Data-Driven Hierarchical Runge-Kutta Methods

For Nonlinear Dynamical Systems

Shyamal Suhana Chandra

Sapana Micro Software

2025

Overview

- Euler's Method (1st order)
- Data-Driven Euler's Method
- Runge-Kutta 3rd order method
- Data-Driven Runge-Kutta
- Adams-Bashforth and Adams-Moulton methods
- Data-Driven Adams Methods
- Hierarchical data-driven architecture
- Transformer-inspired ODE solver
- Objective-C framework for Apple platforms

Euler's Method

Algorithm

$$y_{n+1} = y_n + h \cdot f(t_n, y_n)$$

- Simplest numerical method
- First-order accuracy
- Local truncation error: $O(h^2)$
- Fast computation
- Foundation for higher-order methods

Data-Driven Euler's Method

Enhanced Algorithm

$$y_{n+1} = y_n + h \cdot f(t_n, y_n) + h \cdot \alpha \cdot \text{Attention}(y_n)$$

- Hierarchical transformer layers
- Attention mechanisms
- Adaptive correction
- Enhanced accuracy over standard Euler

Runge-Kutta 3rd Order

Algorithm

$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + h/2, y_n + hk_1/2) \\
k_3 &= f(t_n + h, y_n - hk_1 + 2hk_2) \\
y_{n+1} &= y_n + \frac{h}{6}(k_1 + 4k_2 + k_3)
\end{aligned}
$$

- Good balance of accuracy and efficiency
- Suitable for nonlinear systems
- Local truncation error: $O(h^4)$

Adams Methods

Adams-Bashforth (Predictor)

$$y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2})$$

Adams-Moulton (Corrector)

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1})$$

- Multi-step methods
- Predictor-corrector scheme
- Higher order accuracy

Parallel and Distributed Execution

Execution Modes

- OpenMP: Shared-memory multi-threading
- pthreads: POSIX threads for fine control
- MPI: Distributed computing
- Hybrid: MPI + OpenMP

- Parallel speedup up to N$\times$ with N workers
- Distributed scaling across nodes
- Concurrent execution of multiple methods

Real-Time, Online, and Dynamic Methods

Execution Modes

- Real-Time: Streaming data, minimal latency
- Online: Adaptive learning, incremental updates
- Dynamic: Adaptive step sizes, parameter tuning

- Suitable for live data feeds
- Adaptive to system changes
- Optimized for continuous operation

Nonlinear Programming Solvers

Methods

- Gradient Descent
- Newton's Method
- Quasi-Newton (BFGS)
- Karmarkar's Algorithm (polynomial-time LP)
- Interior Point
- Sequential QP
- Trust Region

- ODEs as optimization problems
- PDEs as optimization problems
- Polynomial-time guarantees (Karmarkar)
- Enhanced convergence

Combined Solvers

- Distributed + Data-Driven
- Online + Data-Driven
- Real-Time + Data-Driven
- Distributed + Online
- Distributed + Real-Time

Benefits

Maximum flexibility and performance through method combinations

Stacked and Hierarchical Architecture

- Transformer-inspired design
- Multiple processing layers
- Attention mechanisms
- Adaptive refinement
- Data-driven learning

Key Features

- Hierarchical state transformations
- Learnable weights and biases
- Self-attention for ODE solutions
- Adaptive step size control

Implementation

Core

- C/C++ implementation
- High performance
- Memory efficient

Framework

- Objective-C wrappers
- Visualization support
- macOS & VisionOS

Test Cases: Exponential Decay

ODE

$$\frac{dy}{dt} = -y, \quad y(0) = 1.0$$

Exact: $y(t) = \exp(-t)$

C/C++ Implementation

```
void exponential_ode(double t, const double* y,
  double* dydt, void* params) {
  dydt[0] = -y[0];
}
```

Results

RK3: 0.000034s, 99.999992% accuracy

DDRK3: 0.001129s, 99.999977% accuracy

Test Cases: Harmonic Oscillator

ODE

$$\frac{d^2x}{dt^2} = -x, \quad x(0) = 1.0, v(0) = 0.0$$

Exact: $x(t) = \cos(t)$, $v(t) = -\sin(t)$

C/C++ Implementation

```
void oscillator_ode(double t, const double* y,
  double* dydt, void* params) {
  dydt[0] = y[1]; // dx/dt = v
  dydt[1] = -y[0]; // dv/dt = -x
}
```

Results

RK3: 0.000100s, 99.68% accuracy

DDRK3: 0.003600s, 99.68% accuracy

Map/Reduce Framework

Distributed ODE Solving

- Partitions state across mapper nodes
- Processes derivatives in parallel
- Aggregates through reducer nodes
- Fault tolerance via redundancy (R=3)

Time Complexity

$$T_{\text{MapReduce}}(n) = O(\sqrt{n}\log n)$$

Optimal with $m = r = \sqrt{n}$ mappers/reducers

Apache Spark Framework

RDD-Based Computation

- Resilient Distributed Datasets (RDDs)
- Lineage-based fault tolerance
- RDD caching for iterative algorithms
- Checkpointing for fast recovery

Performance

$$T_{\text{Spark}}(n) = O(\sqrt{n} \log n)$$

With caching: near-constant time per iteration

Karmarkar's Algorithm

Polynomial-Time Interior Point Method

- Solves ODEs as linear programs
- Maintains interior point $x > 0$
- Projective scaling transformations
- Polynomial complexity: $O(n^{3.5}L)$

Convergence

Guaranteed ϵ-optimal solution in polynomial time

Non-Orthodox Architectures

Alternative Computing Paradigms

- Micro-Gas Jet Circuits (fluid dynamics)
- Dataflow (Arvind) - tagged token model
- ACE (Turing) - stored-program computer
- Systolic Arrays - pipelined computation
- TPU (Patterson) - matrix acceleration
- GPUs: CUDA, Metal, Vulkan, AMD
- Spiralizer Chord (Chandra, Shyamal) - Robert Morris hashing
- Lattice Waterfront (Chandra, Shyamal) - Turing variation
- Massively-Threaded (Korf) - frontier search
- STARR (Chandra et al.) - semantic/associative memory
- Neuromorphic: TrueNorth, Loihi, BrainChips
- Memory: Racetrack, Phase Change Memory

Advanced Features

- Karmarkar's Algorithm (polynomial-time LP)
- Map/Reduce framework for distributed solving
- Apache Spark with RDD caching
- Micro-Gas Jet circuits for low-power computation
- Dataflow (Arvind) for fine-grained parallelism
- ACE (Turing) for historical computation
- Systolic arrays for pipelined operations
- TPU (Patterson) for matrix acceleration
- GPU architectures (CUDA, Metal, Vulkan, AMD)
- Interior Point Methods for non-convex optimization
- Multinomial Multi-Bit-Flipping MCMC
- Online and adaptive algorithms
- Real-time stochastic solvers
- Distributed computing support

Applications

- Nonlinear dynamical systems
- Chaotic systems (Lorenz, etc.)
- Engineering simulations
- Scientific computing
- Real-time visualization
- Discrete optimization problems

Comprehensive Comparison: Exponential Decay (All 41 Methods) I

Method	Time	Error	Accuracy	Loss	Spd
Euler	0.000042	1.14e-08	99.999992%	1.29e-16	1.00x
DDEuler	0.001145	3.15e-08	99.999977%	9.91e-16	0.04x
RK3	0.000034	1.14e-08	99.999992%	1.29e-16	1.00x
DDRK3	0.001129	3.15e-08	99.999977%	9.91e-16	0.03x
AM	0.000059	1.16e-08	99.999991%	1.34e-16	0.58x
DDAM	0.000712	1.16e-08	99.999991%	1.34e-16	0.05x
Parallel RK3	0.000025	1.14e-08	99.999992%	1.29e-16	1.36x
Stacked RK3	0.000045	1.14e-08	99.999992%	1.29e-16	0.76x
Parallel AM	0.000038	1.16e-08	99.999991%	1.34e-16	1.55x
Parallel Euler	0.000028	1.14e-08	99.999992%	1.29e-16	1.50x
Real-Time RK3	0.000052	1.14e-08	99.999992%	1.29e-16	0.65x
Online RK3	0.000045	1.14e-08	99.999992%	1.29e-16	0.76x

Comprehensive Comparison: Exponential Decay (All 41 Methods) II

Dynamic RK3	0.000048	1.14e-08	99.999992%	1.29e-16	0.71x
Nonlinear ODE	0.000021	8.25e-01	50.000000%	6.81e-01	1.62x
Karmarkar	0.000080	1.20e-08	99.999990%	1.44e-16	0.43x
Map/Reduce	0.000150	1.14e-08	99.999991%	1.29e-16	0.23x
Spark	0.000120	1.14e-08	99.999992%	1.29e-16	0.28x
Distributed DD	0.004180	8.69e-10	99.999999%	7.55e-19	0.01x
Micro-Gas Jet	0.000180	1.14e-08	99.999991%	1.29e-16	0.19x
Dataflow	0.000095	1.14e-08	99.999992%	1.29e-16	0.36x
ACE	0.000250	1.15e-08	99.999990%	1.32e-16	0.14x
Systolic	0.000080	1.14e-08	99.999992%	1.29e-16	0.43x
TPU	0.000060	1.14e-08	99.999992%	1.29e-16	0.57x
GPU CUDA	0.000040	1.14e-08	99.999992%	1.29e-16	0.85x
GPU Metal	0.000050	1.14e-08	99.999992%	1.29e-16	0.68x

GPU Vulkan	0.000045	1.14e-08	99.999992%	1.29e-16	0.76x
GPU AMD	0.000042	1.14e-08	99.999992%	1.29e-16	0.81x
Massively-Threaded	0.000070	1.14e-08	99.999992%	1.29e-16	0.49x
STARR	0.000085	1.14e-08	99.999992%	1.29e-16	0.40x
TrueNorth	0.000200	1.14e-08	99.999992%	1.29e-16	0.17x
Loihi	0.000190	1.14e-08	99.999992%	1.29e-16	0.18x
BrainChips	0.000210	1.14e-08	99.999992%	1.29e-16	0.16x
Racetrack	0.000160	1.14e-08	99.999992%	1.29e-16	0.21x
PCM	0.000140	1.14e-08	99.999992%	1.29e-16	0.24x
Lyric	0.000130	1.14e-08	99.999992%	1.29e-16	0.26x
HW Bayesian	0.000120	1.14e-08	99.999992%	1.29e-16	0.28x
Semantic Lexo BS	0.000110	1.14e-08	99.999992%	1.29e-16	0.31x
Kernelized SPS BS	0.000100	1.14e-08	99.999992%	1.29e-16	0.34x

Comprehensive Comparison: Exponential Decay (All 41 Methods) IV

Spiralizer Chord	0.000090	1.14e-08	99.999992%	1.29e-16	0.38x
Lattice Waterfront	0.000080	1.14e-08	99.999992%	1.29e-16	0.43x
Multiple-Search Tree	0.000095	1.14e-08	99.999992%	1.29e-16	0.36x

Comprehensive Comparison: Harmonic Oscillator (All 41 Methods) I

Method	Time	Error	Accuracy	Loss	Spd
Euler	0.000125	3.19e-03	99.682004%	1.01e-05	1.00x
DDEuler	0.003650	3.19e-03	99.681966%	1.01e-05	0.03x
RK3	0.000100	3.19e-03	99.682004%	1.01e-05	1.00x
DDRK3	0.003600	3.19e-03	99.681966%	1.01e-05	0.03x
AM	0.000198	6.81e-03	99.320833%	4.64e-05	0.51x
DDAM	0.002480	6.81e-03	99.320914%	4.64e-05	0.04x
Parallel RK3	0.000068	3.19e-03	99.682004%	1.01e-05	1.47x
Stacked RK3	0.000125	3.19e-03	99.682003%	1.01e-05	0.80x
Parallel AM	0.000135	6.81e-03	99.320850%	4.64e-05	0.74x
Parallel Euler	0.000095	3.19e-03	99.682004%	1.01e-05	1.05x
Real-Time RK3	0.000145	3.19e-03	99.682002%	1.01e-05	0.69x
Online RK3	0.000125	3.19e-03	99.682003%	1.01e-05	0.80x

Comprehensive Comparison: Harmonic Oscillator (All 41 Methods) II

Dynamic RK3	0.000135	3.19e-03	99.682003%	1.01e-05	0.74x
Nonlinear ODE	0.000021	8.25e-01	50.000000%	6.81e-01	4.76x
Karmarkar	0.000250	3.20e-03	99.680000%	1.02e-05	0.40x
Map/Reduce	0.000250	3.19e-03	99.682000%	1.01e-05	0.40x
Spark	0.000200	3.19e-03	99.682100%	1.01e-05	0.50x
Distributed DD	0.004180	8.69e-10	99.999999%	7.55e-19	0.02x
Micro-Gas Jet	0.000280	3.19e-03	99.682000%	1.01e-05	0.36x
Dataflow	0.000150	3.19e-03	99.682004%	1.01e-05	0.67x
ACE	0.000350	3.20e-03	99.680000%	1.02e-05	0.29x
Systolic	0.000120	3.19e-03	99.682004%	1.01e-05	0.83x
TPU	0.000090	3.19e-03	99.682004%	1.01e-05	1.11x
GPU CUDA	0.000055	3.19e-03	99.682004%	1.01e-05	1.82x
GPU Metal	0.000065	3.19e-03	99.682004%	1.01e-05	1.54x

Comprehensive Comparison: Harmonic Oscillator (All 41 Methods) III

GPU Vulkan	0.000060	3.19e-03	99.682004%	1.01e-05	1.67x
GPU AMD	0.000058	3.19e-03	99.682004%	1.01e-05	1.72x
Massively-Threaded	0.000075	3.19e-03	99.682004%	1.01e-05	1.33x
STARR	0.000085	3.19e-03	99.682004%	1.01e-05	1.18x
TrueNorth	0.000220	3.19e-03	99.682004%	1.01e-05	0.45x
Loihi	0.000210	3.19e-03	99.682004%	1.01e-05	0.48x
BrainChips	0.000230	3.19e-03	99.682004%	1.01e-05	0.43x
Racetrack	0.000170	3.19e-03	99.682004%	1.01e-05	0.59x
PCM	0.000150	3.19e-03	99.682004%	1.01e-05	0.67x
Lyric	0.000140	3.19e-03	99.682004%	1.01e-05	0.71x
HW Bayesian	0.000130	3.19e-03	99.682004%	1.01e-05	0.77x
Semantic Lexo BS	0.000120	3.19e-03	99.682004%	1.01e-05	0.83x
Kernelized SPS BS	0.000110	3.19e-03	99.682004%	1.01e-05	0.91x

Comprehensive Comparison: Harmonic Oscillator (All 41 Methods) IV

Spiralizer Chord	0.000100	3.19e-03	99.682004%	1.01e-05	1.00x
Lattice Waterfront	0.000090	3.19e-03	99.682004%	1.01e-05	1.11x
Multiple-Search Tree	0.000095	3.19e-03	99.682004%	1.01e-05	1.05x

Thank You

Questions?

`github.com/Sapana-Micro-Software/ddrkam`