A physical system that solves differential equations inherently, through its physical properties, without executing instructions as a traditional processor.

Analog computers represent variables of differential equations as physical quantities (voltage, current, position and etc) and use the natural laws of physics like Ohm’s law, Kirchhoff’s law and mechanical motions to model and solve differential equations.

**Examples of Physical Systems that Inherently Solve Differential Equations:**

**1. Electrical Analog Computers**

* Use **op-amps, resistors, capacitors**, and **inductors** to model equations.
* Can solve complex systems of differential equations in **real time**.
* Example: Solving a second-order differential equation using an RLC circuit.

**2. Mechanical Analog Systems**

* Systems like **spring-mass-damper models** inherently follow Newton’s laws.
* Used in early analog devices and are still fundamental in engineering design.
* Example: A mass on a spring models a harmonic oscillator (a second-order ODE).

**3. Optical Systems**

* Use **light interference, diffraction, and refraction** to solve wave equations.
* Certain lenses and filters perform mathematical transforms like Fourier transforms **inherently**.
* Used in some **image processing and signal analysis** applications.

**4. MEMS (Micro-Electro-Mechanical Systems)**

* Tiny physical systems that use mechanical motion and feedback to respond to stimuli.
* Some MEMS devices (like accelerometers and gyros) behave according to differential equations.

**5. Quantum Systems**

* Quantum systems evolve according to the **Schrödinger equation**, a partial differential equation.
* Quantum computers and simulators aim to exploit this for solving complex differential systems, such as those in quantum chemistry or materials science.

**Modern Technologies Using These Principles:**

* **Control systems** (e.g., cruise control, autopilot)
* **Signal processing hardware** (e.g., analog filters)
* **Analog neural networks** (e.g., neuromorphic computing)

**More on Neuromorphic computing**

Neuromorphic computing is a fascinating and very active area of research that embodies a physical system that solves problems (often modelled by differential equations) inherently through **its physical structure and dynamics**, rather than via traditional sequential processing.

Neuromorphic computing refers to hardware that is inspired by the structure and functioning of the human brain. It was coined by Carver Mead in the 1980’s. Instead of using conventional von Neumann architecture, these systems often integrate processing and memory in a way similar to how neurons and synapses are interwoven in the brain.

Advantages:

1. Inherent differential equation solving via neuron models.
2. Massive parallelism (thousands or millions of operations at once).
3. Energy efficient by using event-driven, analog computation.
4. Real-time adaptability for robotics, autonomous systems, and edge devices.