On Time-stepping Methods for Gradient-flow Optimization

Arash Sarshar

Computational Science Laboratory
"Compute the Future!"
Department of Computer Science,
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060

CAIMS/SCMAI, June, 2022

Compute the Future!

VIRGINIA TECH

Dr. Adrian Sandu
Director of CSL

Email: sandu@cs.vt.edu
Location: KWII 2222
Phone: (540) 231-2193
Fax: (540) 231-6075

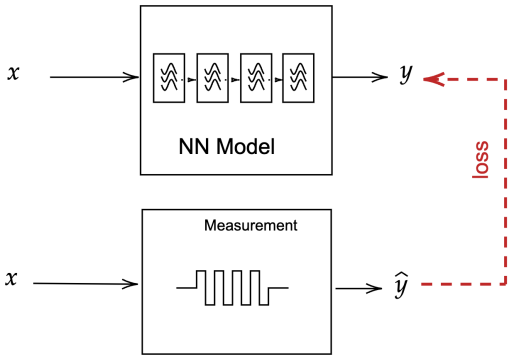
Abhinab Bhattac

Email: abhinab93@
Location: KWII 2201
Research Interests: Data assimilation, numerical methods

Andrey Popov

Email: apopov@vt.edu
Location: KWII 2201
Research Interests: Large-scale dynamical systems, data assimilation, time integration

(Stochastic) Gradient descent and gradient-flow



$$\arg\min_{\theta \in \mathbb{R}^d} L(\theta), \tag{1}$$

$$\theta^{k+1} = \theta^k - \alpha \nabla_\theta L(\theta^k), \tag{2}$$

$$\frac{\partial \theta}{\partial t} = -\nabla_\theta \mathbf{L}(\theta) \tag{3}$$

Inertial optimization methods

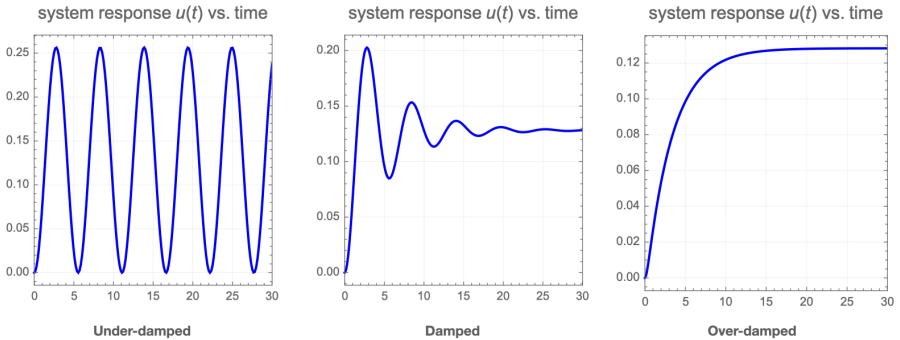
$$\underbrace{m(t)}_{\textit{mass}} \frac{\partial^2 \theta}{\partial t^2} + \underbrace{\gamma(t)}_{\textit{damping}} \frac{\partial \theta}{\partial t} + \underbrace{\nabla_\theta L(\theta)}_{\textit{force}} = 0 \tag{4}$$

- Adding inertia, and stiffness
- Heavy-ball (Polyak 1946)

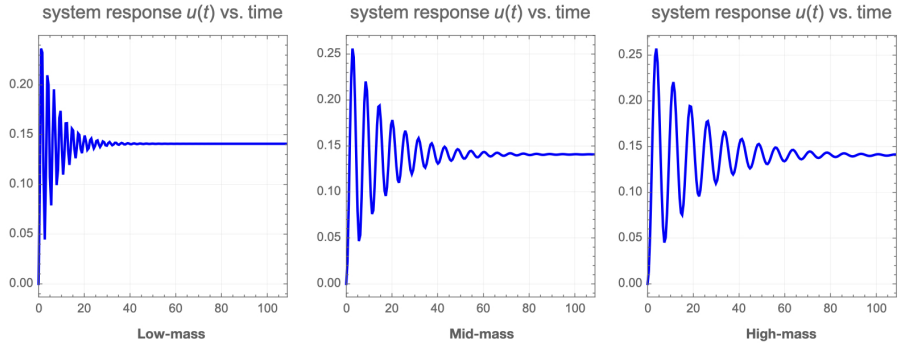
  B. T. Polyak, Some methods of speeding up the convergence of iteration methods, USSR Comput. Math. Math. Phys., 4 (1964), pp. 1–17
- Nesterov acceleration (due to Su et al. 2016)

  W. Su, S. Boyd, and E. J. Candes, A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights, J. Mach. Learn. Res., 17 (2016), pp. 1–43.
- How to enforce constraints on the optimization? (Solver + velocity shocks)

Inertial optimization methods



Inertial optimization methods



Momentum methods through Hamiltonian mechanics part I

$$\dot{\mathbf{x}}_t = \nabla_{\mathbf{z}} \mathcal{H}(\mathbf{x}, \mathbf{z}, t), \tag{5}$$
$$\dot{\mathbf{z}}_t = -\nabla_{\mathbf{x}} \mathcal{H}(\mathbf{x}, \mathbf{z}, t) \tag{6}$$

- Describe the dynamics in terms of pairs $(\mathbf{x}, \mathbf{z})$ (position, momentum) and Hamiltonian function (Total Energy) such that $\frac{\partial \mathcal{H}}{\partial t} = 0$.
- Additive splitting into potential and kinetic components : $\mathcal{H}(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) + \psi^*(\mathbf{z})$
- Norm of the averaged gradient $||\nabla f||$ decays with $\mathcal{O}(1/t)$
- Unconstrained, No convergence guarantees due to preservation of energy.

  Diakonikolas, Jelena, and Michael I. Jordan. 2021. "Generalized Momentum-Based Methods: A Hamiltonian Perspective." SIAM Journal on Optimization 31 (1): 915–44.

Momentum methods through Hamiltonian mechanics part I

$$\begin{aligned}\dot{\mathbf{x}}_t &= \nabla_{\mathbf{z}_t}\mathcal{H}\left(\mathbf{x}_t, \mathbf{z}_t\right) = \nabla\psi^*\left(\mathbf{z}_t\right)\\ \dot{\mathbf{z}}_t &= -\nabla_{\mathbf{x}_t}\mathcal{H}\left(\mathbf{x}_t, \mathbf{z}_t\right) = -\nabla f\left(\mathbf{x}_t\right)\end{aligned} \tag{5}$$

- Describe the dynamics in terms of pairs $(\mathbf{x}, \mathbf{z})$ (position, momentum) and Hamiltonian function (Total Energy) such that $\frac{\partial \mathcal{H}}{\partial t} = 0$.
- Additive splitting into potential and kinetic components : $\mathcal{H}(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) + \psi^*(\mathbf{z})$
- Norm of the averaged gradient $||\nabla f||$ decays with $\mathcal{O}(1/t)$
- Unconstrained, No convergence guarantees due to preservation of energy.

Diakonikolas, Jelena, and Michael I. Jordan. 2021. "Generalized Momentum-Based Methods: A Hamiltonian Perspective." SIAM Journal on Optimization 31 (1): 915–44.

Momentum methods Hamiltonian mechanics part II

$$
\begin{aligned}
&\mathcal{H}_{\mathrm{M}}(\overline{\mathbf{x}}, \mathbf{z}, \tau) = h(\tau) f(\overline{\mathbf{x}}/\tau) + \psi^*(\mathbf{z}), \quad \overline{\mathbf{x}} = \tau \mathbf{x}, \\
&\dot{\mathbf{x}}_t = \frac{\dot{\alpha}_t \left(\nabla \psi^* \left(\mathbf{z}_t\right) - \mathbf{x}_t\right)}{\alpha_t}, \\
&\dot{\mathbf{z}}_t = -h\left(\alpha_t\right) \frac{\dot{\alpha}_t}{\alpha_t} \nabla f\left(\mathbf{x}_t\right).
\end{aligned}
\tag{6}
$$

- $h(\cdot)$ and α_t are strictly increasing functions of t
- Covers most of the acceleration methods
- Hamiltonian system is useful in deriving convergence properties both for f and $\mathbf{x} = \theta$.
- Does not cover CS-favored methods such as `Adam`, `RMSProp`, etc.

Adaptive momentum optimization methods (Adam)

Discrete (original) form of Adam (Kingma and Ba 2014) :

$$\begin{cases} g_k = \nabla f\left(\theta_{k-1}\right) \\ m_k = \mu_k m_{k-1} + \left(1-\mu_k\right) g_k \\ v_k = \nu_k v_{k-1} + \left(1-\nu_k\right) g_k^2 \\ \theta_k = \theta_{k-1} - s m_k / \left(\sqrt{v_k} + \varepsilon\right). \end{cases} \tag{7}$$

- $m(t)$ is a decaying moving average of estimates of the first moment (the mean) of the gradient.
- $v(t)$ is a decaying moving average of estimates of the second raw moment (the uncentered variance) of the gradient.

Adaptive momentum optimization methods (Adam)

Continuous form (Belotto da Silva and Gazeau 2020) :

$$\begin{cases} \dot{\theta} = -m/\sqrt{v+\varepsilon} \\ \dot{m} = h\left(t,\lambda,\alpha_1\right)\left(\nabla f(\theta) - m\right) \\ \dot{v} = h\left(t,\lambda,\alpha_2\right)\left(\nabla f(\theta)^2 - v\right) \end{cases} \tag{7}$$

We recognize it as a partitioned ODE system:

$$\begin{bmatrix} \theta \\ V \end{bmatrix}' = \begin{bmatrix} 0 \\ F(\theta, V) \end{bmatrix} + \begin{bmatrix} G(V) \\ 0 \end{bmatrix} \tag{8}$$

with $V = [m, v]^T$

Generalized-structure Additive Runge-Kutta methods I

- Additively-partitioned ODE system:

$$y' = \sum_{m=1}^{N} f^{\{m\}}(y),$$

- One step of a GARK scheme (Sandu & Günther, 2013):

$$\begin{aligned} Y_i^{\{q\}} &= y_n + h \sum_{m=1}^{N} \sum_{j=1}^{s^{\{m\}}} a_{i,j}^{\{q,m\}} f^{\{m\}} \left(Y_j^{\{m\}} \right), \\ & \qquad i = 1, \ldots, s^{\{q\}}, \quad q = 1, \ldots, N, \\ y_{n+1} &= y_n + h \sum_{q=1}^{N} \sum_{i=1}^{s^{\{q\}}} b_i^{\{q\}} f^{\{q\}} \left(Y_i^{\{q\}} \right). \end{aligned}$$

$$
\begin{array}{cccc}
\mathbf{A}^{\{1,1\}} & \mathbf{A}^{\{1,2\}} & \dots & \mathbf{A}^{\{1,N\}} \\
\mathbf{A}^{\{2,1\}} & \mathbf{A}^{\{2,2\}} & \dots & \mathbf{A}^{\{2,N\}} \\
\vdots & \vdots & & \vdots \\
\mathbf{A}^{\{N,1\}} & \mathbf{A}^{\{N,2\}} & \dots & \mathbf{A}^{\{N,N\}} \\
\hline
\mathbf{b}^{\{1\}} & \mathbf{b}^{\{2\}} & \dots & \mathbf{b}^{\{N\}}
\end{array}
\quad .
$$

- Each component function has a different argument.
- One set of RK coefficients for each component and stage (flexibility).
- The GARK Butcher tableau:

- Individual RKs plus coupling terms.

$$
\begin{aligned}
Y_i^{\{1\}} &= y_n + h \sum_{j=1}^{s^{\{1\}}} a_{i,j}^{\{1,1\}} f^{\{1\}}\left(Y_j^{\{1\}}\right) + h \sum_{j=1}^{s^{\{2\}}} a_{i,j}^{\{1,2\}} f^{\{2\}}\left(Y_j^{\{2\}}\right), \\
Y_i^{\{2\}} &= y_n + h \sum_{j=1}^{s^{\{2\}}} a_{i,j}^{\{2,2\}} f^{\{2\}}\left(Y_j^{\{2\}}\right) + h \sum_{j=1}^{s^{\{1\}}} a_{i,j}^{\{2,1\}} f^{\{1\}}\left(Y_j^{\{1\}}\right), \\
y_{n+1} &= y_n + h \sum_{i=1}^{s^{\{1\}}} b_i^{\{1\}} f^{\{1\}}\left(Y_i^{\{1\}}\right) + h \sum_{i=1}^{s^{\{2\}}} b_i^{\{2\}} f^{\{2\}}\left(Y_i^{\{2\}}\right).
\end{aligned}
$$

GARK methods for the momentum equation

$$\begin{bmatrix} \theta \\ V \end{bmatrix}' = \underbrace{\begin{bmatrix} 0 \\ F(\theta, V) \end{bmatrix}}_{f^{\{1\}}} + \underbrace{\begin{bmatrix} G(V) \\ 0 \end{bmatrix}}_{f^{\{2\}}}, \tag{9}$$

$$\begin{aligned}
Y_i &= y_n + h \sum_{j=1}^{i} a_{i,j}^{\{1,1\}} F(Y_j) + h \sum_{j=1}^{i} a_{i,j}^{\{1,2\}} G(Z_j), \quad i = 1, \ldots, s^{\{1\}}, \\
Z_i &= y_n + h \sum_{j=1}^{i} a_{i,j}^{\{2,1\}} F(Y_j) + h \sum_{j=1}^{s^{\{2\}}} a_{i,j}^{\{2,2\}} G(Z_j), \quad i = 1, \ldots, s^{\{2\}}, \\
y_{n+1} &= y_n + h \sum_{i=1}^{s^{\{1\}}} b_i^{\{1\}} F(Y_i) + h \sum_{i=1}^{s^{\{2\}}} b_i^{\{2\}} G(Z_i),
\end{aligned} \tag{10}$$

GARK methods for the momentum equation

$$\begin{bmatrix} \theta \\ V \end{bmatrix}' = \underbrace{\begin{bmatrix} 0 \\ F(\theta, V) \end{bmatrix}}_{f^{\{1\}}} + \underbrace{\begin{bmatrix} G(V) \\ 0 \end{bmatrix}}_{f^{\{2\}}}, \tag{11}$$

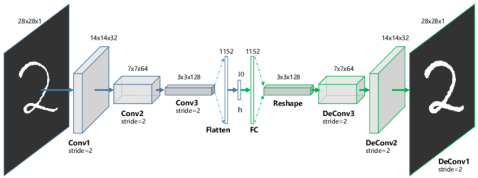
$$\begin{aligned}
Y_i &= \begin{bmatrix} W^Y \\ V^Y \end{bmatrix}_i = y_n + h\sum_{j=1}^{i} a_{i,j}^{\{1,1\}} F(Y_j) + h\sum_{j=1}^{i} a_{i,j}^{\{1,2\}} G(Z_j), \quad i = 1, \ldots, s^{\{1\}}, \\
Z_i &= \begin{bmatrix} W^Z \\ V^Z \end{bmatrix}_i = y_n + h\sum_{j=1}^{i} a_{i,j}^{\{2,1\}} F(Y_j) + h\sum_{j=1}^{s^{\{2\}}} a_{i,j}^{\{2,2\}} G(Z_j), \quad i = 1, \ldots, s^{\{2\}}, \\
y_{n+1} &= y_n + h\sum_{i=1}^{s^{\{1\}}} b_i^{\{1\}} F(Y_i) + h\sum_{i=1}^{s^{\{2\}}} b_i^{\{2\}} G(Z_i),
\end{aligned} \tag{12}$$

$$
\begin{aligned}
W_i^Y &= W_n + h\sum_{j=1}^{i} a_{i,j}^{\{1,2\}} g(V_j^Z), \quad i = 1, \ldots, s^{\{1\}}, \\
V_i^Y &= V_n + h\sum_{j=1}^{i} a_{i,j}^{\{1,1\}} f\left([W_j^Y, V_j^Y]^T\right), \quad i = 1, \ldots, s^{\{1\}}, \\
V_i^Z &= V_n + h\sum_{j=1}^{i} a_{i,j}^{\{2,1\}} f\left([W_j^Y, V_j^Y]^T\right), \quad i = 1, \ldots, s^{\{2\}}, \\
W_{n+1} &= W_n + h\sum_{i=1}^{s^{\{2\}}} b_i^{\{2\}} g(V_i^Z), \\
V_{n+1} &= W_n + h\sum_{i=1}^{s^{\{1\}}} b_i^{\{1\}} f\left([W_i^Y, V_i^Y]^T\right).
\end{aligned}
\tag{13}
$$

note that the term W^Z is not used in any calculation, and can thus be dropped. Notice that the term W_i^Y is quasi-implicit.

$$
\begin{array}{c|c|c}
\mathbf{c}^{\{1,1\}} & \mathbf{A}^{\{1,1\}} & \mathbf{A}^{\{1,2\}} \\
\hline
\mathbf{c}^{\{2,1\}} & \mathbf{A}^{\{2,1\}} & \mathbf{A}^{\{2,2\}} \\
\hline
 & \mathbf{b}^{\{1\}} & \mathbf{b}^{\{2\}}
\end{array}
:=
\begin{array}{c|ccc|cc}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\
1 & 1-\beta & \beta & 0 & \frac{1}{2} & \frac{1}{2} \\
\hline
\frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 \\
\frac{3}{4} & \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{4} \\
\hline
 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \frac{1}{2} & \frac{1}{2}
\end{array}
.
$$

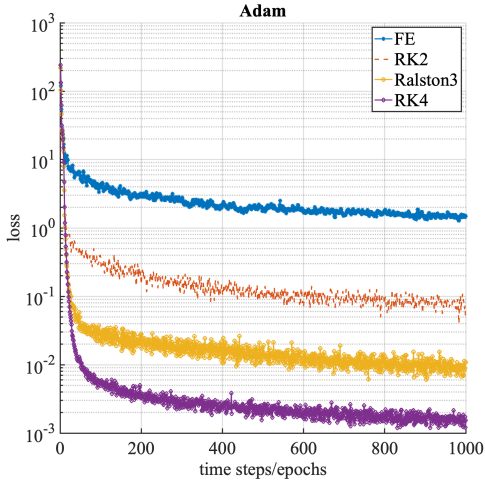
Vanilla `Adam` versus higher order explicit methods



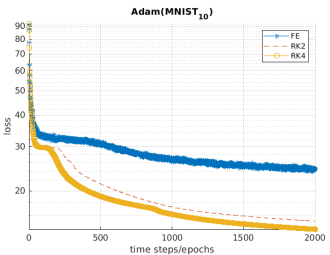
$$X \xrightarrow{\theta} U \xrightarrow{\phi} \widetilde{X},$$

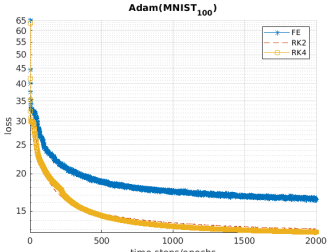
$$L(\widetilde{X}, X) = \left\| X - \widetilde{X} \right\|^2 + \alpha_1 \|\phi\|^2 + \alpha_2 \|\theta\|^2$$

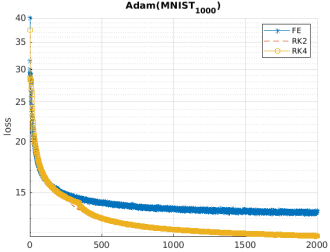
$$\min_{\phi,\theta} L(\widetilde{X}, X) m$$



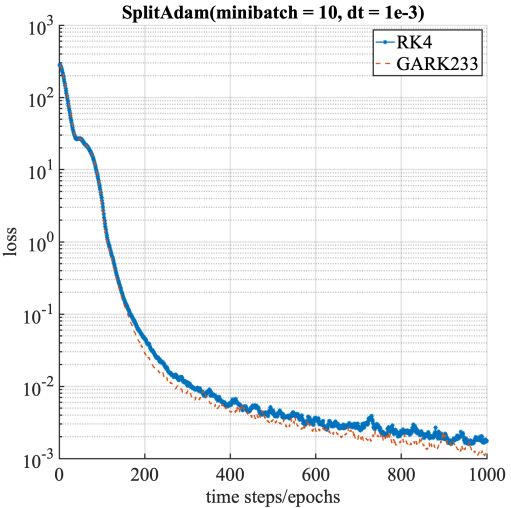
Vanilla `Adam` versus higher order explicit methods

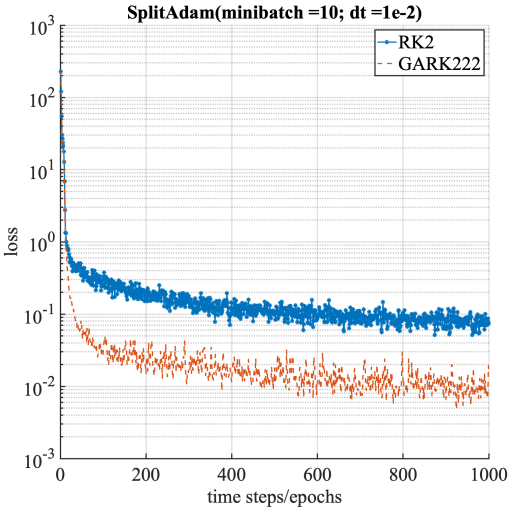






Vanilla `Adam` versus higher order explicit methods





Future directions: Partitioning the linear part

$$\begin{cases} \dot{\theta} = -m/\sqrt{v+\varepsilon} \\ \dot{m} = h\left(t, \lambda, \alpha_1\right)\left(\nabla f(\theta) - m\right) \\ \dot{v} = h\left(t, \lambda, \alpha_2\right)\left(\nabla f(\theta)^2 - v\right) \end{cases}$$

$$\begin{bmatrix} \theta \\ V \end{bmatrix}' = \begin{bmatrix} 0 \\ f(W) \end{bmatrix} + \begin{bmatrix} g(V) \\ BV \end{bmatrix} \tag{14}$$

with $V = [m, v]^T$.

Future directions: Partitioning the linear part

$$\begin{bmatrix} \theta \\ V \end{bmatrix}' = \begin{bmatrix} 0 \\ f(W) \end{bmatrix} + \begin{bmatrix} g(V) \\ BV \end{bmatrix} \tag{15}$$

$$W_i^Y = W_n + h \sum_{j=1} a_{i,j}^{\{1,2\}} g(V_j^Z) \tag{16}$$

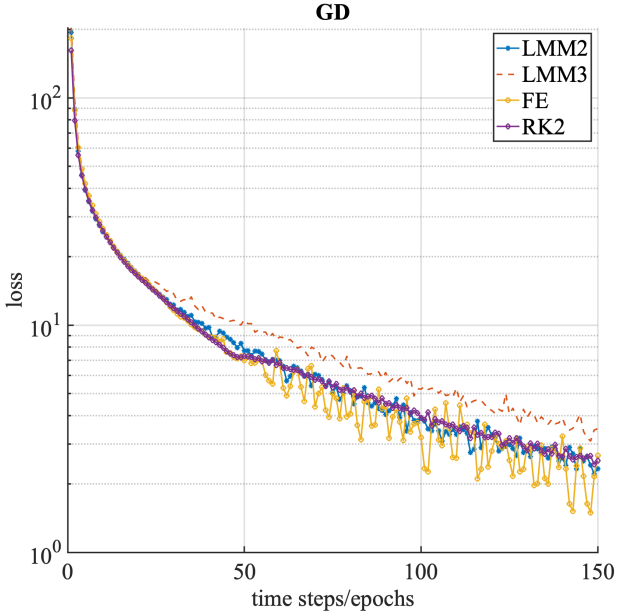
$$V_i^Z = V_n + h \sum_{j=1} a_{i,j}^{\{2,1\}} f(W_j^Y) + h \sum_{j=1}^{i} a_{i,j}^{\{2,2\}} B V_j^Z \tag{17}$$

$$W_{n+1} = W_n + h \sum_{i=1}^{s^{\{2\}}} b_i^{\{2\}} g(V_i^Z), \tag{18}$$

$$V_{n+1} = W_n + h \sum_{i=1}^{s^{\{1\}}} b_i^{\{1\}} f\left(W_i^Y\right) + h \sum_{i=1}^{s^{\{2\}}} b_i^{\{2\}} B V_i^Z \tag{19}$$

Summary and future directions

- ODE form of momentum methods for (stochastic) optimization
- Amenable to IMEX and ADI splittings
- Improved accuracy helps faster convergence to a suitable (local) minimum
- Indications that stability is a stronger requirement than order (Chebyshev explicit methods)
- Extensive testing is required for many machine learning tasks (Classification, Regression, Time-series forcasting, etc.)
- GARK framework allows offers new methods, what are the requirements for a successful one? (SDE analysis)
- arxiv.org/a/sarshar_a_1
- csl.cs.vt.edu


GD
LMM2
LMM3
FE
RK2
loss
10^2
10^1
10^0
0
50
100
150
time steps/epochs