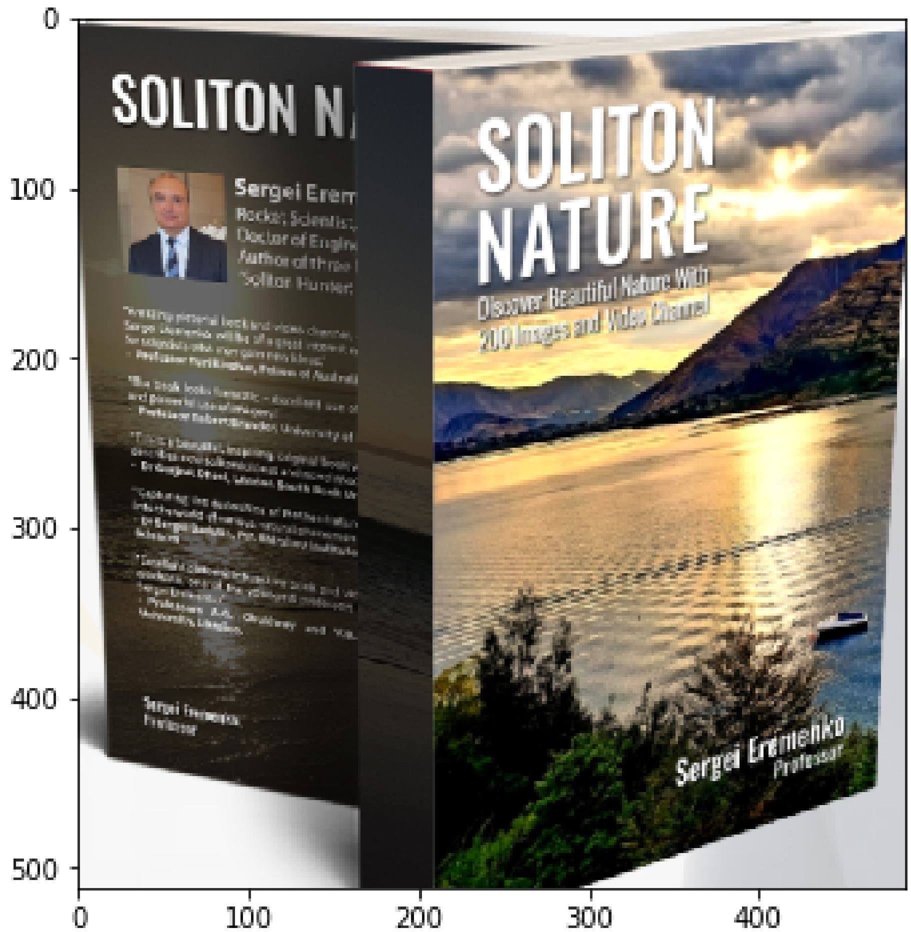

Figure 8: png

6.2. 'Soliton Nature' Video Channel, Book Trailer and Web Site

Video channel https://www.youtube.com/channel/UCexT5iyczZH2HY1-jSafFeQ features amazing solitonic phenomena in nature - welcome to subscribe

Book web site www.solitonnature.com contains book chapters and amazing video-gallery

Book Trailers: https://www.youtube.com/watch?v=cZMZdW_3J84, https://www.youtube.com/watch?v=2lABLpIcevo, https://www.youtube.com/watch?v=hQ3zGFEnSWI

6.3. 'Soliton Nature' book in major bookstores around the globe

- Amazon US https://www.amazon.com/gp/product/1951630777,
- Amazon UK https://www.amazon.co.uk/Sergei-Eremenko/e/B082F3MQ4L,
- Amazon Germany https://www.amazon.de/Sergei-Eremenko/e/B082F3MQ4L,
- Amazon France https://www.amazon.fr/Soliton-Nature-Discover-Beautiful-Channel/dp/1951630777,

- Google Books https://books.google.com.au/books/about/Soliton_Nature.html?id=d2zNDwAAQBAJ,
- Kindle eBooks of your country, like https://www.amazon.com/Soliton-Nature-Discover-Beautiful-Channel-ebook/dp/B082B5PP6R.
- Book web site www.solitonnature.com

7. Online Source Code Repositories

This code is available on GitHub: https://solitonscientific.github.io/AtomicString/AFAString.html

See also

- https://github.com/SolitonScientific
- https://solitonscientific.github.io/AtomicSoliton/AtomicSoliton.html
- https://solitonscientific.github.io/AtomicString/AtomicString1.html
- https://solitonscientific.github.io/AtomicMachineLearning/AtomicMachineLearning.html
- https://notebooks.azure.com/Soliton/projects/AtomicString1
- https://notebooks.azure.com/Soliton/projects/solitonnature
- https://notebooks.azure.com/Soliton/projects/geosolitons

References

1. Eremenko, S.Yu. Atomic Strings and Fabric of Spacetime. Journal Achievements of Modern Radioelectronics, 2018. No.6. https://www.researchgate.net/publication/329455498
2. Eremenko, S.Yu. Atomic solitons as a new class of solitons. Journal Nonlinear World, No.6, Vol.16, 2018, p.39-63. DOI: 10.18127/j20700970-201806-06. https://www.researchgate.net/publication/329455498
3. Eremenko, S.Yu. Atomic solitons as a new class of solitons (English, with Russian Abstract). Journal Nonlinear World, No.6, Vol.16, 2018, p.39-63. DOI: 10.18127/j20700970-201806-06. https://www.researchgate.net/publication/329465767
4. Eremenko, S.Yu. Soliton Nature: Discover Beautiful Nature with 200 Images and Video Channel. ISBN: 978-1-951630-77-5. https://www.amazon.com/gp/product/1951630777; https://www.researchgate.net/publication/32122
5. Eremenko, S.Yu. Atomic Machine Learning. Journal Neurocomputers. 2018, No.3. https://www.researchgate.net/publica
6. ResearchGate project: https://www.researchgate.net/project/Atomic-Strings-Quantum-of-Spacetime-and-Gravitation
7. ResearchGate project: https://www.researchgate.net/project/Atomic-String-and-Atomic-Function-New-Soliton-Candidates
8. ResearchGate project: https://www.researchgate.net/project/Atomic-Strings-Quantum-of-Spacetime-and-Gravitation

By Sergei Eremenko, PhD, Dr.Eng., Professor, Honorary Professor https://www.researchgate.net/profile/Sergei_Eremenko https://www.amazon.com/Sergei-Eremenko/e/B082F3MQ4L https://www.linkedin.com/in/sergei-eremenko-3862079 https://www.facebook.com/SergeiEremenko.Author/