DynexSolve: The Dynex Platform Proof-of-Useful-Work Scheme

Dynex Developers

June 14, 2024
V.2.1

Abstract

Dynex is a DePIN-platform for neuromorphic quantum computing (n.quantum computing)[1,2] based on an innovative new blockchain protocol. It consists of participating Proof-of-Useful-Work (PoUW) miners that constitute a decentralised n.quantum computing network which is capable of computing quantum computing algorithms, featuring quantum entanglement and tunnelling effects, but without the limitations of traditional quantum computing (error-correction, operating temperature, number of qubits) at scale, potentially solving real-world problems[3,4,5,6].

The Dynex neuromorphic chips run on miners' computers and are used to solve computational tasks, that's the algorithm called DynexSolve. Every progress miners contribute is stored in the blockchain.

All this is wrapped in a sustainable and long term business model. Dynex customers are corporates, organisations or research bodies that are not able to solve their complex computational problems due to lack of computing power and high energy consumption of traditional computer networks. These DynexSolve projects are priced with a certain amount of Dynex coins. DynexSolve algorithm is the first mining algorithm which solves real-world computational problems while providing Proof-of-Useful-Work during the mining process.

This document describes the DynexSolve algorithm used on the Dynex Platform since December 1, 2022 including updates up until June 14, 2024.

---

1 https://en.wikipedia.org/wiki/Unconventional_computing

2 Neuromorphic Computing for Computer Scientists: A complete guide to Neuromorphic Computing on the Dynex Neuromorphic Cloud Computing Platform; Dynex Developers; Amazon; 2024; ISBN-13: 979-8874282196

3 https://en.wikipedia.org/wiki/Neuromorphic_engineering

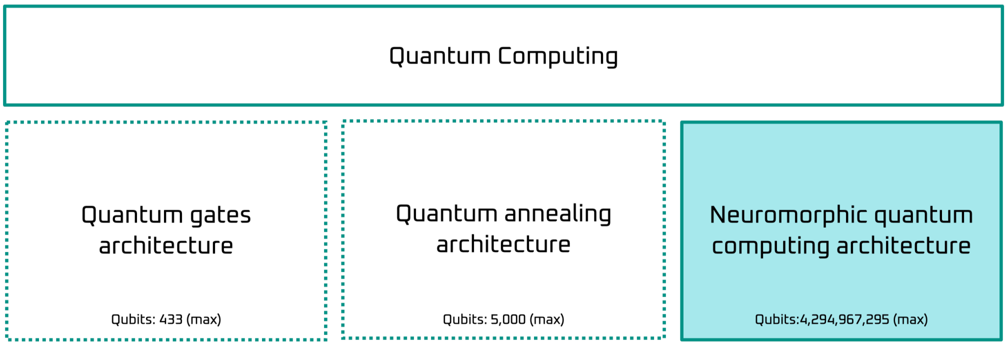
4 Advancements in Unsupervised Learning: Mode-Assisted Quantum Restricted Boltzmann Machines Leveraging Neuromorphic Computing on the Dynex Platform; Adam Neumann, Dynex Developers; International Journal of Bioinformatics & Intelligent Computing. 2024; Volume 3(1):91- 103, ISSN 2816-8089

5 HUBO & QUBO and Prime Factorization; Samer Rahmeh, Cali Technology Solutions, Dynex Developers; International Journal of Bioinformatics & Intelligent Computing. 2024; Volume 3(1):45-69, ISSN 2816-8089

6 Framework for Solving Harrow-Hassidim-Lloyd Problems with Neuromorphic Computing using the Dynex Cloud Computing Platform; Samer Rahmeh, Cali Technology Solutions, Dynex Developers; 112871175; Academia.edu; 2023

Rationale

Neuromorphic Quantum Computing[7,8] (abbreviated as '**n.quantum computing**') is an unconventional computing type of computing that uses neuromorphic computing to perform quantum operations[9,10]. It was suggested that quantum algorithms, which are algorithms that run on a realistic model of quantum computation, can be computed equally efficiently with neuromorphic quantum computing[11,12,13,14,15].



Both, traditional quantum computing and n.quantum computing are physics-based unconventional computing approaches to computations and don't follow the von Neumann architecture. They both construct a system (a circuit) that represents the physical problem at hand, and then leverage their respective physics properties of the system to seek the "minimum". n.quantum computing and quantum computing share similar physical properties during computation[15,16].

---

7 Pehle, Christian; Wetterich, Christof (2021-03-30), Neuromorphic quantum computing, doi:10.48550/arXiv.2005.01533

8 "Neuromrophic Quantum Computing | Quromorphic Project | Fact Sheet | H2020". *CORDIS | European Commission*. doi:10.3030/828826

9 Wetterich, C. (2019-11-01). "Quantum computing with classical bits". *Nuclear Physics B*. **948**: 114776. doi:10.1016/j.nuclphysb.2019.114776. ISSN 0550-3213

10 Pehle, Christian; Meier, Karlheinz; Oberthaler, Markus; Wetterich, Christof (2018-10-24), Emulating quantum computation with artificial neural networks, doi:10.48550/arXiv.1810.10335, retrieved 2024-03-18

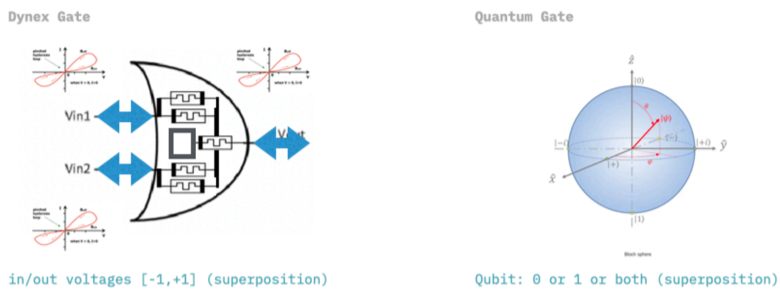
11 Carleo, Giuseppe; Troyer, Matthias (2017-02-10). "Solving the quantum many-body problem with artificial neural networks". *Science*. **355** (6325): 602–606. doi:10.1126/science.aag2302. ISSN 0036-8075

12 Torlai, Giacomo; Mazzola, Guglielmo; Carrasquilla, Juan; Troyer, Matthias; Melko, Roger; Carleo, Giuseppe (2018-05). "Neural-network quantum state tomography". Nature Physics. 14 (5): 447–450. doi:10.1038/s41567-018-0048-5

13 Sharir, Or; Levine, Yoav; Wies, Noam; Carleo, Giuseppe; Shashua, Amnon (2020-01-16). "Deep Autoregressive Models for the Efficient Variational Simulation of Many-Body Quantum Systems". *Physical Review Letters*. **124** (2): 020503. doi:10.1103/PhysRevLett.124.020503

14 Broughton, Michael; Verdon, Guillaume; McCourt, Trevor; Martinez, Antonio J.; Yoo, Jae Hyeon; Isakov, Sergei V.; Massey, Philip; Halavati, Ramin; Niu, Murphy Yuezhen (2021-08-26), TensorFlow Quantum: A Software Framework for Quantum Machine Learning, doi:10.48550/arXiv.2003.02989

15 Di Ventra, Massimiliano (2022-03-23), MemComputing vs. Quantum Computing: some analogies and major differences, doi:10.48550/arXiv.2203.12031



n.quantum gates: superposition of input/output signals

n.quantum Entanglement Effects

Quantum computing utilise certain principles of quantum mechanics, such as entanglement, to enable their components to be interconnected over vast distances. This interconnection means that a change in one component can instantly affect others, no matter how far apart they are. This characteristic facilitates the computer's ability to swiftly determine the most energy-efficient state. Similarly, n.quantum computing exhibits a related phenomenon through long-distance connections stemming from a state known as criticality[16]. In this setup, each gate within the circuit is designed to be responsive to distant gates, creating a network where each gate can influence and be influenced by others across the system. As the circuit evolves, the connections between different gates reach a critical state, making the system prone to sudden, large-scale changes or "avalanches" triggered by minor disturbances anywhere in the circuit. These avalanches, driven by the circuit's long-range correlations, enable the system to quickly find the lowest energy state, significantly speeding up computation compared to traditional methods[17,18].

n.quantum Tunnelling Effects

Quantum computing also harness the principle of quantum tunnelling to find the most efficient solution. This process enables them to bypass energy barriers and access lower energy states that represent improved solutions. In contrast, n.quantum computing circuits[19] utilise the behaviour of electrical currents and voltages to expedite the transition to more favourable states. The concept of an instanton[20], borrowed from the study

---

[16] Di Ventra, Massimiliano; Traversa, Fabio L.; Ovchinnikov, Igor V. (2017-12). "Topological Field Theory and Computing with Instantons". Annalen der Physik. 529 (12). doi:10.1002/andp.201700123

[17] Di Ventra, Massimiliano (2022-03-23), MemComputing vs. Quantum Computing: some analogies and major differences, doi:10.48550/arXiv.2203.12031

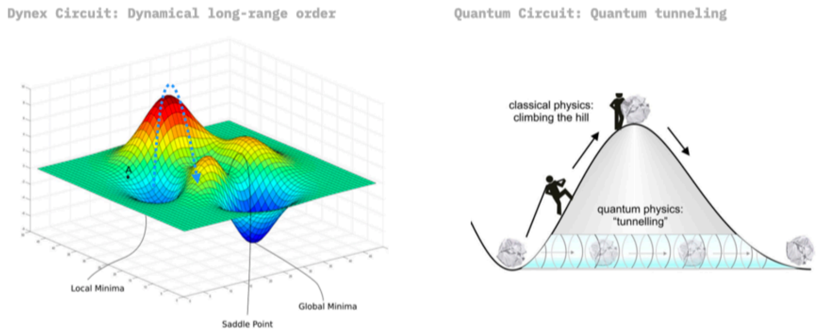
[18] Di Ventra, Massimiliano; Traversa, Fabio L.; Ovchinnikov, Igor V. (2017-12). "Topological Field Theory and Computing with Instantons". Annalen der Physik. 529 (12). doi:10.1002/andp.201700123

[19] Gonzalez-Raya, Tasio; Lukens, Joseph M.; Céleri, Lucas C.; Sanz, Mikel (2020-01). "Quantum Memristors in Frequency-Entangled Optical Fields". Materials. 13 (4): 864. doi:10.3390/ma13040864. ISSN 1996-1944. PMC 7079656

[20] Di Ventra, Massimiliano; Traversa, Fabio L.; Ovchinnikov, Igor V. (2017-12). "Topological Field Theory and Computing with Instantons". Annalen der Physik. 529 (12). doi:10.1002/andp.201700123. ISSN 0003-3804

of dynamical systems, explains how these circuits navigate through energy barriers. This happens as the system encounters "saddle" points within the electrical landscape, which possess characteristics that attract the system initially but then repel it as it gets closer. This interaction results in the system being propelled away from these points at high speed, effectively allowing it to jump from one state to another across the landscape. This phenomenon[20] mirrors quantum tunnelling but occurs in the realm of voltages and currents. In neuromorphic quantum computing circuits, this behaviour emerges from the collective action of memristor based gates, which induce widespread fluctuations in voltages, thereby swiftly steering the circuit towards superior configurations.

These inherent physical mechanisms enable both, n.quantum computing circuits and quantum computing to navigate towards the best possible solution out of a vast array of potential configurations, by effectively mapping the solution into their system[17].



n.quantum Tunneling & Entanglement

Scalability & Real-World Applications

There are currently a number of significant engineering obstacles to construct useful quantum computers capable of solving real-world problems at scale. The major challenge in quantum computing is maintaining the coherence of entangled qubits' quantum states; they suffer from quantum decoherence and state fidelity from outside noise (vibrations, fluctuations in temperature, electromagnetic waves). To overcome noise interference, quantum computers are often isolated in large refrigerators cooled to near absolute zero (colder than outer space) to shield them and in turn, reduce errors in calculation. Although there are error-correcting techniques being deployed, there are currently no existing quantum computers capable of maintaining the full coherence required to solve industrial-sized problems at scale today[21]. Therefore, they are mostly limited to solving toy-sized problems. In contrast to quantum computing, n.quantum computing possesses the capability to be efficiently emulated on contemporary computers through software, as well as to be physically constructed using conventional electrical components. Quantum

[21] Krinner, Sebastian; Lacroix, Nathan; Remm, Ants; Di Paolo, Agustin; Genois, Elie; Leroux, Catherine; Hellings, Christoph; Lazar, Stefania; Swiadek, Francois; Herrmann, Johannes; Norris, Graham J.; Andersen, Christian Kraglund; Müller, Markus; Blais, Alexandre; Eichler, Christopher (2022-05). "Realizing repeated quantum error correction in a distance-three surface code". Nature. 605 (7911): 669–674. doi:10.1038/s41586-022-04566-8

algorithms can be computed efficiently with n.quantum computing at scale, potentially solving real-world problems[22,23,24].

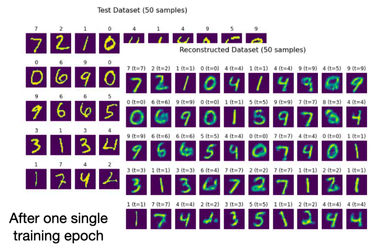
Practical Applications

Utilising n.quantum computing offers a multitude of practical applications, especially as the traditional scaling limitations of traditional quantum computing do not apply:

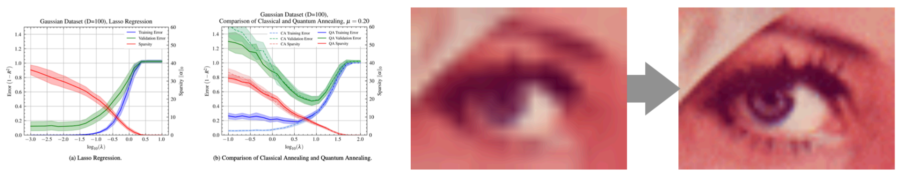
(1) Boosting Artificial Intelligence

• **Quantum-Boltzmann-Machines** (QBM) evolve significantly faster to an attractive Mean Squared Error (MSE) than the traditional RBM, resulting in less training iterations and increased model accuracy)[25].

	RBM	DYNEX QBM
Best MSE at training epoch	158	1
Best Mean Squared Error (MSE)	0,010	0.0097
Training Samples	60,000	60,000
Testing Samples	10,000	10,000
Dataset	MNIST	MNIST



• **Quantum Single-Image-Super-Resolution** (QSISR): The quantum SISR algorithm as a sparse coding optimization problem, which is solved using the Dynex Neuromorphic Computing Platform via the Dynex SDK. This AQC-based algorithm was demonstrated to achieve significantly improved QSISR accuracy[25].



• **Quantum-Support-Vector-Machines** (QSVM) model training is superior to traditional support vector machines[25].

---

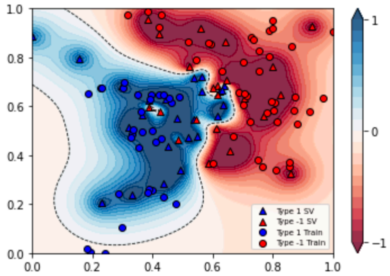
[22] Advancements in Unsupervised Learning: Mode-Assisted Quantum Restricted Boltzmann Machines Leveraging Neuromorphic Computing on the Dynex Platform; Adam Neumann, Dynex Developers; International Journal of Bioinformatics & Intelligent Computing. 2024; Volume 3(1):91- 103, ISSN 2816-8089

[23] HUBO & QUBO and Prime Factorization; Samer Rahmeh, Cali Technology Solutions, Dynex Developers; International Journal of Bioinformatics & Intelligent Computing. 2024; Volume 3(1):45-69, ISSN 2816-8089

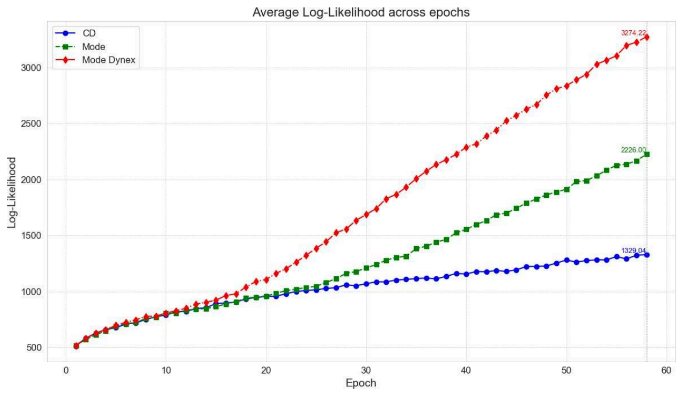
[24] Neuromorphic Computing for Computer Scientists: A complete guide to Neuromorphic Computing on the Dynex Neuromorphic Cloud Computing Platform; Dynex Developers; Amazon; 2024; ISBN-13: 979-8874282196

[25] Neuromorphic Computing for Computer Scientists: A complete guide to Neuromorphic Computing on the Dynex Neuromorphic Cloud Computing Platform; Dynex Developers; Amazon; 2024; ISBN-13: 979-8874282196

	Data 1					Data 2				
Size	$F1$	Prec	Rec	Acc	Lagr	$F1$	Prec	Rec	Acc	Lagr
HQPU										
30	1	1	1	1	7.12	0.66	1	0.5	0.8	7.44
60	0.93	0.88	1	0.95	13.81	0.88	0.8	1	0.9	11.45
90	0.83	1	0.7	0.83	15.98	0.86	0.81	0.92	0.86	16.79
120	0.91	0.95	0.88	0.90	31.04	0.875	0.875	0.875	0.9	21.30
150	0.95	1	0.89	0.94	38.72	0.84	0.88	0.88	0.88	31.85
SA										
30	1	1	1	1	7.13	0.66	1	0.5	0.8	7.45
60	0.93	0.88	1	0.95	13.81	0.88	0.8	1	0.9	11.45
90	0.83	1	0.7	0.83	15.98	0.93	0.81	0.92	0.95	16.8
120	0.91	0.95	0.88	0.90	31.04	0.87	0.875	0.875	0.9	21.30
150	0.95	1	0.89	0.94	38.72	0.84	0.88	0.8	0.88	31.85
Scikit-Learn										
30	1	1	1	1	2.49	0.57	0.4	1	0.7	–4.45
60	1	1	1	1	5.99	0.88	1	0.8	0.9	–15.92
90	0.77	0.71	0.83	0.80	8.30	0.94	0.94	0.94	0.93	–21.18
120	0.91	0.9	0.9	0.9	13.43	0.85	0.82	0.88	0.88	–18.57
150	0.92	0.92	0.92	0.92	4.63	0.85	0.77	0.94	0.88	–34.23
QPU										
30	1	1	1	1	12	0.8	1	0.5	0.9	5.9
60	0.93	0.88	1	0.95	96.6	0.75	0.9	1	0.8	8.4
90	0.77	1	1	0.76	294	0.8	0.75	8	0.8	6.53
DYNEX										
150	1	1	1	1						



• **Mode-Assisted Quantum Restricted Boltzmann Machines**: The integration of neuromorphic computing into the Dynex platform signifies a transformative step in computational technology, particularly in the realms of machine learning and optimization. This advanced platform leverages the unique attributes of neuromorphic dynamics, utilising neuromorphic annealing - a technique divergent from conventional computing methods - to adeptly address intricate problems in discrete optimization, sampling, and machine learning[26].

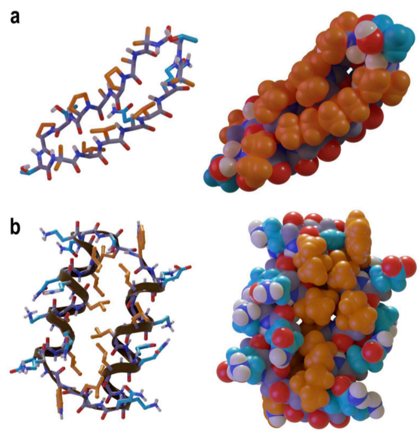


(2) Pharmaceutical Applications

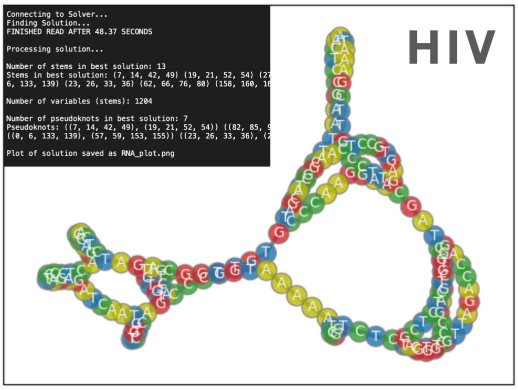
• **Drug Repurposing with 3D molecular quantum methods**: From concept to treating a patient, it can take 10 years for a single treatment. Drug repositioning, repurposing, re-tasking, re-profiling or drug rescue is the process by which approved drugs are employed to treat a disease they were not initially intended/designed for. Virtual screening has become essential at the early stages of drug discovery. The 3D

[26] Advancements in Unsupervised Learning: Mode-Assisted Quantum Restricted Boltzmann Machines Leveraging Neuromorphic Computing on the Dynex Platform; Adam Neumann, Dynex Developers; International Journal of Bioinformatics & Intelligent Computing. 2024; Volume 3(1):91- 103, ISSN 2816-8089

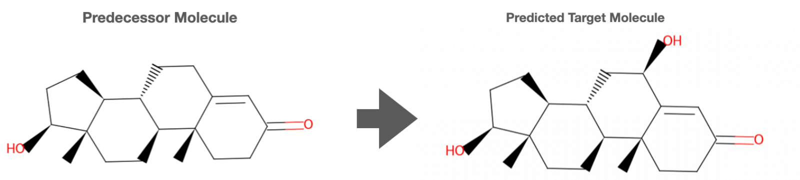
molecular Quantum method is computed efficiently on the Dynex platform and provides a superior virtual screening method.[27]



- **RNA-folding** finds the optimal stem configuration of the RNA sequence from viruses using the Dynex platform. The example takes an RNA sequence and applies a quadratic model in pursuit of the optimal stem configuration[28].



- **Enzyme-target prediction** algorithms predict potential interactions between enzymes and target molecules and leverages the principles of quantum mechanics[28].

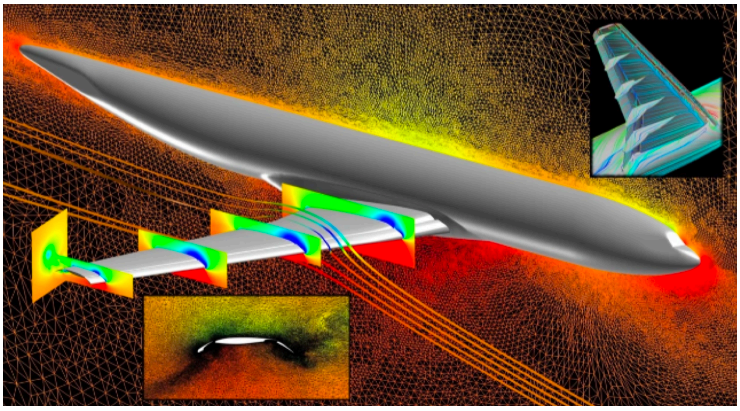


[27] Jimenez-Guardeño JM, Ortega-Prieto AM, Menendez Moreno B, Maguire TJA, Richardson A, Diaz-Hernandez JI, Diez Perez J, Zuckerman M, Mercadal Playa A, Cordero Deline C, Malim MH, Martinez-Nunez RT. Drug repurposing based on a quantum-inspired method versus classical fingerprinting uncovers potential antivirals against SARS-CoV-2. PLoS Comput Biol. 2022 Jul 18;18(7):e1010330. doi: 10.1371/journal.pcbi.1010330. PMID: 35849631; PMCID: PMC9333455.

[28] Neuromorphic Computing for Computer Scientists: A complete guide to Neuromorphic Computing on the Dynex Neuromorphic Cloud Computing Platform; Dynex Developers; Amazon; 2024; ISBN-13: 979-8874282196

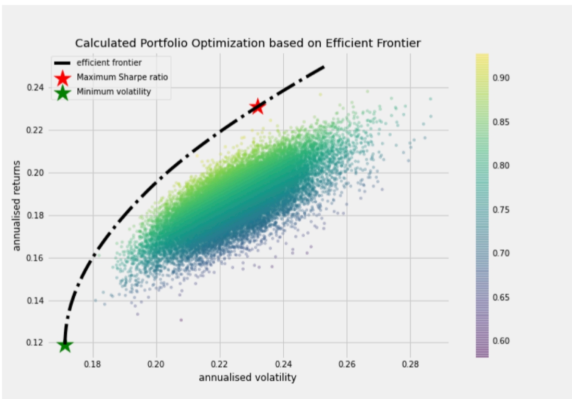
(3) Automotive, Aerospace, Super-Sports & Space Exploration

- **Quantum Computation of Fluid Dynamics** (QCFD): Dynex offers an innovative platform for the efficient simulation of Computational Fluid Dynamics (QCFD), a powerful discipline within engineering and physics. With Dynex, QCFD simulations can be conducted seamlessly, providing engineers and researchers with a robust tool for analysing fluid flow, heat transfer, and related phenomena. This capability is invaluable in numerous industries, including aerospace, automotive, and energy, where understanding and optimising fluid behaviour is crucial. By utilising Dynex's advanced computational capabilities, users can gain insights into aerodynamics, thermal management, and fluid interactions, ultimately aiding in the design and optimization of various systems and devices. Dynex empowers engineers to accelerate the QCFD simulation process, fostering innovation and driving advancements in fields reliant on fluid dynamics analyses[28].



(4) Financial Services

- **Stock portfolio optimisation with quantum algorithms**: Portfolio Optimization (PO) is a standard problem in the financial industry. The computational complexity of such combinatorial optimization problems tend to increase exponentially with the number of variables — here, the number of assets — which at large scale can make solvers incapable of providing only optimal solutions. Instead, the results are likely suboptimal. This article provides detailed introduction on how stock portfolio optimisation can be performed by using quantum computing algorithm on the Dynex Neuromorphic Computing Cloud overcoming these limitations.



n.quantum Circuit Simulation

Herein, there will be many references to modern computers and the computation they perform. It is important to realise that computing is fundamentally a **physical process**[29]. The statement may seem obvious when considering the physical processes harnessed by the electronic components of computers (for example, transistors), however, virtually any physical process can be harnessed for some form of computation. Note, that we are speaking of **Alan Turing's model of computation**[30], that is, a mapping (transition function) between two sets of finite symbols (input and output) in discrete time.

It is important to distinguish between continuous and discrete time: **Dynex circuits** operate in **continuous time**[31], though, their **simulations** on modern computers **require the discretisation** of time. Continuous time is physical time: a fundamental physical quantity. Discrete time is not a physical quantity, and might be best understood as counting time: counting something (function calls, integration steps, etc.) to give an indication (perhaps approximation) of the physical time. In the literature of Physics and other Physical Sciences, physical time has an assigned SI unit of seconds, whereas in Computer Science and related disciplines, counting time is dimensionless.

Granting the infinite resources utilised by a Turing machine[32], universal Dynex circuits (UDCs) have been shown to be Turing-complete, meaning universal Dynex circuits can simulate an universal Turing machine. The universal Dynex circuits class contains digital and analog machines. While analog Dynex circuits theoretically have tremendous computational power, analog systems cannot be engineered for scalability, as their growing size requires growing resources to achieve the same accuracy. It is the **digital Dynex circuit** that is **scalable**, and the focus of this algorithm.

The Fourth Missing Circuit Element

Modern computers rely on the implementation of uni-directional logic gates that represent Boolean functions[33]. Circuits built to simulate Boolean functions are desirable because they are deterministic: A unique input has a unique, reproducible output.

---

29 Massimiliano Di Ventra and Fabio L. Traversa. Perspective: Memcomputing: Leveraging memory and physics to compute efficiently. Journal of Applied Physics, 123(18):180901, 2018.

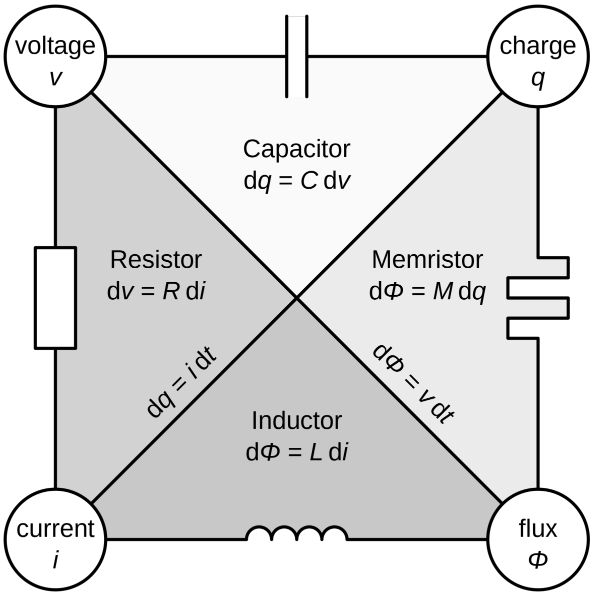
30 Michael R. Garey and David S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA, 1990.

31 A physical process is necessarily continuous in time, as discrete time is not physical, rather a necessary consequence of simulating a physical system.

32 A physical realization of Dynex chips will, of course, have finite resources. However, the “chips” used to study computational complexity are theoretical and impossible to build

33 Behrooz Parhami. Computer arithmetic, volume 20. Oxford university press, 2010.

Modern computers relegate the task of logic to central processing units (CPUs). However, the resources required for the task might exhaust the resources present within the CPU, specifically, cache memory. For typical processes on modern computers, random-access memory (RAM) is the memory used for data and machine code, and is external to the CPU. The physical separation of CPU and RAM results in what is known as the **von Neumann bottleneck**, a slow down in computation caused by the transfer of information between physical locations[34].



Conceptual symmetries of resistor, capacitor, inductor, and memristor

To overcome the von Neumann bottleneck, we propose computing with and in memory, utilising **ideal memristors**[35]. Distinct from in-memory computation[36], it is an efficient computing paradigm that uses memory to process and store information in the same physical location.

A **memristor** is an electrical component that limits or regulates the flow of electrical current in a circuit and remembers the amount of charge that has previously flowed through it. Memristors are important because they are non-volatile, meaning that they **retain memory without power**.

The original concept for memristors, as conceived in 1971 by Professor Leon Chua at the University of California, Berkeley[37], was a nonlinear, passive two-terminal electrical component that linked electric charge and magnetic flux (**"The missing circuit**

---

[34] John Backus. Can programming be liberated from the von neumann style?: A functional style and its algebra of programs. Commun. ACM, 21(8):613–641, August 1978.

[35] Chua, L. (1971). "Memristor-The missing circuit element". *IEEE Transactions on Circuit Theory*. 18 (5): 507–519.

[36] H. Zhang, G. Chen, B. C. Ooi, K. Tan, and M. Zhang. In-memory big data management and processing: A survey. IEEE Transactions on Knowledge and Data Engineering, 27(7):1920–1948, 2015.

[37] L. Chua, "Memristor-The missing circuit element," in *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507-519, September 1971, doi: 10.1109/TCT.1971.1083337.

element"). Since then, the definition of memristor has been broadened to include any form of non-volatile memory that is based on resistance switching, which increases the flow of current in one direction and decreases the flow of current in the opposite direction.

A memristor is often compared to an imaginary pipe that carries water. When the water flows in one direction, the pipe's diameter expands and allows the water to flow faster -- but when the water flows in the opposite direction, the pipe's diameter contracts and slows the water's flow down. If the water is shut off, the pipe retains its diameter until the water is turned back on. To continue the analogy, when a memristor's power is shut off, the memristor retains its resistance value. ***This would mean that if power to a computer was cut off with a hard shut down, all the applications and documents that were open before the shut down would still be right there the screen when the computer was restarted****.*

Memristors, which are considered to be a sub-category of resistive RAM, are one of several storage technologies that have been predicted to replace flash memory. Scientists at HP Labs built the first working memristor in 2008 and since that time, researchers in many large IT companies have explored how memristors can be used to create smaller, faster, low-power computers that do not require data to be transferred between volatile and non-volatile memory.

A **digital Dynex chip** is realised as a **memristor based bi-directional logic circuit**. These circuits differ from traditional logic circuits in that input and output terminals are no longer distinct. In a traditional logic circuit, some input is given and the output is the result of computation performed on the input, via uni-directional logic gates. In contrast, a memristor based bi-directional logic circuit can be operated by **assigning the output** terminals, then **reading the input** terminals.

Operating a logic circuit "backwards" has many applications. An example is integer factorisation: Given an integer, factor it into its prime factors. For simplicity, assume the given integer, b, is the product of two prime numbers, p and q. If given p and q, then a multiplication circuit can be employed to find the product, b, of the two prime numbers. A traditional logic circuit, appropriately designed, can easily perform this task. Now, if given b and told it can be factored into two prime numbers, we take the same multiplication circuit structure (logic gates connected similarly), but design it to be a memristor based bi-directional logic circuit so the logic gates become terminal agnostic, meaning signal can be received and sent on any terminal of the logic gate. However, the new logic gates are not bijective, so the entire circuit will have to self-organiae to produce the values of p and q on the "input" terminals.

Self-organising logic is a recently-suggested framework that allows the solution of Boolean truth tables "in reverse," i.e., it is able to satisfy the logical proposition of gates regardless to which terminal(s) the truth value is assigned ("terminal-agnostic logic"). It can be realised if time non-locality (memory) is present. A practical realisation of self-organising logic gates can be done by combining circuit elements with and without memory. By employing one such realisation, it can be shown numerically, that self-organising logic gates **exploit elementary instantons** to reach equilibrium points. Instantons are classical trajectories of the non-linear equations of motion describing self-organising logic gates, and connect topologically distinct critical points in the phase space. By linear analysis at those points it can be shown that these instantons connect the initial critical point of the dynamics, with at least one unstable direction, directly to the final fixed point. It can also be shown that the memory content of these gates only affects the relaxation time to reach the logically consistent solution. By solving the corresponding stochastic differential equations, since instantons connect critical points, noise and perturbations may change the instanton trajectory in the phase space, but not the initial and final critical points. Therefore, **even for extremely large noise levels**, the gates **self-organise to the correct solution**.

Note that the self-organising logic we consider here has no relation to the invertible universal Toffoli gate that is employed, e.g., in quantum computation[38]. Toffoli gates are truly one-to-one invertible, having 3-bit inputs and 3-bit outputs. On the other hand, self-organising logic gates need only to satisfy the correct logical proposition, without a one-to-one relation between any number of input and output terminals. Instead, it is worth mentioning another type of bi-directional logic that has been recently discussed in[39] using stochastic units (called p-bits). These units fluctuate among all possible consistent inputs. However, in contrast to that work, the invertible logic we consider here is **deterministic**.

With time being a fundamental ingredient, a **dynamical systems approach** is most natural to describe such gates. In particular, non-linear electronic (non-quantum) circuit elements with and without memory have been suggested as building blocks to realise self-organising logic gates in practice[40].

By assembling self-organising logic gates with the appropriate architecture, one then obtains circuits that can **solve complex problems efficiently** by mapping the equilibrium (fixed) points of such circuits to the solution of the problem at hand, as shown in [41,42,43,44]. Moreover, it has been proved that, if those systems are engineered to be point

[38] Tommaso Toffoli. Reversible computing. In International Colloquium on Automata, Languages, and Programming, pages 632–644. Springer, 1980.

[39] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta. Stochastic p-bits for invertible logic. Phys. Rev. X, 7:031014, Jul 2017.

[40] Fabio Lorenzo Traversa and Massimiliano Di Ventra. Polynomial-time solution of prime factorization and np-complete problems with digital memcomputing machines. Chaos: An Interdisciplinary Journal of Nonlinear Science, 27:023107, 2017.

[41] Fabio Lorenzo Traversa and Massimiliano Di Ventra. Polynomial-time solution of prime factorization and np-complete problems with digital memcomputing machines. Chaos: An Interdisciplinary Journal of Nonlinear Science, 27:023107, 2017.

[42] H. Manukian, F. L. Traversa, and M. Di Ventra. Memcomputing numerical inversion with self-organizing logic gates. IEEE Transactions on Neural Networks and Learning Systems, PP(99):1–6, 2017.

[43] Haik Manukian, Fabio L Traversa, and Massimiliano Di Ventra. Accelerating deep learning with memcomputing. Neural Networks, 110:1–7, 2019.

[44] M. Di Ventra, Fabio L. Traversa, and Igor V. Ovchinnikov. Topological field theory and computing with instantons. Ann. Phys. (Berlin), 529:1700123, 2017.

dissipative[45], then, if equilibrium points are present, they do not show chaotic behaviour[46] or periodic orbits[47].

It was subsequently demonstrated[48], using topological field theory (TFT) applied to dynamical systems, that these circuits are described by a Witten-type TFT[49], and they support long-range order, mediated by instantons. Instantons are classical trajectories of the non-linear equations of motion describing these circuits (see, e.g., [50] or [51]).

The intrinsic non-locality of instantons, coupled with the topological character of critical points, is reminiscent of the “rigidity” and topological character of the ground state of some strongly-correlated quantum systems that are currently investigated for topological quantum computation, namely quantum computation that is robust against dephasing and noise[52,53,54]. This analogy is not far-fetched. In fact, in the case of self-organising circuits, instantons, by connecting topologically-distinct critical points in the phase space, correlate elements of the circuit non-locally in space and time[55]. This non-locality is somewhat reminiscent of quantum entanglement. However, self-organising logic gates are circuits that **achieve long-range order without quantum-mechanical effects**.

---

[45] J.K. Hale. Asymptotic Behavior of Dissipative Systems, volume 25 of Mathematical Surveys and Monographs. American Mathematical Society, Providence, Rhode Island, 2nd edition, 2010.

[46] M. Di Ventra and F. L. Traversa. Absence of chaos in Digital Memcomputing Machines with solutions. Physics Letter A, 2017.

[47] M. Di Ventra and F. L. Traversa. Absence of periodic orbits in digital memcomputing machines with solutions. Chaos: An Interdisciplinary Journal of Nonlinear Science, 27:101101, 2017.

[48] M. Di Ventra, Fabio L. Traversa, and Igor V. Ovchinnikov. Topological field theory and computing with instantons. Ann. Phys. (Berlin), 529:1700123, 2017.

[49] E. Witten. Topological quantum field theory. Comms. in Math. Phys., 117:353–386, 1988.

[50] S. Coleman. Aspects of Symmetry, Chapter 7. Cambridge University Press, 1977.

[51] K. Hori, S. Katz, R. Klemm, A. Pandharipande, R. Thomas, C. Vafa, R. Vakil, and E. Zaslow. Mirror symmetry. Clay Mathematics, 2000.

[52] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological quantum computation. Bulletin of the American Mathematical Society, 40(1):31–38, 2003.

[53] Chetan Nayak, Steven H Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. Reviews of Modern Physics, 80(3):1083, 2008.

[54] A Yu Kitaev. Fault-tolerant quantum computation by anyons. Annals of Physics, 303(1):2–30, 2003.

[55] M. Di Ventra, Fabio L. Traversa, and Igor V. Ovchinnikov. Topological field theory and computing with instantons. Ann. Phys. (Berlin), 529:1700123, 2017.

n.quantum Computing on Dynex

The DynexSolve algorithm represents a n.quantum implementation of an efficient sampler to compute **Ising and Quadratic Unconstrained Binary Optimization** (QUBO) problems. Ising/QUBO problems are mapped onto a Dynex circuit and then being computed by the contributing workers. This ensures that traditional quantum algorithms can be computed without modifications on the Dynex platform[56] using the Python based Dynex SDK[57]. It also provides libraries which are compatible with Google TensorFlow, IBM Qiskit, PyTorch, Scikit-Learn and others. DynexSolve's source codes are publicly available[58].

Ising and QUBO problems play a pivotal role in the field of quantum computing, establishing themselves as the de-facto standard for mapping complex optimization and machine learning problems onto quantum systems. These frameworks are instrumental in leveraging the unique capabilities of quantum computers to solve problems that are intractable for classical computers.

The Ising model, originally introduced in statistical mechanics, describes a system of spins that can be in one of two states. This model has been adapted to represent optimization problems, where the goal is to minimise an energy function describing the interactions between spins. Similarly, the QUBO framework represents optimization problems with binary variables, where the objective is to minimise a quadratic polynomial. Both models are equivalent and can be transformed into one another, allowing a broad range of problems to be addressed using either formulation.

The significance of Ising and QUBO problems in quantum computing lies in their natural fit with quantum annealing and gate-based quantum algorithms. Quantum annealers, for instance, directly implement the Ising model to find the ground state of a system, which corresponds to the optimal solution of the problem. This method exploits quantum tunnelling and entanglement to escape local minima, offering a potential advantage over classical optimization techniques. Gate-based quantum computers, on the other hand, use quantum algorithms like the Quantum Approximate Optimization Algorithm (QAOA) to solve QUBO problems. These algorithms use quantum superposition and interference to explore the solution space more efficiently than classical algorithms, potentially leading to faster solution times for certain problems.

The adoption of Ising and QUBO as standards in quantum computing is due to their versatility and the direct mapping of various optimization and machine learning tasks onto quantum hardware. From logistics and finance to drug discovery and artificial intelligence, the ability to frame problems within the Ising or QUBO model opens up new

56 https://live.dynexcoin.org

57 https://github.com/dynexcoin/DynexSDK

58 https://github.com/dynexcoin/DynexSolve

avenues for solving complex challenges with quantum computing. This standardisation also facilitates the development of quantum algorithms and the benchmarking of quantum hardware, accelerating progress in the quantum computing field.

DynexSolve Simulation of Dynex Neuromorphic Quantum Circuits

Dynex's neuromorphic quantum circuits are individually constructed based on the specific computational problem, typically represented in the form of QUBO (Quadratic Unconstrained Binary Optimization), Ising models, Binary Quadratic Models (BQM), or Constrained Quadratic Models (CQM). These circuits are composed of linear and quadratic quantum gates. Given that these quantum gates utilize physical, electric components such as memristors, their behavior can be efficiently simulated by solving the resultant system of equations of motion through Ordinary Differential Equation (ODE) integration. The following differential equations are applied within the Dynex platform.

The optimal values for the applied parameters $\alpha, \beta, \gamma, \delta, \epsilon, \varepsilon, \zeta$ are being automatically identified by the Dynex platform. Note that these are different for each computational problem class.

Linear quantum gates

$$v_n = \sum_m QG_m W_m p_i \qquad \text{(qubit voltage)} \qquad (1)$$

Whereas:

$$QG_m = \frac{1.0 - p_i v_i}{2.0} \qquad \text{(quantum gate state)} \qquad (2)$$

$$W_m \qquad \text{(quantum gate weight)}$$

$$p_i = \begin{cases} 1.0 & i > 0 \\ -1.0 & i <= 0 \end{cases} \qquad \text{(quantum gate polarity)} \qquad (3)$$

Quadratic quantum gates

$$v_n = \sum_m G_{i,j}(W_m a_m b_m) + R_{i,j}\zeta(1 - a_m) \qquad \text{(qubit voltage)} \qquad (4)$$

Whereas:

$$W_m \qquad \text{(quantum gate weight)}$$

$$p_{i,j} = \begin{cases} 1.0 & i,j > 0 \\ -1.0 & i,j <= 0 \end{cases} \qquad \text{(quantum gate polarity)} \qquad (5)$$

$$QG_m = \frac{min((1.0 - p_i v_i), (1.0 - p_j v_j))}{2.0} \qquad \text{(quantum gate state)} \qquad (6)$$

$$G_i = \frac{p_i(1.0 - p_j v_j)}{2.0} \quad \text{(gradient term)} \tag{7}$$

$$G_j = \frac{p_j(1.0 - p_i v_i)}{2.0} \quad \text{(gradient term)} \tag{8}$$

$$R_i = \begin{cases} 0.0 & QG \neq \frac{1.0 - p_i v_i}{2.0} \\ \frac{p_i(1.0 - p_i v_i)}{2.0} & QG = \frac{1.0 - p_i v_i}{2.0} \end{cases} \quad \text{(rigidity term)} \tag{9}$$

$$R_j = \begin{cases} 0.0 & QG \neq \frac{1.0 - p_j v_j}{2.0} \\ \frac{p_j(1.0 - p_j v_j)}{2.0} & QG = \frac{1.0 - p_j v_j}{2.0} \end{cases} \quad \text{(rigidity term)} \tag{10}$$

Auxiliary variables:

$$a_m = \beta(a_m + \epsilon)(QG - \gamma) \quad \text{(memristive function)} \tag{11}$$

$$b_m = \alpha(1.0 + W)(QG - \delta) \quad \text{(memristive function)} \tag{12}$$

Equations 1-12 are being numerically integrated with an adaptive step-size based forward Euler integration scheme.

DynexSolve Proof-of-Useful-Work

The DynexSolve mining algorithm **performs the numerical integration** of all Dynex chips required for the computational job and is therefore classified as a Proof-of-Useful-Work (PoUW) mining algorithm.

Depending on problem size (number of variables *n* and number of clauses *m*) and the memory available on the provided Graphic-Processing-Units (GPUs) the capacity for each miner to run **parallel Dynex chips** is determined. As all miners are working collectively on computational jobs, a job and chip scheduling system is required to assign and balance the work required:

The **Dynex Mallob system**, named after the term *malleable,* which defines a distributed computing environment[59,60,] has been inspired by [61] and [62]. It dynamically assigns jobs with the respective available initial conditions to the individual miners and ensures that all 16*n* initial conditions are being computed for a maximum duration of n^5 integration steps.

DynexSolve combines two algorithms, namely the numerical integration of Dynex Chips as well as a modified CryptoNight hashing function to confirm blocks on the Dynex block chain. It has been designed to spend the **majority** of the computational **energy on the numerical integration** (meaningful work) to ensure that almost no resources are being wasted with hashing:

[59] Desell, T., Maghraoui, K.E. and Varela, C.A., 2007. Malleable applications for scalable high performance computing. *Cluster Computing*, *10*, pp.323-337.

[60] Ghafoor, S.K., 2007. Modeling of an adaptive parallel system with malleable applications in a distributed computing environment. Mississippi State University.

[61] Schreiber, D. and Sanders, P., 2021. Scalable SAT solving in the cloud. In Theory and Applications of Satisfiability Testing–SAT 2021: 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings 24 (pp. 518-534). Springer International Publishing.

[62] Sanders, P. and Schreiber, D., 2022. Mallob: Scalable SAT Solving On Demand With Decentralized Job Scheduling. *Journal of Open Source Software*, *7*(76), p.4591.

Algorithm: DynexSolve

```
Input: Computational problem := DynexMallob()
space := parallel Dynex chips fitting on GPU memory
available Dynex chips := DynexMallob(space)
Build Dynex Chip circuit for numerical integration = n
solved := false
for each CHIP do
    initial conditions_CHIP = available Dynex chips
    while solved = false & integration steps_CHIP < n^5 do
        integration steps_CHIP := 0
        init state_hash, state_nonce and state_diff
        pouw_blob := nonce_blockchain + timestamp + pouw_state
        while integration steps < batch-size do
            state_CHIP,n < Adaptive Forward Euler step
            if solved = true then
                return (solved)
            end if
            state_hash, state_diff := lighthash(pouw_blob, state_nonce)
            pouw_blob,loc := localMinima(state)
            integration steps_CHIP := +1
        end while
        DynexMallob(state)
        eligible_counter := pouw_eligable
        hashing_counter := 0
        while hashing_counter < eligible_counter do
            hash := CryptoNight_modified(blocktemplate, nonce)
            if hash_diff > block_diff then
                submit pouw_blob + pouw_hash + nonce
            end if
            hashing_counter := +1
        end while
    end while
end for
```

Initially, DynexSolve **retrieves the definition** of the assigned **computational task** from the Dynex Mallob system, which allows to calculate the total capacity of all parallel Dynex Chips fitting on the connected Graphic-Processing-Units (GPUs), defined as *space*.

Input: *Computational problem :=* DynexMallob()
space := parallel Dynex chips fitting on GPU memory

Given *space*, DynexSolve retrieves the set of assigned Dynex Chips from the Dynex Mallob system which also define the initial conditions for the Dynex chips to compute. Based on that data, DynexSolve **builds the corresponding system** of equations of motions to be numerically integrated.

available Dynex chips := DynexMallob(*space*)
Build *Dynex Chip circuit* for numerical integration = *n*

Every chip is integrated **in parallel** on each of the connected Graphic-Processing-Units with the different initial conditions provided, as long as either a solution was found or the maximum number of integration steps n^5 has been reached.

solved := false
for each *CHIP* **do**
    $initial\ conditions_{CHIP} = available\ Dynex\ chips$
    while $solved = false\ \&\ integration\ steps_{CHIP} < n^5$ **do**

The numerical integration is performed in **batches** (typically 10,000 integration steps per batch per Dynex chip). Every integration step creates a **unique state** for each Dynex chip:

$state_{CHIP,n} < Adaptive\ Forward\ Euler\ step$

The algorithm also performs hashing on the provided $PoUW_{blob}$ as well as the calculation of the current energy landscape (local minima*):*

$state_{hash}, state_{diff}$:= lighthash($pouw_{blob}, state_{nonce}$)
$pouw_{blob,loc} := localMinima(state)$

This ensures that the performed work is **unique** and **accurate**. The current energy landscape given a current state can be verified quickly with a simple function call for any given computational job.

Each initial condition has a different number of integration steps required to reach the lowest possible global energy level of the underlying computational problem, which represents a ***solution***. As soon as a solution to the job has been found, the Dynex Mallob system is being updated, the solution state submitted to Dynex and the job marked as "finalised".

if *solved = true* **then**
    return *(solved)*
end if

The successful PoUW work continuously defines the overall hash-rate of the miner, also determining the number of eligible hashes DynexSolve can use for the blockchain related $\text{CryptoNight}_{\text{modified}}$ hashing function to confirm blocks in the Dynex block chain:

$eligible_{counter} := pouw_{eligable}$
$hashing_{counter} := 0$
while $hashing_{counter} < eligible_{counter}$ **do**
    $hash := \text{CryptoNight}_{\text{modified}}(blocktemplate, nonce)$
    if $hash_{diff} > block_{diff}$ **then**
        submit $pouw_{blob} + pouw_{hash} + nonce$
    end if
    $hashing_{counter} := +1$
end while

Submitted blocks to the Dynex block chain require a **successful verification** from its PoUW data as well as the calculated block nonce itself. Per definition of the algorithm, both are **entangled** and **uniquely connected** to the underlying **computational job,** which guarantees that block nonces can be found only if the entire Proof-of-Useful-Work scheme has been performed with the DynexSolve algorithm.

Mining renumeration scheme

The renumeration scheme for DynexSolve mining consists of the following elements:

(i) ***Block reward***: Miners who are finding a block nonce of a block are receiving the block reward for this block. The block reward follows a smooth emission curve[63].

(ii) ***Transaction fees***: In addition to the block reward, also the transaction fees included in the mined block are being rewarded.

(iii) ***Block fees***: Dynex customers who post and run computation jobs on the Dynex platform can define a block fee they are paying for the computations. The *block fee* is paid for every block which is being mined working on the computational problem. As a) the network hash-rate, b) the maximum complexity and c) the upper bound of required integration steps known at job creation, customers can allocate a pre-defined amount for any job. Multiple jobs can be run in parallel. The *Dynex Mallob system* is assigning highest paid jobs first, then in descending order. The *block fee* is rewarded to the miners similarly as the block reward. This guarantees **continuity and sustainability** of the **business model** for miners, also for the period after all blocks have been mined. **Miners** are automatically receiving **70% of the block fees** for mined blocks (these are automatically being integrated into the transaction fees of each block).

(iv) ***Solution reward***: As an additional motivation for miners, Dynex customers can define a *solution reward* for the miner who completed the computational job first. It is being automatically rewarded to the first miner completing a job.

[63] https://dynexcoin.org/wp-content/uploads/2023/01/Dynex-whitepaper.pdf

Additional Ressources

Main Dynex website:
https://dynexcoin.org/

Dynex Marketplace:
https://live.dynexcoin.org

Dynex Docs:
https://dynexcoin.org/learn

Dynex Node and CLI Wallet:
https://github.com/dynexcoin/Dynex/releases/

Dynex Blockchain Explorer:
https://blockexplorer.dynexcoin.org