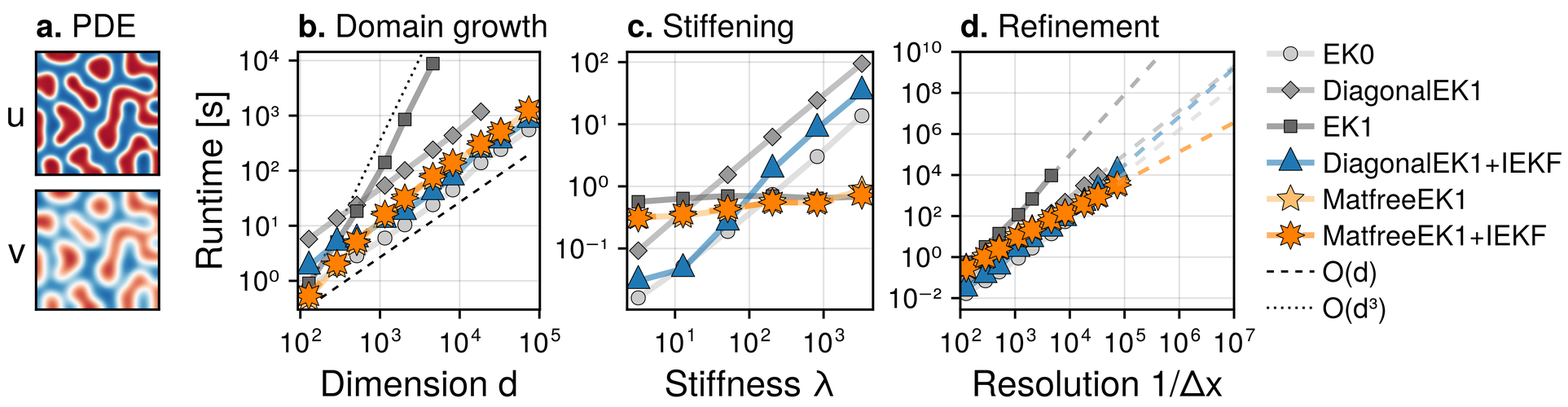
Stable and Scalable Probabilistic Numerical Solvers for Stiff and High-Dimensional ODEs

Nathanael Bosch · Mathematics for Materials Modelling (MatMat), EPFL



Established ODE filters are either *scalable* or *stable*. `MatfreeEK1` achieves both.

The goal: Probabilistic numerical ODE solutions

Given an ***ordinary differential equation*** (ODE)

$$\dot{y}(t) = f(y(t), t), \qquad y(0) = y_0$$

compute a ***probabilistic numerical ODE solution***

$$p\big(y(t) \mid y(0) = y_0, \{\dot{y}(t_n) - f(y(t_n), t_n) = 0\}_{n=1}^{N}\big)$$

The problem: stability vs. scalability

Solver	cost / step	stability	Jacobian F_y
`EK1`	$\mathcal{O}(d^3)$	A-stable	exact
`ExpEK`	$\mathcal{O}(d^3)$	L-stable	exact
`DiagonalEK1`	$\mathcal{O}(d)$	✗	$\approx \mathrm{diag}(F_y)$
`EK0`	$\mathcal{O}(d)$	✗	$= 0$

Classical numerics closes this gap with **iterative linear algebra** and **matrix-free Jacobian products**.
We use these to develop a stable $\mathcal{O}(d)$ ODE filter.

ODE filtering: probabilistic model

Prior: Gauss–Markov process

$$x(0) \sim \mathcal{N}(\mu_0, \Sigma_0) \qquad x(t_n) \mid x(t_{n-1}) \sim \mathcal{N}(\Phi_n x(t_{n-1}), Q_n)$$

typically a q-times integrated Wiener process with dimension $D = d \cdot (q+1)$; models $y(t) = E_0 x(t)$, $\dot{y}(t) = E_1 x(t)$.

Observation model / information operator:

$$z_n = h_n(x(t_n)) := E_1 x(t_n) - f(E_0 x(t_n), t_n) \stackrel{!}{=} 0.$$

ODE filtering: the EKF step

Extended Kalman filtering: Recursively compute

$$\begin{aligned}
\text{Predict:} \quad & \mu_n^- = \Phi_n \mu_{n-1} \\
& \Sigma_n^- = \Phi_n \Sigma_{n-1} \Phi_n^T + Q_n \\
\text{Linearize:} \quad & H_n = E_1 - F_y E_0, \\
& b_n = F_y E_0 \mu_n^- - f(E_0 \mu_n^-, t_n), \\
\text{Update:} \quad & K_n = \Sigma_n^- H_n^T (H_n \Sigma_n^- H_n^T)^{-1}, \\
& \mu_n = \mu_n^- + K_n(0 - (H_n \mu_n^- + b_n)), \\
& \Sigma_n = (I - K_n H_n) \Sigma_n^- (I - K_n H_n)^T.
\end{aligned}$$

This gives $p(x(t_n) \mid z_{1:n}) = \mathcal{N}(\mu_n, \Sigma_n)$.

ODE filter cost analysis

Cheap operations: Multiplication with

- *Selection matrices* $E_0, E_1 \in \mathbb{R}^{d \times D}$ — $\mathcal{O}(d)$
- *Block-diagonal* matrices $\Phi_n, Q_n \in \mathbb{R}^{D \times D}$ — $\mathcal{O}(d)$

Costly operations:

- Forming and storing the Jacobian $F_y \in \mathbb{R}^{d \times d}$ — $\mathcal{O}(d^2)$
- Matrix-vector products with F_y, H_n, Σ_n^- — $\mathcal{O}(d^2)$
- Matrix-matrix products with F_y, H_n, Σ_n^- — $\mathcal{O}(d^3)$
- Solving a linear system with $(H_n \Sigma_n^- H_n^T)$ — $\mathcal{O}(d^3)$

⇒ **Without additional structure, one step costs $\mathcal{O}(d^3)$**

How `EK0` & `DiagonalEK1` are $\mathcal{O}(d)$

With $F_y \approx 0$ (`EK0`) and $F_y \approx \mathrm{diag}(F_y)$ (`DiagonalEK1`), block-diagonal is preserved throughout the solve. **This leads to $\mathcal{O}(d)$; *but the approximate F_y costs stability*.**

Contribution 1: `MatfreeEK1`

1) Matrix-free exact EKF mean updates in linear time.
Computing μ_n requires matrix-vector multiplications with $E_0, E_1, F_y, \Sigma_n^-, (H_n \Sigma_n^- H_n^T)^{-1}$. *This can be done in $\mathcal{O}(d)$:*

- E_0, E_1: *Selection matrices* so multiplication is cheap.
- F_y: Use **JVPs/VJPs**, at cost comparable to f.
- Σ_n^-: If block-diagonal then multiplication is cheap.
- $(H_n \Sigma_n^- H_n^T)^{-1}$: Given the assumptions above about cheap MVPs, solve with iterative linear algebra.

2) Stochastic block-diagonal covariance updates.
The above requires a block-diagonal Σ_n. We therefore compute only the blocks of Σ_n, stochastically with

$$\xi_k = (I - K_n H_n)\sqrt{\Sigma_n^-}\eta_k, \qquad \eta_k \sim \mathcal{N}(0, I)$$

$$[\Sigma_n]_{i,i} \approx \frac{1}{m} \sum_{k=1}^{m} (\xi_k)_i (\xi_k)_i^T,$$

This requires the same matrix-vector products and linear solves as for the mean update (only m of them).

⇒ **$\mathcal{O}(d)$ all while using the full Jacobian.**

Contribution 2: `IEKF`

Iterating the EKF linearization and update steps until convergence is a Gauss–Newton algorithm for computing

$$\mu_n = \operatorname*{argmax}_x p(x \mid z_{1:n}).$$

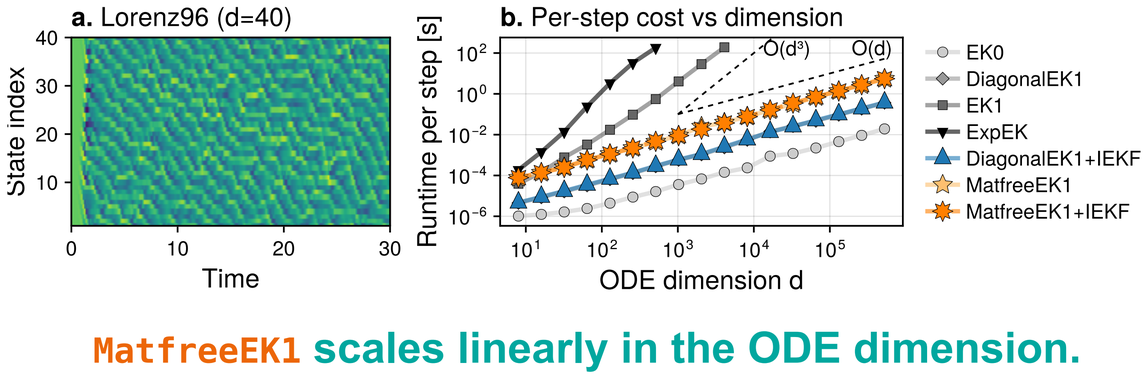
As this is an implicit equation in f, computing the MAP with the IEKF leads to ***fully implicit* ODE filters.**

References

- `EK1`: Tronarp, Kersting, Särkkä & Hennig, *Statistics and Computing*, 2019.
- `ExpEK`: Bosch, Hennig & Tronarp, "Probabilistic exponential integrators", *NeurIPS* 2023.
- `EK0` and `DiagonalEK1` for high-dimensional ODEs: Krämer, Bosch, Schmidt & Hennig, "Probabilistic ODE solutions in millions of dimensions", *ICML* 2022.

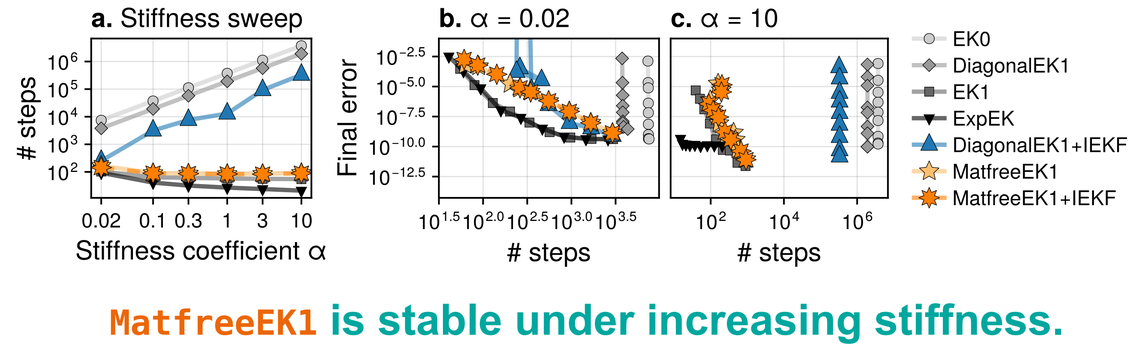
Experiments

Scaling in dimension



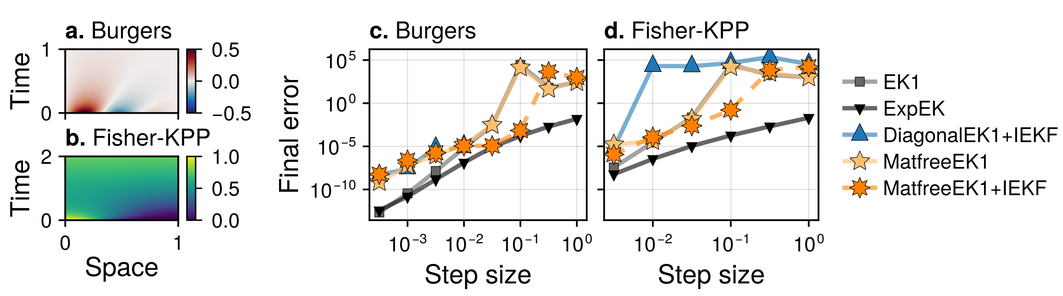
`MatfreeEK1` scales linearly in the ODE dimension.

Stability under stiffness



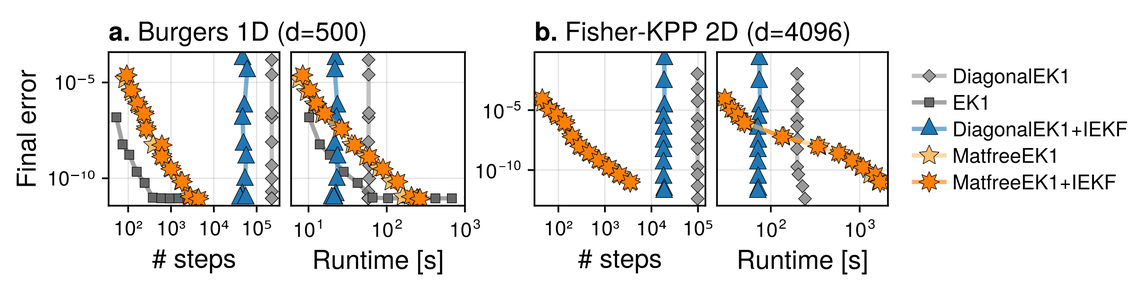
`MatfreeEK1` is stable under increasing stiffness.

Fixed step sizes on stiff PDEs



`MatfreeEK1` is as stable as `EK1`; `IEKF` helps at large Δt.

Adaptive solves on large PDEs



`MatfreeEK1` is stable and scalable on large stiff PDEs.

Nathanael Bosch nathanaelbosch.github.io nathanael.bosch@epfl.ch