Think Analogue!

Prof. Dr. Bernd Ulmann

12. April 2012

"Think Analogue?"

Structure of this keynote:

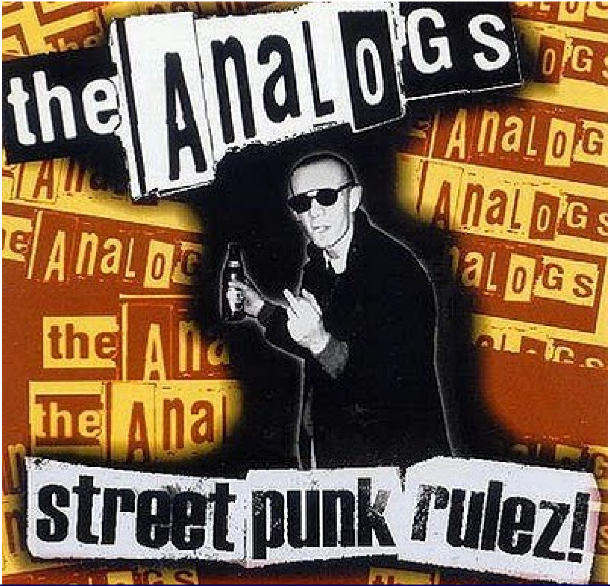
- What it is not about...
- What it is about :-)
- Some historic developments
- The power of analogs
- What can we learn for the future?

"Think Analogue!"

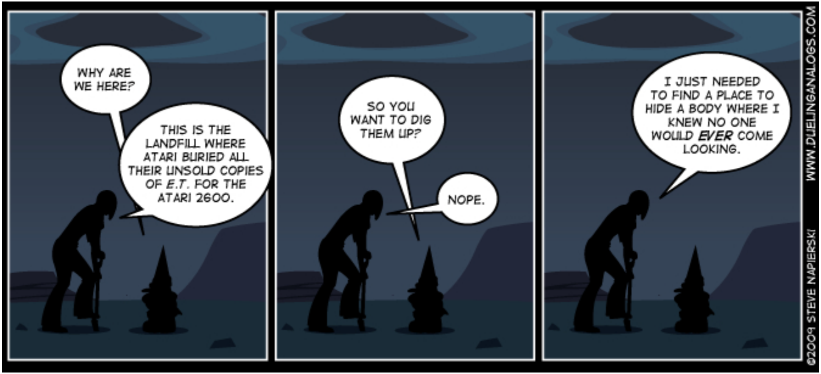
What it is not about!

What it is not about!

…it's not about "The Analogs", a Polish street punk band…



. . . it's also not about the "Dueling Analogs", a web comic by Steve Napierski[1]:



[1] I would like to thank Steve for his permission to use the above comic strip for this talk.

What it is about...

Since we will not talk about Polish street punk or Dueling Analogs (which, by the way, are both rather funny); then, what is the theme of this talk?

It is about computing by employing "models" – so called "analogs" – and the past and future use of this idea in simulation and computation.

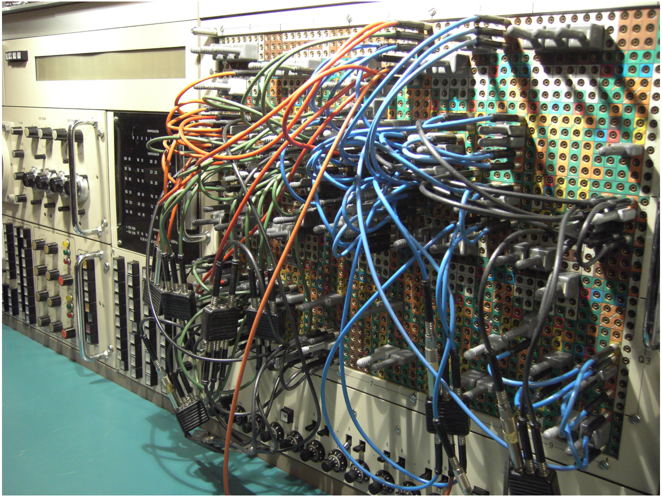
We will distinguish "direct" and "indirect" analogs (like soap bubbles for minimal surfaces and electrical circuits simulating a mechanical system).

The art of **analog computation/simulation** has been largely forgotten (like Atari's ET game mentioned in the Dueling Analogs comic) but this technology has enough potential to rise again to new heights in the future employing state of the art devices like FPGAs ("Field Programmable Gate Array") etc.

We are not too interested in direct analogies although these are very powerful, too – NASA has a program called "Extreme Analogs":

(Cf. `http://www.nasa.gov/exploration/analogs/`)

What we are interested in are "indirect analogies" like this one (but with more modern technology) showing an analog computer setup to compute airflow around a Joukowski air foil:



To create an indirect analog of a given problem the following steps will be executed:

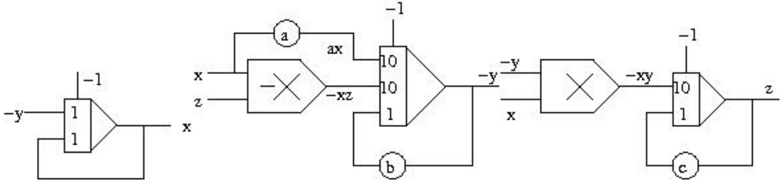
- Create a mathematical description of the problem – this will normally yield coupled differential equations.
- Transform these equations into a corresponding computer setup (a circuit consisting of basic elements like adders, integrators, multipliers etc.). There are standard approaches for this transformation.
- Setup this circuit on an analog computer thus transforming it into a model of the initial problem.

The resulting model, the "analog" will now behave like the problem described initially. In contrast to a memory programmed digital computer the analog computer changes its structure according to the problem that is to be solved while the digital computer keeps its structure and only the algorithm stored in memory will change.

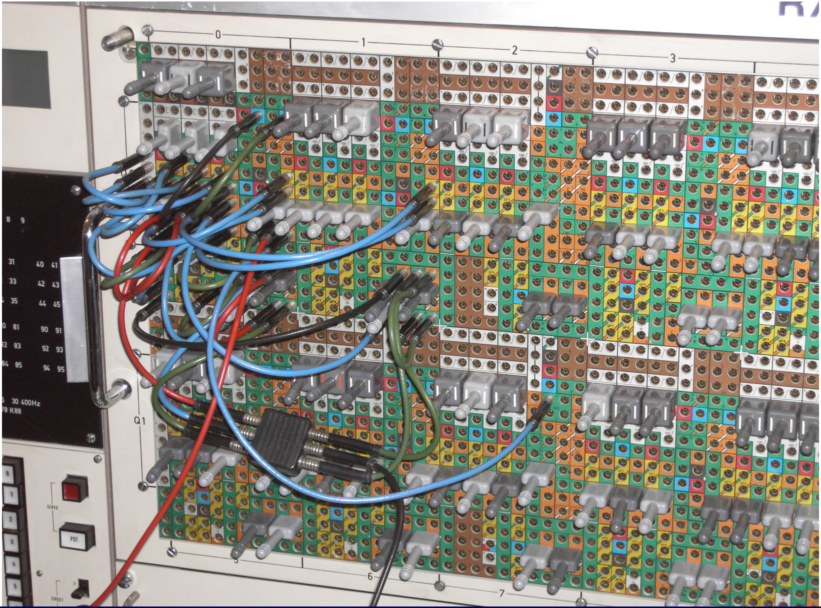
Let us have a look at an example – the Lorenz attractor that is described by the following three DEQs:

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x)\\ \frac{dy}{dt} &= x(\rho - z) - y\\ \frac{dz}{dt} &= xy - \beta z\end{aligned}$$

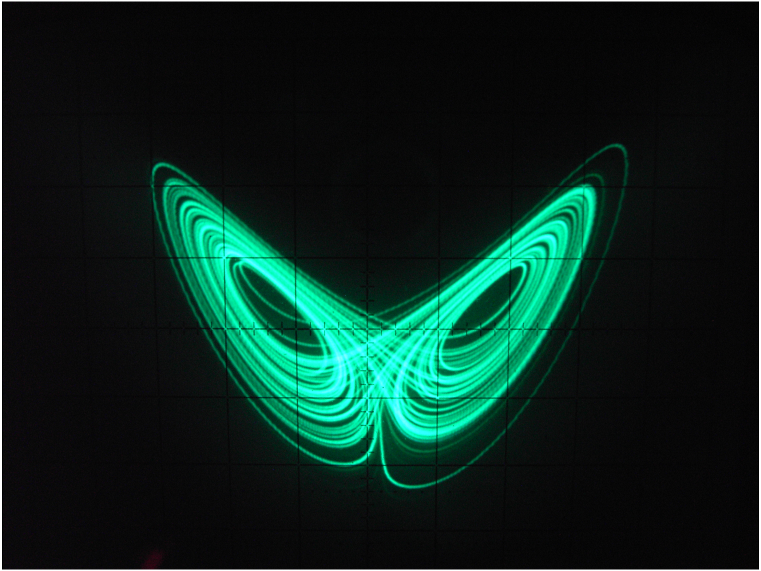
The resulting circuit looks like this:



This yields to the following computer setup:



The result of a computer run:



A short history. . .

The following slides will give an overview of selected developments of historic importance to answer the following questions:

- Why is it important to care about the history of analog computing?
- How did this fascinating technology evolve?
- Why was it forgotten?
- What can we learn from the idea of analog computing for the future?

Why should one spend time to learn about the history of topics like analog computing? George Santayana[2] has answered this basic question once and for all as follows:

> *"[...W]hen experience is not retained, as among savages, infancy is perpetual.* **Those who cannot remember the past are condemned to repeat it.** *In the first stage of life the mind is frivolous and easily distracted; it misses progress by failing in consecutiveness and persistence. This is the condition of children and barbarians, in whom instinct has learned nothing from experience."*

Accordingly we will take a short tour of the past of analog computation:

[2] Cf. [Santayana 1906].

The mechanism of Antikythera is the first mechanical analog computer ever – it dates back to 150 BC[3]:

[3] Cf. http://upload.wikimedia.org/wikipedia/commons/6/66/NAMA_Machine_d%27Anticythre_1.jpg.

What could this lump of gears do[4]?

- It could calculate sidereal, synodic, draconic and anomalistic lunar cycles.
- It took into account the first lunar anomaly (the moon's velocity is higher in its perigee compared with its apogee).
- It predicted lunar and solar eclipses and it even predicted if an ecplise would be visible or not (day/night).
- . . . and more. . . (maybe much more?)

Isn't that impressive – accomplishing all of that with nothing more than cleverly arranged gears?

For those who are interested in more details, there is a simulation of this mechanism described in [McCarthy 2009].

[4] Cf. [McCarthy 2009].

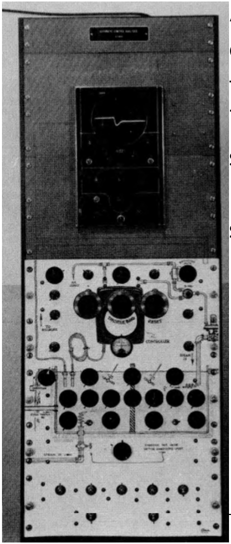
About 2000 years later, in 1876, Lord Kelvin developed the idea of using mechanical analogs to solve differential equations – his "feedback scheme" became the workhorse of the zenith of analog computation in the mid 20th century.

He discovered that a string of integrators in conjunction with other elements could be used to solve DEQs[5]:

> *[...] it seems to me very remarkable that the general differential equation of the second order with variable coefficients may be rigorously, and in a single process solved by a machine.*

Unfortunately, he did not build a real machine which would have become the world's first mechanical differential analyzer.

[5] Cf. [Thomson 1876].

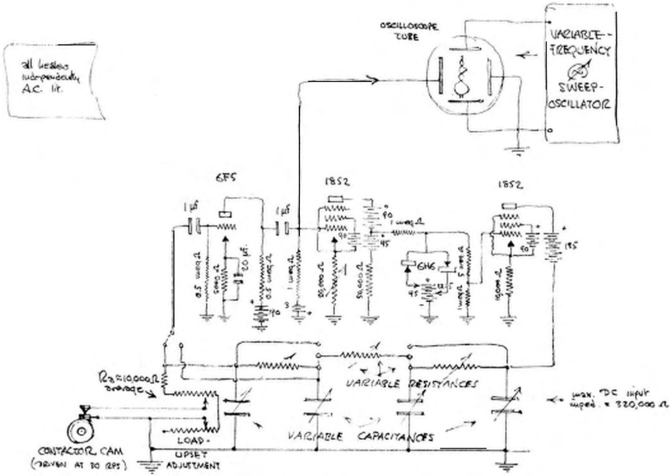


Another 62 years later George A. Philbrick developed "Polyphemus", the first electronic training simulator.

The picture on the left shows a Polyphemus setup with an illustrated process-control model involving a two-stage liquid bath with steam and cold water inflows[a].

[a] Cf. [Holst 1982][p. 152].

This drawing of Philbrick's Electronic Control Analyzer gives an impression of Polyphemus' inner workings[6]:



[6] Cf. [Holst 1982][p. 151].

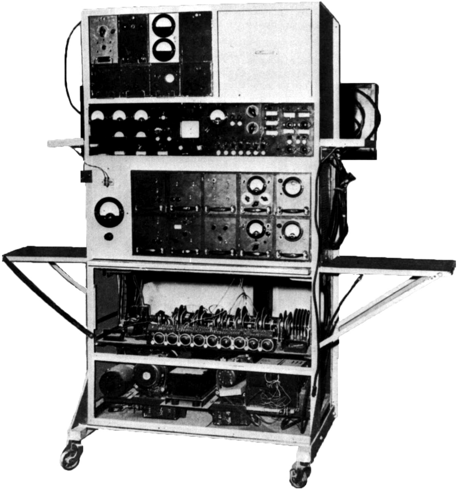
Parallel to Philbrick but completely independent, Helmut Hoelzer developed another analog computer – in fact he developed two:

- The so called "Mischgerät" was the world's first on-board computer and was used in the guidance section of the V2 rocket.
- A truly general purpose electronic analog computer, used by NASA in rocket development at Redstone Arsenal, up to the late 1950s.

The Mischgerät[7]:

[7] Photo by Adri de Keijzer.

Hoelzer's analog computer as it appeared at the end of WWII[8]:



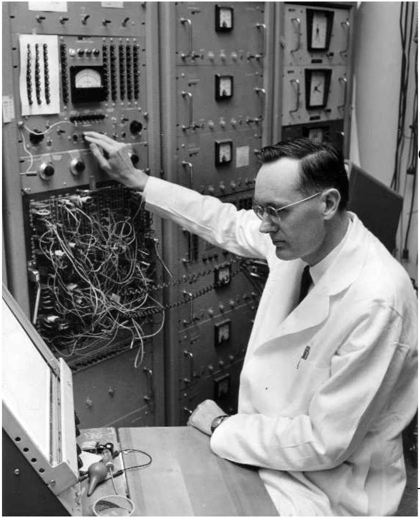
[8]Source: NASA, Marshall Space Flight Center.

After WWII the development of analog electronic analog computers advanced with great pace.

Companies like Electronic Associates Inc., Beckman, Systron Donner, Telefunken and many more developed general purpose analog computers.

These analog computers shaped the world as we know it – without their help many things we take for granted would not have been possible (at least not at this time before the advent of memory programmed digital computers).

The following slides show some archetypical general purpose analog computers to give an impression of their size, power and shape.



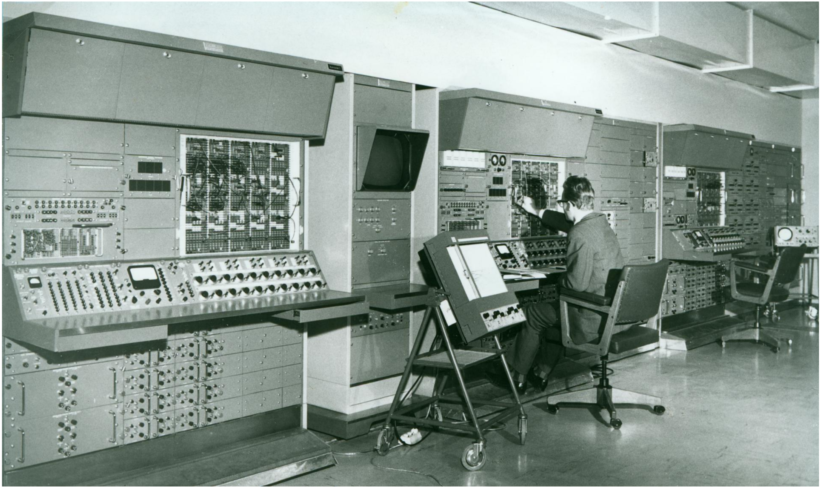
Richard FitzHugh solves the Hodgkin-Huxley equations using a Beckman analog computer[a]:

[a] Source: Izhikevich E. M., FitzHugh R.

EAI 231R-installation at the "Deutsches Elektronensynchrotron", Hamburg[9]:

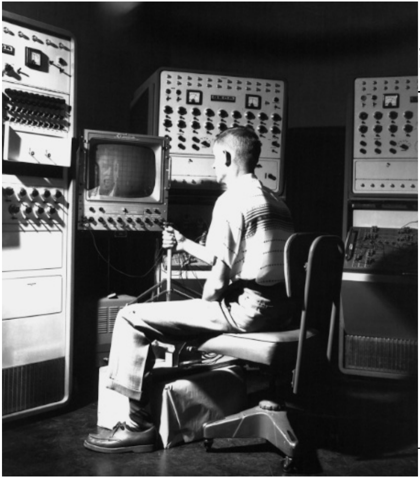
[9] Picture: Inge Borchardt

EAI installation at the "DLR"[10]:



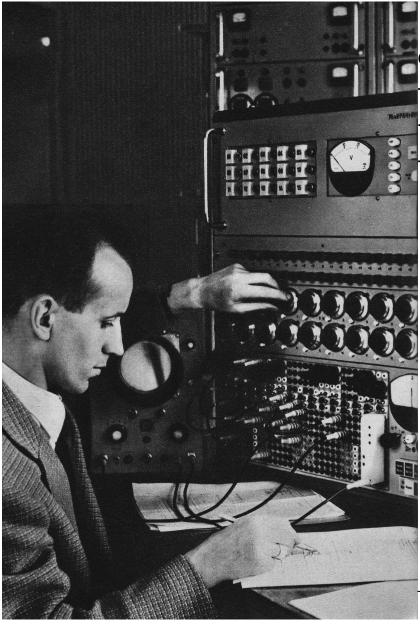
[10] Picture: Dr. Jessika Wichner, DLR, signature FF-557.

EAI Pacer hybrid computer (1970s):



Richard Day runs an inertia-coupling simulation[a].

[a] NASA photo E-1841.



Mr. Winkler, an engineer of the "Laboratory Walter" working with a Telefunken RAT 700 table top analog computer[a].

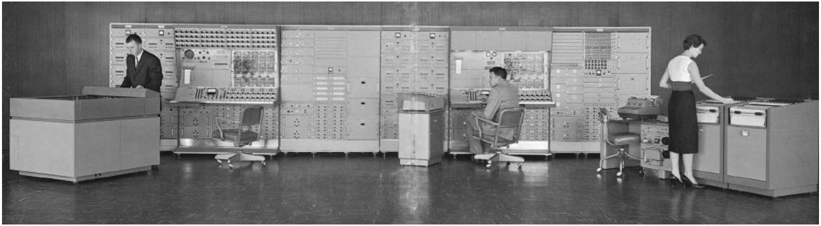
[a] Cf. [Gerwin 1964][p. 84].

Five RAT 700 computers and a minimum RA 800 system:

In the following some typical and influential application areas of analog computers are listed:

- Mathematics (DEQs, partial DEQs, boundary value problems, Monte Carlo techniques, . . .)
- Physics (beam optics, heat transfer, semiconductor physics, . . .)
- Chemistry (reaction kinetics, quantum chemistry, process engineering, . . .)
- Mechanics (analysis of vibrations, rotating systems, plasto mechanics, hydraulics, servosystems, . . .)
- Nuclear physics (reactor simulation, training, control circuits, . . .)
- Biology and medicine (ecosystems, pharmacological research, epidemiology, neurophysiology, . . .)
- Power engineering (power distribution networks, network simulation, frequency regulation, . . .)
- Electronics (filter design, spectrum analysis, . . .)
- Control engineering
- Aerospace engineering (an example follows)

Using analog computer systems like this installation[11]



high performance planes like the Bell X2 were successfully simulated. Initially this dismayed the pilots who regarded it is certainly **not** the "Right Stuff". Dick Day remembers[12]:

> *Well, the simulator was a new device that has never been used previously for training or flight planning. Most pilots had, in fact, expressed a certain amount of distrust in the device.*

[11] Cf. [EAI].

[12] Cf. [Waltman 2000][p. 138 f.].

In a particular impressive example, it was shown by simulation that the X2 would become unstable at around Mach number 3.0 under certain circumstances[13]:

> *We showed [Mel Apt (the test pilot)] if he increased AOA [(angle of attack)] to about 5 degrees, he would start losing directional stability. He'd start this, and due to adverse aileron, he'd put in stick one way and the plane would yaw the other way[...]*

Maybe due to the distrust of pilots concerning analog simulation techniques Mel Apt nevertheless did the maneuver mentioned above when he ran out of fuel and tried to head back to the base... Apt did not survive (as predicted)...

[13] Cf. [Waltman 2000][p. 138 f.]

Mal Apt in the cockpit of the X2[14]:

[14] Cf. http://upload.wikimedia.org/wikipedia/commons/2/2f/Captain_Mel_Apt_in_Bell_X-2_1956.jpg.

The end of this flight which boosted the trust in analog computing[15]:

[15] Cf. [Merlin, Moore 2008][p. 20].

What was the advantage of analog computing that led to this wide spread use?

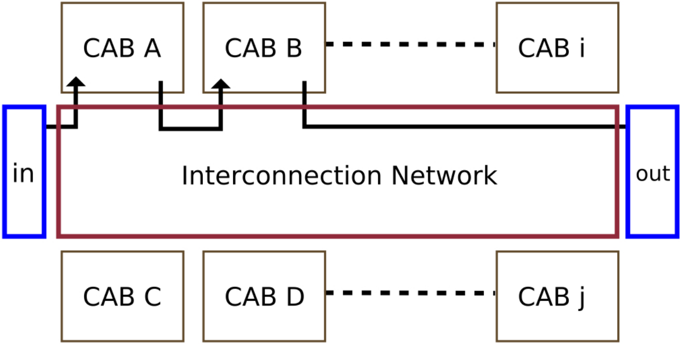
- Programming an analog computer is close to the problem to be solved and not abstract as traditional algorithmic approaches. An analog computer "talks the language of the engineers".
- Analog computers offer an uniquely high amount of interactivity.
- Analog computers are highly parallel machines thus surpassing even modern memory programmed digital computers in terms of parallelism and maybe even in terms of sheer computing power.
- Analog computers do not have a "von Neumann"-bottleneck since they change their structure according to a problem without needing program memory at all.
- Analog computers can be easily expanded – if a problem grows larger the computer can grow with the problem.

Despite these advantages the decline of analog computers started in the 1970s for a number of reasons:

- Memory programmed digital computers became fast enough to compete at least with small and medium analog computers in terms of computing power.
- Memory programmed digital machines became simpler and thus cheaper each month while analog computers continued being highly complex machines – expensive and hard to maintain and run.
- Digital computers offered more flexibility due to time sharing etc. which allowed a better utilization of the machines.
- It is easier to build a very small digital computer than an equally small analog electronic analog computer.
- Over time, the expertise in analog computing was lost – abstract algorithmic approaches won over analog approaches. A change of paradigm occurred.

So what?

So what? The analog electronic analog computer is dead – who cares? We should care because now there are reconfigurable analog integrated circuits, so called "FPAA"s, short for "Field Programmable Analog Arrays", with a structure like this[16]:



[16] Source: http://de.wikipedia.org/w/index.php?title=Datei:Field_Programmable_Analog_Array_diagram.svg&filetimestamp=20070914115453, "CAB"s are "Configurable Analog Block"s.

We should care because the basic strengths of the idea of analog computation can also be readily transferred to other modern technologies like FPGAs[17] and the like.

In fact, it is not necessary to build an analog computer from analog electronic components – digital analog computers are not only possible but combine the best features of both worlds: The speed an inherent fine grain parallelism of analog computers with the high integration, low maintenance overhead and cheap price of digital components.

The idea itself is not even new – there were a number of so called "Digital Differential Analyzers", "DDA"s for short:

[17] Short for "Field Programmable Gate Array"s.

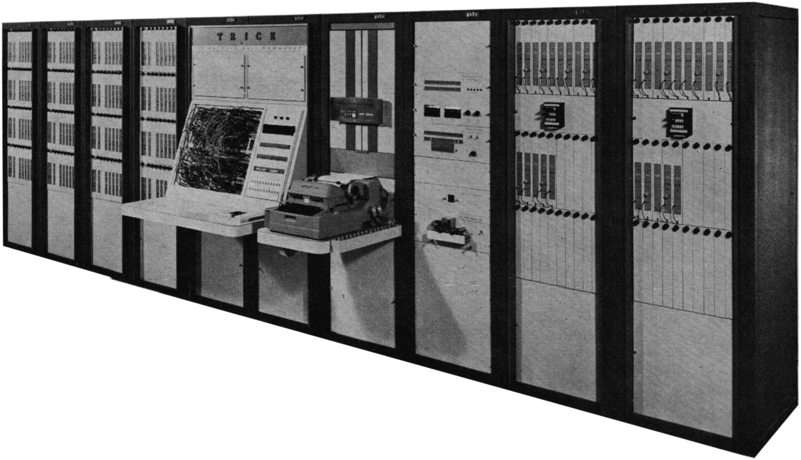
In 1949 Northtrop started development of the "Magnetic Drum Differential Analyzer", "MADDIDA" for short[18]:

[18] Picture: Navy Electronics Laboratory, File Number E1278.

In the early 1950s, Bendix developed the D-12 DDA[19]:

[19] Cf. [Bendix][p. 2].

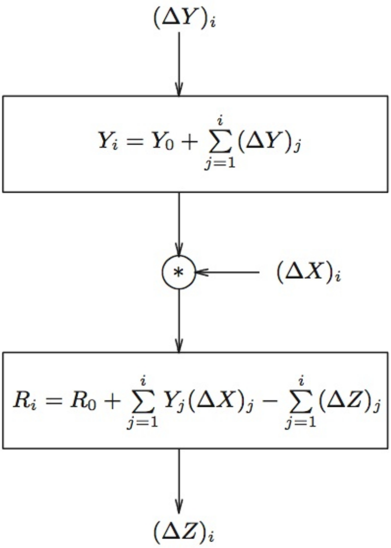
TRICE, early 1960s, Raytheon[20]:



[20] Cf. [Ameling 1963][p. 30].

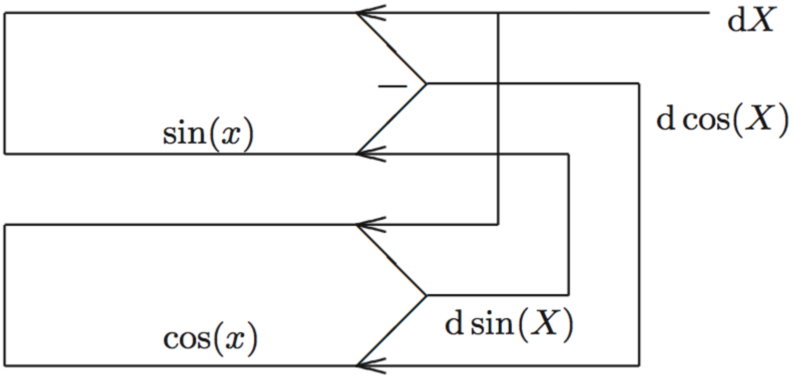
While MADDIDA and the D-12 are sequential DDAs, i.e. they contain only a single ALU that is timeshared to serve all computational elements that make up a computer setup, the TRICE is a truly parallel DDA and might serve as a model for future developments.

The basic computational element of a DDA, the integrator, can be very simple (although more elaborate schemes relying on Runge-Kutta integration or the like will be possible using FPGAs): All that is needed is an accumulator Y that is incremented by inputs ΔY while its output is multiplied with an input $\Delta X \in \{-1; 0; 1\}$. The result of this multiplication is the input to a second accumulator. This accumulator yields a carry output that can be used as input to other integrators etc.

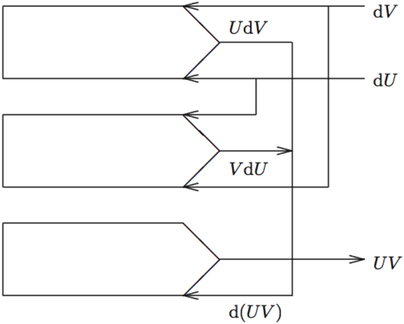


The function to be integrated upon is approximated by a step function yielding the ΔY, while ΔX represents the dx in the integration. The overflow output ΔZ is then used as input to following stages.

The following slide shows the generation of a sine/cosine pair using integrators like this:



$\mathrm{d}X$
$-$
$\mathrm{d}\cos(X)$
$\sin(x)$
$\cos(x)$
$\mathrm{d}\sin(X)$



Such a digital differential analyzer has one advantage over a traditional analog electronic analog computer: Due to its ΔX-inputs it is not limited to the time as the free variable of integration!

Thus multiplication, based on the idea

$$UV = \int_{U_0}^{U_1} V\,\mathrm{d}U + \int_{V_0}^{V_1} U\,\mathrm{d}V + U_0 V_0$$

and other tricks are possible. Furthermore partial differential equations can be solved easily, too (a traditional analog computer requires laborious discretisation).

Using "Field Programmable Gate Arrays" implementing something like TRICE would be easy – one could also abandon the incremental computation scheme described above and work with either long integers (getting all the normalization problems back that plagued analog electronic computers) or floating point numbers (with all their drawbacks in mind).

Due to the lack of program control the analog computing paradigm offers a fine grained parallelism that allows such a system to easily outperform classical memory programmed digital computers in areas where the solution of coupled differential equations is of prime importance. Furthermore such a system would be much more energy efficient – maybe an interesting point, too, in the days of "Green IT".
Several large financial institutions already do basic research concerning the application of FPGAs to simulations etc.[21].

[21] Cf. [Thomas, Luk 2007], [Thomas, Luk 2007], ...

Another approach might be based on highly parallel memory programmed digital computer architectures instead of FPGAs.

An example for such a base system (just brainstorming) is the "Green Arrays GA144" chip[22] which contains 144 independent F18A processors – each of which might implement a group of traditional analog computing elements thus forming high level building blocks (renewing interest in the "Hannauer matrix" – cf. [Hannauer 1968]).

[22] Cf. [Green Array 2010].

Yet another approach might be the implementation of a DDA using a VLIW architecture.

In fact EAI did some research along this path which eventually led to the "Starlight" system (mainly due to the collapse of EAI it was not produced in quantity).

Also graphics cards could be a viable basis for a modern implementation of an analog computer – their high degree of parallelism and their vector architecture seem well suited for the implementation of a parallel DDA.

Thank you for your interest!

Enjoy the conference! :-)

The author can be reached at

`ulmann@analogmuseum.org`

Personal note: If you have any old analog computing equipment, documentation etc. that you might want to donate, please contact the author.

Bibliography

- [Ameling 1963] Walter Ameling, "Aufbau und Arbeitsweise des Hybrid-Rechners TRICE", in *Elektronische Rechenanlagen*, 5 (1963), Heft 1
- [Bendix] Bendix Computer, "Digital Differential Analyzer D-12"
- [EAI] N. N., *PACE 231R analog computer*, Electronic Associates, Inc., Long Branch, New Jersey, Bulletin No. AC 6007
- [Gerwin 1964] Robert Gerwin, "Intelligente Automaten", Chr. Belser Verlag, 1964
- [Green Array 2010] N. N., "Green Array GA144 – 144-Computer Chip", `http://www.greenarraychips.com/home/documents/greg/PB001-100503-GA144-1-10.pdf`
- [Hannauer 1968] George Hannauer, *Stored Program Concept for Analog Computers*, EAI Project 320009, NASA Order NAS8-21228, 1968

- [Holst 1982] Per A. Holst, "George A. Philbick and Polyphemus – The First Electronic Training Simulator", in *Annals of the History of Computing*, Vol. 4, Number 2, April 1982, pp. 143 – 156
- [McCarthy 2009] Jerry McCarthy, "Der Mechanismus von Antikythera", in Tagungsband, 15. Internationales Treffen der Rechenschiebersammler und 4. Symposium zur Entwicklung der Rechentechnik, Universitaet Greifswald, 2009, pp. 55 – 64
- [Merlin, Moore 2008] Peter W. Merlin, Tony Moore, *X-Plane Crashes*, Specialty Press, 2008
- [Santayana 1906] George Santayana, "The Life of Reason or the Phases of Human Progress", Vol. 1 of 4 ("Introduction and Reason in Common Sense"), London, Archibald Constable & Co. Ltd., 1906, S. 284

Bibliography

- [Thomas, Luk 2007] David B. Thomas, Wayne Luk, "Credit Risk Modelling using Hardware Accelerated Monte-Carlo Simulation", Proceedings of FCCM, 2008
- [Thomas, Luk 2007] David B. Thomas, Wayne Luk, "A Domain Specific Language for Reconfigurable Path-based Monte Carlo Simulations", Proceedings of FPT, 2007, pp. 97–104
- [Thomson 1876] William Thomson, "Mechanical integration of linear differential equations of the second order with variable coefficients", Proceedings of the Royal Society, Vol. 24, No. 167, 1876, pp. 269 – 270
- [Waltman 2000] Gene L. Waltman, *Black Magic and Gremlins: Analog Flight Simulations at NASA's Flight Research Center*, NASA SP-2000-4520, 2000

The author would like to thank the following persons for their help in preparing this talk (in alphabetical order):

- Bianca Brunner
- Patrick Hedfeld
- Rikka Mitsam
- Joachim Wagner