Analog- and Hybridcomputing

Prof. Dr. Bernd Ulmann

7. März 2022

Problems of classic digital computers

Why analog/hybrid computing?

Although stored program digital computers are ubiquituous by now, they have a number of drawbacks which get more and more of a real problem for future applications. Some of these problems inherent for digital computers are:

- High power consumption
- Clock frequencies are limited (energy consumption, electron mobility)
- On-chip structures can not be shrinked much further – atoms are quite big at all...
- Parallelism is hard to exploit with digital computers (AMDAHL's law)
- Only a tiny fraction of the silicon used in a digital computer actually performs computations, the vast majority deals with data storage, data management etc.

The following slides depict some of these problems.

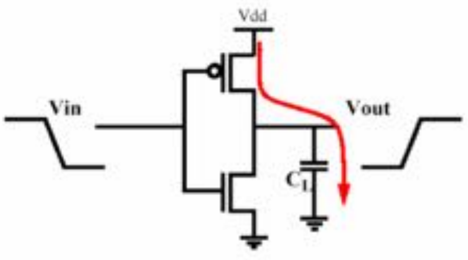
Size

https://upload.wikimedia.org/wikipedia/commons/1/1f/Columbia_Supercomputer_-_NASA_Advanced_

Cooling

http://www.lanl.gov/newsroom/picture-of-the-week/pic-week-1.php, 23.01.2017.

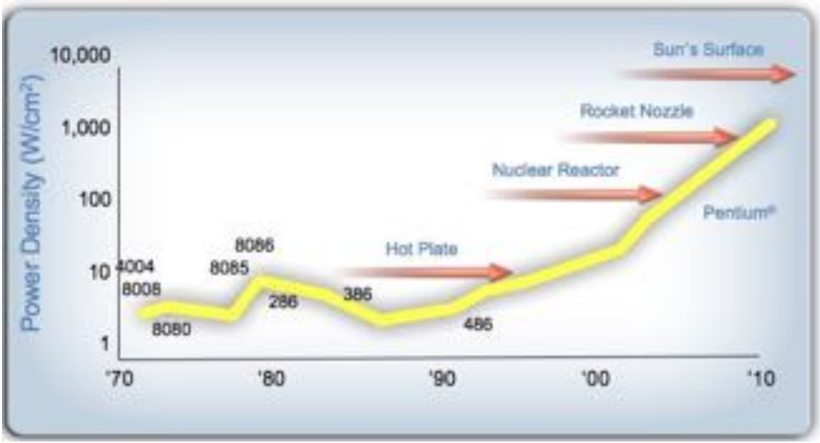
A typical output stage looks like this:



Problems:

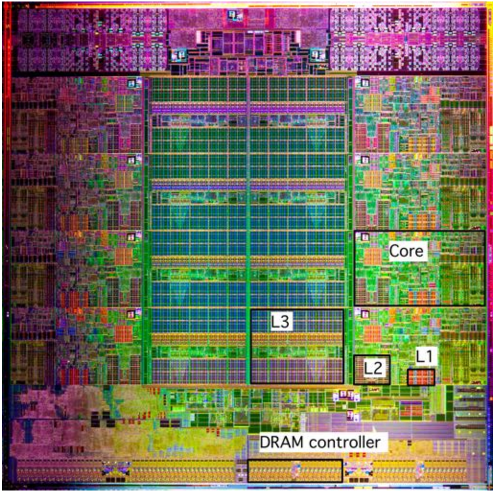
- C_L gets charged/discharged at every change of the output.
- Both output transistors are switched on (very briefly) during a state transition.
- The transistors are not perfect and exhibit leakage currents.
- $P_\text{cpu} = P_\text{dyn} + P_\text{sc} + P_\text{leak}$ is super-linear with respect to the clock frequency.

Power density



http://pages.tacc.utexas.edu/~eijkhout/istc/html/sequential.html, 23.01.2017.

Chip micro-photograph of Sandybridge:



http://pages.tacc.utexas.edu/~eijkhout/istc/html/sequential.html, 23.01.2017.

Typical chip structure

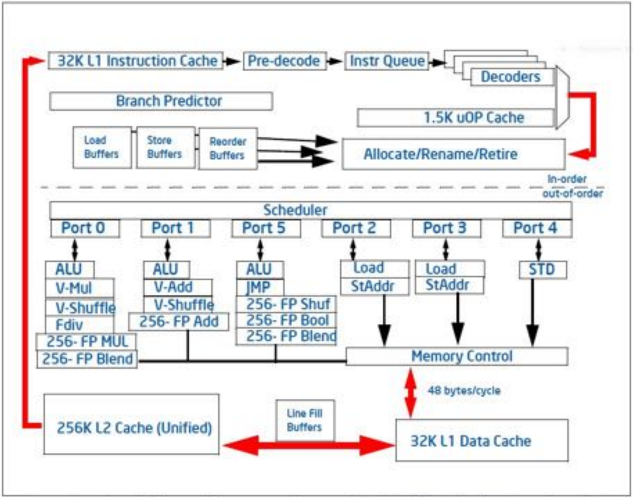


Figure 2-1. Intel microarchitecture code name Sandy Bridge Pipeline Functionality

`http://pages.tacc.utexas.edu/~eijkhout/istc/html/sequential.html`, 23.01.2017.

Fastest digital supercomputer (`top500.org`): Fugaku, RIKEN Center for Computational Science, Japan:

- 442,01 PFLOPS
- 7 630 848 Cores
- Ca. 30 MW of electrical power
- About 14 GFLOPS/W

Most energy efficient digital computer (`green500.org`): Shoubu system B am Advanced Center for Computing and Communication, RIKEN, Japan:

- 842 TFLOPS
- 794 400 Cores
- Ca. 50 kW
- About 17 GFLOPS/W

Human brain:

- Ca. 38 PFLOPS
- 25 W
- Not digital but analog!

Analog computing

There are, in fact, three classes of computers:

- Digital (stored program) computers,
- quantum computers (still far from being a product), and
- analog computers.

Neither analog nor quantum computers will replace digital computers. Instead they will complement digital computers by acting as co-processors to speed up computations way beyond what would be possible by using the digital computer alone.

The following slides cover analog computers (of which also digital implementations exist - but that's for another talk) which should be called "analog electronic analog computers" to be exact. Since this is a bit complicated, so we will just speak of "analog computers" or "AC"s instead.

Analog computers

What are the basic characteristics of an analog computer?

- Analog computers exhibit an extremely high energy-efficiency – much higher than digital computers in many application areas.
- Values are represented as continuous voltages or currents, not as discrete bits.
- Analog computers are inherently parallel and do not suffer from AMDAHL's law etc.
- Analog computers adapt seamlessly to our analog world – no need for data conversions etc.
- Analog computers are not true general purpose computers but excel at problems that can be described by (systems of) differential equations, at optimization problems, and may even be better suited to simulate quantum computers than digital computers.

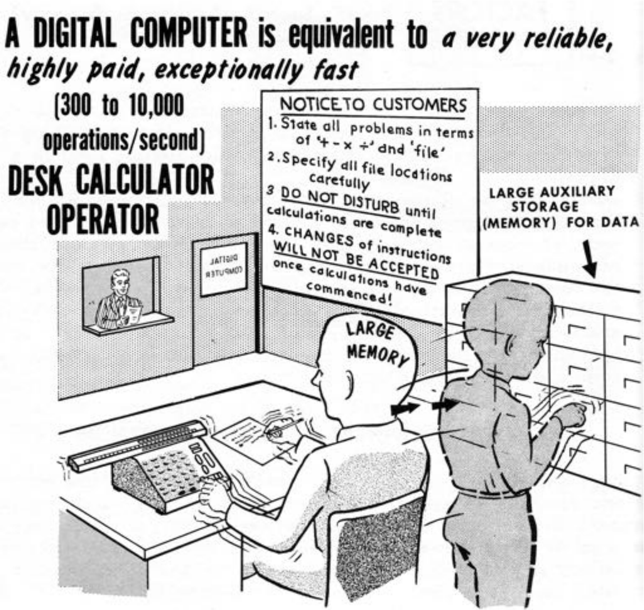
An analog computer

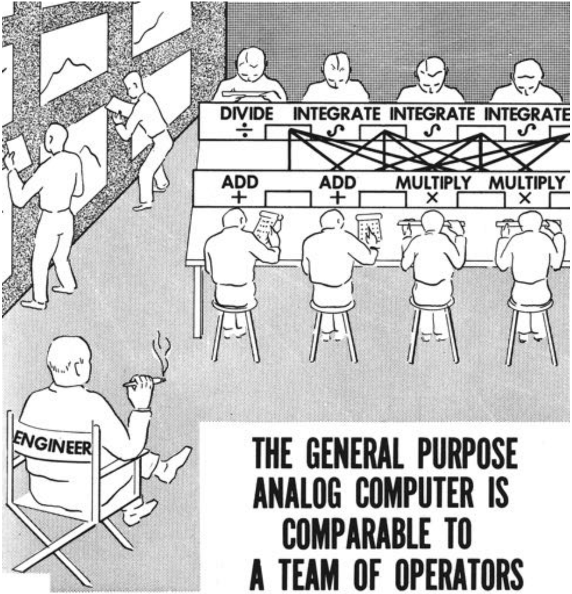
- contains a plethora of computing elements such as integrators (!), summers, multipliers etc. (not only a few computing units as in a digital computer!).
- These computing elements are connected to each other in order to form an *analogue* for the problem to be solved (that's where the name stems from). So there is **no algorithm** and **no memory** at all.
- All computing elements work in full parallelism exchanging time varying values by means of their connections.

The following (historic) pictures show the main differences between a digital computer and an analog computer:[1]

[1] Cf. [Truitt et al.(1960), pp. 1-40/41].

Digital computer




DIVIDE
÷
INTEGRATE
INTEGRATE
INTEGRATE
ADD
+
ADD
+
MULTIPLY
×
MULTIPLY
×
ENGINEER
THE GENERAL PURPOSE ANALOG COMPUTER IS COMPARABLE TO A TEAM OF OPERATORS

Disadvantages

Since there is nothing like a free lunch analog computers have some disadvantages, too:

- Limited precision – typically about 3 to 4 decimal places (it turns out that this is often not a problem at all).
- Only time is directly available as free variable of integration, so solving partial differential equations requires some transformations/tricks.
- Generating arbitrary functions, especially those having more than one argument, requires special circuitry.
- Values are limited to the interval $[-1, 1]$, so problems must typically be *scaled* accordingly.

What is a hybrid computer?

In modern applications, analog computer will be coupled with traditional digital computers forming *hybrid computers*:

- A hybrid computer is a combination of a traditional stored program digital computer and an analog computer.
- The analog computer offloads the digital computer from compute intensive tasks and thus acts as a co-processor.
- The digital computer controls the configuration and parametrization of the analog computer.
- The analog co-processor must be integrated in a seamless fashion requiring an extensive and complex software stack including an abstract programming language for the analog computer, associated compilers, libraries etc.

Applications

What are typical applications of an analog computer on chip (either used standalone or as part of a hybrid computer setup)?

- High performance computing
- Artificial intelligence (implementation/training of ANNs etc.) and machine learning
- Medical applications (cardiac pacemakers, brain pacemakers, insulin sensing/pump control) – the power consumption could be sufficiently low to power some implants with energy harvesting thus getting rid of clumsy batteries which limit life span and have to be recharged etc.
- Industrial control systems (basically stateless, not hackable in a traditional sense)
- Signal pre-/postprocessing for mobile devices and the like
- trigger word detection for smart devices
- Monte-Carlo simulations, financial mathematics
- Optimization problems

Historic analog computers

- Analog computers were dominant machines from the late 1940s to about the mid-1970s.
- They were (and still are) much faster than stored-program digital computers and were used extensively in aerospace technology, vehicle design, chemistry, medicine etc.
- Analog computers were superseded by stored-program digital computers mainly due to maintenance issues, price and programmability

Telefunken RA800, 1960

Telefunken RA770, 1966

VW car simulator, 1970s

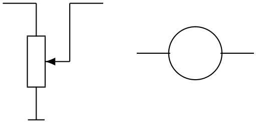
Why were classic analog computers displaced by digital computers?

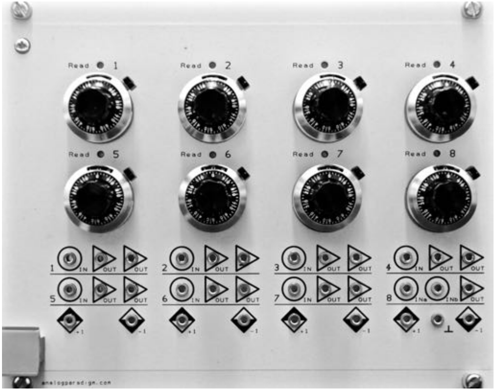
- Programming was very time consuming, switching from one program to another also took a long time.
- There was no way to use such a computer in a time sharing fashion.
- Digital computers became increasingly cheaper while these classic analog computers were quite expensive.

All of these problems can be easily overcome by modern integrated circuit technology making the inherent advantages of analog computers available for the 21st century.

Examples

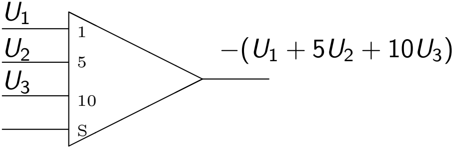
- Coefficient units (e. g. a voltage divider/multiplying DAC etc.)

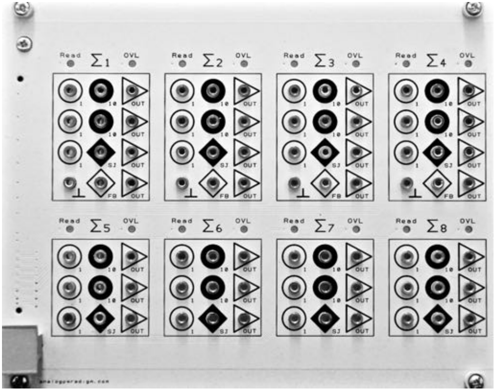




Typical computing elements

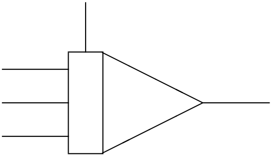
- Summers, computing $-U_{\text{out}} = \sum_{i=1}^{n} a_i U_i$:





- Integrators, computing[2]

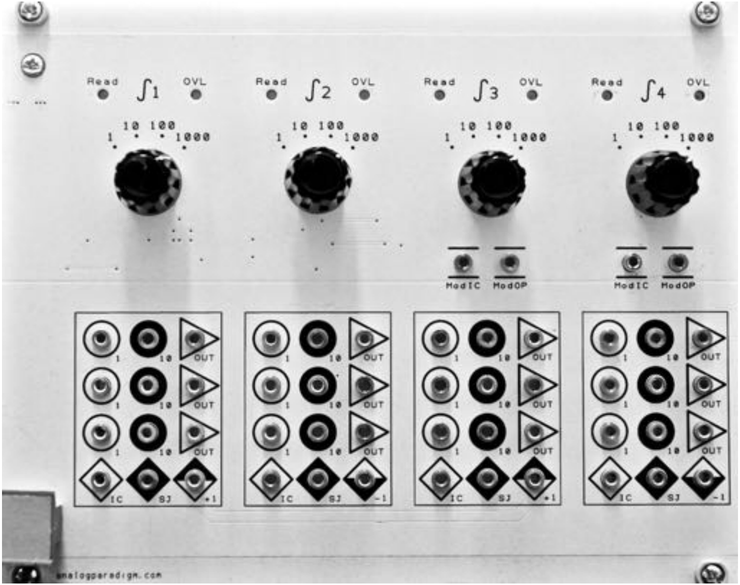
$$-U_{\text{out}} = \int\limits_0^t \sum_{i=1}^n a_i U_i(t)\, \mathrm{d}t + U(0)$$



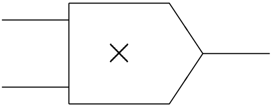
An integrator has three modes of operation: *initial condition*, *operate*, *halt*.

[2] The "magic component" – (near) perfect integration with a resistor, a capacitor and an operational amplifier.

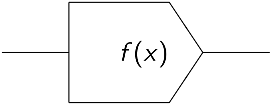
Typical computing elements



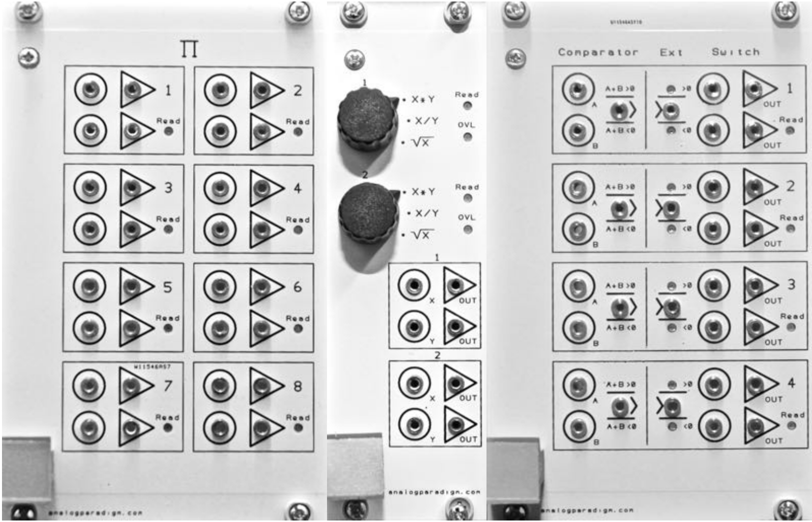
- Multipliers

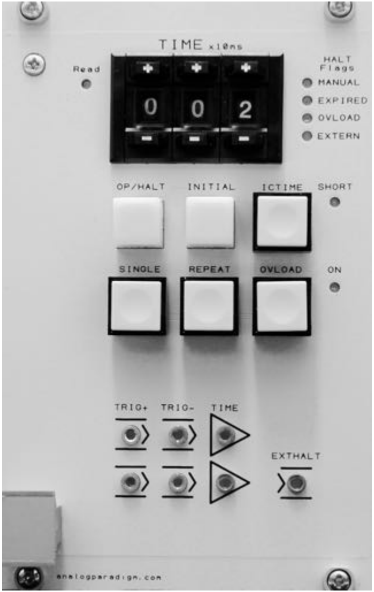


- *Open amplifiers* which can be used to implement inverse functions such as square root, division as well as special functions like limiters etc.
- Comparators with electronic switches
- Function generators (traditionally using polygonal interpolation, nowadays table-lookups using ADCs and DACs)



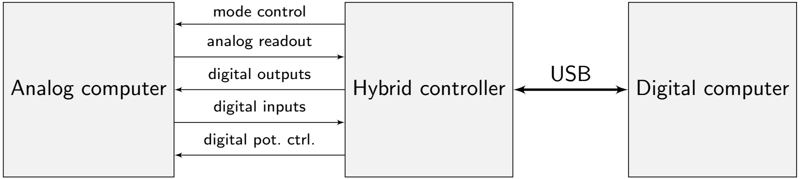
Typical computing elements



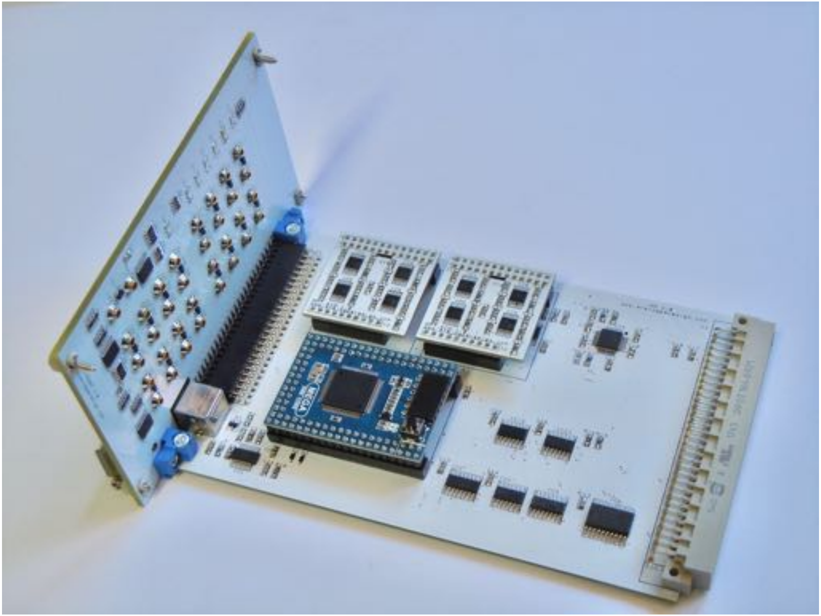


Shown on the left is a typical control unit. It allows full manual control of the analog computer (initial condition, operate, halt) and also supports single run operation as well as repetitive operation.

Hybrid controller



Hybrid controller



Two modern analog computers

Model-1

THE-ANALOG-THING

Basic programming – ODEs

Programming an analog computer is much simpler than programming a stored program digital computer as there is nothing like an intricate algorithm controlling the system.

An analog computer programm, i. e. the interconnection scheme for its computing elements, is derived from the problem equations.

A classic way to develop an analog computer setup is due to Lord Kelvin:

1. Solve the equation for the highest derivative.
2. Generate all lower derivatives by repeated integrations.
3. Derive all terms on the right hand side of the equation based on these lower derivatives.
4. Tie all these terms together. According to the equation this process started with, this must be equal to the highest derivative, so this value can be fed back into the circuit as this highest derivative.

Programming

A simple example

Often, a sine/cosine signal pair is required in a simulation. A typical way to generate this on an analog computer is by solving the DEQ

$$\ddot{y} + \omega^2 y = 0 \tag{1}$$

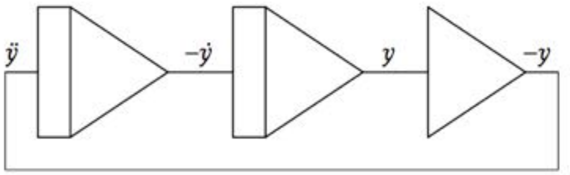
with the initial conditions

$$y_0 = a \sin(\varphi) \text{ und}$$
$$\dot{y}_0 = a\omega \cos(\varphi)$$

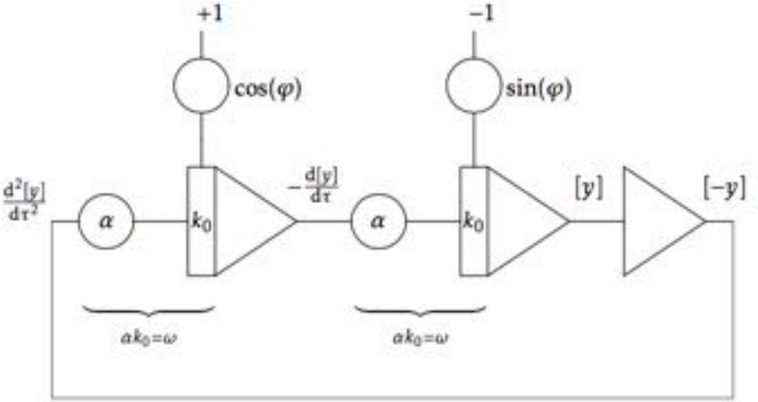
Solving (1) for $\ddot{y}$ and assuming $\omega = 1$ yields

$$\ddot{y} = -y$$

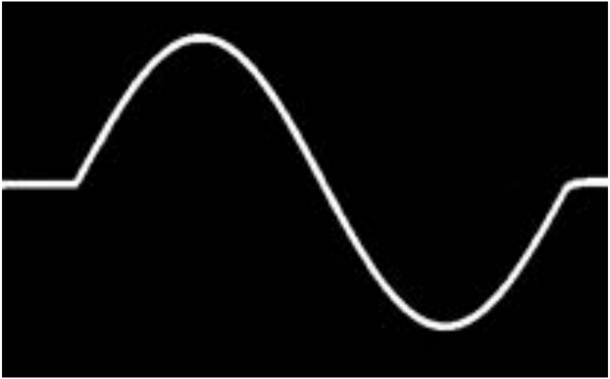
which leads to the following program sketch:



Taking ω into account as well as the initial conditions yields the following complete program:



This yields the following output at the second integrator (actual screen shot – the analog computer was running for the time of one period as determined by ω):



Two more complex examples:

- van der Pol-equation
- the heat equation

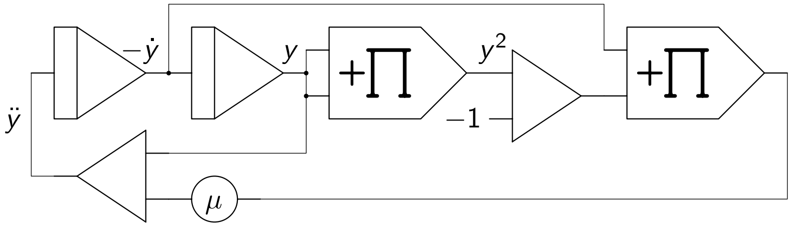
The VAN DER POL equation looks like

$$\ddot{y} + \mu\left(y^2 - 1\right)\dot{y} + y = 0,$$

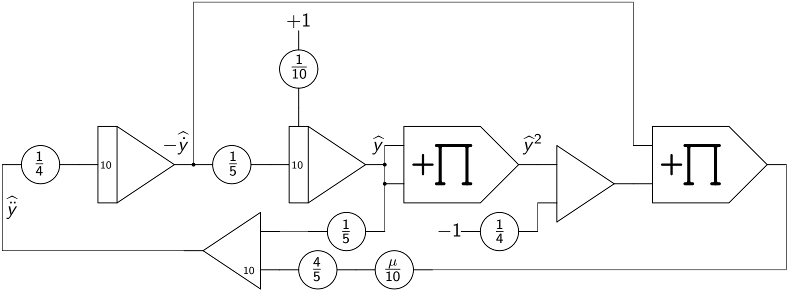
which can be rearranged, yielding

$$\ddot{y} = -y - \mu\left(y^2 - 1\right)\dot{y}.$$

The unscaled analog computer program looks like this:

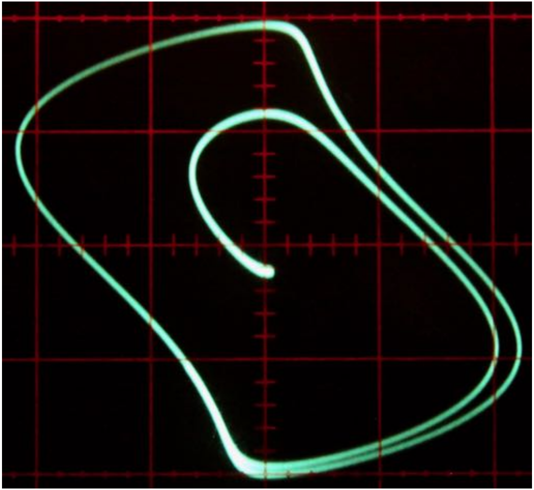


This problem must be scaled so that no variable exceeds the interval $[-1, 1]$ yielding the following program:



A typical phase space plot generated with this setup looks like this:

van der Pol equation



An analog computer setup for the two-dimensional heat-equation starts with

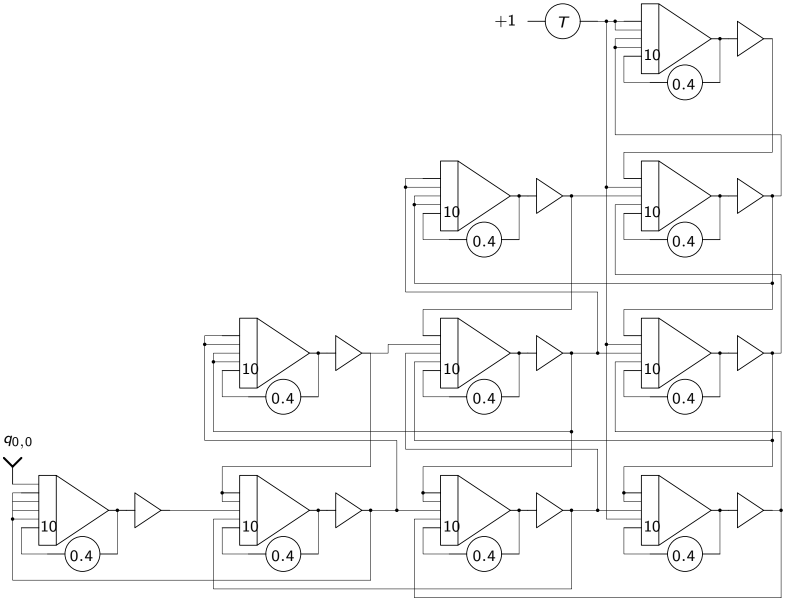
$$\dot{u} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right).$$

Since an analog computer can only integrate with respect to time, an obvious aproach would be to replace the right-hand side of the above equation by a difference term:

$$\dot{u}_{i,j} = \alpha \left(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} \right) + q_{i,j}.$$

This can now be treated as shown before. In the following one octant of a quadratic sheet is modelled (taking a symmetry argument into account):

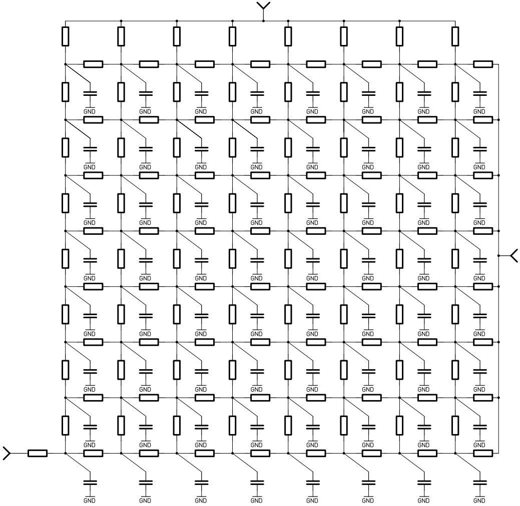
The heat equation



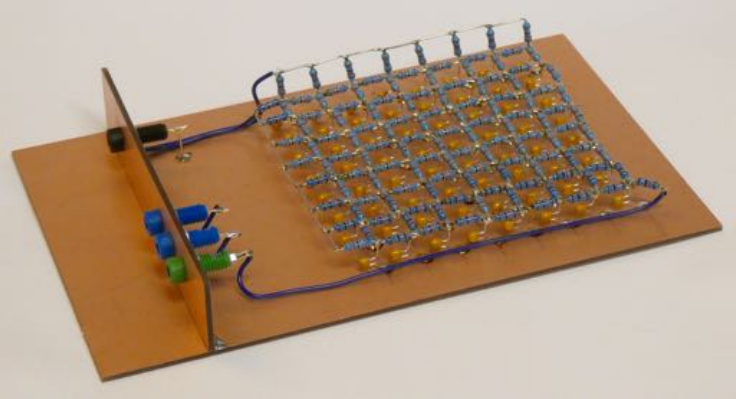
Passive indirect Analogies – the heat equation

A passive analogue

The following example shows a simple passive model for the two-dimensional heat-equation:

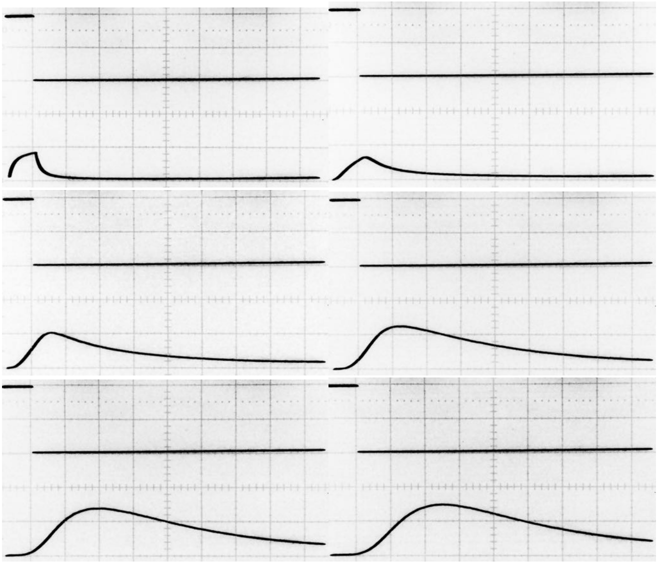


A passive analogue

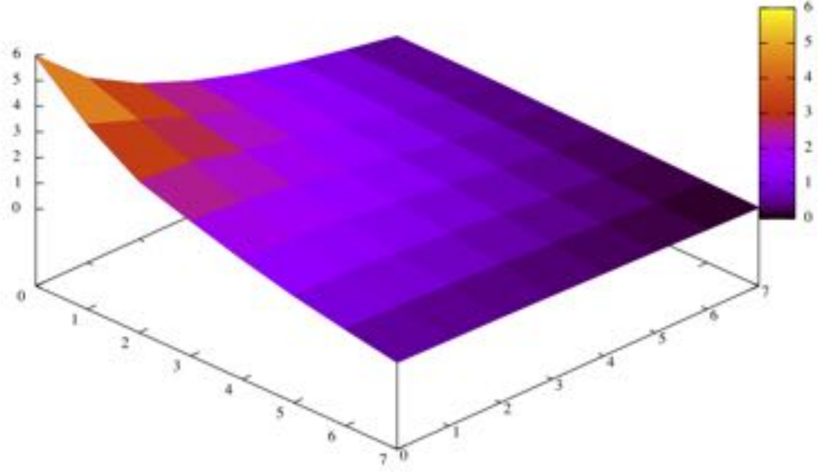


A passive analogue

Response along the diagonal elements to an input pulse at $u_{0,0}$:



Steady state solution:



Modern developments

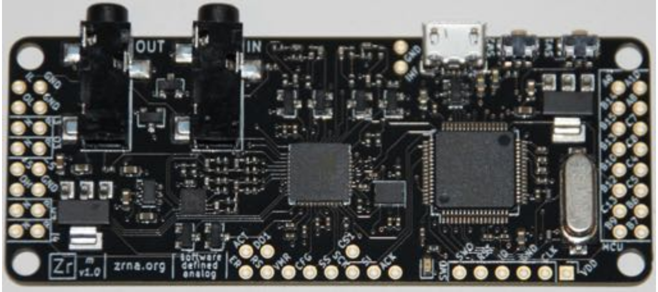
Historic analog computers have some severe drawbacks that must be overcome for future applications:

- These machines are bulky due to their implementation based on discrete electronic components. This will be easily overcome by using modern integrated circuit technology.
- Setting up a traditional analog computer is way too cumbersome:
  - The patch panel programming takes too much time.
  - Changing programs is a mechanical process (removable patch panels).
  - Setting up coefficients and function generators is time consuming and error prone.

There are already some recent developments such as these:

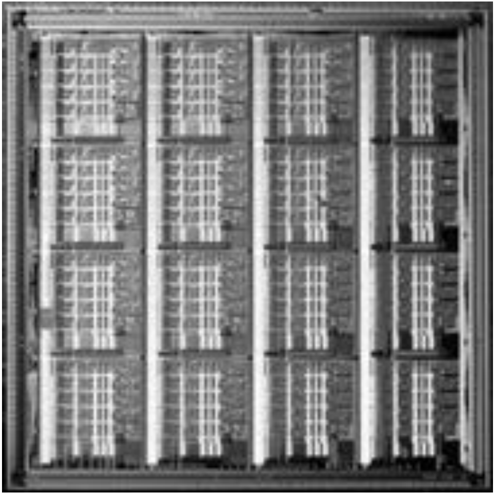
Anadigm

Anadigm manufactures FPAAs (using switched capacitors for the central interconnect structure) mainly aimed at the signal processing market. A typical application example is shown here (`zrna.org`):



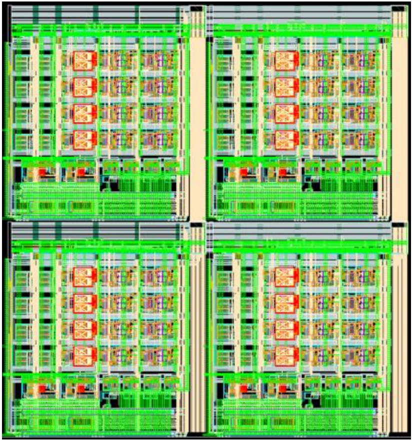
Glenn E. R. Cowan (2005)

First fully reconfigurable analog computer on chip – mainly used for stochastic differential equations:[3]



[3] See [Cowan(2005)].

More advanced analog computer on chip:[4]



(Picture source: http://yipenghuang.com/wp-content/uploads/2016/07/v2_chip.png, retrieved 2019-09-19.

[4]See [Guo et al.(2016)].

Problems

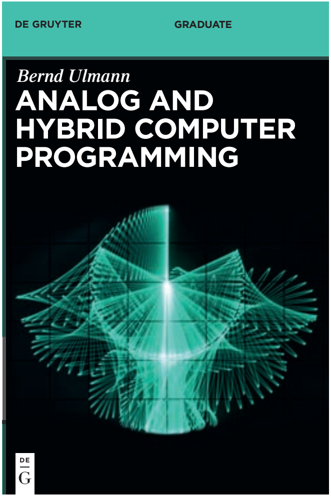
- These and other chips are often heavily specialized (signal pre-/postprocessing, artificial intelligence etc.).
- Most of these chips are academic projects with no intentions to turn them into production ready hardware.
- Software support is minimal to say the least...
- The computing elements often exhibit rather low bandwidth (about 20 kHz).
- Static and dynamic errors are pretty large.

anabrid GmbH (Germany) has been founded in 2020 with the goal of developing, producing and marketing a high-performance reconfigurable analog computer on chip.

Based on this chip, which will find applications in areas such as high-performance-computing etc., more specialised analog computer chips will be developed with main targets being artificial neural networks, signal per- and postprocessing etc.

More details...

More details about analog and hybrid computer programming can be found in the following recent publication:[5]



[5] See [Ulmann(2020)].

Bibliography

- Glenn Edward Russell Cowan, *A VLSI Analog Computer / Math Co-processor for a Digital Computer*, Columbia University, 2005
- N. Guo, Y. Huang, T. Mai, S. Patil, C. Cao, M. Seok, S. Sethumadhavan, and Y. Tsividis, “Energy-Efficient Hybrid Analog/Digital Approximate Computation in Continuous Time”, in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1514-1524, July 2016
- Thos. D. Truitt, A. E. Rogers, *Basics of Analog Computers*, John F. Rider Publisher, Inc., New York, December 1960
- Bernd Ulmann, *Analog and hybrid computer programming*, DeGruyter, 2020