Aden, J Inform Tech Softw Eng 2016, 6:4
DOI: 10.4173/2165-7866.1000187

Journal of
Information Technology & Software Engineering

Research Article Open Access

Thermodynamic and Quantum Mechanical Limitations of Electronic Computation

Fayez Fok Al Adeh*

The Syrian Cosmological Society, Damascus, Syria

Abstract

The role of entropy and uncertainty in electronic computations is used to derive a fundamental thermodynamic equation. The thermodynamic limitations of electronic computations are expressed in terms of macroscopic disorder, while quantum mechanical limitations are formulated through uncertainty relationships. Electronic computations are described quantum-mechanically, and hence provide a reasonable argument in favour of the wave interpretation of quantum mechanics.

Keywords: Entropy; Electronic computations; Uncertainty; Quantum mechanics

Introduction

The second law of thermodynamics is a universal law. The galactic spiral structures, for example [1], arise in response to the interaction of that law and the law of gravitation. The second law states that some forms of transformation of one kind of energy to another do not occur in natural processes. The allowed transformations in a closed system are always characterized by a non-decreasing entropy (introduction of macroscopic disorder). In open systems where the entropy is kept constant [1], the allowed transformations are always characterized by a decrease in the amount of free energy available to do useful work.

Another profound principle of nature is worth mentioning here. The Heisenberg's uncertainty principle asserts that nature forbids knowledge beyond a certain limit [2]. This is not the result of the restricted abilities of the available tools or the contemporary theorizing methods.

I will now attempt to apply both the second law of thermodynamics and Heisenberg's uncertainty principle to electronic computations. I will conclude by giving a quantum mechanical description of these computations.

Floating - Point Numbers

$$0.1<|m|<1$$

By a normalized floating-decimal representation of a number *a,* we imply representation of the form:

a = mx $a = m \text{ x } 10^q$

where q is an integer. Such a representation is possible for all numbers, and unique if $a \neq 0$ [3]. The variable m is the fractional part or mantissa and q is the exponent.

In a computer, the number of digits for q and m is limited. The number of digits characterizes a given computer. This means that only a finite set of numbers can be represented in the machine. The numbers in this set (for a given q and m) are called floating-point numbers. The limited number of digits in the exponent implies that a is limited to an interval which is called the machine's floating-point variable range.

In a computer, a is represented by the floating number $\overline{a} = \overline{m} \times 10^q$

Where $\overline{m}$ is the mantissa m, rounded off to t decimals. The precision of the machine is said then to be t decimal digits. Now suppose that the floating numbers in a machine have base B (ten for the decimal number system) and a mantissa with t digits. (The binary digit which gives the sign of the number is not counted) Then, every real number in the floating-point range in the machine can be represented by a relative error which does not exceed the machine unit (round-off unit) u which [3] is defined by:

$$u = \begin{cases} 0.5 \times B^{1-t} & \text{if rounding is used} \\ B^{1-t} & \text{if truncation is used} \end{cases}$$

The floating-point set of numbers is not a field. It is also not a ring. Besides, it is neither a group, nor a semi-group. The elementary operations in the real number system, are not well defined in this set. If we denote the result of addition in this set by fl(x + y), then associativity does not, in general, hold for floating addition. Consider floating addition using seven decimals in the mantissa,

$a = 0.1234567 \times 10^0$, $b = 0.4711325 \times 10^4$, $c = -b$

$$\text{fl}(b + c) = 0$$

$$\text{fl}(a + \text{fl}(b + c) = 0.1234567 \times 10^0$$

$$\text{fl}(a + b) = 0.4711448 \times 10^4$$

$$\text{fl}(\text{fl}(a + b) + c) = 0.0000123 \times 10^4 = 0.123 \times 10^0$$

Hence: $\text{fl}(\text{fl}(a + b) + c) \neq \text{fl}(a + \text{fl}(b + c))$

In general, the usual laws and operations of the real number system are not applicable to the floating-point set of numbers. Strictly speaking, we have to define special new laws and operations for each computer and each problem. Moving the problem to a new computer forces a redefinition of these laws and operations. Such a definition necessitates a premature prediction of the results obtained after solving the problem. This situation constrains us to a vicious circle; defining the aforementioned laws and operations is conditional, we must know beforehand the results based on using them. These results represent what we expect upon employing a given computer in solving a definite problem. Here we face a sort of a contradiction; it is the problem of formulating a definition and then reformulating a new one and then

***Corresponding author:** Fayez Fok Al Adeh, President of The Syrian Cosmological Society, Damascus, Syria, Tel: 00963112776729; E-mail: hayfa@scs-net.org

Received August 02, 2016; **Accepted** August 20, 2016; **Published** August 30, 2016

Citation: Adeh FFA (2016) Thermodynamic and Quantum Mechanical Limitations of Electronic Computation. J Inform Tech Softw Eng 6: 187. doi:10.4173/2165-7866.1000187

Citation: Adeh FFA (2016) Thermodynamic and Quantum Mechanical Limitations of Electronic Computation. J Inform Tech Softw Eng 6: 187. doi:10.4173/2165-7866.1000187

another one and so on ad infinitum. The task looks impossible, and the impossibility appears to be fundamental. It is not the result of a lack of abilities of a given machine, and it has nothing to do with the insufficiencies inherent in the chosen algorithm.

In real life, we seek a separate fundamental [4] algorithm and apply a suitable algorithm to each problem. The applied algorithm depends on the machine and the operating system used. An amalgamation of these two types of algorithms leads to a resultant program. I call this the resultant algorithm. The whole process is based on supplying concrete proofs assuring that the algorithms (fundamental, applied, and resultant) that are looked for will solve the problem during an acceptable period of time, using a given machine and a definite operating system.

The various sources of errors are important in formulating my results [5]. Among these are:

- errors due to rounding;
- errors due to the truncation of series; unjustified simplification of formulae;
- a complex of correct logical propositions resulting in a machine default; mathematical instability;
- catastrophic cancellation; and
- Exaggerated sensitivity of an algorithm.

It is a good programming practice, however, to seek a limit [3] for the relative error in the output data. Such a limit is given by:

$$= C_p(r + C_A u) \qquad (1)$$

where: r is a bound for the relative error in the input data; u is the machine unit; C_p is the condition number for the problem p. With given input data, it is the largest relative change, measured with u as a unit, that the exact output data of the problem can have, if there is a relative disturbance in the input data of size u; C_A is the condition number for the algorithm A. By means of backward error analysis, the output data which the algorithm produces (under the influence of round-off error) is the exact output data of a problem of the same type in which the input data has been changed relatively by a few u. That change measured with u as a unit, is called the condition number of the algorithm. Hence, the condition number has a small value for a good algorithm and a large value for a poor algorithm.

Entropy

The equivalence of information and entropy [6] is no more surprising than the equivalence of mass and energy implied by Einstein's formula: $E = m\,c^2$, where $c = 3 \times 10^{10}$ cm s^{-1} is the velocity of light; 1 erg is equivalent to a mass of 10^{-21} g. Likewise, 1 bit of information is equivalent to $Kln\ 2 = 10^{-16}$ erg/degree kelvin of entropy (where $K = 1.38 \times 10^{-16}$ erg/ degree Kelvin, is Boltzmann's constant).

In general, let W represent the number of different micro-states which correspond to the same macrostate. The entropy S of a macrostate is equal to [6]:

Boltzmann's constant K times the natural logarithm of W:

$$S = K\ \text{In W} \qquad (2)$$

Since the natural logarithm is a monotonic function, S attains its maximum value when W does.

The equivalence of entropy and information means that new information is obtained at the price of increased entropy (in a different part of the system). In other words, some energy must be dissipated. The minimum energy consumption per one bit of information obtained is $K\ T$. In 2, where T is the absolute temperature. Suppose that we are running a moderate program, the output of which consists of 10000 decimal numbers. Assume also that to each one of these numbers there corresponds a set of configurations of the internal memory equivalent to 10 decimal numbers.

We would then have a total of 110 000 decimal numbers. An increase in accuracy of one decimal digit per number will entail a corresponding increase in global entropy given by:

$$10^{-16} \times 16 \times 11 \times 10^4 \times 300 \times 4 \simeq 10^{-8}\ \text{erg}$$

assuming an absolute temperature of 300 K and four bits of information for the additional decimal digit. This is still a very small amount of entropy. In the next section, we will formulate a conclusion about the temperature of the memory and its role in the energy consumption for a computation.

I consider as a microstate, a set of real numbers representing a theoretical solution to a given problem. It is assumed that such a solution exists but is not necessarily attained. Any computerized solution of the problem is only an approximation of this set of real numbers. The microstate is the sequence of configurations of the internal memory of the computer leading to the approximate solution which can be included within the microstate. A program is executed by triggering a series of pulses [7]. Each pulse gives rise to a new configuration of the memory. The set of all these configurations constitute a microstate of the sought for theoretical solution. Trying to solve the same problem using another computer and a different algorithm provides a new microstate. Each pulse, in a sense, assigns the memory. Each assignment, in turn, can be translated into some set of floating-point numbers. In some programming languages (such as APL/360) [8], one can write a program using assignments only.

An increase in accuracy forces an extension of the mantissa. I can now introduce the notion of a dynamic computer. This is a computer whose set of floating-point numbers can be extended in response to any demand for more accuracy, and it must be ever developing. This is equivalent to the continuous employment of successive generations of computers.

In what follows, 'computer' will define a dynamic computer which is necessary to attain an increase in accuracy. A dynamic computer allows new microstates to develop and produce corresponding increases in global entropy. This is easily verified using Equation 2 which tells us that the continuous search for more accuracy entails a gradual increase in global entropy. Moreover, an indefinite search for accuracy will lead to infinite global entropy. If we refer to the formula [9]:

$$I = I_0\ e^{-S/K}$$

Here I and I_0 are the information content at ordinary temperatures, and at absolute zero respectively, of a closed local system. More accuracy means an increase of local informa-tion content and a corresponding decrease in local entropy. This is balanced by a parallel increase in global entropy.

Thermodynamic Equation

The second law of thermodynamics can be written in the form:

$$Tds \geq du + pdv + ydx \qquad (3)$$

or by using equation (2)

Citation: Adeh FFA (2016) Thermodynamic and Quantum Mechanical Limitations of Electronic Computation. J Inform Tech Softw Eng 6: 187. doi:10.4173/2165-7866.1000187

$$TKdw / w \geq du + pdv + ydx \tag{4}$$

where *T,* is the absolute temperature; *S,* the entropy of the system; *U,* the internal energy of the system; *P,* the pressure within the system; *V,* the volume of the system; y, a generalized external force; and *x,* a generalized coordinate

The inequality corresponds to a nonequilibrium state of the system; when the system is still on the way to equilibrium, and the equality corresponds to a system already in equilibrium.

$$Tds = du + pdv + ydx \tag{5}$$

or again by using equation (2)

$$TKdw / w = du + pdv + ydx \tag{6}$$

The different components of a computer communicate with each other by signals. The reception and interpretation of a signal constitutes a physical measurement, which is assigned to a given location of the memory. The degree of assignment is characterized by the total sum per unit volume of the number of different assigned locations of the memory:

$$j = \frac{1}{V}\sum_{i=1}^{n} ji \tag{7}$$

where j_i refers to a distinct location of the memory, and *V* (the volume) is the total number of accessible locations of the memory.

If *M* is the mass of the memory, then:

$$j = A \ \ M \tag{8}$$

A is the mass specific degree of assignment, or the mass specific number of distinguishable messages. If the assignment is the outcome of a physical measurement, then the faster the measurement the larger is the energy that is required to make the signal readable with sufficiently small error probability. If the total signalling energy is limited, then there is a trade-off between the number of distinguishable signals that can be sent and the time required to identify them.

Let *H* denote the total signalling energy. It is expressed in units of energy. But *H* affects an assignment which is interpreted as information. Hence we can refer also to *H* in bits of information.

Now *H* changes *j* to *j* + dj. The work done in such a change is:

$$dL = -Hdj \tag{9}$$

The minus sign shows that when the number of assigned locations increases, work is expended on the memory.

Replacing y dx in equation (5) by –H dj, and get:

$$Tds = du + pdv - Hdj \tag{10}$$

The enthalpy of the computer in this case is:

$$I = u + pV - Hj \tag{11}$$

Maxwell's thermodynamic equations give us:

$$\left(\frac{\partial j}{\partial T}\right) S,V = \left(\frac{\partial S}{\partial H}\right) j,v \tag{12}$$

$$\left(\frac{\partial j}{\partial S}\right) H,P = \left(\frac{\partial S}{\partial H}\right) S,P \tag{13}$$

$$\left(\frac{\partial j}{\partial S}\right) T,V = \left(\frac{\partial T}{\partial H}\right) J,V \tag{14}$$

$$\left(\frac{\partial j}{\partial T}\right) H,P = \left(\frac{\partial S}{\partial H}\right) T,P \tag{15}$$

From Equation 2 by taking differentials:

$$dS = K\frac{dW}{W} \tag{16}$$

substituting in Equation 10 from Equation 16 we get

$$TKdW = \left(du + pdV - Hdj\right) \tag{17}$$

By combining Equations 17 and 11 with Equations 12-15 it is easy to find the relation between the entropic properties of the memory (i.e. its macroscopic disorder properties), its internal energy and enthalpy, and *j* and *H*. Equation 17 is the fundamental thermo-dynamic equation of electronic computations.

In the case of the dynamic computer, we should rewrite Equation 17 in the form of an inequality:

$$TKdW \geq \left(du + pdV - Hdj\right) W \tag{18}$$

At absolute zero where *T* = 0, d W = 0, Equation 17 gives us:

$$Hdj = du + pdv \tag{19}$$

substituting in Equation 11, we deduce that the enthalpy of the memory remains constant at absolute zero. Equation 10 yields:

$$\left(\frac{\partial u}{\partial j}\right) T,V = \left(\frac{\partial S}{\partial j}\right) T,V + H \tag{20}$$

Using Equation 14 we find that:

$$\left(\frac{\partial u}{\partial j}\right) T,V = H - T\left(\frac{\partial H}{\partial T}\right) j,V \tag{21}$$

The physical meaning of the derivative shows how much *H* must be increased with increasing temperature of the memory so that the assignment *j* remains the same, regardless of the increase in temperature (which will affect the assignment).

Note that the memory's macroscopic disorder is not meant to be realistic in that it would be practically possible for such disorder to be attained. The entropy of the memory is merely a yardstick beyond which macroscopic disorder surely takes place. It is comparable to saying that astronauts cannot travel at speeds exceeding the velocity of light, though in practice their speeds are much more limited.

If a result is to be formulated, then it will be a sort of summary asserting that electronic computations are not out of the grasp of the second law of thermodynamics. A computer is only a limited ordering machine. Increasing its ordering tasks without bounds is impossible, for such an increase will end sooner or later in macroscopic disorder and chaos.

Heisenberg's Uncertainty Principle

According to quantum mechanics, the existence of a physical system is related to measurement. A measurement by definition, is any physical interaction, and physical interactions obey Heisenberg's uncertainty principle. The so-called energy [10] version of the principle states that if a limited time *T* is available for making the measurement, then the energy of the system cannot be determined better than to within an amount of order *h* / *T*.

Citation: Adeh FFA (2016) Thermodynamic and Quantum Mechanical Limitations of Electronic Computation. J Inform Tech Softw Eng 6: 187. doi:10.4173/2165-7866.1000187

This is equivalent to:

$$ET \geq \hbar \tag{22}$$

where: *E* is the energy of the system; *T*, the period or measurement; and $\hbar$ planck's constant divided by 2 .

In particular, if a system is unstable having a finite lifetime *t*, its energy cannot be measured to within an accuracy better than about $\hbar$ /*t*.

Let the measurement in our case be a signalling process producing a bit of information. Denote by *F* the rate of signal flow in a computer (in bits per second). Then *F* and *T* would be reciprocals. We can deduce that no closed computer system, however constructed, can have *F* exceeding E/ $\hbar$.

Let m be the total mass of the system: which includes the mass equivalent of the energy of signals employed in the computer, as well as the mass of the materials of which the computer and its power supply are made. In computers the structural mass outweighs the mass equivalent of the signal energy, However, the mass equivalent of the energy of signals can be computed by applying Einstein's formula: Energy = (mass) × (square of velocity of light).

In other words, the total mass equivalent of the energy that is invested in signals cannot exceed *m,* the total mass of the system.

We would have:

$$F \geq \frac{mc^2}{\hbar} \tag{23}$$

I now consider the electro weak theory which unifies electromagnetism and weak forces. It indicates that, during infinitesimal periods of time, huge amounts of energy can be created from nothing and even transformed into particles [11]. But before such periods end, everything created from nothing must vanish, and the situation must return to normal.

Imagine a hypothetical computer which accomplishes all of its tasks during an infinitesimal period. The result will not be transferred to the usual output devices, but they will be transmitted by telepathy to the mind of the operator. Such a hypothetical computer is called a quantum computer, and should be developed during the coming centuries. Remember, any concrete scientific model is based on stirring of the imagination!

Consider the momentum version of Heisenberg's uncertainty principle. It says that one cannot know both where something is and how fast it is moving. The uncertainty of the momentum and the uncertainty of the position are complementary and the product of the two is constant [12]. We can write the law as:

$$\Delta x \, \Delta p \geq \hbar \tag{24}$$

where: x is the uncertainty in the position; p, the uncertainty in momentum.

Let *(x, t)* be the wave function of an electron contributing to an assignment at a location of the memory given by the coordinate x. (We assume here a linear memory). This assignment takes place at the moment *t*.

If we assume a gaussian wave packet, then it is easy to verify that [13]:

$$(\Delta x)^2 = \frac{1}{4}a^2 + h^2t^2 / m^2a^2 \tag{25}$$

where *a* is a constant.

Equality 25 says that assignments occurring at later times are prone to higher uncertainties. Since x refers to a general location of the memory, we conclude that no closed computer system, however constructed, can run programs of infinite lengths (practically very long programs). Compare with the very long program that governs the lifetime of a human individual, perhaps it violates the principles of quantum mechanics?

Consider now the case in which the probability density at each location x of the memory is independent of time. It is called a stationary state, and is characterized by a definite energy.

If we substitute zero for *t* in Equation 25, we obtain:

$$\Delta x = \frac{a}{2} \tag{26}$$

Using Inequality 24 we arrive at:

$$\Delta p \geq \frac{2\hbar}{a} \tag{27}$$

Since uncertainty in momentum induces a corresponding uncertainty in assignment, a similar conclusion to the previous one can be drawn from Inequality 27.

As in the case of entropy in electronic computations, uncertainty limitations may not emerge in practical situations, yet they form another yardstick beyond which improvement cannot be made: improvements mean extending tlie memory, as well as increasing the lengths of programs.

Quantum Mechanical Description

This section is not strictly devoted to a quantum mechanical description of electronic computations, rather it looks at what such a description can provide in terms of the many-worlds interpretation of quantum mechanics [14].

The interpretations of quantum mechanics are, in brief:

1. The popular interpretation: here the wave function is regarded as objectively characterizing the single system, obeying a deterministic wave equation when the system is isolated but changing probabilistically and discontinuously under observation.
2. The Copenhagen interpretation.
3. The 'hidden variables' interpretation.
4. The stochastic process interpretation: this point of view holds that the fundamental processes of nature are stochastic.
5. The wave interpretation: this is the interpretation we are interested in. A correlation is established between the observer and the observed. In this situation, any borderline between the two disappears.

In our case the observer is the mind, and the observed is the computer.

A computer that is operated for a finite period of time, and a future computer will be equivalent in the sense that the fundamental characteristics of the floating-point set of numbers will be the same in both. This is because any computer has a limited space for the mantissa. This quantity is the precision of the machine *t*.

Suppose now that one of the following two scenarios occurs. In the first case, we partition a given problem and all the operations included therein, so that the available space for the mantissa is used iteratively so that all the input, intermediate, and output digits are accommodated without truncation. From a theoretical point of view, we will need an infinite period of time to process the whole problem. In the second case, we wait for an infinite period of time and use the computer that will be available afterwards to solve the same problem.

In both cases, the floating-point set of numbers will converge and coincide in the limit with the real number system. Besides, all of the aforementioned sources of errors will be eliminated. Specifically, we do not need simplification of formulae without justification. Also we can avoid use of algorithms with exaggerated sensitivity. Lastly, we will be in a position to choose the simplest possible logical propositions.

Under our assumptions, the bound for the relative error in the output data, given by Equation 1, should become independent of the machine unit.

Differentiating with respect to *u*, we get:

$$\Delta' = C_p'\left(r + C_A u\right) + C_p\left(r + C_A u\right)' \tag{28}$$

$$\frac{\Delta'}{\Delta} = \frac{C_p'}{C_p} + \frac{\left(r + C_A u\right)'}{\left(r + C_A u\right)} \tag{29}$$

putting $\Delta' = 0$ in Equation 29 gives:

$$\frac{C_p'}{C_p} + \frac{\left(r + C_A u\right)'}{\left(r + C_A u\right)} = 0 \tag{30}$$

Integrating this differential equation, we obtain:

$$C_p\left(r + C_A u\right) = \Delta = \text{constant} \tag{31}$$

Since *r* is arbitrary, Equation 31 cannot hold unless $u = 0$, and hence $C_p = C_A = 0$.

Since u is proportional to B^{1-t}, in both cases t will equal infinity. Hence, either of our proposed cases is equivalent to operating a computer built so that the space designed to hold the mantissa is of infinite size, i.e. the precision of the machine equals infinity. Such a computer will coincide exactly with the real number system.

Since the real number system is what the mind tries to test on observing the computer, we conclude that the borderline vanishes completely between the mind and the computer in the aforementioned situation. One of the main characteristics of the wave interpretation of quantum mechanics is its specific viewpoint of measurement. According to this interpretation, any interaction between two entities is a measurement. We might reasonably assert from this that either the first quantity measures the second, or vice-versa. For finite times of interaction the measurement is only approximate, approaching exactness as the time of interaction increases indefinitely. But through our mental experiment we have shown that this amounts to the same thing. Thus we have succeeded in providing a reasonable argument in favor of the wave interpretation of quantum mechanics.

References

1. Shu FH (1982) The Physical Universe, An Introduction to Astronomy. University Science Books.
2. d'Espagnat B (1976) Conceptual Foundations of Quantum Mechanics. Massachusetts: Benjamin, Inc.
3. Dahlquist G, Bjorck A (1974) Numerical Methods. New York: Prentice-Hall 43-59.
4. Rader R (1979) Advanced Software Design Techniques. New York: McGraw-Hill 105-110.
5. Rifai G (1977) Supreme Council of Sciences- SYRIA 17th Science Week Publications, pp: 324-54.
6. Makarov IM (1987) Cybernetics of Living Matter. Moscow: Mir Publishers 83-86.
7. Nashelsky L (1988) Introduction to Digital Computer Technology. New York: Wiley pp: 389-420.
8. Hellerman H, Smith IA (1976) APL/360 Programming and Applications. New York: McGraw-Hill pp: 35-122..
9. Stonier T (1990) Information and the Internal Structure of the Universe. Berlin: Sprin ger-Verlag pp: 37-41.
10. Tarasov L (1980) Basic Concepts of Quantum Mechanics. Moscow: Mir Publishers pp: 25-48.
11. Davies P (1984) Superforce. New York: Simon and Schuster pp: 117-124.
12. Landshoff P, Metherell A (1979) Simple Quantum Physics. Cambridge University Press pp: 1-40.
13. Feynman R, Leighton R, Sands M (1963) The Feynman Lectures on Physics. New York: Addison -Wesley pp: 6-7.
14. Dewitt BS, Graham N (1973) The Many-World Interpretation of Quantum Mechanics. Princeton University Press pp: 167-218.