Algorithms and the Shift in Scientific Thought

CCCS 2016

W. Brian Arthur

External Professor, Santa Fe Institute
Senior Advisor to the President, NTU
Science is shifting from equation-based expression to algorithm-based expression
An earlier shift in 1600s. From geometric expression to equation-based expression.
Si parallelogrammi latera quatuor infinite producta tangant sectionem quamcunque Conicam, & abscondantur ad tangentem quamvisquin-tam; sumantur autem abreisse terminae ad angulos oppositos parallelogrammi: di-sco quod absccissa minimus lateris sit ad latum illud ut pars lateris contempti inter punctum contactus et latus tertium et abscissas.
All nature is but art, unknown to thee;
All chance, direction, which thou canst not see;
All discord, harmony not understood;
All partial evil, universal good.
And, spite of pride, in erring reason's spite,
One truth is clear, 'Whatever is, is right.'
The world in question is

1. Orderly
2. Equation-based
3. Predictable
4. Usually in stasis or equilibrium
“Chemistry is a science but not Science. The criterion of true Science lies in its relation to mathematics.”
Biology is challenging these 4 pillars

Evolution, speciation, embryology, protein expression, epigenetics, genetic regulatory networks are:

- Ordered, but open systems
- Not generally expressed by equations
- Not generally predictable
- Generally not in stasis
The equation-based setup

Equations define an updating rule:

\[
\frac{dX}{d\tau} = -\sigma X + \sigma Y
\]
\[
\frac{dY}{d\tau} = -XZ + rX - Y
\]
\[
\frac{dZ}{d\tau} = XY - bZ
\]
The updating rule depends on where system currently is

Like a ball on a smooth curved surface, or a toboggan moving down a well-defined path
Lorentz equations’ “attractor”
ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

[Received 28 May, 1936.—Read 12 November, 1936.]

The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable numbers, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbersome technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers π, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In §§8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel. These results...
Turing's idea

Standard setup: Updating rule (in equation form)

Turing's setup: Updating Rule + Inner State of the System

E.g. Addition.
Inner state = Carry one, or not carry one
Inner state of the system

The “state of mind” of the person calculating

This can be complicated (e.g. “mind” itself)
Lends itself to conditional logic:

If A, B, F, not G are currently true
then execute R, and S, and T

So we can model the changing “logic of the situation”
We can also model processes

Can model events-trIGGERING-events. This is a world of possibly parallel processes, highly context dependent

Note: This is the way life works.
So: We have a equation-based setup that is position dependent

And a (Turing) algorithmic setup that is position dependent and context dependent
Caveats

An algorithmic system can in principle be expressed mathematically, but this is normally cumbersome.

Equations can pick up some “context” too.

Turing wasn’t first to think of algorithms.

Note: Standard equation setup is a special case of algorithmic one.
Some consequences of this
We recognize equations as a mathematical entity.
The *algorithm* becomes a basic mathematical entity

```
Outcome = Computer [Algorithm | Data]
```

Cf. Algorithmic Information Theory
The computer also becomes a mathematical entity

“The computer is a powerful new mathematical concept.

… It is a revolutionary new kind of mathematics with profound philosophical consequences. It reveals a new world.”

- Greg Chaitin 2012
The algorithm becomes the basic mathematical unit for expressing context-dependent systems.

Outcome = Computer [Context-Dependent System]

This widens greatly what science can capture and express.
How does this world relate to complexity?

Complexity studies systems whose elements react to the pattern they create, i.e. to the context they create.

Algorithmic updating depends on inner state of system, i.e. the context. This allows systems to react to the context they create.

Complexity is the natural study of systems that react to their context.
Complexity is closely related to computation
Side note: when probability enters

Standard math-based systems tend to add outcomes Therefore they have normal deviations

Event-causing-event systems are like dominoes that can pass on happenings with prob. $p$

Length of such cascades is distributed geometrically: these multiply probabilities. Leads to power laws
What happens the 4 Pillars in Turing’s World?
Algorithmic expression captures systems that are interrelated, parallel, highly context dependent.

Ordered, but complicated and open.

© 2016 W. Brian Arthur
Predictable?
E.g. Cellular Automata
In general, algorithmic systems do not lead to stasis or equilibrium (Turing, Chaitin, Wolfram).

Stasis is very much a special case
Equation-based expression
Allows noun-based science

Algorithmic expression
Allows verb-based or procedural science
No longer prim dreams of pure order ...
Rather, a world of messy vitality