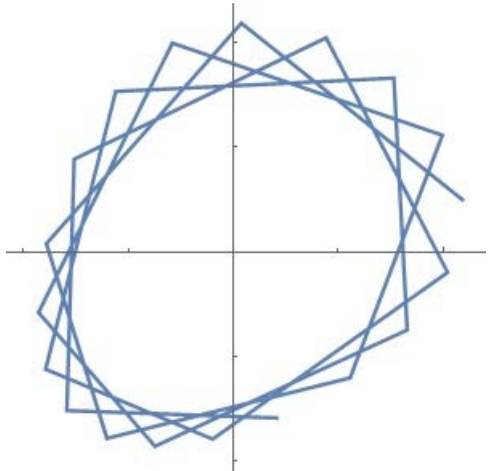
Information entropy as a measure for complexity

Notes for CAS

Feijs, Delbressine



How to measure complexity?

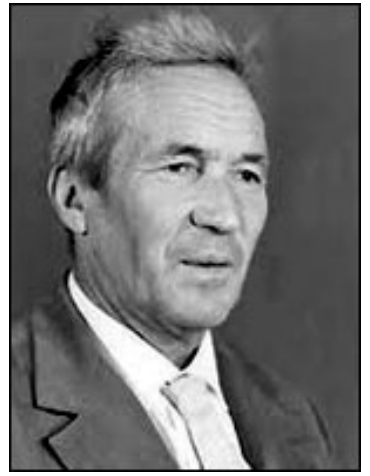
- Shannon's aproach:
  assume a source which produces random messages,
  how many bits do we need on average to code a message?

  A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, pp. 379-423 & 623-656, 1948

Claude Shannon in 1948
Source: www.i-programmer.info

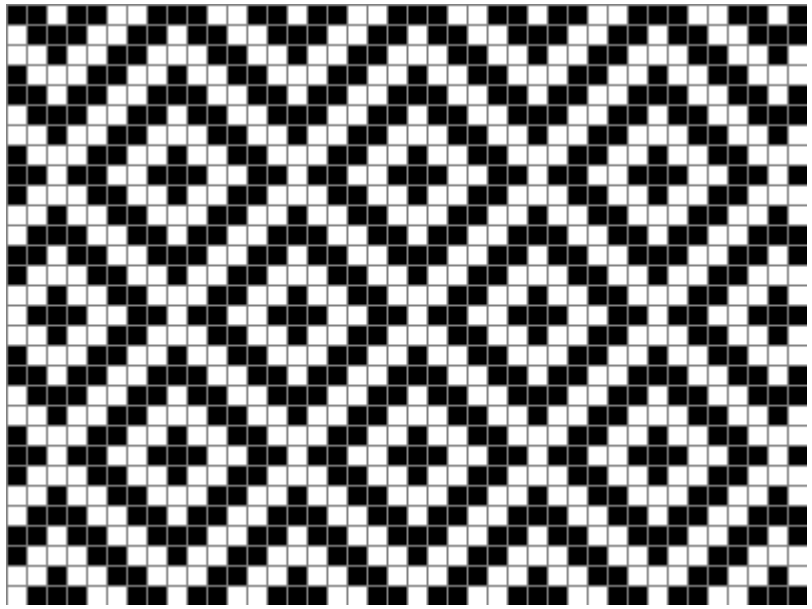
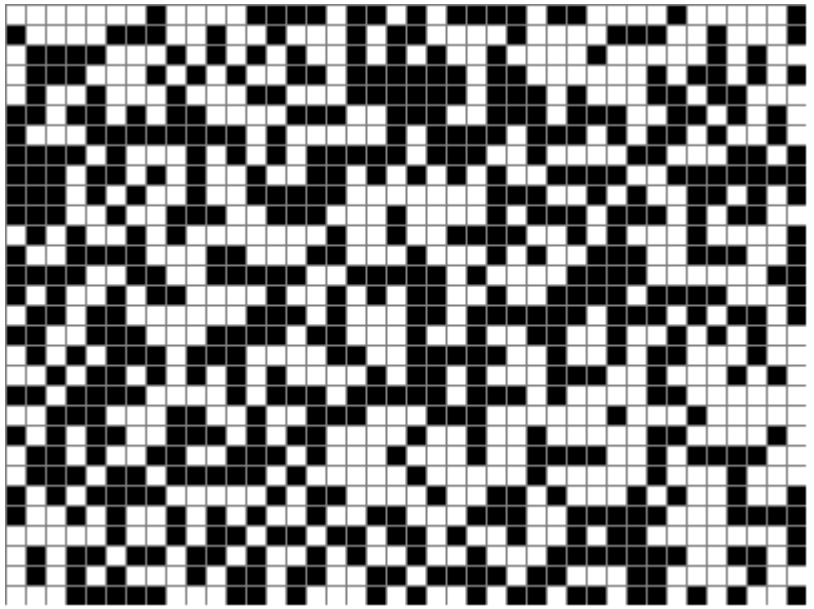
- Kolmogorov's approach:
  assume a fixed message
  (a tekst, or an image, for example),
  how long is the shortest program
  that will reproduce the message?

  Kolmogorov, A. (1968). Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, *14*(5), 662-664.



Andrei Kolmogorov
Source: http://www.mi.ras.ru

Examples:

- “0100110101101001100100111001010010011001”
- “0000000000001000000000000100000000000001”
- int tm(int i, int N){int im = i%(2*N); return im<N? im : (2*N)-im;}
- “a[i][j] = (tm(i,N)+1000+1 - tm(j-1,N+1))%4 <2? true : false;”
- 
- 

Shannon's aproach:

Assume source X
with alphabet {A,B,C,D}
and probabilities P(A)=0.5, P(B) = 0.25, P(C)= 0.125, P(D) = 0.125

Information per letter
$H(A) = -\,{}^{2}\log 0.5 = -(-1) = 1$ bit
$H(B) = -\,{}^{2}\log 0.25 = -(-2) = 2$ bit
$H(C) = -\,{}^{2}\log 0.125 = -(-3) = 3$ bit
$H(D) = -\,{}^{2}\log 0.125 = -(-3) = 3$ bit

Complexity of this source
$H(X) = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1\frac{3}{4} = 1.75$ bit per letter

Shannon’s aproach:

Claim: we can code these letters in 1.75 bit (on average)

“Huffman coding”

A → 0

B → 10

C → 110

D → 111

BABADACA ←10010011101100

DDDDDDDD ← 11111111111111111111111

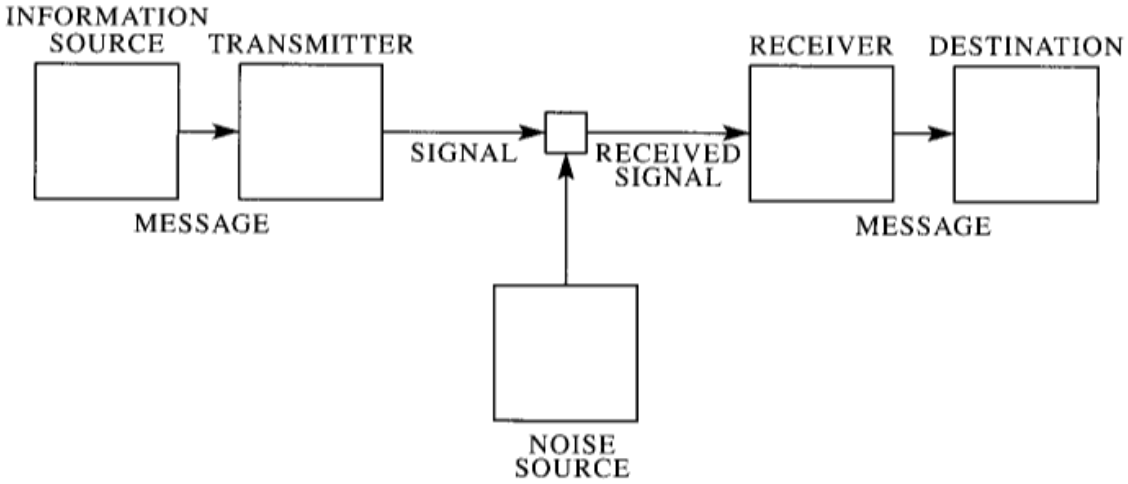


Fig. 1—Schematic diagram of a general communication system.

Source: A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, pp. 379-423 & 623-656, 1948

$$H(X) = \sum_i -p_i \, {}^2\log p_i$$

Special case:
two-letter alphabet
$P(A) = p_1 = p$
$P(B) = p_2 = (1 - p)$

$$H(X) = -\, p \, {}^2\log p - (1 - p) \, {}^2\log (1 - p)$$

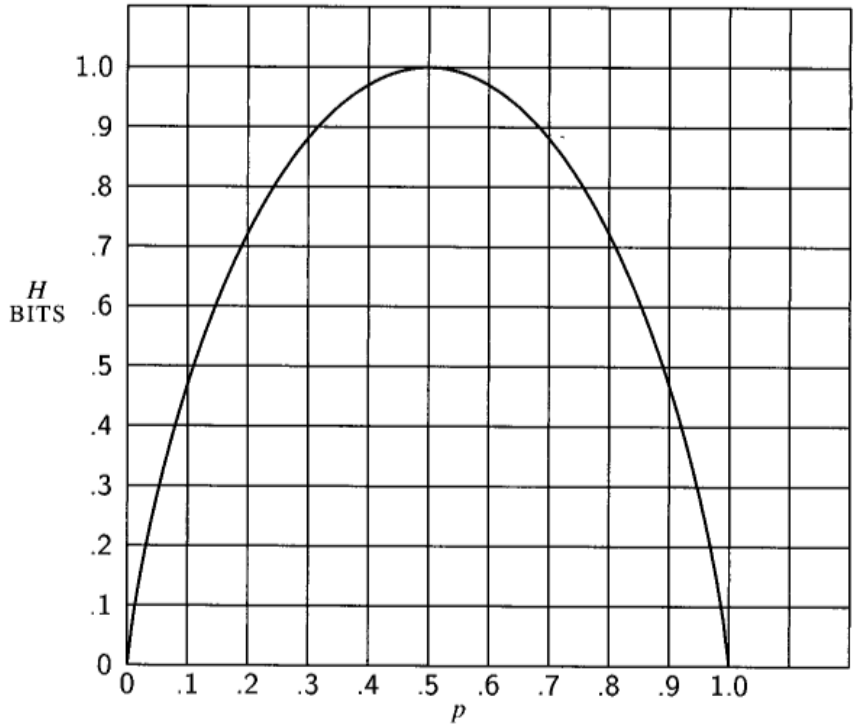


Fig. 7—Entropy in the case of two possibilities with probabilities p and $(1 - p)$.

Source: A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, pp. 379-423 & 623-656, 1948

III. The Series of Approximations to English

To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

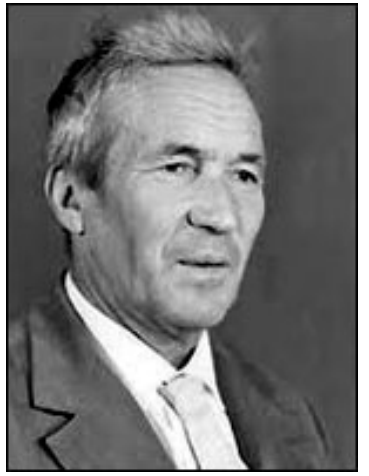
4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

Source: A Mathematical Theory of Communication",
Bell System Technical Journal, vol. 27, pp. 379-423
& 623-656, 1948

How to measure complexity, 2nd approach?

- Kolmogorov's approach:
  assume a fixed message
  (a tekst, or an image, for example),
  how long is the shortest program
  that will reproduce the message?



Andrei Kolmogorov
Source: http://www.mi.ras.ru

Kolmogorov, A. (1968). Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, *14*(5), 662-664.

Logical Basis for Information Theory and Probability Theory

ANDREI N. KOLMOGOROV

Abstract—**A new logical basis for information theory as well as probability theory is proposed, based on computing complexity.**

Section I

WE SHALL be concerned with the main basic concepts of information theory, beginning with the traditional concept of the conditional entropy of x when the value of y is known, $H(x \mid y)$, which can be interpreted as the quantity of information required for computing ("programming") the value x when the value y is already known. By using ϕ to denote a particular given known value, we get the unconditional entropy

$$H(x \mid \phi) = H(x).$$

Information given by y concerning the value of x can, as is well known, be expressed:

$$I(x \mid y) = H(x) - H(x \mid y).$$

It is evident that

$$I(x \mid x) = H(x).$$

The ordinary definition of entropy uses probability concepts, and thus does not pertain to individual values, but to random values, i.e., to probability distributions within a given group of values. In order to stress this difference, we will denote random values by Greek letters.

cepts. I believe that the need for attaching definite meaning to the expressions $H(x \mid y)$ and $I(x \mid y)$, in the case of individual values x and y that are not viewed as a result of random tests with a definite law of distribution, was realized long ago by many who dealt with information theory.

As far as I know, the first paper published on the idea of revising information theory so as to satisfy the above conditions was the article by Solomonov [1]. I came to similar conclusions, before becoming aware of Solomonov's work, in 1963–1964, and published my first article on the subject [2] in early 1965. A young Swedish mathematician, Martin-Löf, who worked in Moscow during 1964–1965, began developing this concept. His lectures [3] which he gave in Erlangen in 1966 represent a better introduction to the subject of my paper.

The meaning of the new definition is very simple. Entropy $H(x \mid y)$ is the minimal length of the recorded sequence of zeros and ones of a "program" P that permits construction of the value of x, the value of y being known,

$$H(x \mid y) = \min_{A(P,y)=x} l(P). \tag{2}$$

This concept is supported by the general theory of "computable" (partially recursive) functions, i.e., by the theory of algorithms in general. We will return again to the interpretation of the notation $A(P, y) = x$.

$H(x) = \min_{P \to x} \text{length}(P)$

length(P_1) = 100

```
P1 = print
(“aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
”);
```

$P_1 \to$

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

length(P_2) = 32

```
P2 = for(int i=0;i<90;i++)print("a");
```

$P_2 \to$

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

An analogous situation exists in the principles of information theory. Essentially, it is applicable to large quantities of information, when the initial information (contained in the method on which the theory is based) is infinitesimal. Our basic formula (1) implies a "universal programming method" A, which exists because there are programming methods A possessing the quality

$$H_A(x) \leq H_{A'}(x) + C_{A'}$$

They allow the programming of anything with a program length that exceeds the length of any other programming method by not greater than a constant and is dependent only on this second programming method and not on values of x

From Kolmogorov, 1968, simplified, no y, LF

Kolmogorov, A. (1968). Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, *14*(5), 662-664.

Example (Mandelbrot):

- simple program
- fascinating complexity

mandelbrot_simple | Processing 2.2.1

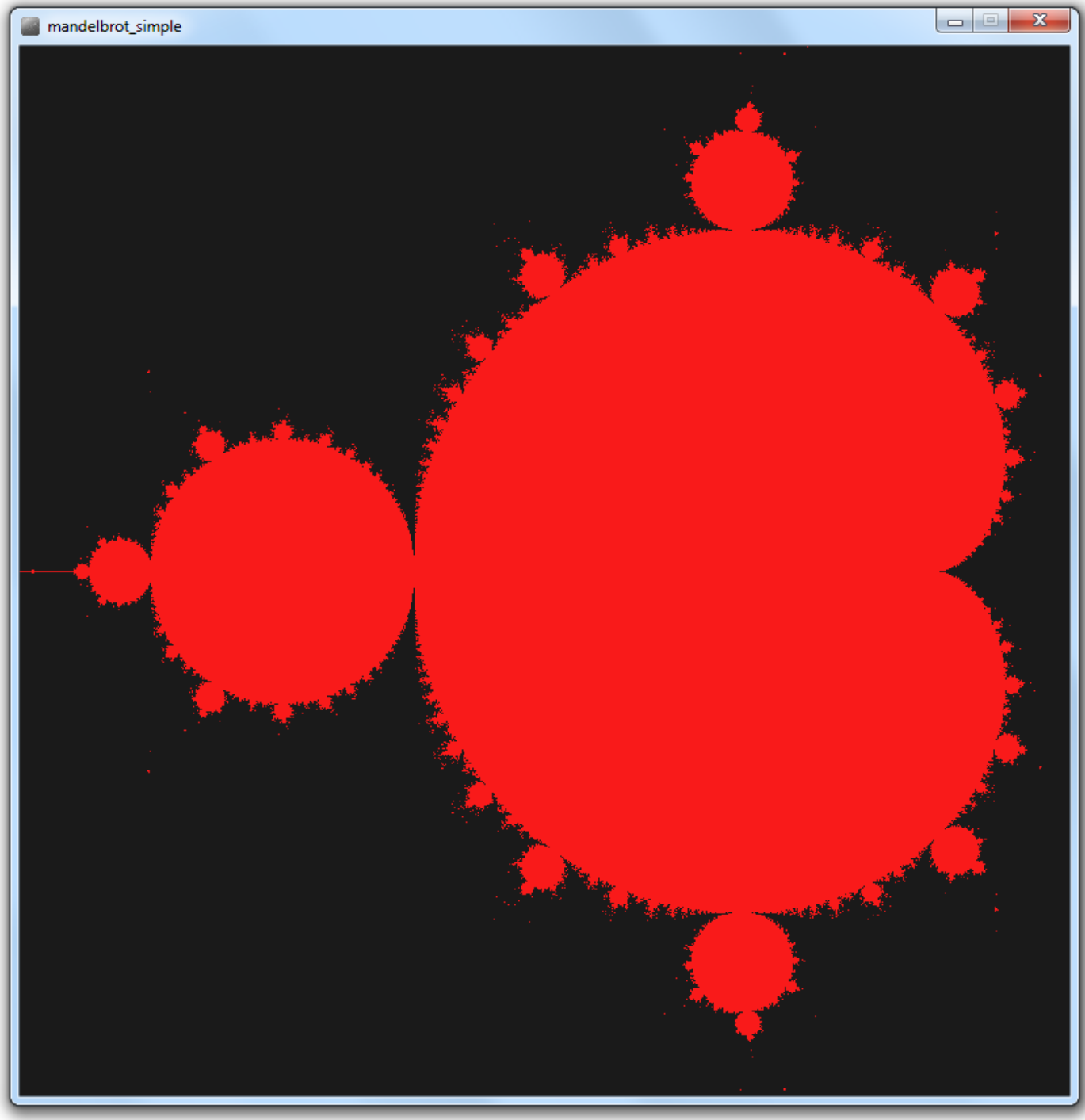
File Edit Sketch Tools Help

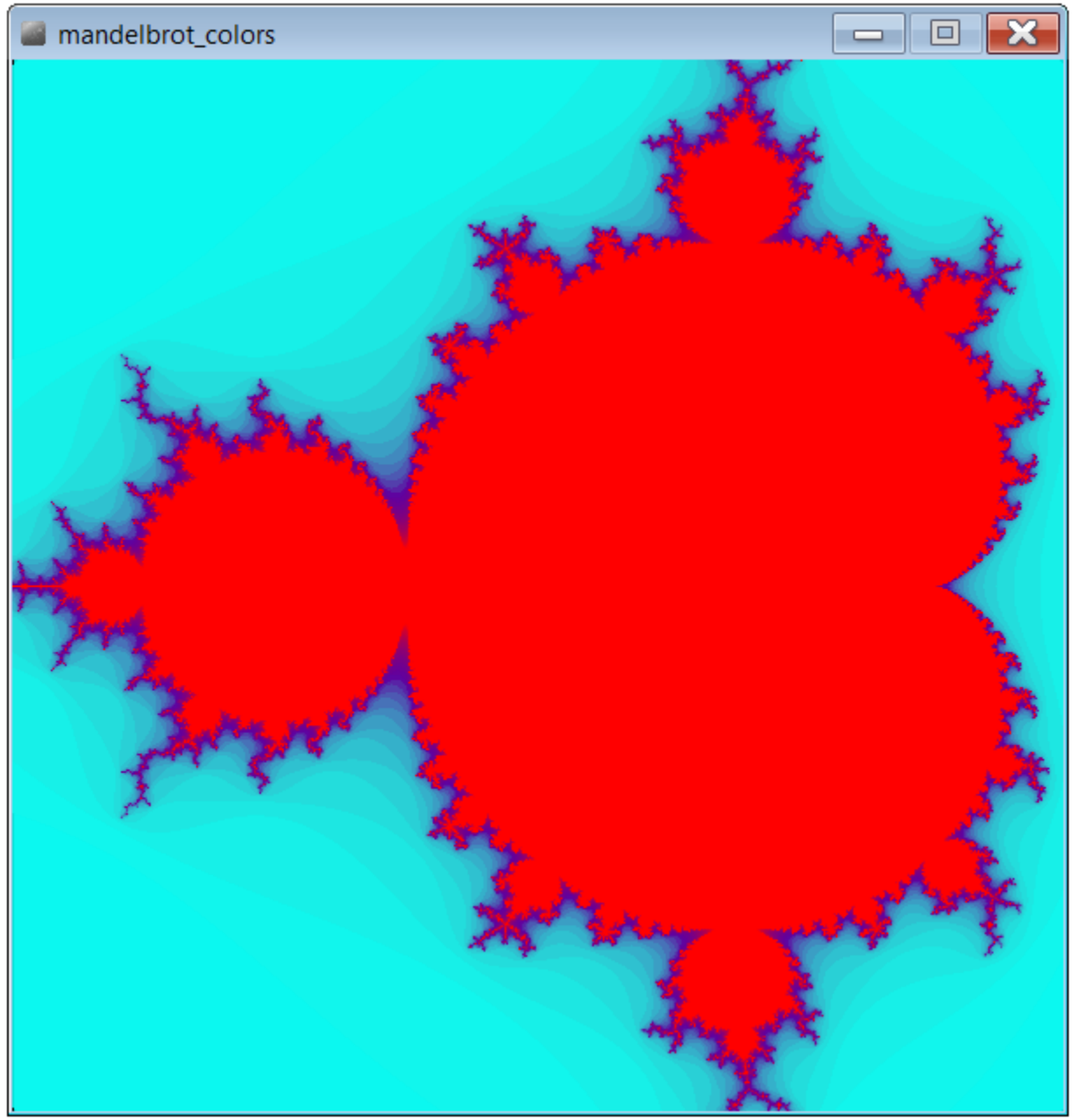
Java

mandelbrot_simple complex

```
Complex  F(Complex c, int n){
         if (n == 0)
             return new Complex(0,0);
         else { Complex z = F(c,n - 1);
                return z.mul(z).add(c);
         }
}

void setup(){
     size(400,400);
     float scale = width / 2;
     for (float x = 0; x < width; x++)
          for (float y = 0; y < height; y++){
               float zx = 2*x / width  - 1.5;
               float zy = 2*y / height - 1.0;
               Complex zn = F(new Complex(zx,zy),100);
               stroke(zn.modulus() < 2? 255 : 0,0,0);
               point(x,y);
     }
}
```




mandelbrot_colors