Computing with Cellular Automata

Kees van Berkel

MPSoC 2024, July 7-11

Kanazawa, Japan

A 2D Cellular automaton: Game of Life

The Game of Life [Conway, 1970].

- It is based on
  a finite two-dimensional grid of cells.
- Each cell has two states: dead or alive.
- A transition from dead to alive occurs if there are exactly 3 alive neighbors.
- A transition from alive to dead occurs if fewer than 2 or more than 3 neighbors are alive.
- All cells transit in synchrony.

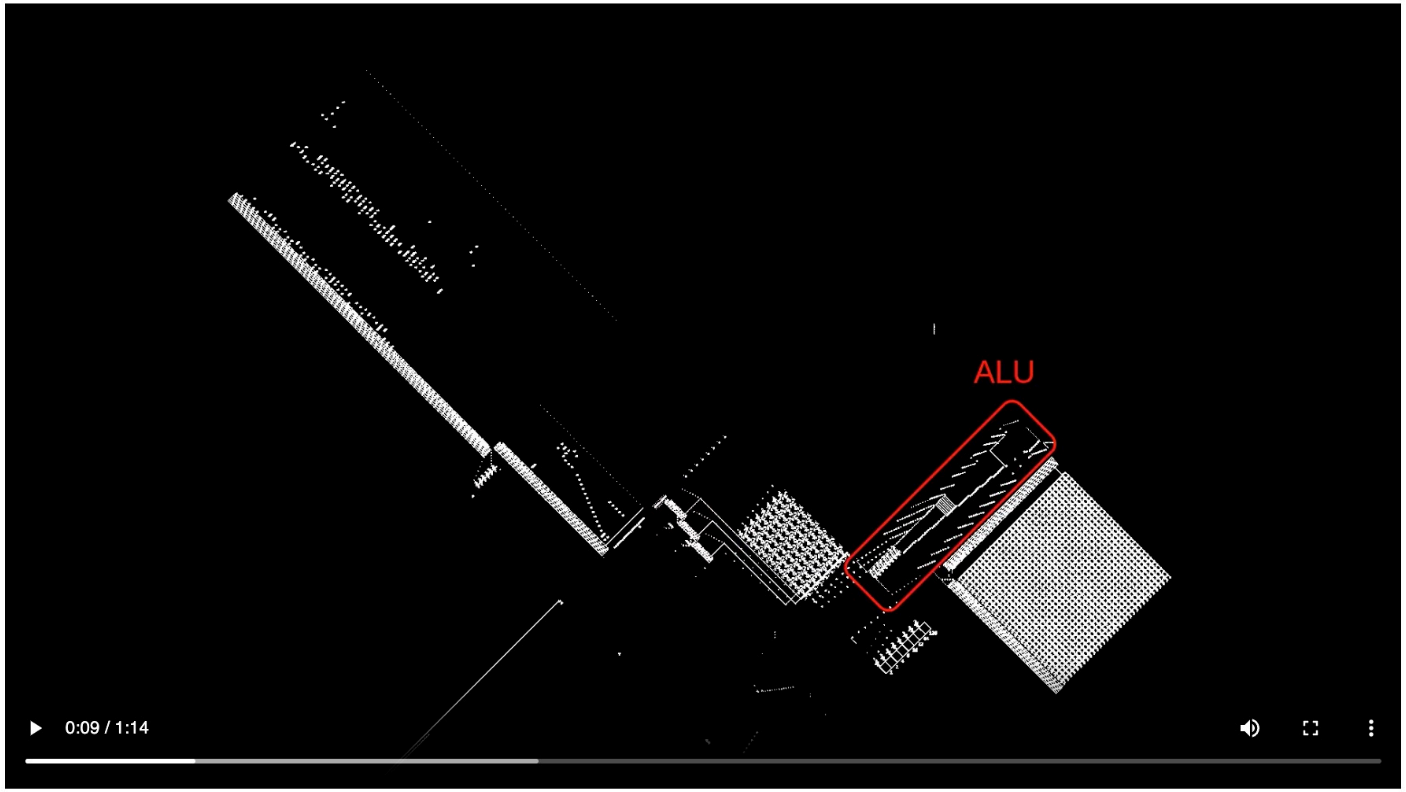
It is an example of a cellular automaton .

The animation shows a Gosper glider gun .

It disproves Conway's original conjecture that no pattern can grow indefinitely. [wikipedia]

A 2D CA: Game of life, a programmable computer

Out[2]:

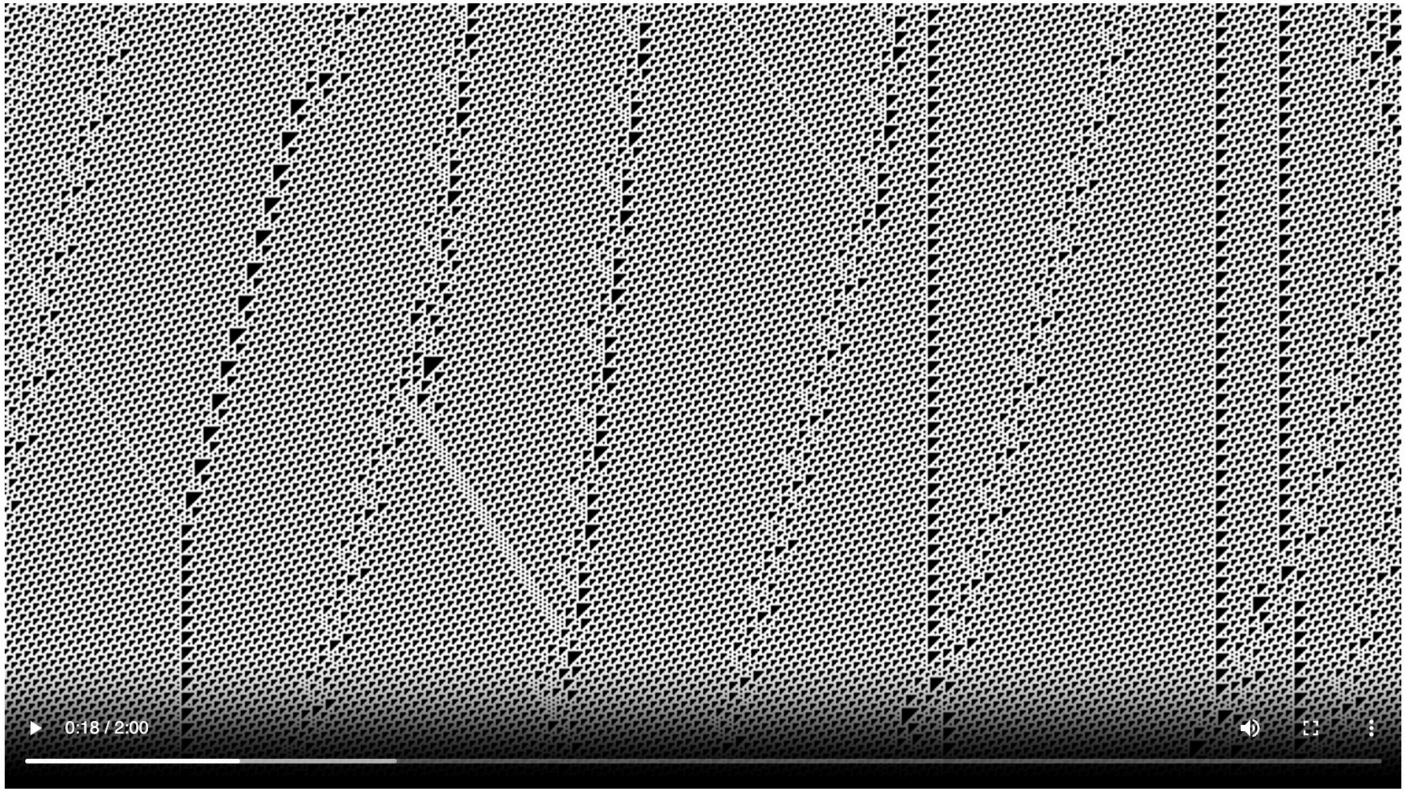


Nicolas Loizeau, 2018, https://www.nicolasloizeau.com/gol-computer.

Paul Rendell built a Turing machine in GoL [2000] and a universal Turing machine [2009].

A 1D cellular automaton: Rule 110

Out[4]:

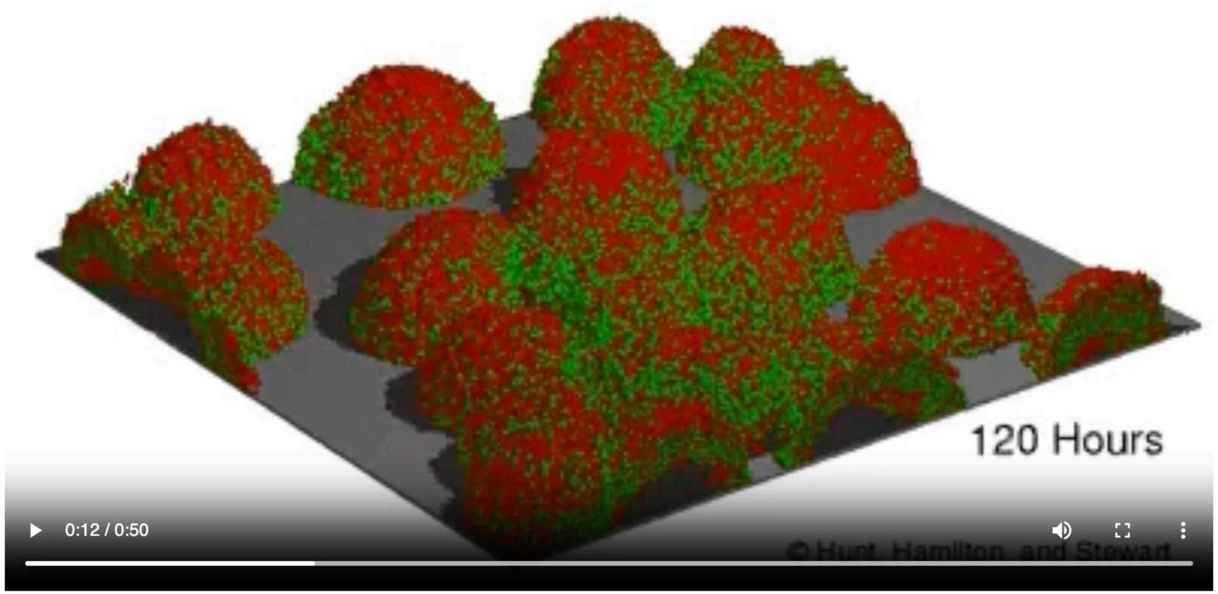


Rule 110 is a 1-dimensional "borderline chaotic" cellular automaton [Wolfram, 2003].

With a particular repeating background pattern it is Turing complete [Cook, 2004].

A 3D cellular automaton: a model of biofilm dynamics

Out[5]:

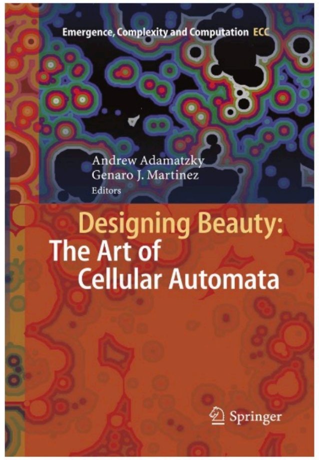


Simulates the response of a microbial biofilm to antimicrobial treatment.
Live cells are shown in green and dead cells in red [Hunt, 2005].

Cellular automata ?

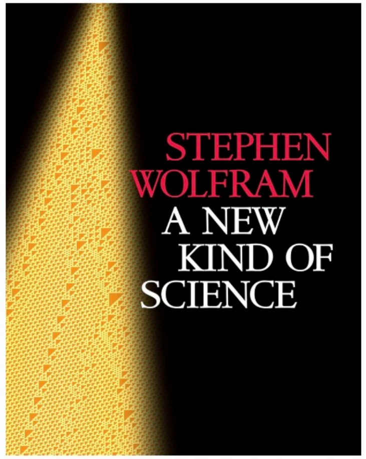
A form of art

[Adamatzky & Martinez, 2016]



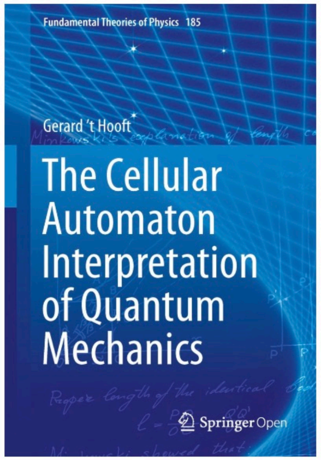
A new kind of science

[Wolfram, 2002]



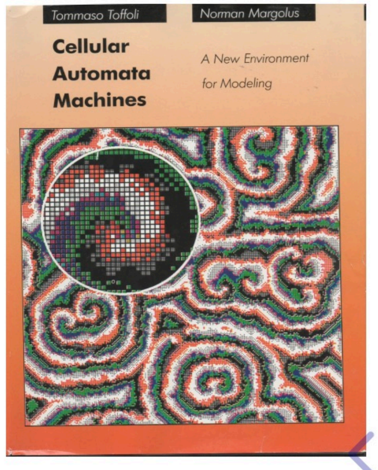
A view on the universe

['t Hooft, 2016]



A model of computation for practical use

[Toffoli & Margolus, 1987]



"Useful" cellular automata: (research) questions

1. In what sense are cellular automata a model of computation?
2. What are (and could be) practical/useful applications of cellular automata?
3. What are typical/potential workloads of such applications?
4. How to exploit the (intrinsic) parallelism of cellular automata?
   What are the limits to these forms of parallelism?
   How well do these forms of parallelism scale?
5. How well would CA run on a typical GPU accelerator?
   What would a dedicated/tailored hardware architecture look like?
6. The 2020s is the decade of accelerators (GPUs, NPUs, quantum computers).
   Could a cellular-automaton accelerator offer a viable path beyond exascale computing?

The CA model of computation: a brief history

- 1940s : Stanislaw Ulam and John von Neumann discover cellular automata , while working on the problem of self-replicating systems.
- 1969 : Konrad Zuse proposes Rechnender Raum: the universe as a cellular automaton.
- 1970 : John Conway discovers the Game of Life .
- 1982 : Richard Feynman suggests to quantize cellular automata, now known as Quantum Cellular Automata .
- 1987 : Norman Margolus proposes block cellular automata , the key to time-reversibility and conservation laws.
- 2004 : Matthew Cook shows that the 1D CA *Rule 110* is Turing complete .
- 2009 : Paul Rendell constructs a Turing machine in the Game of Life.

The CA model of computation: diversity

Cell data type : 1 bit, integer, real, complex, vector of ...

Cell grid : 1D, 2D, 3D,.. (finite/infinite), +optional 1D history.

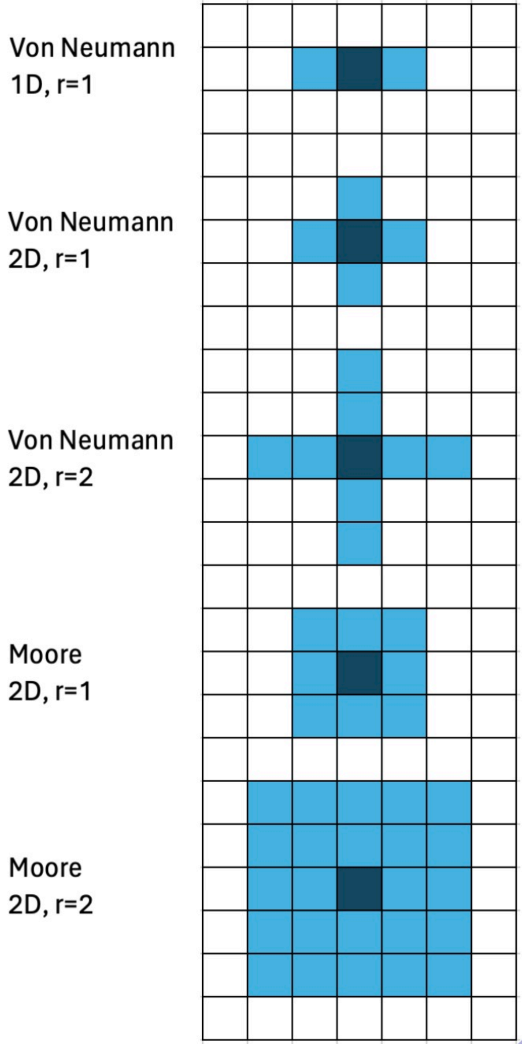
Neighborhood : e.g. Von Neumann/Moore, range. See ⟶

Transition rules : homogeneous vs inhomogeneous,
deterministic vs probabilistic,
synchronous vs asynchronous,
linear vs non-linear.

The standard Game of Life:
1bit, 2D (no history), Moore (r=1), homogeneous, sync., non-linear.

Special tilings: e.g. a 2D tiling with triangular or hexagonal cells,
or a 3D (layered) tiling of a sphere.

Time-reversible cellular automata ("partitioning CA"),
to be discussed later.



The CA model of computation: versus FSM

A finite, synchronous, and deterministic cellular automaton (with discrete cell states) can be viewed as a deterministic FSM.

A deterministic finite-state machine is a quintuple $(\Sigma, S, s_0, \delta, F)$.

- The input alphabet Σ consists of a single symbol τ,
  hence a CA is a so-called "generator FSM".
- The state space S is structured, e.g. $[0, 30) \times [0, 40) \times \{dead, alive\}$.
- The initial state $s_0 \in S$.
- Transition δ = combined effect of all cell transitions.
- The final states $F \subset S$, e.g. the F consists of a single state "all cells dead".

If the CA is also linear then transition δ can be represented by a matrix multiplication.

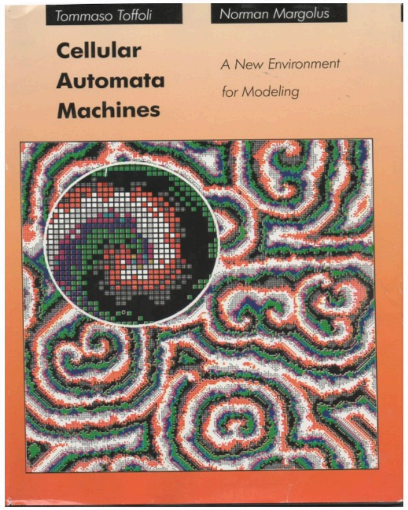
The CA model of computation: key properties

1. Versatile, universal:
   As a model of computation it is Turing complete.
2. Highly regular:
   (Nearly) all cells have the same neighborhood, with possibly (periodic) boundaries.
   All cells have the same (or similar) transition function.
3. Abundantly parallel:
   All cells transit simultaneously.
4. Strictly local:
   The transition function depends on a local neighborhood.

Is this the ideal model of computation for High Performance Computing?

Applications of cellular automata

> *Cellular automata are discrete dynamical systems*
> *whose behavior is completely specified in terms of a local relation,*
> *much as is the case for a large class of continuous dynamical systems*
> *defined by partial differential equations.*
> *In this sense, cellular automata are the computer scientist's*
> *counterpart to the physicist's concept of ``field.''*



Also, book by J. Schiff: *Cellular Automata: A Discrete View of the World.*

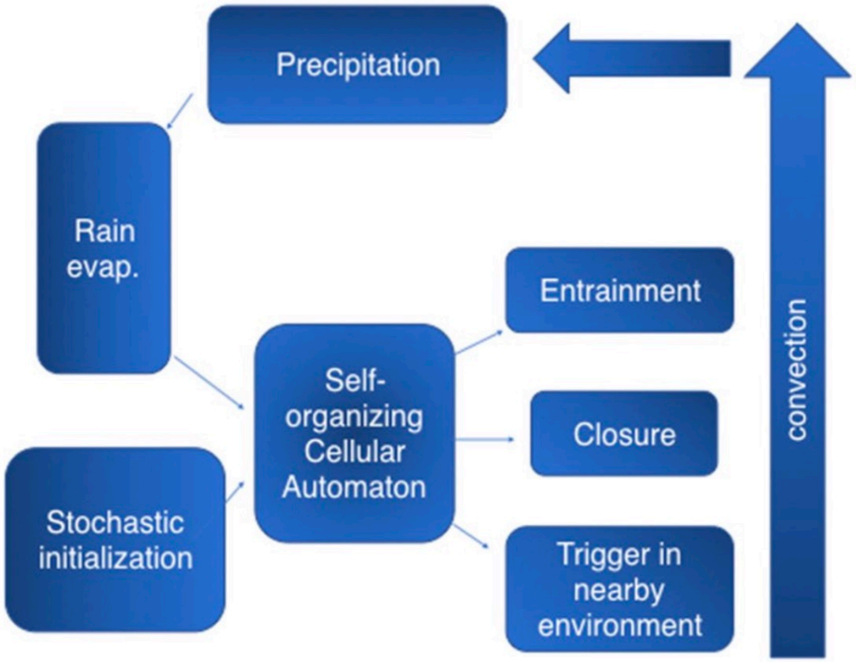
Physical processes :
diffusion, heat flow, lattice gasses, crystal formation, fluid dynamics, spin glasses, ...

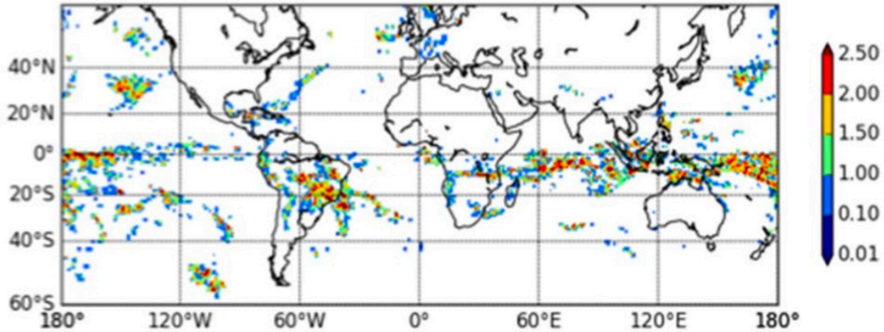
Also [Google]: chemistry, biology, urban planning, weather/climate models, cryptography, ...

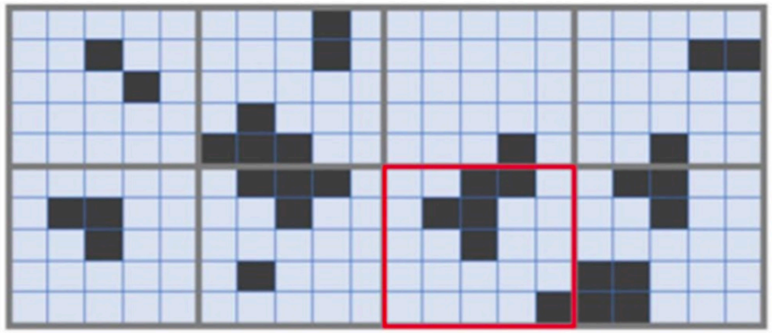
ACRI 2024 : *16th Cellular Automata for Research and Industry conference*, AUTOMATA 2024.

However, the status of actual *deployment* of cellular automata is unclear.
Not many documented examples.

Applications of cellular automata: for global forecasts







This study explores the impact of representing convective organization in weather and climate models using cellular automata. One cell ≈ 1 square degree.

[National Oceanic and Atmospheric Administration, NOAA, Bengtsson, 2020]

Schrödinger Unitary Cellular Automata

Joint work with Jan de Graaf and Kees van Hee. See arXiv 2406.08586 [quant-ph] .

A 1D linear cellular automaton:

- The CA state of N cells is a vector of length N: $\Psi(t)$.
- The CA transition is a multiplication by a matrix $\mathbf{U}$: $\Psi(t+1) = \mathbf{U}\ \Psi(t)$.

In a 1D Schrödinger cellular automaton for a single particle:

- $\Psi(x, t)$ is a complex number, the value of the wave function of cell x at time t.
- Probability density $P(x, t)$ denotes the probability that the particle is in cell x at time t.
- Born rule: $P(x, t) = |\Psi(x, t)|^2$, $P(t) = \sum_x P(x, t) = 1$.

Evolution matrix $\mathbf{U}$ must be:

1. unitary : $\mathbf{U}\mathbf{U}^\dagger = \mathbf{I}$, to preserve $P(t) = 1$.
2. band structured : to support the locality required for cellular automata.

The 1D Schrödinger equation: continuous time and space

The Schrödinger equation is a linear partial differential equation
that governs the wave function Ψ of a quantum-mechanical system.

In 1 dimension, for a single particle:

$$i\hbar \frac{\partial \Psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x,t)}{\partial x^2} + V(x)\Psi(x,t) \ .$$

- $\Psi(x,t)$ is the wave function, $\hbar$ is the Planck constant,
- m is the particle's mass, $V(x)$ is a potential-energy function.

Research question:

- What if the Schrödinger equation is a continuous approximation of a discrete universe?
- What if, e.g. at the Planck scale,
  quantum dynamics occurs on a discrete lattice and in discrete time steps?
- What if the universe is a cellular automaton?

The 1D Schrödinger equation: discrete time and space

In discrete time (step τ) and space (cell size a), ignoring $V(x)$:

$$\begin{aligned} i\frac{\hbar}{\tau}\Big(\Psi(x,t+1)-\Psi(x,t)\Big) &= -\frac{\hbar^2}{2m}\frac{1}{a^2}\Big(\Psi(x+1,t)-2\Psi(x,t)+\Psi(x-1,t)\Big) \\ &= \delta\hat{\mathbf{H}}\,\Psi\,, \end{aligned}$$

Hamiltonian $H=\delta\hat{\mathbf{H}}$, for $N=8$ cells:

$$\delta=\frac{\hbar^2}{2m}\frac{1}{a^2}\,,\qquad \hat{\mathbf{H}}=\begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

The 1D Schrödinger equation: solution

The discrete-time evolution for *integer* time $t, 0 \leq t$ and fixed time step τ

$$|\Psi((t+1)\tau)\rangle \;=\; \mathbf{U}\,|\,\Psi(t\tau)\rangle\;,$$

where

$$\mathbf{U} = \exp\left(-i\theta\hat{\mathbf{H}}\right)\,, \qquad \text{with} \qquad \theta = \frac{\tau}{\hbar}\delta\,,$$

and matrix exponential

$$\exp\left(-\,i\theta\hat{\mathbf{H}}\right) \;=\; \sum_{k=0}^{\infty}\frac{1}{k!}\left(-i\theta\hat{\mathbf{H}}\right)^k\,.$$

Evolution matrix $\mathbf{U}$ must be

1. unitary : $\mathbf{U}\mathbf{U}^\dagger = \mathbf{I}$, to preserve $\sum_x P(x,t) = 1$, and
2. band structured : to support the locality required for cellular automata.

Unfortunately, matrix $\mathbf{U} = \exp\left(-i\theta\hat{\mathbf{H}}\right)$ is dense: all its elements are nonzero.

The 1D Schrödinger equation: split evolution

Let Hamiltonian $\hat{\mathbf{H}} = \hat{\mathbf{H}}_0 + \hat{\mathbf{H}}_1$, where

$$\hat{\mathbf{H}}_0 = \mathbf{I}_m \otimes \mathbf{B}\,, \qquad \mathbf{B} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \qquad \hat{\mathbf{H}}_1 = \mathbf{S}^{-1}\hat{\mathbf{H}}_0\,\mathbf{S}\,.$$

Here $\otimes$ denotes the Kronecker matrix product, $2m = N$,
and matrix $\mathbf{S}$ is the so-called *circular shift* matrix. Furthermore, let

$$\mathbf{U}_0 = \exp\left(-i\theta\hat{\mathbf{H}}_0\right), \qquad \mathbf{U}_1 = \exp\left(-i\theta\hat{\mathbf{H}}_1\right).$$

Then

$$\begin{aligned} \exp\left(-i\theta\hat{\mathbf{H}}\right) &= \exp\left(-i\theta(\hat{\mathbf{H}}_0 + \hat{\mathbf{H}}_1)\right) \\ &= \exp\left(-i\theta\hat{\mathbf{H}}_1\right)\exp\left(-i\theta\hat{\mathbf{H}}_0\right) + \mathcal{O}(\theta^2) \\ &= \mathbf{U}_1\mathbf{U}_0 + \mathcal{O}(\theta^2)\,. \end{aligned}$$

State $\Psi(x,t)$ can be evolved to $\Psi(x,t+1)$ by multiplication with $\mathbf{U}_1\mathbf{U}_0$.

Matrix $\mathbf{U} = \mathbf{U}_1\mathbf{U}_0$ is both unitary and band structured , so are $\mathbf{U}_0$ and $\mathbf{U}_1$.

The 1D Schrödinger equation: split evolution

$$\mathbf{U}_1 = \exp(-i\theta) \times \begin{bmatrix} \cos(\theta) & 0 & 0 & 0 & 0 & 0 & 0 & i\sin(\theta) \\ 0 & \cos(\theta) & i\sin(\theta) & 0 & 0 & 0 & 0 & 0 \\ 0 & i\sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & i\sin(\theta) & 0 & 0 & 0 \\ 0 & 0 & 0 & i\sin(\theta) & \cos(\theta) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos(\theta) & i\sin(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & i\sin(\theta) & \cos(\theta) & 0 \\ i\sin(\theta) & 0 & 0 & 0 & 0 & 0 & 0 & \cos(\theta) \end{bmatrix}.$$

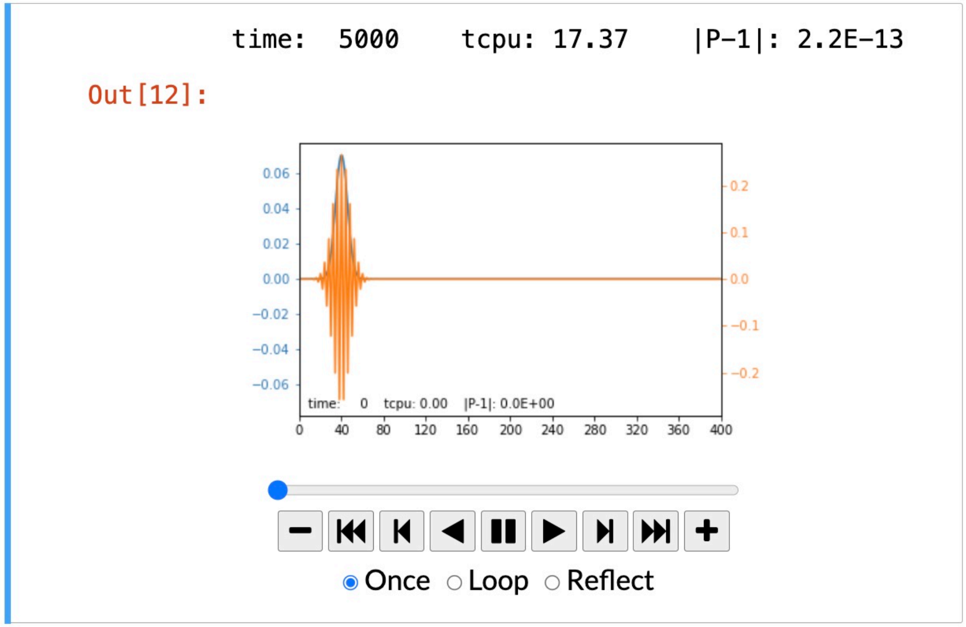
Note: elements $\mathbf{U}_1[0,7]$ and $\mathbf{U}_1[7,0]$ are nonzero $\Leftrightarrow$ periodic boundary conditions .

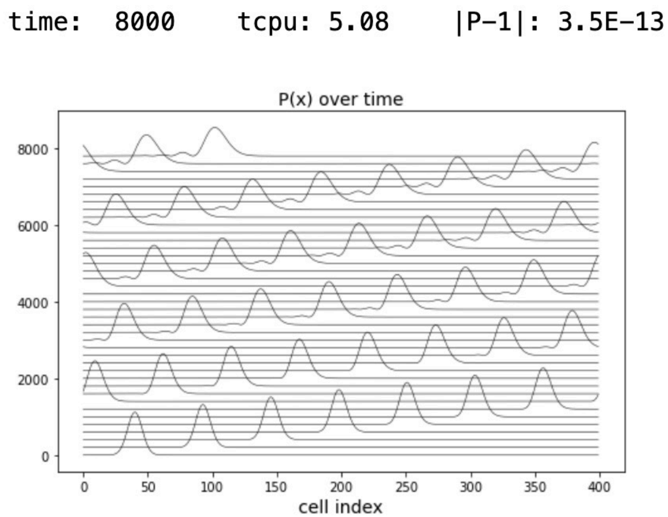
This split evolution yields a so-called partitioning cellular automaton
a.k.a. a block cellular automaton [Toffoli & Margolus, 1987, pp 119-120].
These are *reversible* in time.

A 1D Schrödinger UCA

The cellular automaton consists of 400 cells, and has periodic boundaries.

The intial state $\Psi(x, 0)$ is a wavepacket .





The measured *group velocity* is ≈ 0.26 cells per cycle.
After thousands of cycle, the *dispersion* of the wavepacket becomes visible.

A 2D Schrödinger UCA

Let unitary evolution matrices $\mathbf{U}_H$ and $\mathbf{U}_V$ denote two homogeneous one-dimensional CA, where $\mathbf{U}_H$ and $\mathbf{U}_V$: same {particle mass m, cell size a, time step τ}.

Kronecker product $\mathbf{U}_H \otimes \mathbf{U}_V$ defines a homogeneous two-dimensional cellular automaton :

$$\mathrm{vec}\Big(\Psi(t+\tau)\Big) = (\mathbf{U}_H \otimes \mathbf{U}_V)\ \mathrm{vec}\Big(\Psi(t)\Big)\ .$$

Vector $\mathrm{vec}(A)$ = stack the columns of matrix A on top of one another.

A two-step execution: $\mathbf{U}_H \otimes \mathbf{U}_V = (\mathbf{I} \otimes \mathbf{U}_V)(\mathbf{U}_H \otimes \mathbf{I})$.

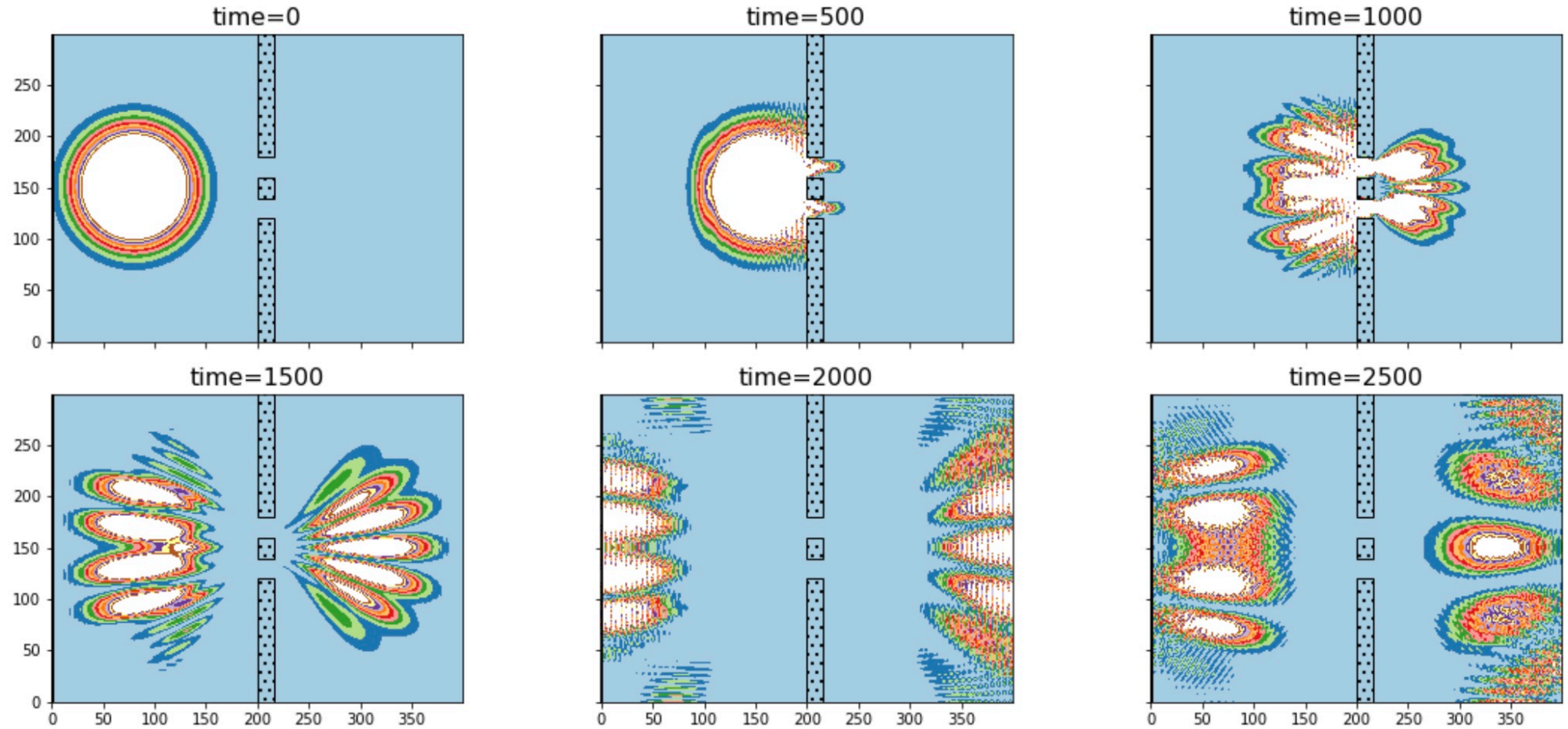
1. apply $\mathbf{U}_H$ to all rows of matrix $\Psi(t)$, with Ψ' as result.
2. apply $\mathbf{U}_V$ to all columns of matrix Ψ', with $\Psi(t+\tau)$ as result.

With split evolution:

$$\begin{aligned}
\mathbf{U}_H \otimes \mathbf{U}_V &= \Big(\mathbf{U}_{H,1}\mathbf{U}_{H,0}\Big) \otimes \Big(\mathbf{U}_{V,1}\mathbf{U}_{V,0}\Big) && \text{(used for experiments)} \\
&= \Big(\mathbf{U}_{H,1} \otimes \mathbf{U}_{V,1}\Big) \cdot \Big(\mathbf{U}_{H,0} \otimes \mathbf{U}_{V,0}\Big) && \text{("Margolus neighborhood")}\ .
\end{aligned}$$

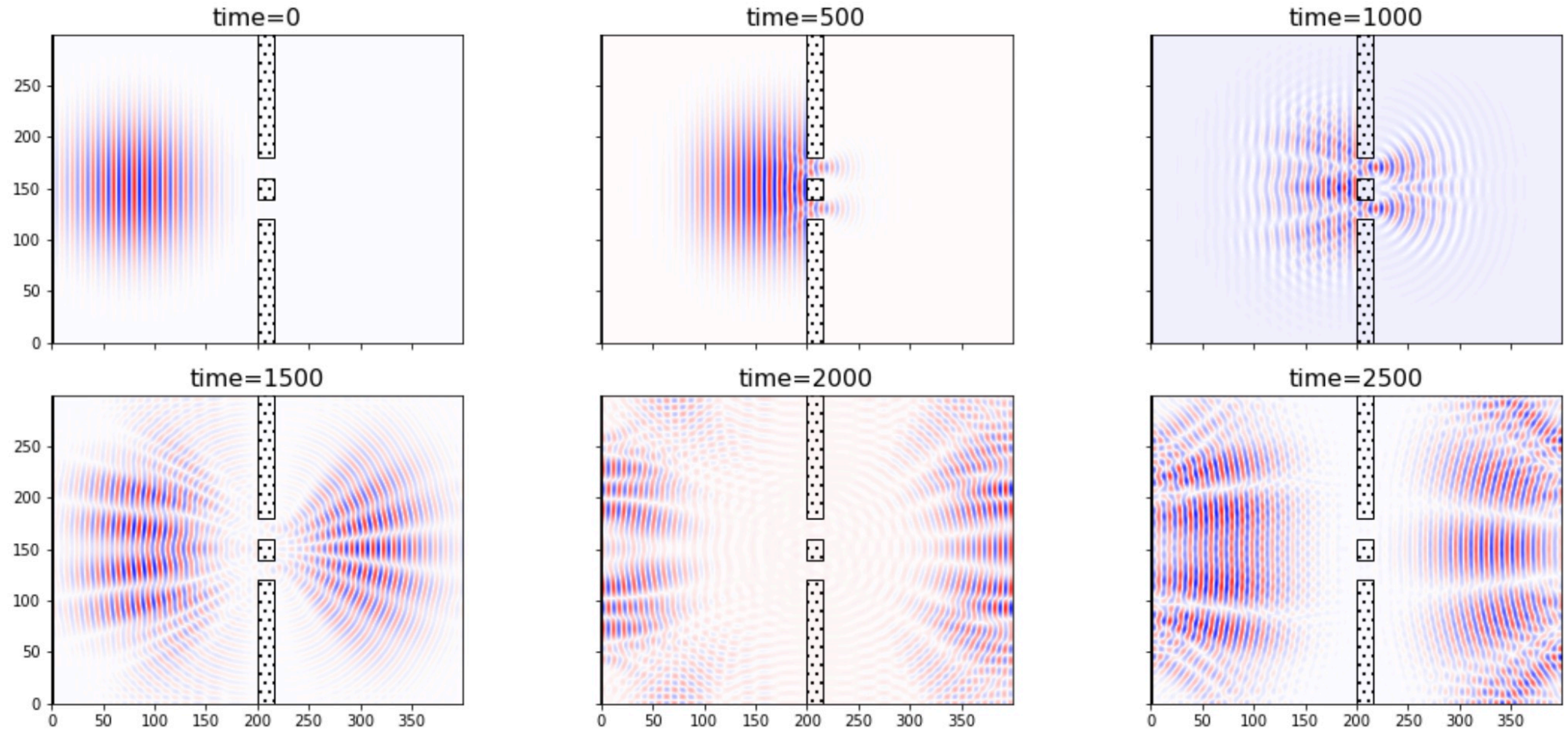
A 2D Schrödinger UCA: double-slit experiment

Single-particle double-slit interference. Feynman: ``*a phenomenon which is impossible, absolutely impossible, to explain in any classical way, and which has in it the heart of quantum mechanics*''.



Probability density $P(x, y, t) = |\Psi(x, y, t)|^2$.
See also https://www.youtube.com/watch?v=Igv0igKdDJg.

A 2D Schrödinger UCA: double-slit experiment



$Re(\Psi)$: red for positive, blue for negative value.

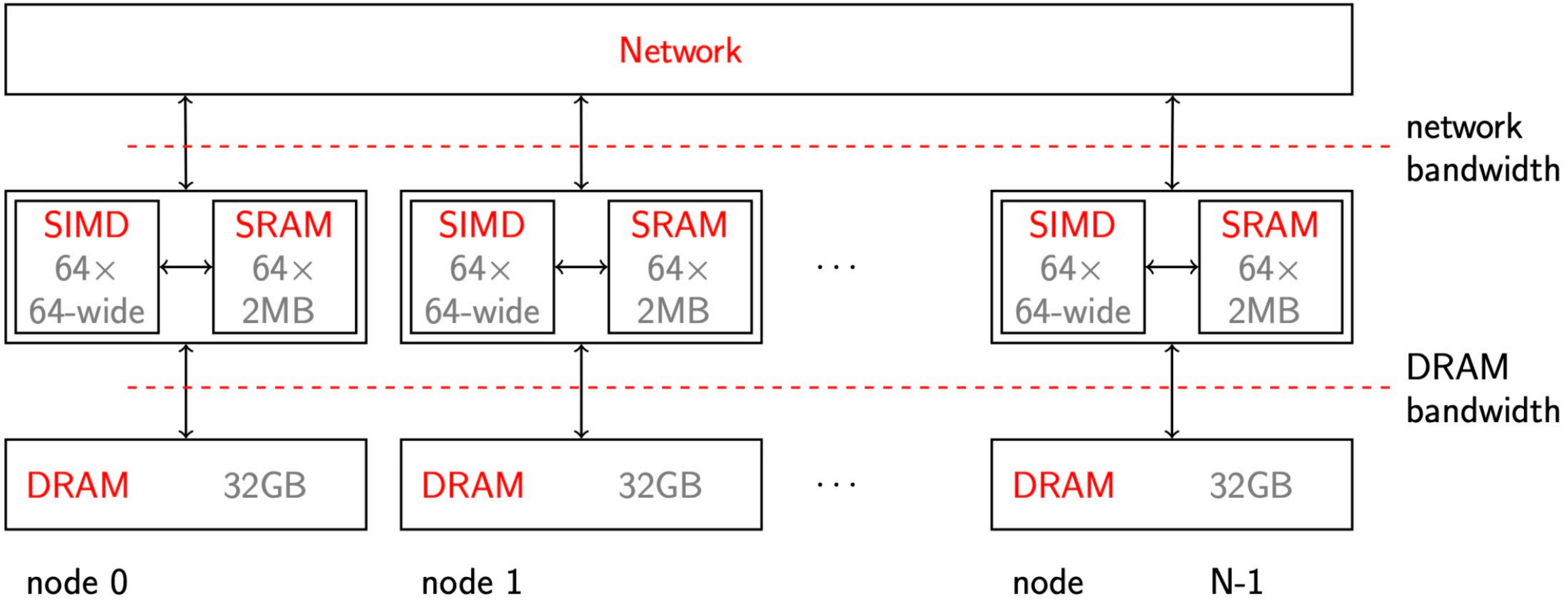
Schrödinger UCA

- The discretization of space causes spatial aliasing of Ψ, with intriguing effects.
  E.g., phase and group velocities are periodic in wavenumber k.
- Next: Klein Gordon equation and Dirac equation, for relativistic behavior and spin?
  Next-next: Quantum Field Theory (QFT) for multiple particles?
- UCA and QCA: a new tool for quantum-physical experiments?
  Pure speculation: ultimately, a "Virtual Hadron Collider"?
- The requirements for cellular-automata computing are currently a bit of guesswork:
  - a 3D cellular automaton: $(16k)^3$ cells $\times$ 1M cycles?
  - $\approx$ hundred 64b FLOPS per cell per cycle?

State $\Psi(x, y, z)$ is measured in many tens of TB
and the compute load in many Peta FLOPs.

... exascale computing .

High performance CA computing



@1.25GHz: 1 node delivers
$\approx$ 10 TFLOPS FP64 *peak performance.*

Scalable to many 1000s of nodes.

Goal: schedule large cellular automata:

- ▱ high SIMD utilization, $\gg$ HPCG 3%
- ▱ low network bandwidth, $\ll$ InfiniBand
- ▱ low DRAM bandwidth, $\ll$ 5x HBM3

Highly parallel CA-evolution: schedules

Case study: 2D double-slit experiment, 16k×16k cells. (The findings are more general.)

Partition the CA cells over a 2D grid of macrocells , one 256×256 macrocell per SIMD unit.

A schedule is a (structured) sequence CA blocks (2×1 or 1×2 cell-pairs):

- sequential: $((x_0, y_0), (x_1, y_1))^*$,
- SIMD parallel: $\left(((x_0, y_0), (x_1, y_1))^{64}\right)^*$,
- machine parallel: $\left(\left(((x_0, y_0), (x_1, y_1))^{64}\right)^{64N}\right)^*$.

Constraints:

1. The schedule (CA-block order) must respect e.g. $\left(\mathbf{U}_{H,1} \otimes \mathbf{U}_{V,1}\right) \cdot \left(\mathbf{U}_{H,0} \otimes \mathbf{U}_{V,0}\right)$.
2. The 2 cells in each CA block of the schedule must "live in the same time zone".

Macrocell boundaries: if a cell-pair is split over two different machine nodes (different DRAMs) then the cell states must be shared, across the network.

Highly parallel CA-evolution: multiple passes per iteration

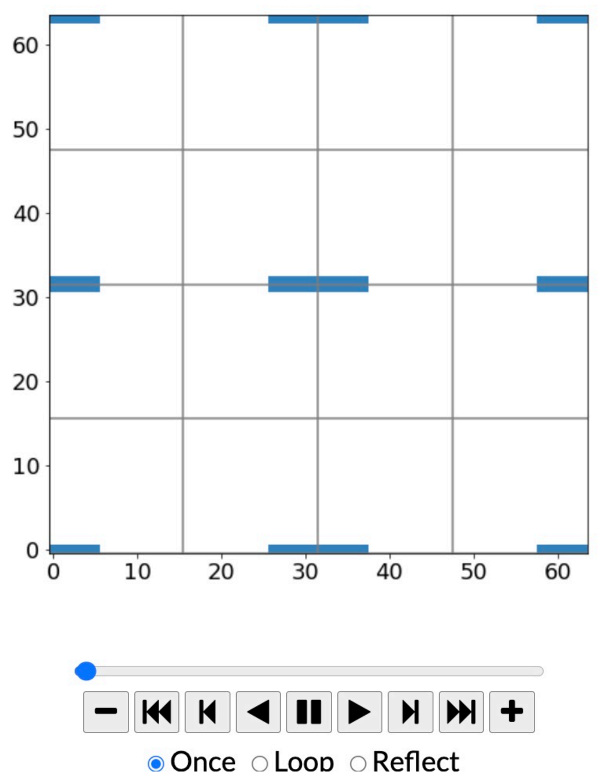
$\left(\mathbf{U}_{H,1} \otimes \mathbf{U}_{V,1}\right) \cdot \left(\mathbf{U}_{H,0} \otimes \mathbf{U}_{V,0}\right)$:

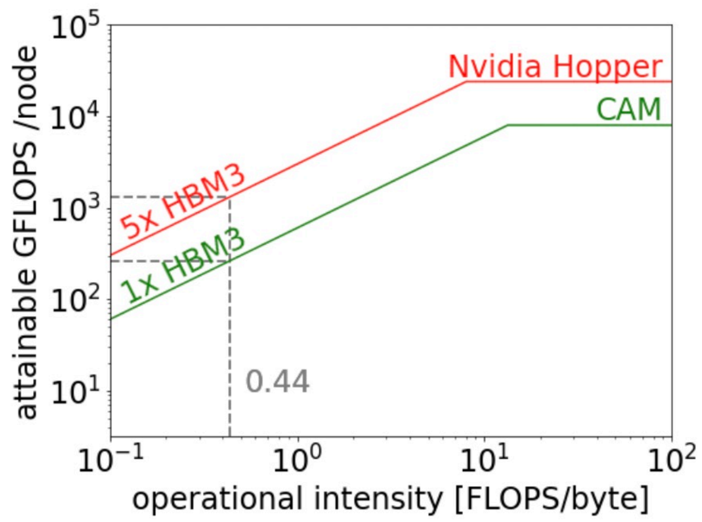
4 *DRAM passes* per 1 iteration .

Operational intensity OI:

= #operations / 1 byte-DRAM-access.

Out[67]:





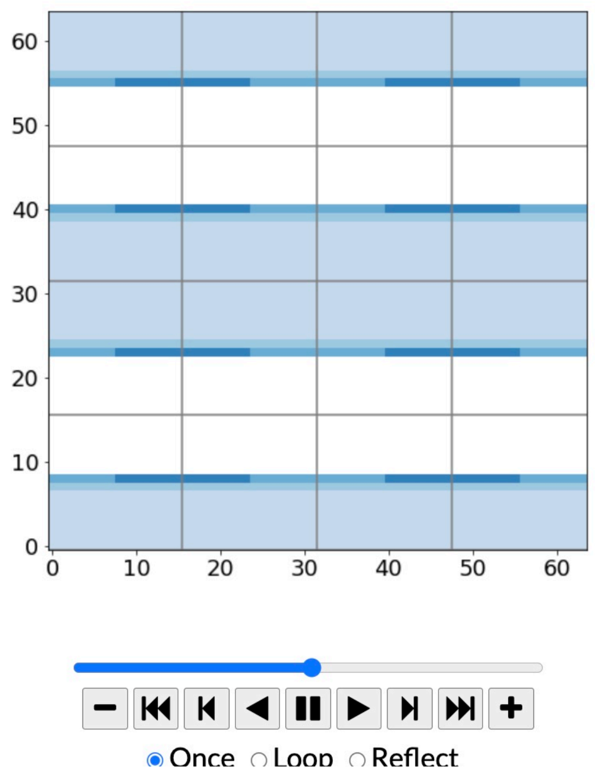
$\mathrm{OI} \;=\;$ 28 FP64 ops / 64 B $\;\approx\; 0.44$

Highly parallel CA-evolution: multiple iterations per pass

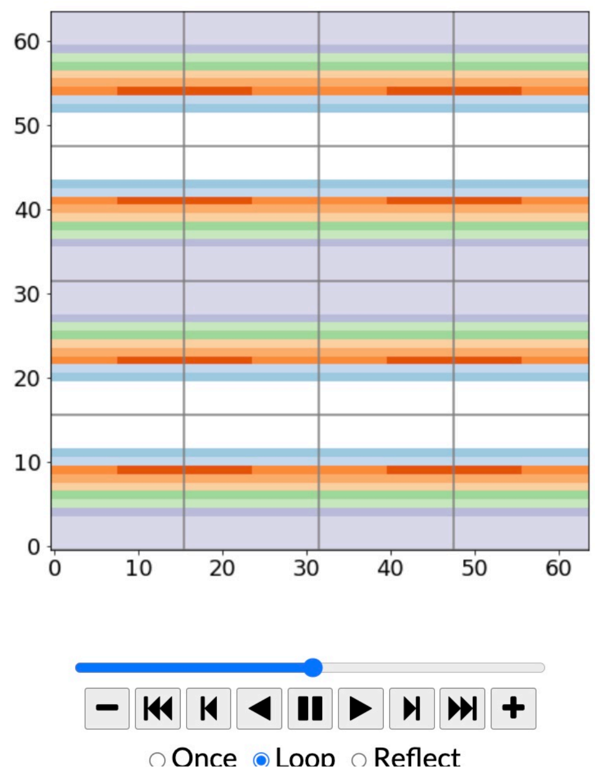
$\left(\mathbf{U}_{H,1}\mathbf{U}_{H,0}\right) \otimes \left(\mathbf{U}_{V,1}\mathbf{U}_{V,0}\right)$, incremental:
$I_{PP} = 1$ iteration per *DRAM pass*.

$\left(\mathbf{U}_{H,1}\mathbf{U}_{H,0}\right) \otimes \left(\mathbf{U}_{V,1}\mathbf{U}_{V,0}\right)$, incremental[4]:
$I_{PP} = 4$ iterations per *DRAM pass*.

Out[69]:



Out[72]:



Highly parallel CA-evolution

Intra-macrocell parallelism, SIMD, e.g. 64-wide.

- Store "wavefront" in local SRAM to reduce DRAM bandwidth.
- A high wave front $\Rightarrow$ high operational intensity OI. $I_{PP} = 16 \Rightarrow \mathrm{OI} = 28$.
- (GPUs use available SRAM mostly for register files and L2 cache, limiting the OI).

Inter-macrocell parallelism, both inside a node and across nodes.

- Neighbor boundary cells must be kept in sync.
- There is ample room for network-latency hiding.

Operational intensity OI:

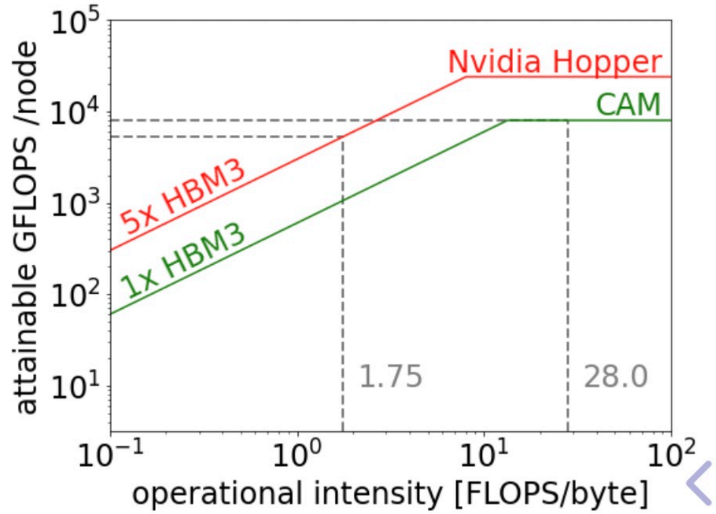
$= 1.75 \times I_{PP}$ (iterations / *DRAM-pass*)

Local SRAM needed to store the "wavefront":

$= 2I_{PP}$ cell rows per macrocell

$= 64 \times 2I_{PP} \times 256 \times 16$ Byte.

$= \frac{1}{2} I_{PP}$ MB.



Highly parallel CA-evolution: a 3D usecase in numbers

Usecase: Schrödinger UCA, 3 dimensions, $(16k)^3$ cells, 1M cycles

Memory view, assuming machine size N =16k nodes:

	SIMD unit	node	machine	note
cells	64×64×1k = 4M	256M	$16k^3$ = 4T	macrocell: $Z = 16X$ to fit SRAM
macro cells	1	64	1M	N = 16k nodes
SRAM	2MB	128 MB	2TB	state of wavefronts
DRAM		4 GB	64TB	state of cellular automaton

Time view, assuming no DRAM bottleneck (sufficient SRAM for 16x wavefront):

	FP64 ops	cycles	time	note
per cell-pair update	28			2x2-matrix × vector, complex
per cell /iteration	84	84		50% FMA utilization
per 64 cells /iteration		84		SIMD
per macro cell /iteration		6M	5 msec	1.25GHz
... 16 k nodes wide			5 msec	assumes network-latency hiding
per run of 1M iterations			1.5 h	

Conclusion

1. Cellular automata (CA) as a model of computation:
   versatile (universal), highly regular, abundantly parallel, and strictly local.
2. However, there are not many compelling CA practical applications, yet.
   Conjecture:

   $$\text{compelling CA applications} \iff \begin{cases} \text{compelling CA benefits} \\ \text{powerful CA tools + libraries} \end{cases}$$

3. Candidates for compelling CA applications include
   (quantum) physical processes, chemistry, and weather/climate modelling.
4. Compelling CA benefits include, potentially 10x flops/$ and 10x flops/W, and scalability.
   These stem from: high PU utilization, low DRAM bandwidth, low network bandwidth.
5. Needed: powerful CA tools + libraries, free and open-source:
   for describing, analyzing, interpreting, mapping, scheduling, ..., CA,
   not unlike TensorFlow and PyTorch for machine learning and artificial intelligence.