On linkage between incompressible integers as distance and 1/r potential (v1)

Dara O Shayda
dara@compclassnotes.com
February 14 2025

Due to its novelty and incompleteness, this technical note is versioned.

TODO:
1. Reinstate the results for Real numbers Kolmogorov complexity
2. In the 2.0 embedding in $\mathbb{R}^3$, the last computations 2.0.1 require better rigor
3. Further research into the $1/r$ and Laplacian functional forms and decompositions into wave functions [5]
4. Revisit the Wave Functions section and offer better applicable examples to further the relationship to Kolmogorov Complexity
5. Add a new computational addendum for Taylor and Fourier expansions and harmonic waves decompositions
6. Categorify incompressible distance integers and the $1/r$ potential
7. Try to see the forest for the trees [6]

Abstract

The main result: assuming distances are numericized as incompressible integers, given two objects, one stationary and the other moving, the rate of change of the measure of their **distantial randomness** is that of the potential form $1/r$. This form is known as the Newtonian potential. If the incompressible assumption dropped then the potential form vanishes as well (conjecture). The supplementary results by Whittaker: for any force varying as the inverse square of the distance, the potential of such a force satisfies both Laplace's equation and the wave equation, and can be analyzed into simple plane waves propagating with constant velocity. The sum of these waves, however, does not vary with time, i.e. standing waves. Therefore, the $1/r$ potential can be defined as summation of waves. Thus the linkage between the incompressible integers and particular standing waves in physics.

Keywords: Kolmogorov Complexity, inverse distance potential, incompressible integer, Bessel functions, Laplacian equation, wave functions, standing waves, potential functional form, arbitrary precision arithmetic, Vandermonde determinant, graph embedding, Fourier Taylor series.

Software

Scripts: Arbitrary Precision Arithmetic, Symbolic integration and series expansions for Taylor and Fourier provided in Wolfram Mathematica 14.2 .

Conspectus

● *c1*: Incompressible integer distances of moving objects are intrinsically imbued with $1/r$ potential

● *c2*: Given any graph with vertices indicating moving objects and the edges labeled with Incompressible integer distances, is fully edge-length-preserving embeddable in to $\mathbb{R}^3$ [3,4]

● *c3*: Such graph embeddings coordinatize the $1/r$ potential form namely:

$$\frac{1}{r} = \frac{1}{\sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2}}$$

● *c4*: The latter coordinatized potential form is always decomposable into wave functions [5]:

$$\frac{1}{r} = \frac{1}{\sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2}} = \pm \frac{1}{2\pi} \int_0^{2\pi} \frac{1}{((z-c) + i\,(x-a)\,\mathrm{Cos}(u) + i\,(y-b)\,\mathrm{Sin}(u))}\, du$$

● *c5*: Incompressible integer distances are intrinsically mechanical bodies [8]

1.0 $1/r$ potential derivation

● 1. Incompressible word of any language is governed by the following identity:

$$|\,w\,| = K(w)$$

Remark 1.0.1: We assume the length of w and its Kolmogorov complexity to be quite high and as such no need to worry about the customary O-notations and additions of constants and prefix obfuscations. All the reported results remain the same regardless of the said choices.

● 2. Assuming *r* being the distance between two objects and further assume binary base for the numerization of the distance:

$$|\,r\,| = K(r) \text{ and } |\,r\,| = \log_2(r)$$

Remark 1.0.2: Assume *r* being the actual integer number and expressed as a list of integer tokens in some base e.g. base 2 or base 10 in our later examples.

● 3. The above identities state: Due the incompressible nature of the integer-token word then the shortest program that outputs the said integer would simply be a list of print calls to output each integer token. The total number of such print calls is then the length of the input integer-word namely $|\ r\ | = K(r)$.

● 4. Assuming the binary numericization of the integer word then logarithm base 2 of the distance r encodes the length of the word. If we introduce the . decimal token to avoid overflow and underflow in rudimentary arithmetic operations, the said length will be a Real number with digits after the decimal point token. While the numericization of the r might differ the relationships between the terms remains the same and the inferences firmly valid.

● 5. By using the single variable derivative applied to the length of the integer $|\, r\, |$

$$\frac{d\,(K(r))}{d\,r} = \frac{d\,(|r|)}{d\,r} = \frac{d\,(\log_2(r))}{d\,r} = \frac{1}{\log_e(2)\,r}$$

and the $1/r$ potential appears! Therefore:

Theorem 1.0.3: Assuming distances are numericized as incompressible integers, given two objects, one stationary and the other moving, the rate of change of the measure of their **distantial randomness** is of the form $1/r$, namely the potential form.

Remark 1.0.4: The computations above and the inferences did not assume any understanding of physics and almost no geometry save the scalar concept of distance. In conclusion, the $1/r$ form in this theoretical setting primarily concerns itself with the randomness of the distances. Paraphrase: the attractive forces with $1/r$ potential are direct consequences of the incompressible numericization of distances.

● 6. There are many ways of measuring straight-line distances. For example, it can be done directly using a ruler, or indirectly with a radar (for long distances) or interferometry (for very short distances). The measurement of distance requires an Emitter and an Absorber, in real physical spaces Emitter and Absorber are made from matter with attractive force of gravity.

● 7. No matter how near or far distances are measured the distance numbers are always without any exceptions noisy. There is no absolute distance between an Emitter and an Absorber we can numerically be sure of.Therefore the distance numericization contains considerable measure of randomness or **distantial randomness**.

Given the two arguments 6-7 we can cautiously infer:

Corollary 1.0.5: Assuming an incompressible integer for distance between two objects then an attractive force between them appears.

Conjecture 1.0.6: In absence of an incompressible integer for distance between two objects then the attractive force does not appear.

● 8. Our understanding of the numericization of distances we measure is incomplete. What we think these numbers are and what actually they do or how they were obtained are far far apart.

Distantial: Pertaining to distance.
1885 July 4, T. Oughton, “On the Secondary Nature of Monocular Relief”, in The Lancet, volume 2, page 9:
In order for the hypothesis to cover this fact, it would be indispensable that such objects should subtend a uniform visual angle at every distance, and that they do not do so is proof that an element has been introduced which may regulate **distantial** perception.
Source: https://en.wiktionary.org/wiki/distantial

1.1 Calculus of Incompressible Integers

Our aim is to compute the Difference Quotient of the K(r) given r and the K(r) are integers. Furthermore, per Kolmogorov’s own requirements for “amount of information content” [7] the length of r is required to be substantial.

To that end arbitrary precision arithmetic is provided below, for non-binary base of integer representation, to validate the concepts of scaling the integers into decimal rationals for computing a meaningful Difference Quotient.

Compute a 200 digit random integer in base 10 as String, left pad with “1.” :

```
In[1]:= seed1 = "incompressible/random integer base 10 distance";
        SeedRandom[seed1];

        SetPrecision[
          m = 200;
          r1 = ToExpression[
            "1." <> (r0 = (StringJoin@Table[ToString@RandomChoice@Range[0, 9], m]))],

          m + 1
         ];

        SetPrecision[r1, m + 1]

Out[4]= 1.885906200938984865703794776759323287562447947659917125107708444332020318957305 7
        521597122835950569386525062523987551236683338098557734488704968021199282325866 43
        06818029125857867559905844160003761409011
```

The integer is not transformed, digit by digit to a number of format 1.ddd where ddd is the list of the digits of the integers.

This avoids underflow and overflow arithmetic anomaly and standard practice for such computations.

```
In[5]:= StringLength@ToString[SetPrecision[r1, m + 1]]

Out[5]= 202
```

- The 200 digits are placed after the decimal point or else overflow and or underflow in arithmetic shall occur.
- 1 is added before the decimal point in order to avoid negative logarithms. This addition is not necessary and can be handled differently. That increments the length by 1 to 201.
- **.** the decimal point increments the length again by 1 to 202
- The position of the decimal point is of no consequence to the final inferential results

Now we compute a Δ which will be added to r1 to compute r2. Δ has left 0-padding of length 3 to reduce its magnitude, and additional 8 random digits padded to the right to serve as r2 having a larger length as an integer.

r1 has the length m+1 while r2 has the length m+n+3.

The final Δ is ≈ 0.00088004... format

In[6]:=
```
seed2 = "incompressible/random integer base 10 distance 2";
SeedRandom[seed2];

SetPrecision[
  n = 8;
  Δ = (ToExpression@(("." <> "000" <>
        ((StringJoin@Table[ToString@RandomChoice@Range[0, 9], m + n]))))),
  m + n + 3
 ];

SetPrecision[Δ, m + n + 3]
```

Out[9]=
```
0.000880042085362235460626015609623560792549448046925141444037957095341831662170744151017880874433429388985274421459991121269834916929706776608949155983809980652301677752678526536233454747901654331501001860423401500
```

Before we compute the r2, carefully review the precision digits of the r1 namely m+1 vs. m+n+3 precision digits. Note that none of the digits of r1 are missing between the two precisions namely no truncations:

In[10]:=
```
SetPrecision[r1, m + 1]
```

Out[10]=
```
1.885906200938984865703794776759323287562447947659917125107708444332020318957305752159712283595056938652506252398755123668333809855773448870496802119928232586643068180291258578675599058441600037614090 11
```

In[11]:=
```
SetPrecision[r1, m + n + 3]
```

Out[11]=
```
1.8859062009389848657037947767593232875624479476599171251077084443320203189573057521597122835950569386525062523987551236683338098557734488704968021199282325866430681802912585786755990584416000376140901 10000000000
```

Right 0-padding counts to 10 since the padding length should be m+n+3 - (m + 1)

In[12]:=
```
m + n + 3 - (m + 1)
```

Out[12]=
```
10
```

Compute r2 = 1 + Δ however r1 with precision m+n+3. This really not necessary but mostly was added for manual inspection to make sure the integers have the lengths as expected:

In[15]:=
```
SetPrecision[
  r2 = SetPrecision[r1, m + n + 3] + SetPrecision[Δ, m + n + 3],
  m + n + 3
 ];

SetPrecision[r2, m + n + 3]
```

Out[16]=
```
1.8867862430243471011644207923689468483549973957068422665517464014273621506194764
 96310730164469490368041491526820215114789603644772703155647105751275912042567295
 369858043937105211832513189501691945591111860423401
```

Make sure the length of r2 is correct:

In[17]:=
```
StringLength@ToString[SetPrecision[r2, m + n + 3]]
m + n + 3 + 1
```

Out[17]=
```
212
```

Out[18]=
```
212
```

● for the length of r2 we obtain m+n+3 +1 due to the increment of 1 for decimal .

Finally compute the Log of both r1 and r2 however with m+1 precision since the r1 had the same length and that sets the precision of our arithmetic model. This argument is unnecessary and does not impact the final results, but necessary for clearer review of the computations.

l1 is the log of r1 and r1 has format 1.ddd:

In[19]:=
```
SetPrecision[l1 = N@Log[10, SetPrecision[r1, m + n + 3]],  m + 1];

SetPrecision[l1, m + 1]
```

Out[20]=
```
0.2755200884926984339351463404454989358782768249511718750000000000000000000000000
 00000000000000000000000000000000000000000000000000000000000000000000000000000000
 000000000000000000000000000000000000000000
```

l2 is the log of r2 and r2 has format 1.ddd:

In[21]:=
```
SetPrecision[l2 = N@Log[10, SetPrecision[r2, m + n + 3]],  m + 1];

SetPrecision[l2, m + 1]
```

Out[22]=
```
0.2757227010481211326720085708075203001499176025390625000000000000000000000000000
 00000000000000000000000000000000000000000000000000000000000000000000000000000000
 000000000000000000000000000000000000000000
```

l2 in Log-scale is ≃ 0.000202612... digits longer:

In[23]:=
```
SetPrecision[l2, m + 1] - SetPrecision[l1, m + 1]
```

Out[23]=
```
0.0002026125554226987368622303620213642716407775878906250000000000000000000000000\
 00000000000000000000000000000000000000000000000000000000000000000000000000000000\
 000000000000000000000000000000000000000000
```

Difference Quotient: subtract the two logs and divide by the actual two integers corresponding to the two distances, with m+1 precision:

In[24]:=
```
(SetPrecision[l2, m + 1] - SetPrecision[l1, m + 1]) /
 (SetPrecision[r2, m + 1] - SetPrecision[r1, m + 1])
```

Out[24]=
```
0.2302305296448420133295778386848709098056815215112886161320217017340691486921265\
 1106374979976783452119691754249263618211470312835344983990948666516144545447628\
 9754473896994958769855497338808292055 00
```

Difference Quotient is ≈ 0.230230...

Plugin the r1 in derivative of log and the **form 1/r results**:

In[25]:=
```
N[1 / (Log[10] * r1)]
```

Out[25]=
```
0.230284
```

1 / (Log[10] * r1 is ≈ 0.230284...

Notice that the (l2-l1)/(r2-r1) and 1/(Log[10]*r1) are quite close and equal at the limit.

2.0 $\mathbb{R}^3$ length preserving graph embedding

In order to compute the wave functions for 1/r we need to coordinatize the incompressible integer distances. And that we can accomplish with the following graph embedding algorithm. (for more complete treatments see [3,4]).

💡 The author added this simple algorithm here to demonstrate that: any finite graph can be embedded $\mathbb{R}^3$ while edge-length preserved. Therefore if have a theory or computation about the distances of objects, as long as their count is finite, we can extend the theory and computations into $\mathbb{R}^3$ and make it closer to our physical reality and our instrumentation.

● Define $\mathscr{C}$ be the curve $\{p(t) : t \in \mathbb{R}\}$ in $\mathbb{R}^3$:

$$p : \mathbb{R} \longrightarrow \mathbb{R}^3$$
$$t \longrightarrow p(t) = \left(t,\ t^2,\ t^3\right)$$

● For any four distinct t_1, t_2, t_3, t_4 in $\mathbb{R}$, the volume of the tetrahedron $\mathscr{T}$ formed by $p\,(t_i) \in \mathscr{C}$ is proportional to a Vandermonde determinant:

The general formula for the volume of the tetrahedron:

$V = 1/(3!) \mid a \cdot (b \times c) \mid$

In coordinates:

$$V = \frac{1}{3!}\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix}$$

In the case of 4 distinct points lying on the curve $\mathscr{C}$ the Vandermonde determinant is formed and evaluates non-zero:

$$\det\begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \\ 1 & t_4 & t_4^2 & t_4^3 \end{bmatrix} \neq 0$$

The volume V is proportional (1/(3!)) to the non-zero Vandermonde determinant .

Therefore any 4 distinct choices of t_1, t_2, t_3, t_4 in $\mathbb{R}$ results to a unique tetrahedron with non-zero volume.

● **2.0.1**: To preserve the lengths for any t_i place a Sphere at $p(t_i) \in \mathscr{C}$ with the radius equal to the required length, compute the intersections with $\mathscr{C}$, select any intersection and obtain the $p(t_{i+1}) \in \mathscr{C}$, and then so on to get 4 embedded vertices . This computes the tetrahedron with 4 embedded points with the required distances preserved. Continue through all the vertices, partitions of 4, in the same similar fashion to complete the embedding of the vertices in $\mathbb{R}^3$ while preserving their distances. (**TODO**: *better computations and rigorous arguments*)

● A larger formulation could easily guarantee a complete graph embedding in $\mathbb{R}^3$ with all the lengths/distances preserved

3.0 1/*r* potential decomposition into wave functions

These computations for the most part are inspired by Whittaker [5] from 122 years ago!

Let's compute the Laplacian of the 1/*r* :

$$\frac{1}{r} = \frac{1}{\sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2}}$$

```
In[1]:= Simplify@Laplacian[1/Sqrt[(x - a)^2 + (y - b)^2 + (z - c)^2], {x, y, z}]
Out[1]= 0
```

After simplification the Laplacian is 0.

For ease of reading, compute the integral of $1/(Z + i * X * \text{Cos}[u] + i * Y * \text{Sin}[u])$ from 0 to 2π:

```
In[2]:= Integrate[1/(Z + i * X * Cos[u] + i * Y * Sin[u]), {u, 0, 2*π}]
```

$$\text{Out[2]=} \begin{cases} -\frac{2\pi}{\sqrt{X^2+Y^2+Z^2}} & \text{Abs}\left[\frac{Z-\sqrt{X^2+Y^2+Z^2}}{X-i\,Y}\right] \geq 1 \;\&\&\; \text{Abs}\left[\frac{Z+\sqrt{X^2+Y^2+Z^2}}{X-i\,Y}\right] < 1 \\ \frac{2\pi}{\sqrt{X^2+Y^2+Z^2}} & \text{Abs}\left[\frac{Z-\sqrt{X^2+Y^2+Z^2}}{X-i\,Y}\right] < 1 \;\&\&\; \text{Abs}\left[\frac{Z+\sqrt{X^2+Y^2+Z^2}}{X-i\,Y}\right] \geq 1 \\ 0 & \text{True} \end{cases} \quad \text{if condition}$$

Finally replace X by (x - a) and Y by (y - b) and Z by (z - c) and establish the identity:

$$\frac{1}{r} = \frac{1}{\sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2}} = \pm \frac{1}{2\pi} \int_0^{2\pi} \frac{1}{((z-c) + i\,(x-a)\,\text{Cos}(u) + i\,(y-b)\,\text{Sin}(u))}\,du$$

The term $i\,(x-a)\,\text{Cos}(u) + i\,(y-b)\,\text{Sin}(u)$ is an indicator of underlying wave structure that in full summation appears as the resultant $1/r$.

Or you might fathom that $1/r$ is an emergent macro computational concept comprised of vast micro waving substructures.

Oddly enough in the publications only the $+\frac{1}{2\pi}$ is mentioned.

We can easily find values for X, Y and Z to force a negative value for the integration which its absolute value is the $+1/r$:

```
In[3]:= Integrate[(1/(Z + i * X * Cos[u] + i * Y * Sin[u])) /. {X → 46, Y → -8/5, Z → -6},
          {u, 0, 2*π}]/(2*Pi)
        (*+1/r*)
        1/Norm[{X, Y, Z} /. {X → 46, Y → -8/5, Z → -6}]
```

$$\text{Out[3]=} -\frac{5}{2\sqrt{13466}}$$

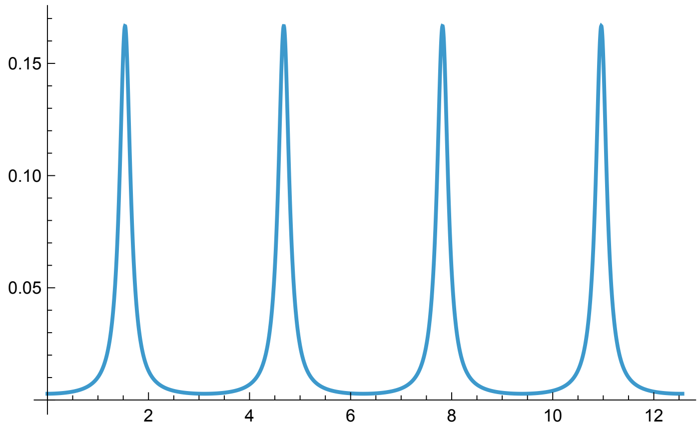
$$\text{Out[4]=} \frac{5}{2\sqrt{13466}}$$

3.1 wave functions

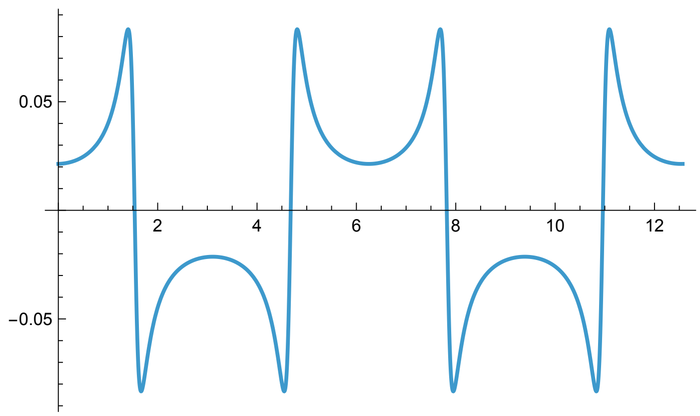
Re and Im part of $1/(Z + i * X * \text{Cos}[u] + i * Y * \text{Sin}[u])$ are periodic and this is the first indication of the underlying Wave structures:

```
In[1]:= re = Re[(1 / (Z + i * X * Cos[u] + i * Y * Sin[u])) /. {X → -46, Y → +8/5, Z → +6}];
im = Im[(1 / (Z + i * X * Cos[u] + i * Y * Sin[u])) /. {X → -46, Y → +8/5, Z → +6}];
Plot[re, {u, 0, 4 * Pi}, PlotRange → Full]
Plot[im, {u, 0, 4 * Pi}, PlotRange → Full]
```

Out[3]=



Out[4]=

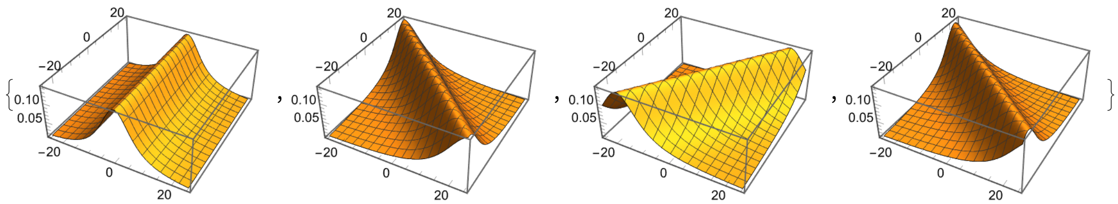


Fixed Z=7 and X, Y bound between -8π, 8π . Below snapshots of 3D plot assuming fixed Z and few variations of the u value both Re and Im part of the

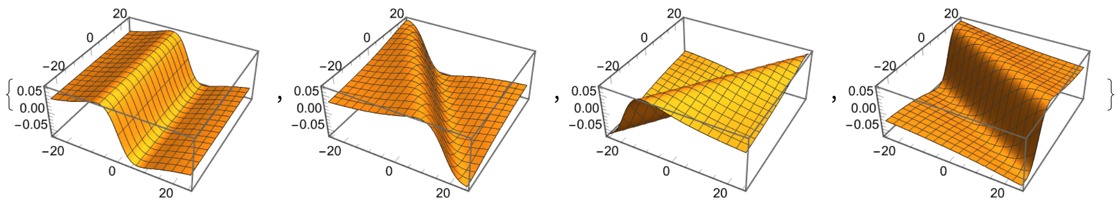
$$\frac{1}{1/(Z+i\ X\ \mathrm{Cos}[u]+i\ Y\ \mathrm{Sin}[u])}$$

u = 0 at the left most plot and progressively increasing towards the right:

Re



Im



Appendix A: *K*omplexity

Definition A.1: The Kolmogorov Complexity $C_{\mathcal{U}}(x)$ of a string x with respect to a universal computer (Turing Machine) $\mathcal{U}$ is defined as

$$C_{\mathcal{U}}(x) = \min_{p:\mathcal{U}(p)=x} \ell(p)$$

the minimum **length** program p in $\mathcal{U}$ which outputs x.

Therefore we assign the dimension L of length to the said Complexity integer (A.1.1).

Theorem A.2 (Universality of the Kolmogorov Complexity): *If $\mathcal{U}$ is a universal computer, then for any other computer $\mathcal{A}$ and all strings x,*

$$C_{\mathcal{U}}(x) \leq C_{\mathcal{A}}(x) + c_{\mathcal{A}}$$

where the constant $c_{\mathcal{A}}$ does not depend on x.

Corollary A.3 : $\lim_{\ell(x)\to\infty} \frac{C_{\mathcal{U}}(x) - C_{\mathcal{A}}(x)}{\ell(x)} = 0$ *for any two universal computers.*

Remark A.4: *Therefore we drop the universal computer subscript and simply write C(x).*

Theorem A.5: $C(x) \leqslant \ell(x) + c.$

A string x is called incompressible if $C(x) \geqslant \ell(x)$.

Definition A.6: Self-delimiting string (or program) is a string or program which has its own length encoded as a part of itself i.e. a Turing machine reading Self-delimiting string while knowing when exactly when to stop reading the tape.

Definition A.7: The Conditional or Prefix Kolmogorov Complexity of self-delimiting string x given string y is

$$K(x \mid y) = \min_{p:\mathcal{U}(p,y)=x} \ell(p)$$

The length of the shortest program that can compute both x and y and a way to tell them apart is

$$K(x, y) = \min_{p:\mathcal{U}(p)=x,y} \ell(p)$$

Remark A.8: *x, y can be thought of as concatenation of the strings with additional separation information.*

Assume Prefix K:

Theorem A.9: $K(x) \leq \ell(x) + 2\log \ell(x) + O(1), \quad K(x \mid \ell(x)) \leq \ell(x) + O(1)$.

Theorem A.10: $K(x, y) \leq K(x) + K(y)$.

Theorem A.11: $K(f(x)) \leq K(x) + K(f)$, *f* is computble function

Let's assume the Prefix Kolmogorov Complexity from now on and further assume K(x) = l(x) while assuming l(x) being astronomically large!

References

[1] Tetrahedron: https://mathworld.wolfram.com/Tetrahedron.html

[2] Vandermonde matrix: https://en.wikipedia.org/wiki/Vandermonde_matrix

[3] Tetsuo Asano et al, A linear-space algorithm for distance preserving graph embedding, Computational Geometry: Theory and Applications, Elsevier

[4] Sean Dewar et al, Edge-length preserving embeddings of graphs between normed

https://arxiv.org/pdf/2405.02189

[5] E.T Whittaker , On the Partial Differential Equations of Mathematical Physics, Mathematische Annalen Vol 57, 1903, p. 333-355

[6] J. Haywood https://en.wiktionary.org/wiki/see_the_forest_for_the_trees

[7] A. N. Kolmogorov, Three approaches to the quantitative definition of information, https://doi.org/10.1080/00207166808803030

[8] Categorification: https://en.wikipedia.org/wiki/Categorification

[9] Mechanical Bodies: https://en.wikipedia.org/wiki/Mechanics