Welcome to Three Days of Deep Learning on Neural ODEs

Stéphane Gaudreault[1] Seth Taylor[2] Siqi Wei[3]

[1]Research Manager, Environment and Climate Change Canada
[2]Postdoctoral Fellow, University of Saskatchewan
[3]Postdoctoral Fellow, Environment and Climate Change Canada (ECCC)

June 23, 2025

Hackathon Overview

Our Mission: Systematically explore Neural ODEs through hands-on experimentation

Neural ODEs:

$$\frac{d\mathbf{z}(t)}{dt} = f(\mathbf{z}(t), t, \boldsymbol{\theta}) \tag{1}$$

How do different integration methods and training strategies affect performance?

Key Investigations:

- Time integration methods (explicit, implicit, adaptive)
- Training approaches: Adjoint method vs. backpropagation through time
- Performance across problem types and complexities
- Efficiency-accuracy trade-offs

Your Learning Journey: From Simple to Sophisticated

Day 1: Foundation Building

- MNIST classification with Neural ODEs
- Compare integration methods systematically
- Measure: accuracy, memory, wall-clock time, solver steps

Day 2: Dynamical Systems Deep Dive

- Van der Pol oscillator: $\ddot{x} - \mu(1 - x^2)\dot{x} + x = 0$
- Parameter sweeps: stiffness $\mu \in \{0.5, 1.0, 2.0, 5.0, 10.0\}$
- Long-term prediction accuracy analysis

Day 3: Advanced Extensions

- **Option A:** Implicit methods for stiff systems
- **Option B:** PDEBench - 1D Burgers equation

Success Strategy

- Start simple, build complexity gradually
- Document everything: code, results, insights
- Ask questions early and often!
- Focus on understanding, not just getting code to run

Philosophy: Deep insight from simple examples
Quality over quantity

Key Technical Comparisons You will Explore

Integration Methods:

- **Explicit:** Euler, Runge-Kutta (RK4, Dormand-Prince)
- **Implicit:** Backward Euler, BDF methods
- **Adaptive:** Automatic step-size control

Training Strategies:

- **BPTT:** Backpropagate through all solver steps
- **Adjoint:** Solve backward ODE

What You will Measure:

- Computational efficiency (time, memory, NFE counts)
- Numerical accuracy and stability
- Training convergence behavior
- Performance on different problem types

Day 1 - Task 1: Classification Experiments

- Introduction to Neural ODEs and the torchdiffeq package
- Project planning, environment setup, testing torchdiffeq examples
- Implement Neural ODE models for MNIST and fashion classifications
- Compare different time integration methods
- Compare adjoint method and backpropagation through time
- Document solver step counts, wall-clock time, memory usage, and accuracy
- **Progress check-in and discussion**

Problem 1: MNIST Classification

- Input: $\mathbf{z}(0) =$ image features (28×28 pixels)
- Evolution: $\frac{d\mathbf{z}}{dt} = f(\mathbf{z}(t), t, \boldsymbol{\theta})$
- Output: $\mathbf{z}(T) \rightarrow$ classifier

Day 2 - Task 2: Van der Pol Oscillator Experiments

- Implement reference solutions using scipy.integrate
- Generate synthetic training data with varying time horizons
- Train Neural ODE models with different integrators
- Systematic comparison of training approaches
- Analysis of long-term prediction accuracy
- Vary stiffness parameter μ (0.5, 1.0, 2.0, 5.0, 10.0)
- Analyze efficiency vs. accuracy for each configuration
- Study solver behavior as stiffness increases

Problem 2: Van der Pol Oscillator

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} y \\ \mu(1 - x^2)y - x \end{bmatrix} \tag{2}$$

- Varies from simple ($\mu = 0.5$) to stiff ($\mu = 10.0$)
- Test long-term prediction capabilities
- Analyze solver step counts vs. stiffness

Day 2 - Additional Dynamical Systems

Additional dynamical systems (teams can choose):

- Lorenz system
- Lotka-Volterra predator–prey
- Double pendulum

- Add noise to the system and analyze robustness
- **Progress check-in and discussion**

Remember: Choose depth over breadth!

Day 3 - Advanced Extensions

Advanced extensions (teams select based on progress):

Option A: Implicit methods and stiff systems

- Implement implicit Euler or backward differentiation formulas
- Compare performance on stiff Van der Pol ($\mu > 10$)

Option B: PDEBench - 1D Burgers equation

- Introduction to Neural PDE solvers
- Implement Neural ODE approach for 1D Burgers equation
- Compare with traditional numerical methods (finite differences)

Day 3 - Wrap-up

- **Presentation preparation**
- **Team presentations**

Project Leader Availability

- Check-ins at 9-10am, 1:30-2:30pm, 4:30-5:00pm daily
- Always available for questions!

Ready to explore the fascinating world of Neural ODEs?