

Analog Computing for Molecular Dynamics

SVEN KÖPPEL¹, ALEXANDRA KRAUSE² AND BERND ULMANN^{1,3,4,*}

¹*anabrid GmbH, Am Stadtpark 3, 12345 Berlin, Germany*

²*FU Berlin, Arnimallee 14, 14195 Berlin, Germany*

³*Institute of Medical Systems Biology Ulm University, Germany*

⁴*FOM University of applied science*

Received: February 28, 2022. Accepted: March 3, 2022.

Modern analog computers are ideally suited to solving large systems of ordinary differential equations at high speed with low energy consumption and limited accuracy. In this article, we survey N -body physics, applied to a simple water model inspired by force fields which are popular in molecular dynamics. We demonstrate a setup which simulate a single water molecule in time. To the best of our knowledge such a simulation has never been done on analog computers before. Important implementation aspects of the model, such as scaling, data range and circuit design, are highlighted. We also analyze the performance and compare the solution with a numerical approach.

Keywords: Analog computing, computational physics, quantum simulation

1 INTRODUCTION

General purpose analog computers are highly efficient computers with respect to energy consumption and time to solution, particularly suitable for solving problems formulated as differential equations, albeit with limited accuracy. Analog computers were popular up to the 1980s and are experiencing a comeback as vital parts of hybrid computers, which consist of a digital and an analog computer. The analog computer in such a setup acts as a co-processor, speeding up simulations substantially. The digital computer can reconfigure and parameterize the analog computer. In this paper, we will show a simple model for molecular dynamics, demonstrating how

* Contact author: E-mail: ulmann@anabrid.com

analog computing can be used to address relevant yet challenging computational problems.

1.1 The need for beyond-von-Neumann computing

Where digital computers operate intrinsically serial by means of an algorithm, an analog computer operates intrinsically parallel by means of an equivalent circuit of the mathematical model. This is distinct from any attempts of algorithmic parallelization, which are fundamentally limited by non-parallelizable parts (Amdahl's law). Partitioning problems and engineering distributed algorithms for high performance simulations in scientific computing is an ever ongoing task for generations of scientists. Networking and memory access are bottlenecks which make it hard to find optimal algorithms which are only bound by processing speed (CPU-bound). The architectural flaws of digital computer design (such as the Von-Neumann bottleneck) are fundamentally based in the serial nature of algorithms.

Problems governed by partial differential equation systems are generally mapped onto high performance computer clusters by means of distributing the physical simulation space accross the available compute nodes, in a way that each node holds a small box or physical domain where it operates on. This approach is decades old and generally scales well. Challenges like unbalanced problem sizes or stiff equations covering different physical scales are encountered with sophisticated schemes for distributing work. This is part of traditional molecular dynamics codes such as GROMACS, CHARMM, Amber and many more [5, 10, 13, 14, 17, 18, 21].

Non-traditional computer architectures (frequently called, beyond Von-Neumann computing") try to overcome the fundamental limitations in order to increase the predictive capabilities and/or decrease the time to solution and power requirement of future computational systems. When it comes to analog computers as candidate systems, we have shown that analog and digital computers are *orthogonal* with respect to their solution time and power consumption, and a hybrid computer system can do a trade between these two [16].

2 AN *N*-BODY MODEL FOR WATER

Molecular dynamics is a popular approach in chemistry, biophysics and life-sciences for describing the dynamics of atoms and molecules. Being a classical *N*-body particle simulation with effective physical forces between the constituent atoms (or molecules, respectively), it has traditionally been solved numerically, using digital computers. The first such simulations date back to the 1950s (see for instance the reviews [11, 20]).

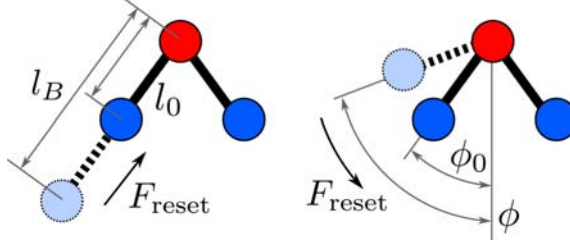


FIGURE 1

Intramolecular degrees of freedom in the water molecule modeled as harmonic displacements: Bond stretching (Eq. 2) and angle bending (Eq. 3) between an oxygen atom (red) and an associated hydrogen atom (blue).

Mathematically, N -body physics is described by a set of coupled ordinary differential equations (ODEs),

$$m_i \frac{d^2 \vec{r}_i}{dt^2} = -\vec{\nabla}_i V(\vec{r}_0, \dots, \vec{r}_N) = \vec{F}_i(\vec{r}_0, \dots, \vec{r}_N), \quad (1)$$

for N distinct particles with positions $\vec{r}_i = \vec{r}_i(t)$, masses m_i and conservative forces \vec{F}_i generated by the potential energy V with $\nabla_i^d = \partial/\partial r_i^d$ being the d th component of the gradient. The system (1) is an ODE of second order in time and first order in space.

2.1 Physical content: CHARMM force fields

The physical content encoded in the potential V is a macroscopic theory where no quantum effects are involved. Water models are among the most popular parts of theoretical biophysics, as water is a very important solvent and a rather complex system. Different water models for digital computers have been suggested, implementing different properties that have been experimentally studied in real water. These water models vary in their computational complexity and accuracy and need to be chosen for the specific purpose of a study. Typically, the potential is a sum of bounded (intramolecular) and unbounded (intermolecular) contributions. Here, the typical constituents for the CHARMM (short for *Chemistry at HARvard Macromolecular Mechanics*) force fields will be reviewed [?, 6].

First, for intramolecular contributions, one typically considers *bond stretching* with a quadratic potential (c. f. Figure 1)

$$V_b = \sum_{bonds} k_b/2 (l_b - l_0)^2 \quad (2)$$

with force constant k_b and bond length difference $l_b - l_0$ indicating deflection from equilibrium. *Angle bending* also models the bond angle deflection as a quadratic potential

$$V_\phi = \sum_{\text{angles}} k_\phi / 2 (\phi - \phi_0)^2. \quad (3)$$

Other contributions are obtained from *dihedral angles*, which are clockwise angles between two planes, where the planes are spanned between three respective atoms. The potential terms include quadratic contributions for proper and improper dihedrals. Since they only apply for molecules with $N \geq 4$ atoms, we won't consider them here.

Next, for intermolecular contributions, one typically considers Van-der-Waals forces, which are modeled by Lennard-Jones potentials

$$V_{vdW} = \sum_{i,j|i < j} 4\varepsilon_{i,j} \left[\left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right]. \quad (4)$$

The polynomial potential (4) for uncharged atoms or molecules combines attracting and repelling forces. Indices i and j count over the atoms in the sum, $r_{i,j}$ is defined as the modulus of the position vectors r_i and r_j for atoms i and j . $\varepsilon_{i,j}$ denotes the y -coordinate of the minimum in the potential and $\sigma_{i,j}$ describes the x -coordinate of the position where $V_{vdW} = 0$.

Both inter- and intramolecular electrostatic interactions of the charged nuclei are modeled in a straightforward fashion with the Coulomb potential

$$V_{el} = \sum_{i,j|i < j} \frac{q_i q_j}{4\pi \varepsilon_0 r_{i,j}}, \quad (5)$$

which describes the interaction between atoms i and j with charges q_i and q_j and their position vector difference $r_{i,j}$ as in the Lennard-Jones potential. ε_0 denotes the dielectric constant.

Note that the older and currently most widely used models are rigid models which do not allow for polarization of the water molecule. All internal degrees of freedom are fixed and there are no dummy charges which would allow for polarization. They differ in their parameters for the force field and are purposely fitted to model-specific physical properties of water such as the diffusion coefficient, viscosity and others. Prominent examples here are the TIP3P and SPC/E water models [15].

Newer generations of water models introduce dummy charges to allow for a change in polarization and dipole moment within the water molecule. All

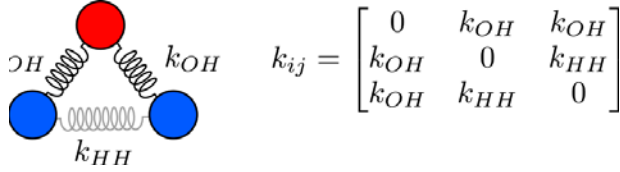


FIGURE 2

Harmonic intermolecular model for maintaining a rest position within the Coulomb potentials. The attracting k_{HH} counteracts the repelling $q_H q_H$ Coulomb potential, where the repelling k_{OH} counteracts the attracting $q_O q_H$ Coulomb potential. This is implemented in the equations of motions (6).

internal degrees of freedom are still held static but the introduction of the dummy charges allows for the negative charges to be not entirely focused on the oxygen atom but to be located on the oxygen atom and a dummy charge below it (TIP4P) or dummy charges simulating the valence electron pairs of the oxygen atom (TIP5P). These approaches are computationally less costly than the introduction of flexible bounds but are still not used extensively as their computational costs are much higher than for models without polarization.

2.2 A single molecule demonstrator model

We want to simulate a *single* H_2O molecule with internal dynamics described by Coulomb forces, bond stretching and angle bending. Due to the limited size of the analog computer available, the problem is reduced to $d = 2$ spatial dimensions instead of a full $d = 3$ model. Furthermore, the angle bending is modelled using a *small-angle approximation*, i. e. replaced by a spring between the two hydrogen atoms, which is valid for small angular excitations.

For a single molecule in two dimensions, we have $N = 3$ particles (H_1, H_2, O) and $D = 2$ dimensions (x, y). Generally, the indices are given by $i \in (0, 1, 2) \equiv (O, H_1, H_2)$ and $d \in (0, 1) \equiv (x, y)$. The equations of motions for the miniature force field are

$$\frac{d\vec{v}_i}{dt} = \sum_{j=0}^{N-1} \left(\left(1 - \frac{l_{ij}}{r_{ij}} \right) k_{ij} + \frac{q_i q_j}{r_{ij}^3} \right) \frac{\vec{r}_i - \vec{r}_j}{m_i}, \quad (6a)$$

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i. \quad (6b)$$

The spring rate matrix k_{ij} models that either bond stretching (k_{OH}) happens between an oxygen and an hydrogen atom or angle bending (k_{HH}) is taking place between the two hydrogen atoms, c. f. Figure 2. The rest positions l_{ij} follow a similar definition with bond length l_{OH} and the HH distance l_{HH} .

2.3 Physical units

We have chosen a suitable natural unit system where all constants are within a reasonable size between zero and one. We measure the electric charges in multiples of the elementary charge $e = 1$ and thus have $q_O = q_0 = 0.417$ and $q_H = q_{1,2} = -0.834$. The masses are measured in units of atomic mass units, $m_O = m_0 = 16$ and $m_H = m_1 = m_2 = 1$. The vacuum permittivity is set to $4\pi\epsilon_0 = 1$. Lengths are measured in Å, and therefore the distance at rest (bond length) between OH is given by $l_B = 0.9572 := l_{OH}$. The opening angle at the Oxygen atom is $104^\circ = 0.58\pi$, which determines the distance at rest between HH as $l_{HH} = 2l_{OH} \sin(0.58\pi/2)$.

2.4 Properties and limitations

The *characteristic frequencies* or *normal modes* of molecules [2] are often taken into account when reasoning about the validity of a force field or its numerical simulation. In $d = 3$ spatial dimensions, a single water molecule has 9 well-known modes, such as the symmetric, anti-symmetric and bending mode. Due to the non-orthogonal geometry of our spring model (6), one cannot expect these modes to appear when $k_{HH} \neq 0$. This is mainly because the small angle approximation is violated by the HH spring. On the other hand, with $k_{HH} = 0$, nothing prevents the *OH* binding from rotating around the oxygen atom. The angle bending approximation is a computationally cheap choice for avoiding these non-physical oscillations.

For solvent models, a number of statistically or thermodynamically derived quantities are typically considered in benchmarks [19], such as the temperature, potential energy and equilibrium stability of a large molecular ensemble. The self-diffusion coefficient and the radial distribution functions for water can be tested and are meaningful properties of a physical model (force field) and the applied computational methods which have been used. With a single molecule, there is no meaningful equivalent.

This water model is simplified, but it is not a toy model. For instance, TIP3P, the most currently used water model in molecular dynamics simulations, does not include charge transfer or induced dipole moment within the water molecule. All bond lengths are fixed as is the HOH angle at 104.5° . Therefore, the bond stretching and angle bending potential contributions vanish in TIP3P, and the intermolecular forces from the Lennard-Jones potential and the Coulomb potential determine the dynamics of the system. Our model is derived from TIP3P and its forecasting power is basically capped (limited) by the computational resources available (see next Section 3). Coincidentally, with the same amount of computing elements, a two molecule system could have been simulated, with only a single vectorial degree of freedom per molecule (basically representing molecules as spins). It is worth being noted that simple elastic mass-and-spring networks are also established in

molecular modelling, such as the Gaussian network model (GNM), see for instance [8, 12].

This section has shown that the extension of models is mathematically straightforward and agnostic to the number of particles simulated. The discussion on circuit scaling (Section 4) is done on the demonstrator model (6) for reasons of brevity.

3 ANALOG COMPUTER IMPLEMENTATION

The classical equations of motions (1) are coupled ordinary differential equations. In contrast to classical fluid dynamics, where the time evolution of some field $F(\vec{x}, t)$ is determined by a partial differential equation, ordinary differential equations of distinct particle trajectories $\vec{r}(t)$ can be directly mapped to electrical circuits. The traditional direct approach represents all vectorial components i of the involved particles j by one connection between computing elements per degree of freedom $\mathbf{r}_j^i(t)$. Following this approach, all intermediate computing results are also represented this way, such as velocities $\mathbf{v}_j^i(t)$ and any algebraic (sub-)expression encountered in the particular analytically defined equation of motion. Note that time is kept continuous while the discretization follows the concepts of discrete and distinguishable particles in classical physics.

3.1 Naming all constitutional parts of the equation

In order to implement the equation of motions on the analog computer, one has to decompose the equation into the primitive analog computing elements. In general, upper indices will indicate the vector index \cdot^d , while lower indices \cdot_i indicate the particle index. Bold face is used to separate the name of a variable (connection) whereas indices are given in regular form, which is particular relevant for the multiplicative inverses, such as the distance matrix \mathbf{r}_{ij}^{-1} . It is computed as following:

$$\mathbf{r}_{ij}^d = r_i^d - r_j^d, \quad \mathbf{s}_{ij}^d = (\mathbf{r}_{ij}^d)^2 \quad \text{and} \quad (7a)$$

$$\mathbf{t}_{ij} = \sum_k^D \mathbf{s}_{ij}^k, \quad \text{so that } \mathbf{r}_{ij}^{-1} = 1/\sqrt{\mathbf{t}_{ij}}. \quad (7b)$$

Note that the coordinate difference components $\mathbf{r}_{ij}^d = -\mathbf{r}_{ji}^d$ are antisymmetric. The distance matrix \mathbf{r}_{ij} and its multiplicative inverse \mathbf{r}_{ij}^{-1} are symmetric ($\mathbf{r}_{ij} = \mathbf{r}_{ji}$) and traceless ($\mathbf{r}_{ii} = 0$). Thus one only has to compute $\mathbf{r}_{ij} \in \mathbb{R}^{M \times D}$ many elements, with $M = (N^2 - N)/2 = 3$. The actual force components

$$\mathbf{F}_{ij}^d = \left(\left[l_{ij} \mathbf{r}_{ij}^{-1} - 1 \right] k_{ij} + \frac{q_i q_j}{\varepsilon_0} (\mathbf{r}_{ij}^{-1})^3 \right) \mathbf{r}_{ij}^d \quad (7c)$$

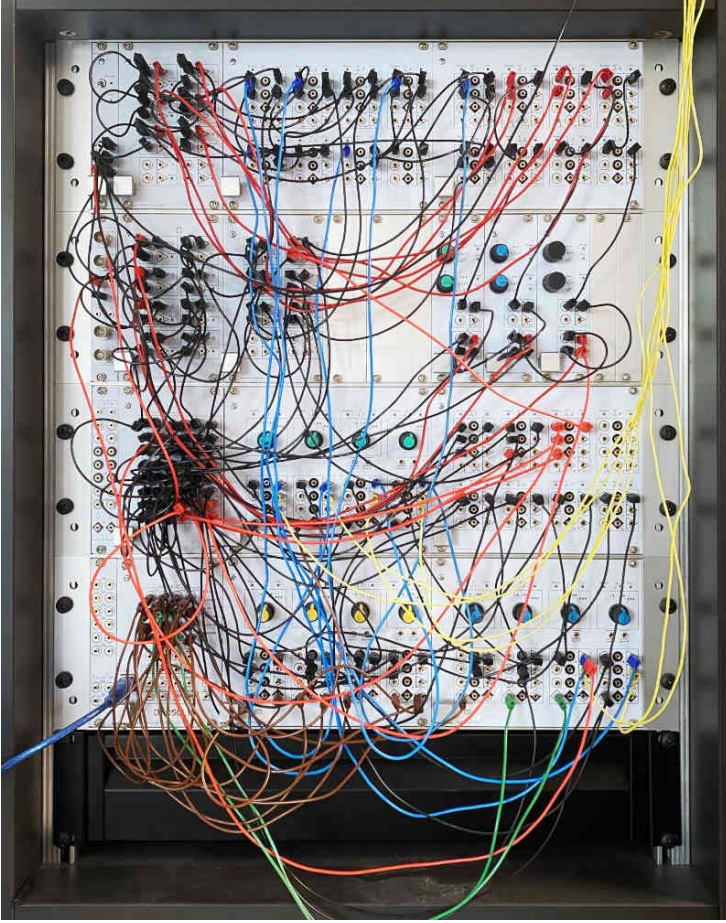


FIGURE 3

Photography of the front panel of a large analog computer of type *Analog Paradigm Model-1*, implementing equations (7) in a scaled manner (c.f. section 3). See main text for a rough navigation in computing elements.

are antisymmetric and traceless ($\mathbf{F}_{ij}^d = -\mathbf{F}_{ji}^d$ and $\mathbf{F}_{ii}^d = 0$ for all i, j, d). Also note that the bond stretching is only applied between H and O atoms of a single molecule, which is encoded in Δ_{ij} . The velocity and position components can be determined by analog time integration

$$\mathbf{v}_i^d = - \int_t dt \sum_j^N \left(-\frac{\mathbf{F}_{ij}^d}{m_i} \right), \quad \mathbf{r}_i^d = - \int_t dt (-\mathbf{v}_i^d), \quad (7d)$$

where the notation is oriented on the negating nature of the analog computing elements.*

3.2 Machine realization

The set of equations (7) is directly mapped to the general purpose analog computing elements of the *Analog Paradigm Model-1* computer we used in this simulation [24]. The overall circuit is given in Section 5 and is a 1:1 translation of the mathematical expressions (7). We would like to emphasize that on this level of abstraction, the abstract syntax tree of the mathematical expressions is *identical* to the equivalent circuit. The physical realization of this particular analog computer program is shown in figure 3. The machine is equipped with a *Hybrid Controller* (it can be identified by its USB cable in the lower left of figure 3). The system can be fully configured from an attached digital computer to set the model's constants and parameterers; there are more then 30 digital potentiometers involved, each with a resolution of 10 bit.

3.3 Properties

The mapping of the mathematical equations (7) onto the analog computing elements (as shown schematically in Section 5) by no means assumes *ideal* electronics. In fact, the limits of the system in the particular application are studied and discussed in Section 6.

4 CIRCUIT SCALING

The question of value ranges and domain sizes is crucial in any (analog or digital) computer simulation of a physical problem. This topic is typically not given much attention in presentations of novel numerical codes. When implementing analog circuits, ensuring the correct scaling of the solution is an inherent part of the problem mapping. Various aspects of this scaling process are discussed here.

4.1 Analog number representation

For number representation on computers, it is essential that quantities do not exceed the valid range of values. For instance, a signed 32 bit integer can store values between 0 and $2^{31} \approx 10^{10}$. In contrast, the IEEE 754 floating point number representation allows to represent numbers up to $2^{127} \approx 10^{38}$, which comes with a loss of precision. When doing numerical computations, any number outside the valid data range results in an overflow and an invalid

* Analog computer modules such as summers and integrators perform an implicit change of sign which is due to the actual technical implementation of these computing elements.

result. However, on a digital computer, there is always the possibility to use longer bit sequences to represent numbers. For instance, one could use a double-precision floating point number representation with a width of 64 bit, allowing numbers up to $2^{1023} \approx 10^{308}$. The same applies for numbers which are bounded by nature, for instance percentages or saturations. If a number $p \in [0, 1]$ is to be represented on a digital computer, then the 32 bit signed integer will naturally represent p with a uniform accuracy of $\Delta p \approx 10^{-38}$. In contrast, 32 bit single-precision floating point has a precision $\Delta p \approx 10^{-20}$ for values down to 10^{-12} and up to $\Delta p \approx 10^{-8}$ for values close to 1.

The success and widespread use of floating point numbers makes number representation a topic, which numerical scientists only have to deal in special cases, for instance when it comes to instabilities, catastrophic cancellation or the like. In day-to-day work, the occurrence of an out-of-bound value (represented as *NaN*, not a number, in the IEEE754 standard) is typically a sign of a programming error, such as computing $1/0$ resulting from an uninitialized or in some way “illegal” denominator.

In electronic analog computers, numbers are typically represented as voltages or currents. Given the fixed physical structure of such a computer, these quantities are bounded in principle. On an information theoretical level, such computing elements are called “bounded in, bounded out” (BIBO) [26]. The illegality of non-bounded values is similar as to the behaviour of number representations in digital computing. For the time being, let’s assume the valid range for a value in an analog computer to be $[-1, +1]$.

Analog computers have an inherent limited precision of about three to four decimal places [23]. In the above convention, this yields a precision $\Delta p \approx 10^{-4}$ which is roughly the resolution of a half-precision floating point number or 10 to 16 bits of information.

In order to answer the question of whether simulating water molecules on an analog computer can compare to current digital simulations, given the limitations in resolution of the analog computer, we assume the Coulomb potential terms to be the limiting factor. Keep in mind that in the present study we can neglect the Lennard-Jones potential as we are only modelling a single molecule.

4.2 Coordinate units

To improve readability, we assume a physical square simulation domain of size $\Omega := [-L, +L]^D$ with some dimensionful length scale L in D spatial dimensions with any physical coordinate $\mathbf{r}_i^d \in \Omega$. Scaled and dimensionless units can be easily recovered as $\hat{\mathbf{r}} := \mathbf{r}/L \in \hat{\Omega}$ contained in a unit domain $\hat{\Omega} = [-1, +1]^D$. In practice that means that initial values have to be scaled according to $\hat{r}_i^d(0) = r_i^d(0)/L$ and $\hat{v}_i^d(0) = v_i^d(0)/L$. Also the distances at rest

l_{OH} , l_{HH} and the de facto Coulomb coupling constant ε_0 have to be chosen in physical units before transformation.

Assuming that each mapping in (7a) has a bounded input on the right hand side of the definition, the following scaling has to be applied to ensure bounded output:

$$\hat{\mathbf{r}}_{ij}^d := \frac{\hat{r}_i^d - \hat{r}_j^d}{\alpha}, \quad \hat{\mathbf{s}}_{ij}^d := (\hat{\mathbf{r}}_{ij}^d)^2, \quad (8)$$

$$\hat{\mathbf{t}}_{ij} := \frac{\sum_k^D \hat{\mathbf{s}}_{ij}^k}{\delta^2}, \quad \hat{\mathbf{r}}_{ij}^{-1} := 1 / \left(\beta \sqrt{\hat{\mathbf{t}}_{ij}} \right). \quad (9)$$

Three factors were introduced: Two geometric ones (α and δ) resulting from the dimensionless coordinate representation and an internal one (β) to deal with the multiplicative inverse (reciprocal). These scaling factors sum up to

$$\hat{\mathbf{r}}_{ij}^{-1} := \gamma \mathbf{r}_{ij}^{-1} \quad (10)$$

with overall scaling factor $\gamma := \alpha\delta/\beta$. Note that applying this scaling means that $\hat{\mathbf{r}}_{ij}^{-1} \neq 1/|\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j|$, i. e. a clear relationship between the “hatted” quantities $\hat{\mathbf{r}} \dots$ no longer exists. The necessity and implications of the individual factors are discussed below.

4.2.1 Geometric scaling

The factor α has two contributions determined by the maximum possible distance of two particles in both D dimensions ($d_{\max} = \sqrt{D}$) and one dimension, where $d_{\max} = 2$ and $1 \leq \alpha \leq 2$ is a sensible choice. The factor δ results from the sum of D scaled quantities. A useful choice might be $\delta = \sqrt{D}$. Both factors α and δ are *optional* as they might be applied but their underlying scaling problems can also be neglected in practice if no overflow problems occur. In order to simplify the reasoning, the optional factors are neglected for the rest of the text, so $\alpha = \delta = 1$ and thus $\gamma = 1/\beta$.

4.2.2 Inverse-square law scaling

The factor β is due to the necessity of computing multiplicative inverses on BIBO unit domains. It is crucial and must not be omitted. Displaying an inverse-square law such as the Coulomb potential $V_{ij}(d) \sim 1/r_{ij}$ or the electrostatic force $F_{ij}(d) \sim 1/r_{ij}^2$ with $r_{ij} = |\vec{r}_i - \vec{r}_j|$ being the distance between two particles i and j on an analog computers is challenging, since it is immediately $1/r_{ij} > 1$ for $r_{ij} < 1$. The approach in (8) is to apply a scaling factor $1/\beta \ll 1$ on \mathbf{r}^{-1} . This allows the distance component \mathbf{t}_{ij} to become as small

as β^{-2} (because for smaller $\mathbf{r}_{ij} < \beta^{-2}$, again $\mathbf{r}^{-1} > 1$). Naturally, this introduces an *ultraviolet cutoff* at β^{-1} . If two particles come closer than β^{-1} , their distances can no longer be represented on the computer. Cutoffs are a well-understood phenomenon in theoretical physics and also occur in numerical simulations.

4.3 In-circuit scaling

The scaling coefficients α, β, δ are not relevant outside the computation. Instead, they only affect the potentiometer values. By convention, the scaling factors (lowercase greek letters as α) are typically outside the BIBO range $[-1, +1]$, while the potentiometer values (uppercase Fraktur letters as \mathfrak{A}) are within.

4.3.1 Force scaling

The system force (7c) is scaled as

$$\hat{\mathbf{F}}_{ij}^d := \zeta \frac{\mathbf{F}_{ij}^d}{\beta^3} \quad (11)$$

The factor β^3 results from the linear dependency on \mathbf{r}^{-3} , which implies carrying on the scaling. The factor $\zeta > 1$ is a non-nphysical scaling factor introduced in order to allow force amplification in the subsequent steps.

By replacing unscaled quantities in (7c) by their scaled counterparts of the previous section, the potentiometer values are determined as:

$$\hat{\mathbf{F}}_{ij}^d := \left[\left(\frac{l_{ij}}{\beta^2} \frac{\hat{\mathbf{r}}_{ij}^{-1}}{\beta} - \frac{1}{\beta^3} \right) k_{ij} + \frac{q_i q_j}{\varepsilon_0} \frac{\hat{\mathbf{r}}_{ij}^{-3}}{\beta^3} \right] \zeta \hat{\mathbf{r}}_{ij}^d \quad (12)$$

With the intention to define parameters \mathfrak{P} , which directly map on a single potentiometer with range $0 \leq \mathfrak{P} \leq 1$ on the *Model-I*, we introduce the quantities

$$\mathfrak{R}_{ij} = \zeta \frac{l_{ij} k_{ij}}{\beta^2}, \quad \mathfrak{U}_{ij} = \zeta \frac{k_{ij}}{\beta^3}, \quad \mathfrak{C}_{ij} = \zeta \frac{|q_i q_j|}{\varepsilon_0}. \quad (13)$$

The scaled force is then

$$\hat{\mathbf{F}}_{ij}^d = \left(\mathfrak{R}_{ij} \frac{\hat{\mathbf{r}}_{ij}^{-1}}{\beta} - \mathfrak{U}_{ij} \pm \mathfrak{C}_{ij} \frac{\hat{\mathbf{r}}_{ij}^{-3}}{\beta^3} \right) \hat{\mathbf{r}}_{ij}^d. \quad (14)$$

One has to take care of the sign in front of \mathfrak{C}_{ij} , which is -1 for OH interactions and $+1$ for HH interactions. The required negation is indicated by

\pm in (14). When following the scaling argument δ of the previous section, (14) would also require a factor $1/3$ to compensate the sum of three scaled quantities. However, due to the different orders of magnitude of the coupling constants, this scaling choice was not considered here.

4.3.2 Time integration

The standard analog time integration [25] reads

$$e_0 = - \left(\int_0^t k_0 dt' \sum_{r=1}^n a_r e_r(t') \right) - e(0) \quad (15)$$

with initial condition $e(0)$, inputs e_r , output e_0 , input weights a_r , time scale factor k_0 and machine time $t' = t/k_0$. On the Model-1 analog computer, possible input weights are $a_i \in \{1, 10\}$ and $k_0 \in \{10, 100, 1000\}$. These have to be chosen at programming time. For the velocity and coordinate integration, we can compute physical quantities $\hat{\mathbf{v}}_i^d \equiv \mathbf{v}_i^d$ and $\hat{\mathbf{r}}_i^d \equiv \mathbf{r}_i^d$ by carefully reversing all scaling operations. We do so by casting (7d) as

$$-\hat{\mathbf{v}}_i^d = - \int_0^t k_0 dt \sum_j^N \left(a_i \frac{1}{m_i} \mathbf{F}_{ij}^d \right) \quad (16)$$

$$= - \int_0^t k_0 dt \sum_j^N \left(a_i \frac{1}{m_i} \frac{\beta^3}{\zeta} \hat{\mathbf{F}}_{ij}^d \right) \quad (17)$$

$$= - \int_0^t k_0 dt \sum_j^N \left(\mathfrak{F}_i \hat{\mathbf{F}}_{ij}^d \right). \quad (18)$$

We introduced the potentiometer value

$$\mathfrak{F}_i = \frac{1}{m_i} \frac{\beta^3}{\zeta} \frac{1}{a_i}. \quad (19)$$

Thanks to moving the integrator weight a_i into \mathfrak{F}_i , it can be ensured that $\mathfrak{F}_i \leq 1$ for many weight choices. If this is not sufficient, one can still make use of the timescale factors k_0 for an additional degree of freedom in scaling. In our setup all integrators have the same k_0 which is thereby solely responsible for the speed of integration (or simulation vs. realclock time ratio). On future chip-level generations, $k_0 \approx 10^7$ is expected [16]. The velocity integration is

then trivially given by

$$\hat{\mathbf{r}}_i^d = - \int_0^t k_0 dt (-\hat{\mathbf{v}}_i^d). \quad (20)$$

4.3.3 Numerical potentiometer values

With $\beta = 5$ and the physical units given in section 2.3, we obtain $\Re_{HH} = 0.036\zeta$, $\Re_{OH} = 0.04\zeta$, $\mathfrak{C}_{ij} = (\pm)0.0139\zeta$ and $\mathfrak{U}_{ij} = 0.008\zeta$. These small numbers suggest to upscale the forces with $\zeta = 30$. This results in $\mathfrak{F}_0 = \beta^3/\zeta/m_O = 125/16/30 = 0.2594$ with $a_0 = 1$ and $\mathfrak{F}_1 = \mathfrak{F}_2 = \beta^3/m_H/\zeta/10 = 125/30/10 = 0.4166$ with $a_1 = a_2 = 10$.

4.4 Conclusions on the dynamical range

In the last section, we came up with a value scaling, which introduces a cutoff at a length scale β^{-1} . With typical choices $\beta \sim 10$, this introduces basically a minimal length the order of $L_{\min} = 1/\beta \sim 0.1$. Equally, the maximum length in the simulation domain is determined by the BIBO domain size, i. e. $L_{\max} = 1$. Note that the minimal length is *not* a rounding/resolution error (as a Δx would be in a numerical approach). The computational exactness (rounding error) of an analog computer is of the order of $10^{-5} \ll L_{\min}$.

L_{\min} provides a minimal distance between two atoms, so it is the limiting factor in terms of precision. Chemical bonds like the O-H distance are of the order of magnitude of 1 Å while the valid range of the entire water box runs from -1 analog units to $+1$ analog units. Identifying L_{\min} analog units with 1 Å in the water box a maximum side length for a water box of 20 Å or 2 nm is possible. Most water boxes are in the nanometer range, so this value representation might be applicable to tackle real world simulations. Nevertheless we acknowledge that it might be too small to contain large macromolecules and different value representations or digital assisted approaches may be necessary in future work.

5 COMPLETE CIRCUIT

The overall circuit of the setup shown in Figure 3 is described below. All indices and sums are written out and are fully scaled. Note that we moved some constants between the definitions (for instance between F and C).

A triangular shape denotes a summer (with implicit change of sign), while a polygon with a large Π represents a multiplier, divider or square root component. Circles represent coefficient potentiometers and rectangles with attached triangles are integrators.

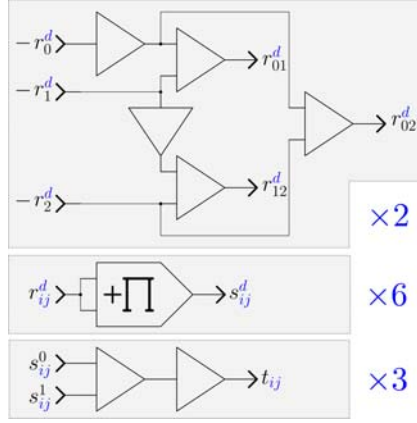


FIGURE 4

Computing r_{jk}^i , s_{jk}^i , and t_{jk} , as in equation (21).

5.1 Computing distance components r_{jk}^i , s_{jk}^i and t_{jk}

$$\begin{aligned}
 r_{01}^0 &= -(-r_0^0) + (-r_1^0) & s_{01}^0 &= (r_{01}^0)^2 \\
 r_{02}^0 &= -(-r_0^0) + (-r_2^0) & s_{02}^0 &= (r_{02}^0)^2 \\
 r_{12}^0 &= -(-r_1^0) + (-r_2^0) & s_{12}^0 &= (r_{12}^0)^2 \\
 r_{01}^1 &= -(-r_0^1) + (-r_1^1) & s_{01}^1 &= (r_{01}^1)^2 \\
 r_{02}^1 &= -(-r_0^1) + (-r_2^1) & s_{02}^1 &= (r_{02}^1)^2 \\
 r_{12}^1 &= -(-r_1^1) + (-r_2^1) & s_{12}^1 &= (r_{12}^1)^2
 \end{aligned} \tag{21}$$

$$t_{01} = s_{01}^0 + s_{01}^1 \quad t_{02} = s_{02}^0 + s_{02}^1 \quad t_{12} = s_{12}^0 + s_{12}^1$$

5.2 Computing inverse distances r^{-1} and their cubes

The scaling with β is applied in the circuit (Figure 5) *before* applying the division. A single potentiometer for β^{-1} is sufficient as an input for all three divisions.

$$\begin{aligned}
 \frac{r_{01}^{-1}}{\beta} &= \beta^{-1} / \sqrt{t_{01}} & \frac{r_{01}^{-3}}{\beta^3} &= (r_{01}^{-1})^3 / \beta^3 \\
 \frac{r_{02}^{-1}}{\beta} &= \beta^{-1} / \sqrt{t_{02}} & \frac{r_{02}^{-3}}{\beta^3} &= (r_{02}^{-1})^3 / \beta^3 \\
 \frac{r_{12}^{-1}}{\beta} &= \beta^{-1} / \sqrt{t_{12}} & \frac{r_{12}^{-3}}{\beta^3} &= (r_{12}^{-1})^3 / \beta^3
 \end{aligned} \tag{22}$$

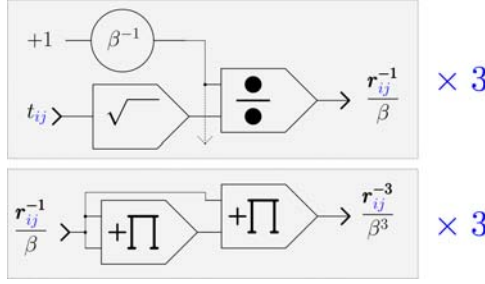


FIGURE 5

Computing $\frac{r_{jk}^{-1}}{\beta}$ and $\frac{r_{jk}^{-3}}{\beta^3}$, according to equation (22).

5.3 Computing the forces

$$\begin{aligned}
 \frac{\mathbf{F}_{01}^0}{\beta^3} &= \left(\Re_{01} \frac{\mathbf{r}_{01}^{-1}}{\beta} - \mathfrak{U}_{01} - \mathfrak{C}_{01} \frac{\mathbf{r}_{01}^{-3}}{\beta^3} \right) r_{01}^0 \\
 \frac{\mathbf{F}_{02}^0}{\beta^3} &= \left(\Re_{02} \frac{\mathbf{r}_{02}^{-1}}{\beta} - \mathfrak{U}_{02} - \mathfrak{C}_{02} \frac{\mathbf{r}_{02}^{-3}}{\beta^3} \right) r_{02}^0 \\
 \frac{\mathbf{F}_{12}^0}{\beta^3} &= \left(\Re_{12} \frac{\mathbf{r}_{01}^{-1}}{\beta} - \mathfrak{U}_{12} + \mathfrak{C}_{12} \frac{\mathbf{r}_{12}^{-3}}{\beta^3} \right) r_{12}^0 \\
 \frac{\mathbf{F}_{01}^1}{\beta^3} &= \left(\Re_{01} \frac{\mathbf{r}_{01}^{-1}}{\beta} - \mathfrak{U}_{01} - \mathfrak{C}_{01} \frac{\mathbf{r}_{01}^{-3}}{\beta^3} \right) r_{01}^1 \\
 \frac{\mathbf{F}_{02}^1}{\beta^3} &= \left(\Re_{02} \frac{\mathbf{r}_{02}^{-1}}{\beta} - \mathfrak{U}_{02} - \mathfrak{C}_{02} \frac{\mathbf{r}_{02}^{-3}}{\beta^3} \right) r_{02}^1 \\
 \frac{\mathbf{F}_{12}^1}{\beta^3} &= \left(\Re_{12} \frac{\mathbf{r}_{02}^{-1}}{\beta} - \mathfrak{U}_{12} + \mathfrak{C}_{12} \frac{\mathbf{r}_{12}^{-3}}{\beta^3} \right) r_{12}^1
 \end{aligned} \tag{23}$$

5.4 Computing velocities and positions

Note how the signs in front of \mathfrak{F}_i result from symmetries carried out (c. f. section 3.1).

$$\begin{aligned}
 -\mathbf{v}_0^0 &= -\int_0^t k_0 \, dt \left(+\mathfrak{F}_0 \cdot \beta^3 \frac{F_{01}^0}{\beta^3} + \mathfrak{F}_0 \cdot \beta^3 \frac{F_{02}^0}{\beta^3} \right) \\
 -\mathbf{v}_1^0 &= -\int_0^t k_0 \, dt \left(-\mathfrak{F}_1 \cdot \beta^3 \frac{F_{01}^0}{\beta^3} + \mathfrak{F}_1 \cdot \beta^3 \frac{F_{12}^0}{\beta^3} \right) \\
 -\mathbf{v}_2^0 &= -\int_0^t k_0 \, dt \left(-\mathfrak{F}_1 \cdot \beta^3 \frac{F_{02}^0}{\beta^3} - \mathfrak{F}_1 \cdot \beta^3 \frac{F_{12}^0}{\beta^3} \right) \\
 \Leftrightarrow +\mathbf{v}_2^0 &= +\int_0^t k_0 \, dt \left(+\mathfrak{F}_1 \cdot \beta^3 \frac{F_{02}^0}{\beta^3} + \mathfrak{F}_1 \cdot \beta^3 \frac{F_{12}^0}{\beta^3} \right)
 \end{aligned} \tag{24}$$

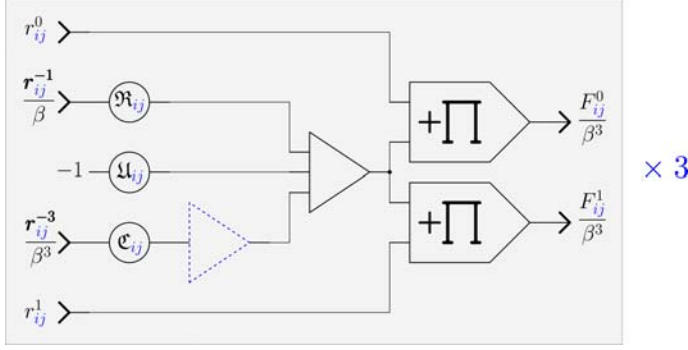


FIGURE 6

Computing $\frac{F_{ij}^k}{\beta^3}$ as in equation (23). Note that the negator indicated in dotted lines is only placed when computing F_{01}^d and F_{02}^d but not F_{12}^d .

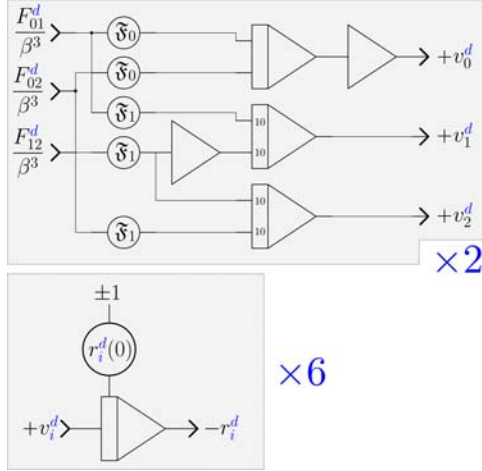


FIGURE 7

Computing v_j^i and r_i^d , as in equation (26).

$$\begin{aligned}
 -v_0^1 &= -\int_0^t k_0 dt \left(+\mathfrak{F}_0 \cdot \beta^3 \frac{F_{01}^1}{\beta^3} + \mathfrak{F}_0 \cdot \beta^3 \frac{F_{02}^1}{\beta^3} \right) \\
 -v_1^1 &= -\int_0^t k_0 dt \left(-\mathfrak{F}_1 \cdot \beta^3 \frac{F_{01}^1}{\beta^3} + \mathfrak{F}_1 \cdot \beta^3 \frac{F_{12}^1}{\beta^3} \right) \\
 -v_2^1 &= -\int_0^t k_0 dt \left(-\mathfrak{F}_1 \cdot \beta^3 \frac{F_{02}^1}{\beta^3} - \mathfrak{F}_1 \cdot \beta^3 \frac{F_{12}^1}{\beta^3} \right) \\
 \Leftrightarrow +v_2^1 &= +\int_0^t k_0 dt \left(+\mathfrak{F}_1 \cdot \beta^3 \frac{F_{02}^1}{\beta^3} + \mathfrak{F}_1 \cdot \beta^3 \frac{F_{12}^1}{\beta^3} \right)
 \end{aligned} \tag{25}$$

In our circuit, the initial data for the velocities are always zero, while initial data for the positions are fed in as initial conditions for the integrators (see Figure 7).

$$\begin{aligned}
 \mathbf{r}_0^0 &= - \int_0^t (-v_0^0) k_0 dt - r_0^0(0), \\
 \mathbf{r}_1^0 &= - \int_0^t (-v_1^0) k_0 dt - r_0^1(0), \\
 \mathbf{r}_2^0 &= - \int_0^t (-v_2^0) k_0 dt - r_0^2(0), \\
 \mathbf{r}_0^1 &= - \int_0^t (-v_0^1) k_0 - r_1^0(0) dt, \\
 \mathbf{r}_1^1 &= - \int_0^t (-v_1^1) k_0 - r_1^1(0) dt, \\
 \mathbf{r}_2^1 &= - \int_0^t (-v_1^1) k_0 - r_1^2(0) dt.
 \end{aligned} \tag{26}$$

6 BENCHMARKING THE ANALOG IMPLEMENTATION

The model produces reasonable results for small perturbations, as shown in Figure 8 for a simulation time of $t_{\text{final}} = 60\text{ms}$.

6.1 Energy and time demand

The bending patterns have roughly a wall clock time frequency of $f \sim 100\text{Hz}$, while the water bending normal modes have a frequency of $\nu \sim 32\text{GHz}$ in nature. This gives an effective ratio of about $k \sim 10^{-9}$ for the analog implementation, or a simulation time per real clock time ratio of $100/32 \sim 3\text{ns/s}$ or $260\mu\text{s/day}$. This number has to be compared with typical simulations on digital computers, which achieve a few hundred ns per day (see for instance a review [9] and references therein, such as [22]).

It is a crucial property of analog computing that this speed factor is invariant of the size of the equation (and circuitry, respectively)! That it, it is virtually independent of both the number of particles simulated as well as the physics included (c. f. Section 2.1). In contrast, the power (and thus energy) requirements scale linearly with the size of the circuit [16]. The *Analog Paradigm Model-1* power requirement for the present circuit is measured as $P_A = 21\text{W}$. The overall energy requirement for a single bending wave is therefore $E_A = P_A T \sim 210\text{mJ}$.

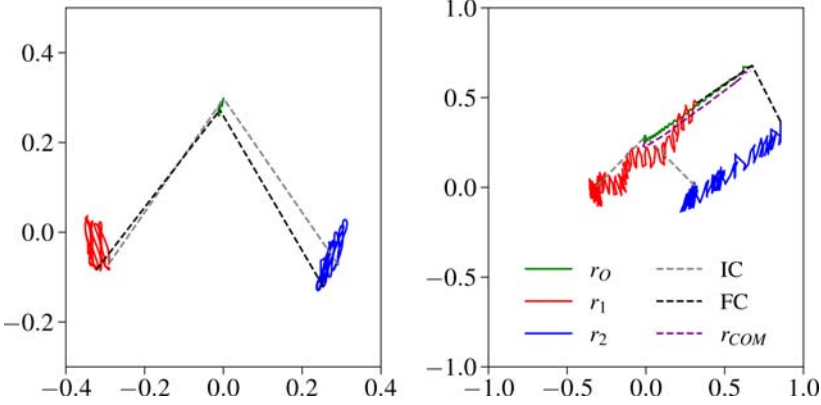


FIGURE 8

Analog water molecule simulation results. The left panel shows a closeup of a small excitation around the resting positions. The red and blue trajectories represent the two hydrogen atoms. The oxygen atom (green trajectory) is almost at rest. The OH binding is indicated by the dashed lines. The grey dashed lines show the initial condition (IC), while the dark dashed lines indicate the final condition (FC). The right panel shows a long time run which clearly shows the center of mass (r_{COM} , purple line) drifting along with the oxygen r_O .

6.2 Upscaling: Towards real-life molecular dynamics

The overall number of computing elements required for this simulation can be obtained from the system equation (7) or the circuit schematics (Appendix 5). It scales with spatial dimensions D , number of atoms N and reduced number of degrees of freedom $M = (N^2 - N)/2$ as

$$\begin{aligned}
 (2D + 1)M + 2 &= 5 \cdot 3 + 2 = 17 && \times \text{SUM } (\Sigma) \\
 (2D + 2)M &= 6 \cdot 3 = 18 && \times \text{MUL } (\Pi) \\
 M &= 3 && \times \text{SQRT } (\sqrt{\cdot}) \\
 2ND &= 2 \cdot 3 \cdot 2 = 12 && \times \text{INT } (f).
 \end{aligned}$$

Ignoring the different types of computing elements, one comes up with $\mathcal{O}(8N^2)$ computing elements in $D = 3$ dimensions. For instance, a typical water box in $D = 3$ with a size of $(30\text{\AA})^3$ and $N \sim 10^4$ atoms requires 7×10^6 computing elements.

One can estimate the performance of such a simulation carried out on a future analog computer on a chip. To do so, the reasoning of [16] is employed. Moving from an *Analog Paradigm* Model-1 analog computer to a proposed analog computer on chip will decrease energy consumption by at least a factor of 40 with an increase in speed by a factor of at least 150. This results in a power demand for a fully integrated analog computer simulating $N \sim 10^4$

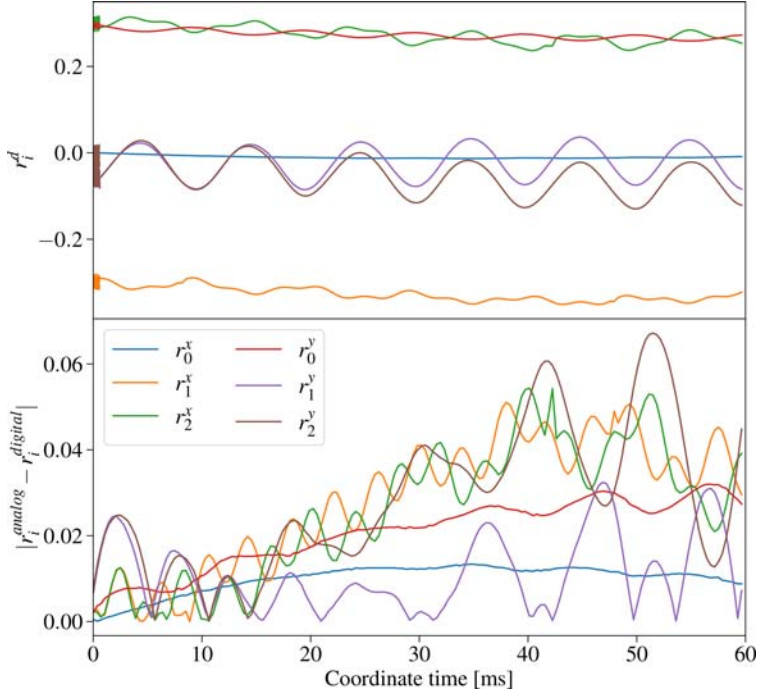


FIGURE 9

Upper panel: Analog (solid) vs. digital (dashed) time evolution. Lower panel: Absolute error (L1 norm) of digital vs. analog results for all particle coordinates r_i^d .

atoms of $P = 10^4/3 \times 20\text{W}/40 \approx 1600\text{W}$ with a simulation per real clock time ratio of more than 10^7 ns/day!

While this number of computing elements is definitely in the ballpark of future analog computing chips, the number of interconnections is certainly the limiting factor. The quadratic dependency on N is in fact dominated by the all-to-all force coupling F_{ij} in Newtonian mechanics. Typically, a long distance cutoff is introduced in N -body simulations involving inverse square law forces (such as molecular dynamics or galaxy simulations in cosmology). A similar approach has to be chosen in an analog implementation. For instance, the authors of [19] do a spherical cutoff at $L = 12\text{\AA}$. Concepts for implementing cutoffs on an analog computer (where the wiring is „hard coded“ at programming time) are left as an open question for the future.

6.3 Comparison with numerical time evolution

A classical algorithmic, i. e. numerical, integration scheme for (7) is to evaluate the time integrals in an iterative manner, for instance with the well-known

primitive Euler scheme:

$$y(t) = \int f(y(t), t) dt \rightsquigarrow y_{k+1} = y_k + f(y_k)\Delta t. \quad (27)$$

Here, k is an index denoting the values at a discretized time $t = k\Delta t$. The time discretization Δt can be basically chosen arbitrarily small, which is unique to digital computers.

In practice, a higher order (still explicit) Runge-Kutta scheme may be adopted to integrate both the velocities \mathbf{v}_i^d and subsequently the positions \mathbf{r}_i^d in time. Such an attempt has an intrinsic inexactness since the underlying ODE (1) is of second order in time. Spurious damping or excitation of the solutions (in general an exponential error) will be introduced at any convergence order [7]. Since classical 3-body problems are known to exhibit chaotic behaviour, long-term evolutions are inevitably unstable and unsuitable for long time evolutions. This includes phenomena such as drifting of the center of mass.

In practice, we observe similar stability problems on our analog computer as with traditional explicit numerical integration schemes. For instance, for longer simulation times (such as the right panel in Figure 8, which shows $t_{\text{final}} = 300\text{ms}$), we see a drift of the center of mass. This systematic error is reflected in the error plot in Figure 9, which shows a comparison with a high resolution numerical time evolution ($\Delta t = 0.005$ with a first order Euler scheme).

For this long standing problem in computational physics, methods have been developed to improve accuracy and maintain the validity of the physical solution. A remarkable scheme for Hamiltonian systems such as (1) are *geometric* or *symplectic* integrators which, among other things, preserve the phase space volume of the solution and the geometric structure in space-time [1]. The popular symplectic splitting method is basically still explicit. This approach has not been used in this work due to limitations in size of the available analog computers. However, in principle approaches like this can be implemented on future analog computing circuits in a straightforward manner.

7 CONCLUSION

We have demonstrated the first analog computer implementation of a simple water model aimed towards solvent theory and molecular dynamics. In order to implement an actual analog computer circuit, a simple model was chosen, which implements electrostatic interactions and which models

molecular stretching and bonding in a quadratic potential in Cartesian coordinates. Since the computer used in this study has just enough computing elements to implement a single water molecule, intermolecular interactions such as Lennard-Jones potentials were not taken into account. Special focus was given to the issue how to rigorously scale such a problem for an analog computer. Thanks to hardware-supported scientific (floating point) number representation on digital computers, this topic is discussed only rarely in numerical approaches, for instance in the context of catastrophic cancellation (loss of significance). Number representation and loss of resolution are problems not yet solved within this work, but approaches exist to improve these properties without losing the advantages such as fully parallel evaluation and exceptionally good energy efficiency. We also expect that a number of techniques (and experiences) in numerically solving molecular dynamics can be applied on analog-digital hybrid approaches, such as more robust higher order integration schemes or implementing long-distance effects (spatial cut-offs).

It should be noted that in molecular dynamics, the individual particle positions are of no direct interest and only used to derive effective, thermodynamical quantities. We can project the computational power of this attempt, since it scales mathematically and in terms of electrical engineering, but we do not go beyond the particle level in this work.

There are also completely different approaches, such as the application of artificial intelligence approaches in molecular dynamics [3]. For instance, in density functional theory (DFT) approaches, Behler and Parrinello [4] introduce a new breed of neural network model of DFT. This gives the energy as a function for all atomic positions in systems with arbitrary size and of various order of magnitude, which is faster than traditional DFT approaches. The high level of accuracy of the neural network method is shown for bulk silicon compared with DFT. It may turn out that implementing analog neural networks which solve DFT or MD can be a very fruitful approach combining the energy efficiency and fast time to solution shown by analog computers without the issues of stability, convergence and resolution.

8 THANKS

The authors thank Chris Giles for linguistic corrections.

REFERENCES

- [1] (2006). *Geometric Numerical Integration*. Springer-Verlag.
- [2] (2007). Chapter 7 - motion of nuclei. In Lucjan Piela, editor, *Ideas of Quantum Chemistry*, pages 275–323. Elsevier, Amsterdam.

- [3] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. (November 2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938.
- [4] Jörg Behler and Michele Parrinello. (April 2007). Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters*, 98(14).
- [5] B. R. Brooks *et al.* (2009). Charmm: The biomolecular simulation program. *Journal of Computational Chemistry*, 30(10):1545–1614.
- [6] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, S. Swaminathan, and Martin Karplus. (1983). Charmm: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4(2):187–217.
- [7] J.C. Butcher. (1996). A history of runge-kutta methods. *Applied Numerical Mathematics*, 20(3):247–260.
- [8] Bahar I Cui Q. (2006). *Normal Mode Analysis: Theory and applications to biological and chemical systems*.
- [9] Michael Gecht, Marc Siggel, Max Linke, Gerhard Hummer, and Jürgen Köfinger. (October 2020). MDBenchmark: A toolkit to optimize the performance of molecular dynamics simulations. *The Journal of Chemical Physics*, 153(14):144105.
- [10] O Guvench and Jr. MacKerell. (2008). Comparison of protein force fields for molecular dynamics simulations. *Methods Mol Biol.*, (443):63–88.
- [11] Kevin Hadley and Clare McCabe. (07 2012). Coarse-grained molecular models of water: A review. *Molecular simulation*, 38:671–681.
- [12] Turkan Haliloglu, Ivet Bahar, and Burak Erman. (October 1997). Gaussian dynamics of folded proteins. *Physical Review Letters*, 79(16):3090–3093.
- [13] Berk Hess, Carsten Kutzner, David van der Spoel, and Erik Lindahl. (February 2008). GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. *Journal of Chemical Theory and Computation*, 4(3):435–447.
- [14] Konrad Hinsén. (January 2000). The molecular modeling toolkit: A new approach to molecular simulations. *Journal of Computational Chemistry*, 21(2):79–85.
- [15] William L. Jorgensen, Jayaraman Chandrasekhar, Jeffry D. Madura, Roger W. Impey, and Michael L. Klein. (1983). Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935.
- [16] Sven Köppel, Bernd Ulmann, Lars Heimann, and Dirk Killat. (February 2021). Using analog computers in today’s largest computational challenges. *arXiv e-prints*, page arXiv:2102.07268.
- [17] Pedro E. M. Lopes, Olgun Guvench, and Alexander D. MacKerell. (September 2014). Current status of protein force fields for molecular dynamics simulations. In *Methods in Molecular Biology*, pages 47–71. Springer New York.
- [18] Alexander D. Mackerell. (2004). Empirical force fields for biological macromolecules: Overview and issues. *Journal of Computational Chemistry*, 25(13):1584–1604.
- [19] Pekka Mark and Lennart Nilsson. (November 2001). Structure and dynamics of the TIP3p, SPC, and SPC/e water models at 298 K. *The Journal of Physical Chemistry A*, 105(43):9954–9960.
- [20] Aneesur Rahman and Frank H. Stillinger. (1971). Molecular dynamics study of liquid water. *The Journal of Chemical Physics*, 55(7):3336–3359.
- [21] D. C. Rapaport. (April 2004). *The Art of Molecular Dynamics Simulation*. Cambridge University Press.

- [22] Romelia Salomon-Ferrer, David A. Case, and Ross C. Walker. (September 2012). An overview of the amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3(2):198–210.
- [23] Rahul Sarpeshkar. (October 1998). Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Computation*, 10(7):1601–1638.
- [24] Bernd Ulmann. (2019). *Model-1 Analog Computer Handbook/User Manual*.
- [25] Bernd Ulmann. (2020). *Analog and Hybrid Computer Programming*. De Gruyter.
- [26] L. C. Westphal. (1995). *BIBO stability and simple tests*, pages 351–370. Springer US, Boston, MA.