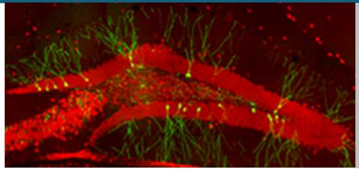
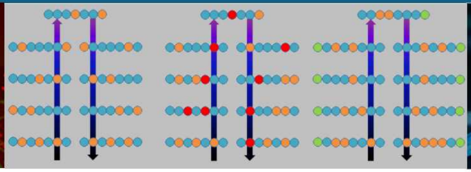
SAND2020-7022C

Solving a Steady-State PDE using Spiking Networks and Neuromorphic Hardware





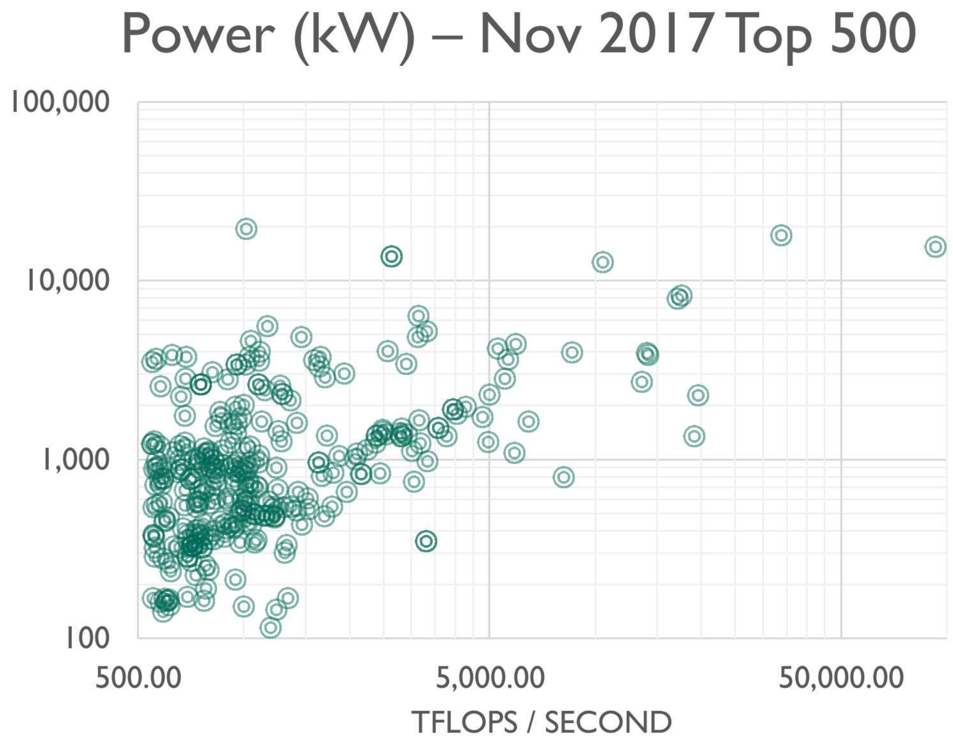
Presented by

J. Darby Smith, William Severa, Aaron Hill, Leah Reeder, Brian Franke, Richard Lehoucq, Ojas Parekh, Brad Aimone

Sandia National Laboratories is a multimission laboratory managed and op Technology & Engineering S LLC, a wholly owned subsic International Inc., for the U Energy's National Nuclear Se under contract DE-N

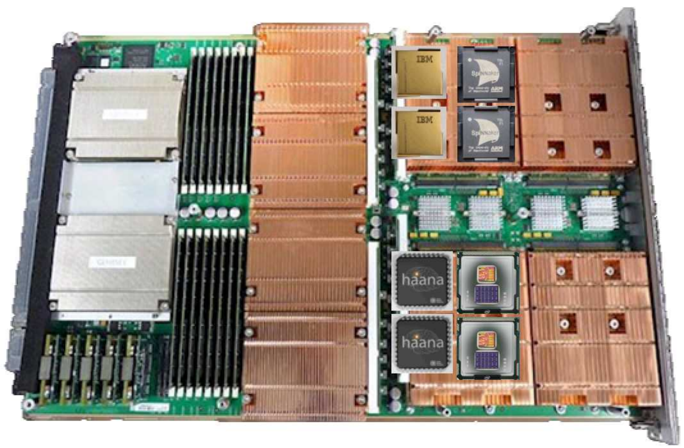
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The Cost of High Performance Computing



- Supercomputers are increasingly limited by power consumption.
- Exascale systems are forecast to be ~100MW
- Neuromorphic could be a radical advance for HPC.

Beyond Machine Learning/Deep Learning



Imagine fully integrated neuromorphic chips on HPC platforms.

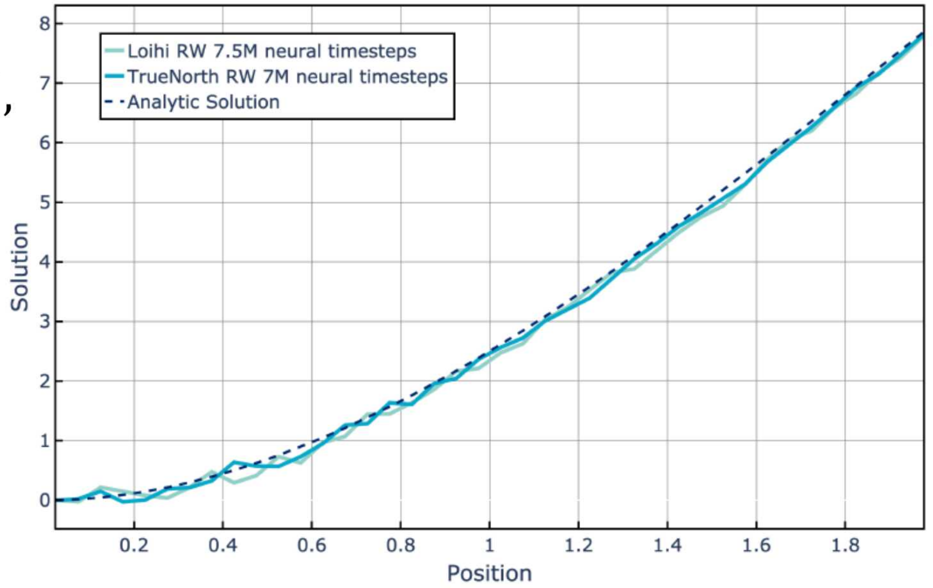
Biological-inspired neural algorithms

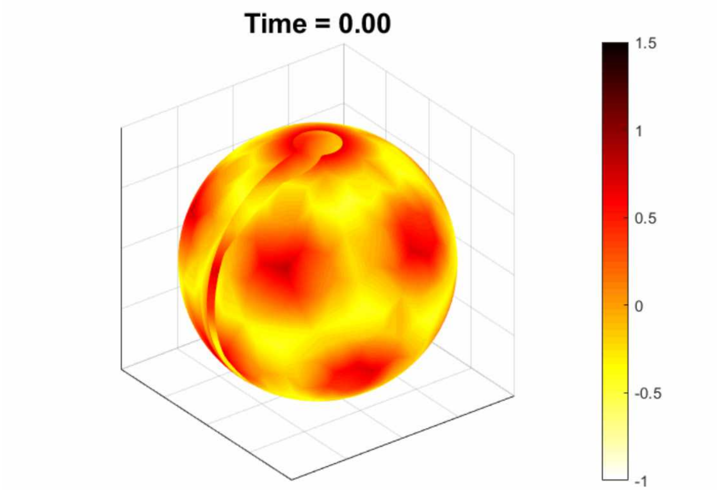
Machine Learning / Deep Learning

Neural-implemented numerical and scientific computing

Leveraging Spiking Neuromorphic to Solve Differential Equations

$$0 = \frac{\mathrm{d}}{\mathrm{d}x^2} u - F(\ell - x), \qquad x \in [0, \ell],$$
$$u(0) = 0,$$
$$u'(0) = 0.$$





Loihi Approximate Solution

$$\frac{\partial}{\partial t} u(t, x, y, z) = \alpha \nabla u(t, x, y, z), \qquad (x, y, z) \in \mathcal{S}^2$$
$$u(0, x, y, z) = g(x, y, z)$$

Random Walks and the Heat Equation

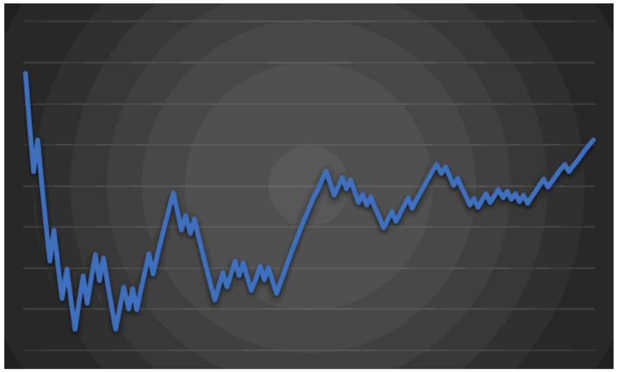
- Random walk methods for PDEs stem from the heat equation:

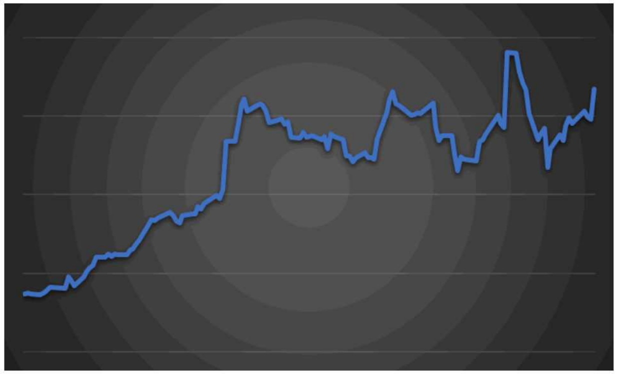
$$\frac{\partial}{\partial t}u = \frac{1}{2}\frac{\partial^2}{\partial x^2}u, \qquad u(x,0) = f(x)$$

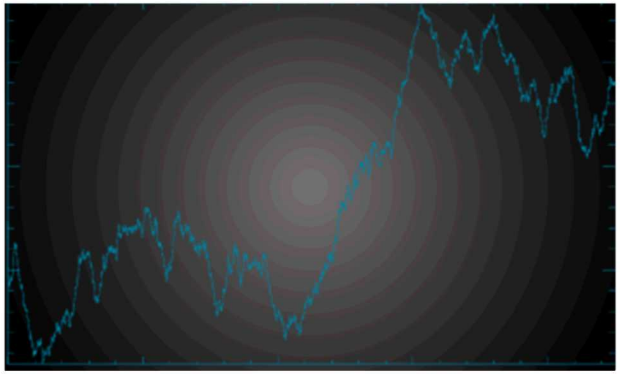
- Let W_t be a standard Brownian motion. Then W_t is normally distributed and

$$\mathbb{E}[f(W_t)|W_0 = x] = \frac{1}{\sqrt{2\pi t}}\int f(y)\exp\left(-\frac{(y-x)^2}{2t}\right)\mathrm{d}y\,.$$

This is the solution to the heat equation.





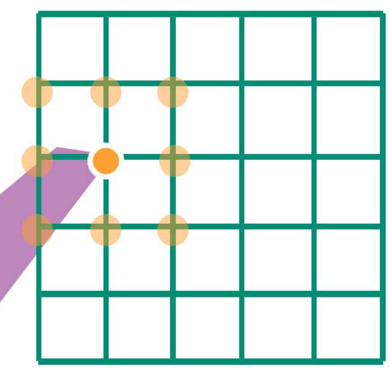


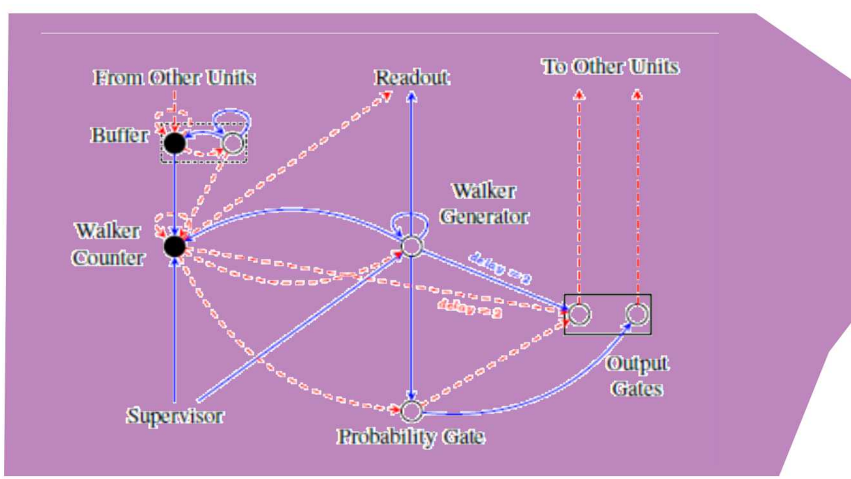
Spiking Networks and Random Walks

- In previous work (Severa et al. 2018) we efficiently implemented a density-based approach to calculating random walks.
  - Each vertex encodes density of particles in the internal potential of certain nodes
  - Each time step "hands off" particles to connected vertices according to probabilistic maps

Density Method

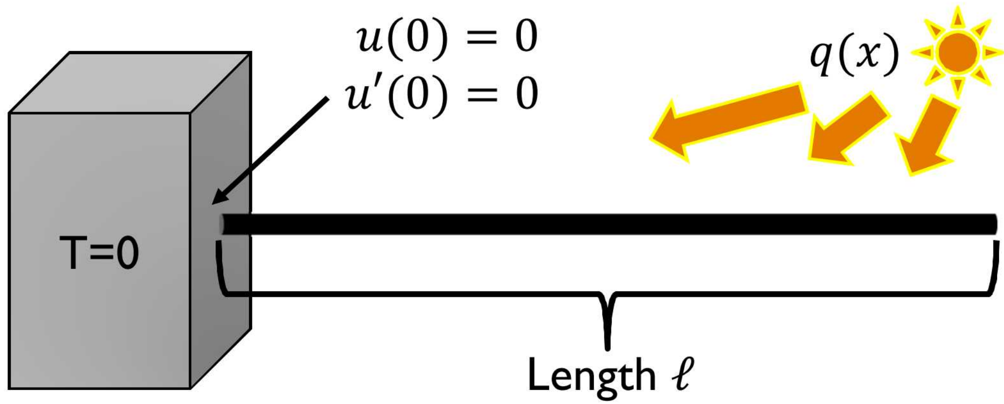
Circuit per position





Measure	Cost (for k locations, simulating N walkers; 1-D case)
Walker memory	O(1)
Connection memory	O(k)
Total neurons	O(k)
Time per physical timestep	O(max(ρ_i)), where ρ_i is the density of walkers at each location
Position energy per timestep	O(N)
Update energy per timestep	O(N)

A Specific Problem: Steady-State Heat Conduction



- Taking $q(x) = -F(\ell - x)$, the situation can be described by the PDE

$$0 = \frac{\mathrm{d}}{\mathrm{d}x^2} u - F(\ell - x), \qquad x \in [0, \ell],$$
$$u(0) = 0,$$
$$u'(0) = 0.$$

- This problem has an easy to obtain analytic solution. Can it be solved via random walks on spiking networks?

$$u(x) = \frac{F\ell}{2} x^2 - \frac{F}{6} x^3$$

A Probabilistic Solution for a Class of Steady-State PDEs

- Under certain conditions and assumptions (Grigoriu 2013), the PDE

$$0 = \sum_{i=1}^{d} \alpha_i(\boldsymbol{x}) \frac{\partial u(\boldsymbol{x})}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^{d} \beta_{ij}(\boldsymbol{x}) \frac{\partial^2 u(\boldsymbol{x})}{\partial x_i \partial x_j} + p(\boldsymbol{x}), \qquad \boldsymbol{x} \in D \subset \mathbb{R}^d$$

$$u(\boldsymbol{x}) = \xi(\boldsymbol{x}), \qquad \boldsymbol{x} \in \partial D.$$

has local solution given by

$$u(\boldsymbol{x}) = \mathbb{E}\left[\xi\big(\boldsymbol{X}(T)\big) + \int_0^T p\big(\boldsymbol{X}(s)\big)\mathrm{d}s \,\middle|\, \boldsymbol{X}_0 = \boldsymbol{x}\right],$$

$$d\boldsymbol{X}_t = \boldsymbol{\alpha}(\boldsymbol{X}_t)dt + \boldsymbol{\sigma}(\boldsymbol{X}_t)d\boldsymbol{W}_t,$$

$$T = \inf\{t > 0 | X_t \notin D\}.$$

- This particular result doesn't apply directly to our chosen problem because:
  - Our boundary condition is only specified for a part of our boundary ($u(0) = 0$). We do not specify anything for $x = \ell$.
  - We have a mixed condition and require $u'(0) = 0$ as well.

A Probabilistic Solution for Our Steady-State Equation

- Absorb $u'(0) = 0$ into the random process.
  - Make X_t a reflective process at zero.
- After making zero reflecting, we can't evaluate $\mathbb{E}[\xi(\boldsymbol{X}(T))|\boldsymbol{X}_0 = \boldsymbol{x}]$.
  - We only know the value of ξ at zero, and the process will no longer exit at zero.
- We still must enforce $u(0) = 0$.
  - Define

$$u_0^* = \mathbb{E}\left[-\int_0^T F(\ell - X(s))\mathrm{d}s \middle| X_0 = 0\right].$$

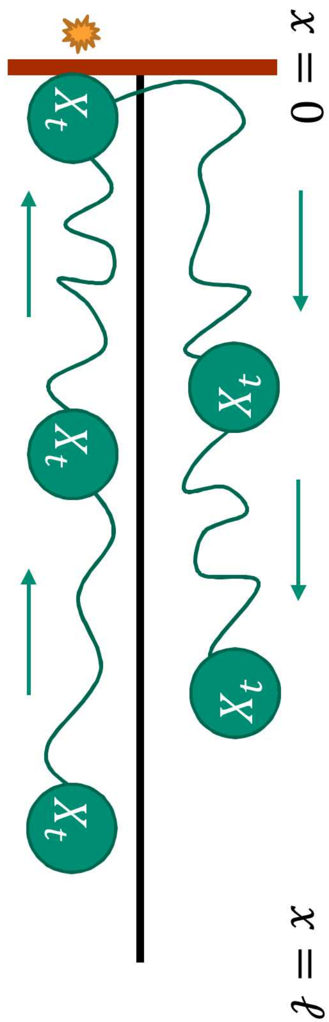
  - Then, the probabilistic solution to our heat transport equation is

$$u(x) = \mathbb{E}\left[-\int_0^T F(\ell - X(s))\mathrm{d}s \middle| X_0 = x\right] - u_0^*,$$

  where X_t is a process reflecting at zero and has law

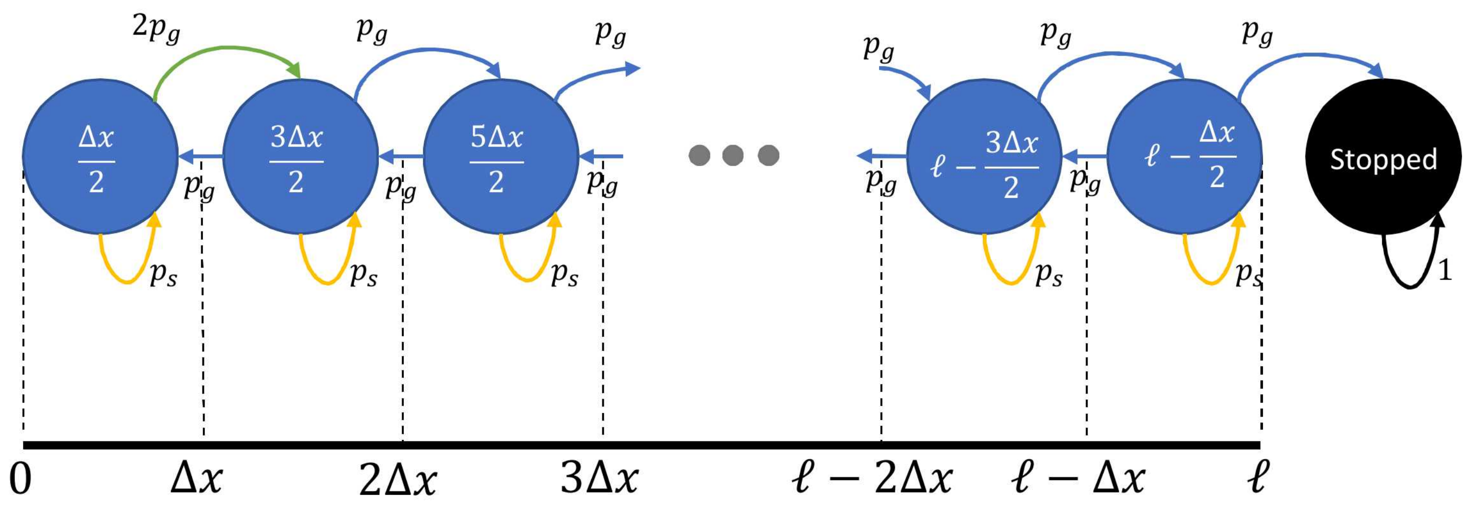
$$dX_t = \sqrt{2}\mathrm{d}W_t$$

  elsewhere.



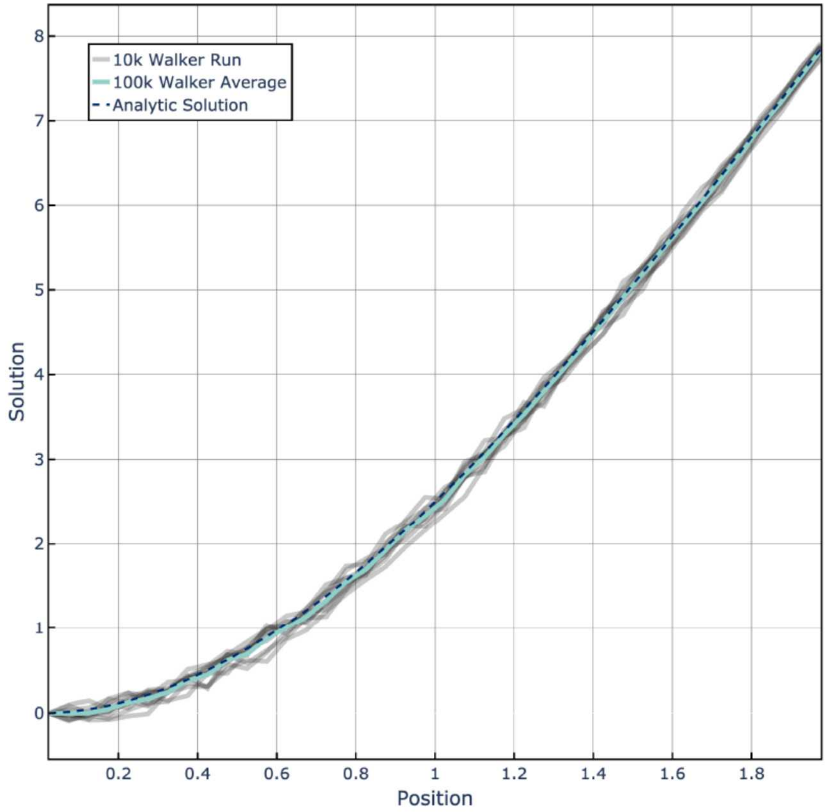
Defining a Random Walk from X_t

- To define a random walk from the process X_t, we:
  - Choose a spatial discretization unit Δx;
  - Choose a time discretization size Δt so that we can be reasonably sure the process X_t will not move further than $3\Delta x/2$ to the left or right during the increment;
  - Calculate the transition probabilities, taking care when dealing with the reflective boundary at zero.
- Random walks are assumed to occupy the midpoint of the divisions.
- Random walks must continue until they reach the absorption position.



Calculating $u(x)$ from samples of X_t

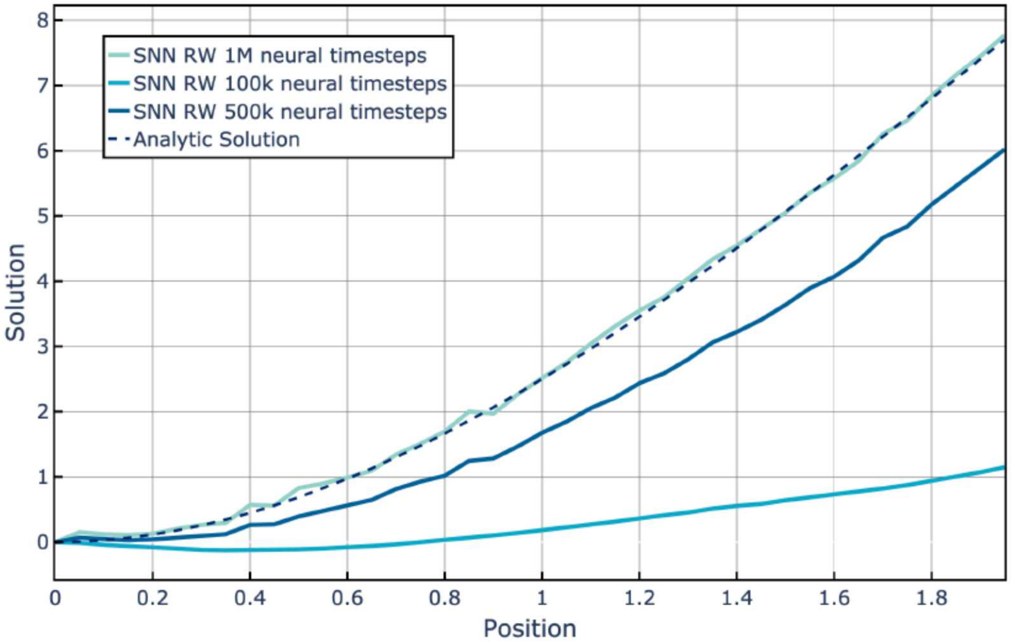
1) For a position x_i on the mesh, initialize M random walkers at x_i.
2) Simulate each of the M walkers, keeping track of the cumulative number n_{ij} of walkers on node x_j that began on x_i. End when all have absorbed. **Do not include the initialization as part of the cumulative count.**
3) Repeat or parallelize for all positions x_i.
4) Use a right end-point approximation to assign
$$\mathbb{E}\left[-F\int_0^T \ell - X(s)\,\mathrm{d}s\right] \approx -\frac{F\Delta t}{M}\sum_j n_{ij}(\ell - x_j) \coloneqq u_i.$$
5) Then,
$$u(x_i) \approx u_i - u_0.$$



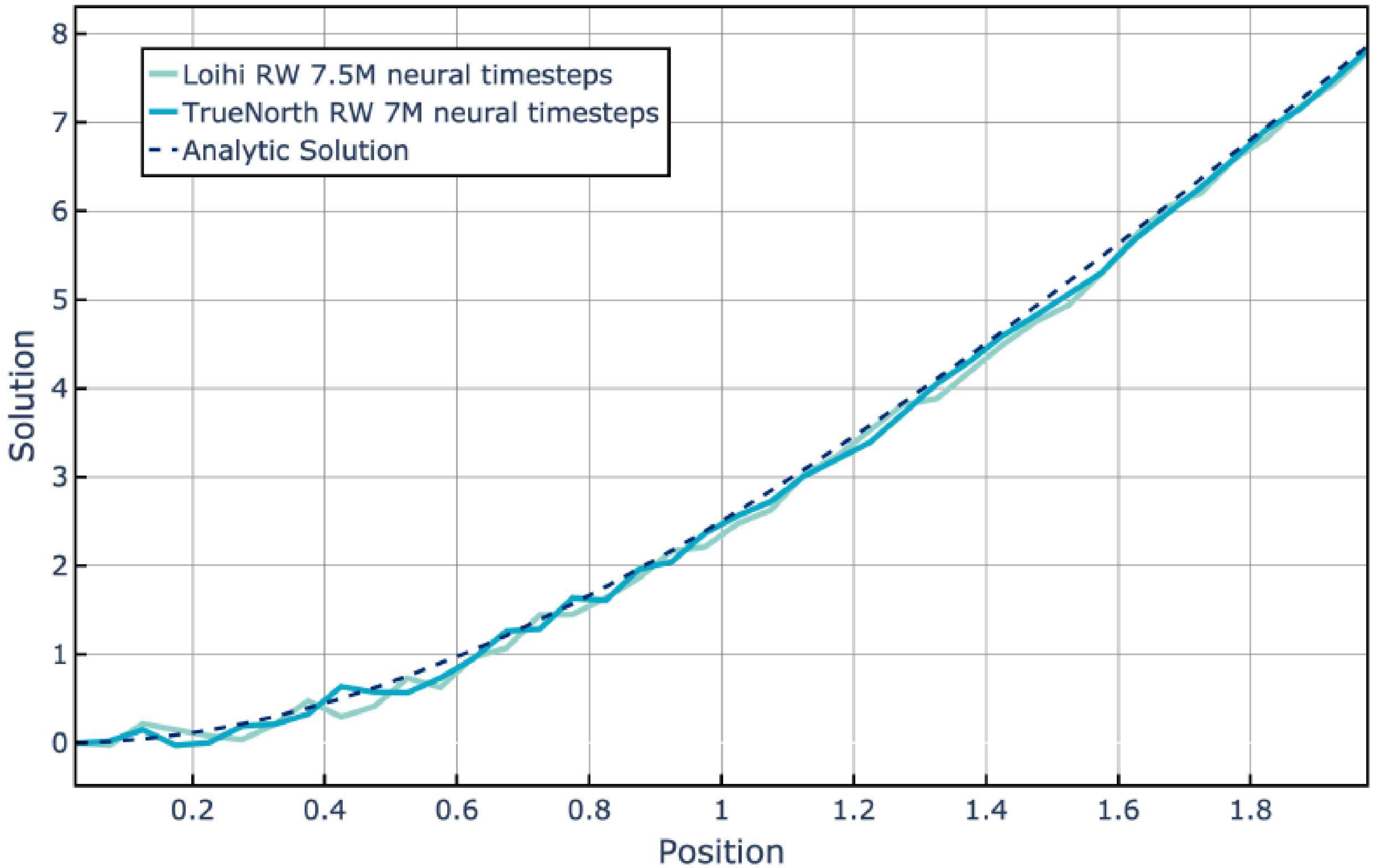
Matlab implementation of ten runs with $M = 10{,}000$, $F = 3$, $\ell = 2$, $\Delta x = 0.05$, and $\Delta t = 0.0001$.

Spiking Net Simulator Results

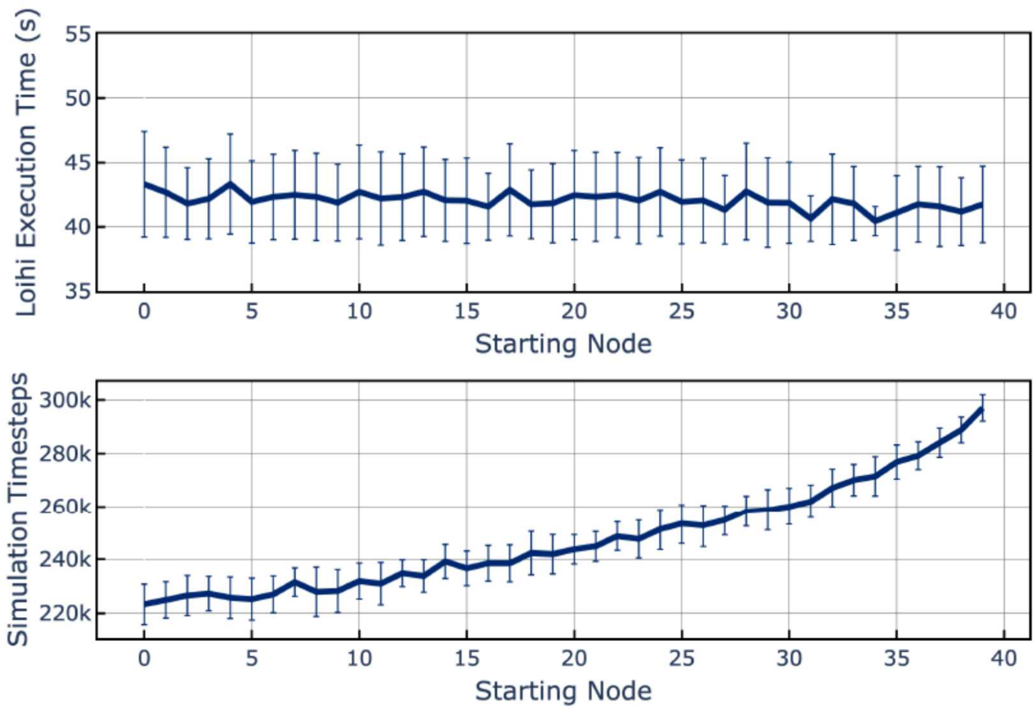
- We run for a fixed number of neural timesteps.
- Any walkers not finished by then increase the error of approximation.



Loihi and TrueNorth Results



Loihi Implementation



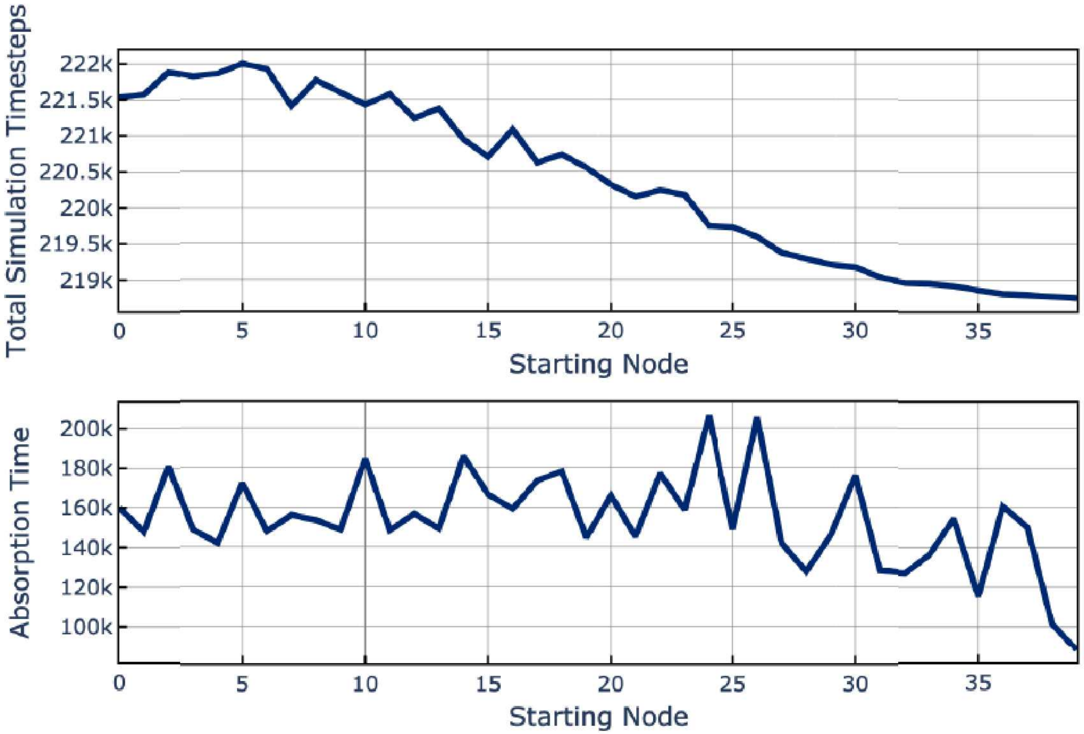
Top: Execution time for 250 walkers/starting location for 7.5 million neural timesteps.

Bottom: Number of simulation timesteps gained from 7.5 million neural timesteps per starting location.

- We deployed a neural circuit representing this random walk onto an 8-chip Nahuku platform.
- Walkers are removed from simulation once they reach the absorption node.

TrueNorth Implementation

- We implemented the random walk on a single chip of the TrueNorth hardware.
- Walkers are not removed once they reach the absorption node.

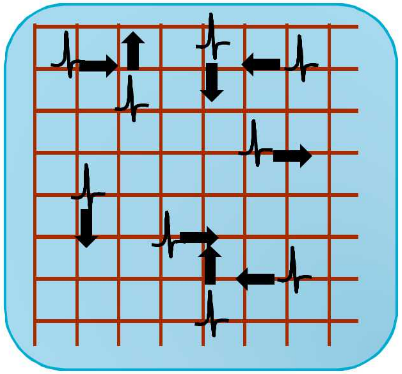


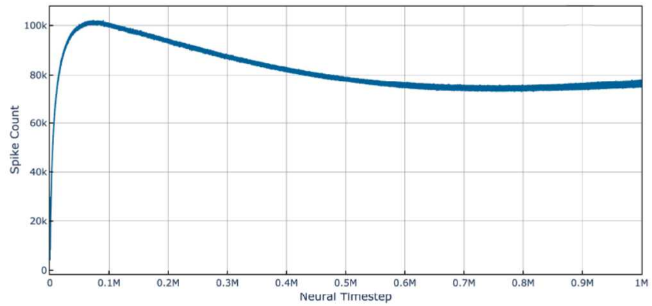
Top: Number of simulation timesteps gained from 7 million neural timesteps per starting position.

Bottom: Number of simulation timesteps until absorption of all walkers per starting position.

Steady-State Heat Transport as a Neuromorphic Benchmark

- Fully self-contained.
- Has an easy-to-grasp analytic solution.
- Low requirements on the neuron model.
- Scales in the number of nodes (neurons) and the number of walkers (spikes).
- Connectivity is local.
- Simple pattern.

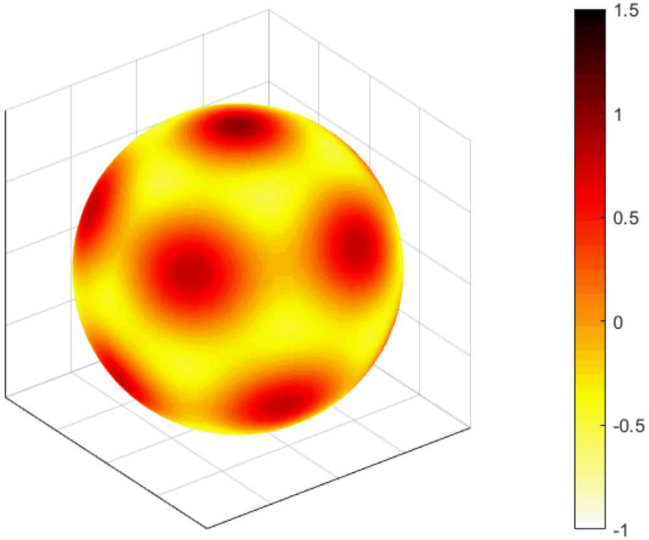




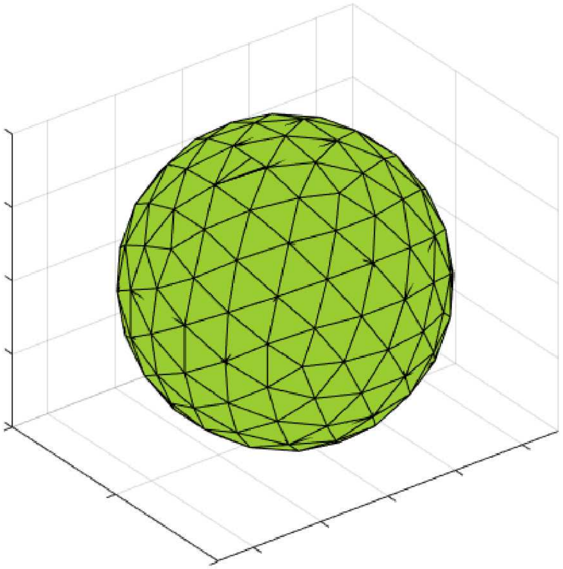
Ongoing Work and Future Directions

Non-Euclidean Heat Flow

- How would we calculate the flow over the surface of the sphere with this initial condition?



- Discretize the sphere into roughly equal pieces

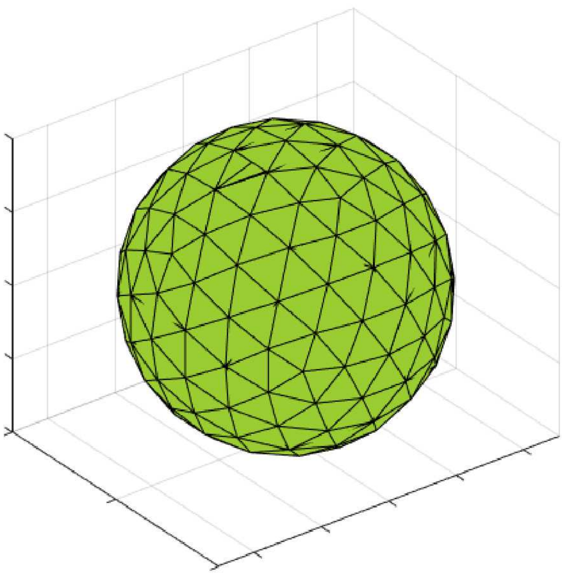


$$\frac{\partial}{\partial t}u(t,x,y,z) = \alpha\nabla u(t,x,y,z), \qquad (x,y,z) \in \mathcal{S}^2$$

$$u(0,x,y,z) = g(x,y,z)$$

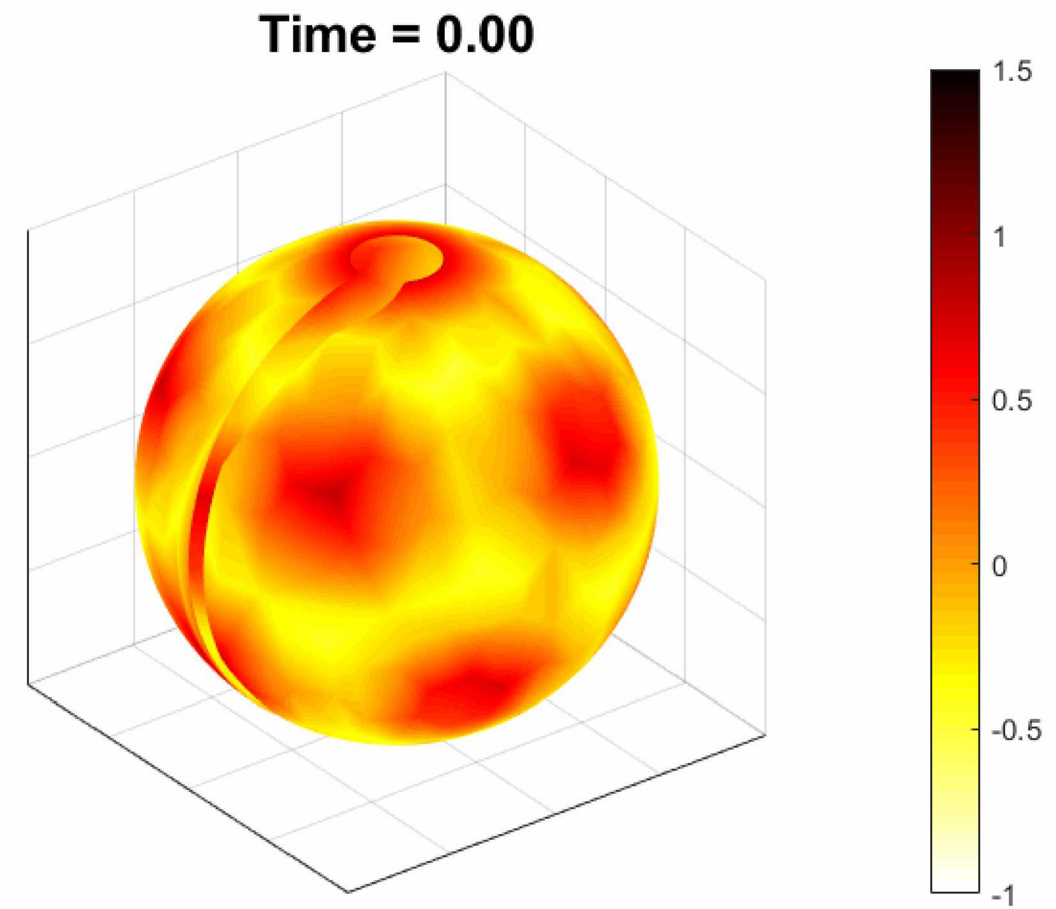
Non-Euclidean Heat Flow

- The solution is given by
$u(t, x, y, z) = \mathbb{E}[g(\boldsymbol{X}_t) | \boldsymbol{X}_0 = (x, y, z)]$.
- But what is the appropriate process for $\boldsymbol{X}_t$?



- There are some options.
  1. Use the von Mises-Fisher distribution.
  $$f(\boldsymbol{x}; \mu, k) = \frac{k}{2\pi(e^k - e^{-k})} e^{x\mu^\top \boldsymbol{x}}$$
  2. Use spherical coordinates.
  $$d\Theta_t = \alpha \cot \Theta_t + \sqrt{2\alpha} dW_1(t)$$
  $$d\Phi_t = \sqrt{2\alpha \csc^2 \Theta_t} \, dW_2(t)$$
  3. Project to the tangent plane.

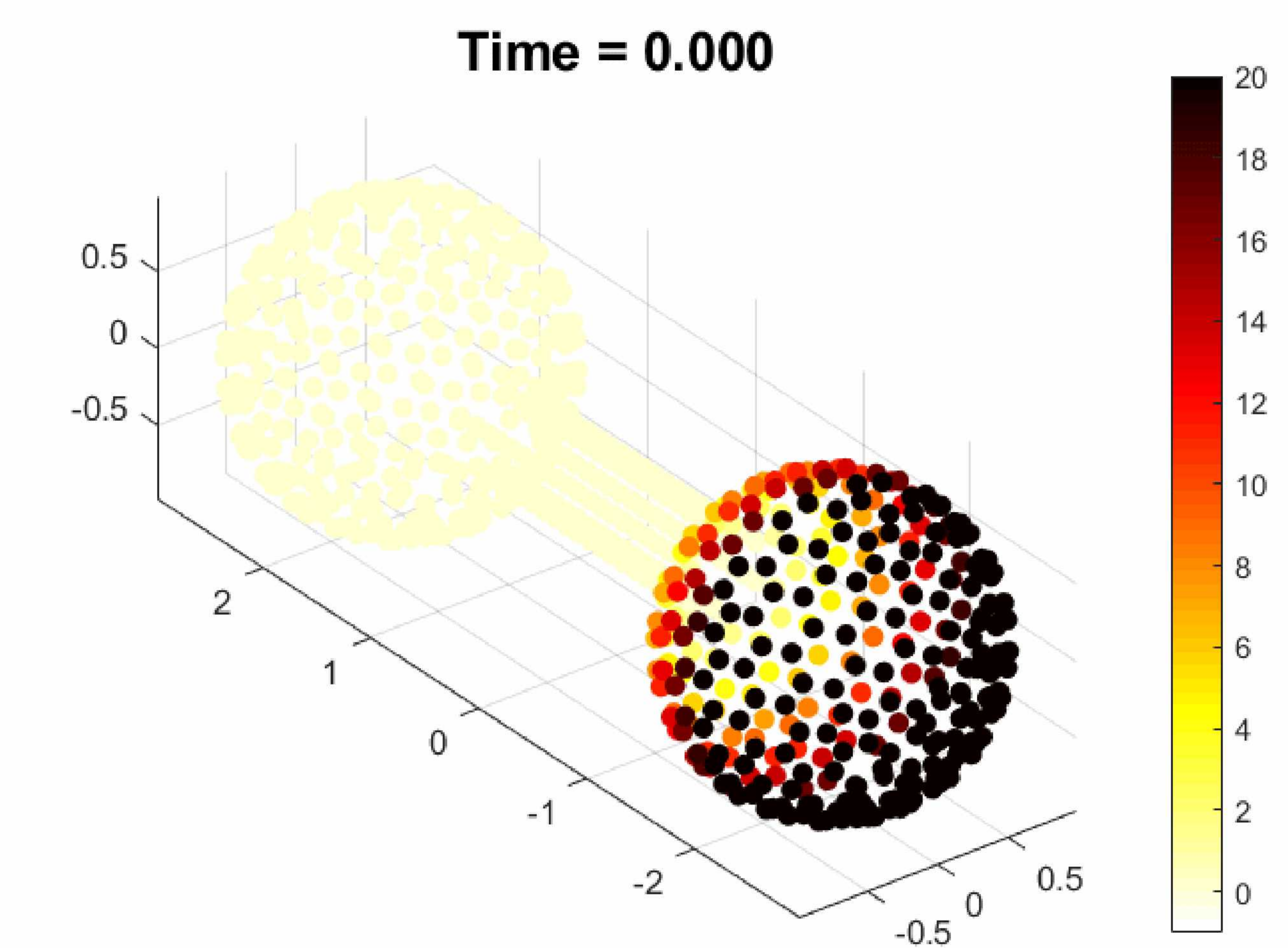
Non-Euclidean Heat Flow



$$\frac{\partial}{\partial t}u(t,x,y,z) = \alpha \nabla u(t,x,y,z), \qquad (x,y,z) \in \mathcal{S}^2$$
$$u(0,x,y,z) = g(x,y,z)$$

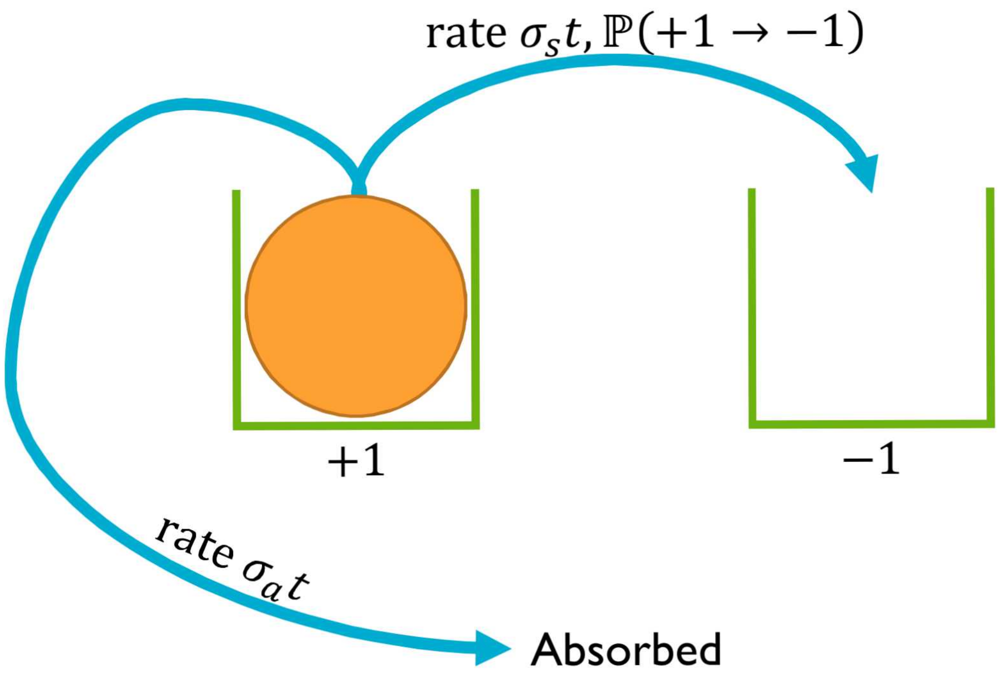
*result in preparation for future publication

Non-Euclidean Heat Flow



*result in preparation for future publication

Boltzmann Transport Equation for Simple Particle

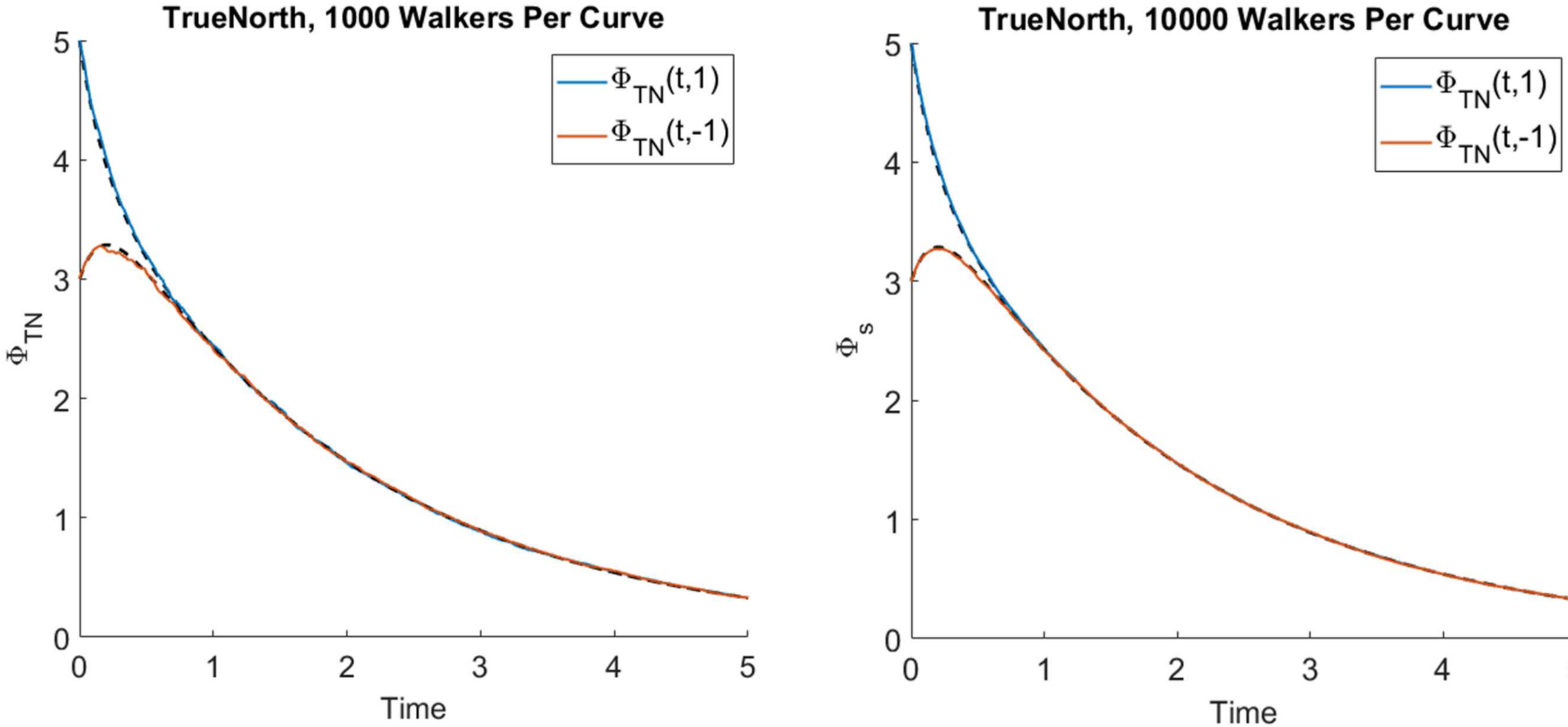


$$\frac{\partial}{\partial t}\Phi(t,\Omega) = -(\sigma_a + \sigma_s)\Phi(t,\Omega) + \int \sigma_s \Phi(t,\Omega')\mathbb{P}(\Omega' \to \Omega)\mathrm{d}\Omega'$$

$$\Phi(\Omega,0) = g(\Omega) = \begin{cases} 5 & \text{if } \Omega = 1 \\ 3 & \text{if } \Omega = -1 \end{cases}$$

$$\Phi(t,\Omega) = \begin{cases} \frac{1}{2}g(1)\left(e^{-\sigma_a t} + e^{-(\sigma_a+\sigma_s)t}\right) + \frac{1}{2}g(-1)\left(e^{-\sigma_a t} - e^{-(\sigma_a+\sigma_s)t}\right) & \text{if } \Omega = 1 \\ \frac{1}{2}g(1)\left(e^{-\sigma_a t} - e^{-(\sigma_a+\sigma_s)t}\right) + \frac{1}{2}g(-1)\left(e^{-\sigma_a t} + e^{-(\sigma_a+\sigma_s)t}\right) & \text{if } \Omega = -1 \end{cases}$$

Boltzmann Transport Equation for Simple Particle



$$\frac{\partial}{\partial t}\Phi(t,\Omega) = -(\sigma_a + \sigma_s)\Phi(t,\Omega) + \int \sigma_s \Phi(t,\Omega')\mathbb{P}(\Omega' \to \Omega)\mathrm{d}\Omega'$$

$$\Phi(\Omega,0) = g(\Omega) = \begin{cases} 5 & \text{if } \Omega = 1 \\ 3 & \text{if } \Omega = -1 \end{cases}$$

*result in preparation for future publication

Selected References

- Random Walks and the Steady-State Heat Equation
  - Smith, J.D., Severa, W., Hill, A., Reeder, L., Franke, B., Lehoucq, R., Parekh, O., and Aimone, J.B., Solving a steady state PDE using spiking networks and neuromorphic hardware. In *International Conference on Neuromorphic Systems 2020 (ICONS 2020)*, ACM, 8 pages.
  - This paper can already be found on arXiv!
- Random Walks with Spiking Neuromorphic Hardware
  - Severa, W., Lehoucq, R., Parekh, O. and Aimone, J.B., Spiking Neural Algorithms for Markov Process Random Walk. in 2018 *International Joint Conference on Neural Networks (IJCNN)* (2018), IEEE, 1-8.
- Random Walk Methods for PDEs
  - Grigoriu, M. (2013), *Stochastic Calculus: Applications in Science and Engineering*, Springer Science & Business Media.
- Stochastic Calculus
  - Protter, P. (1990), *Stochastic integration and differential equations: a new approach,* Springer-Verlag.
  - Wiersema, U. F. (2008), *Brownian motion calculus*, John Wiley & Sons.