Fokker-Planck Prediction on the Cylindric Manifold using Tensor Decomposition of a Regular Grid

Daniel Frisch and **Uwe D. Hanebeck**

SDF 2025, Bonn, Germany

Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

isas.iar.kit.edu

Fokker-Planck Prediction

Input

Given

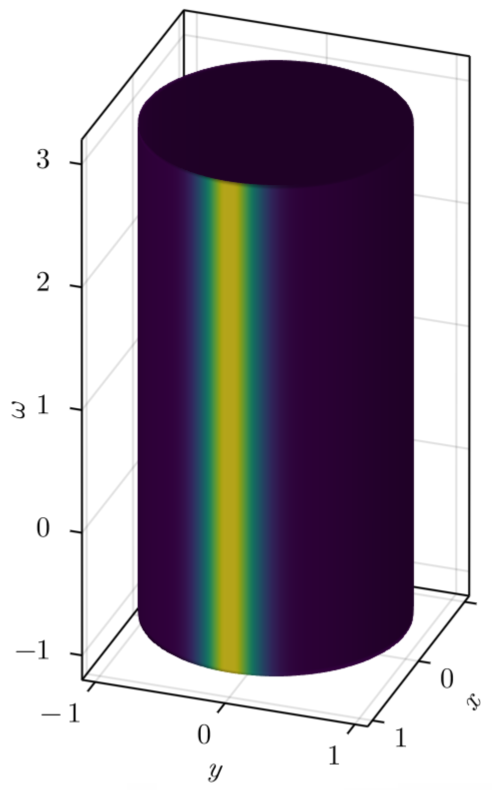
- System Model
- Density (Low-Rank Matrix)

→ **Predictor** →

Output

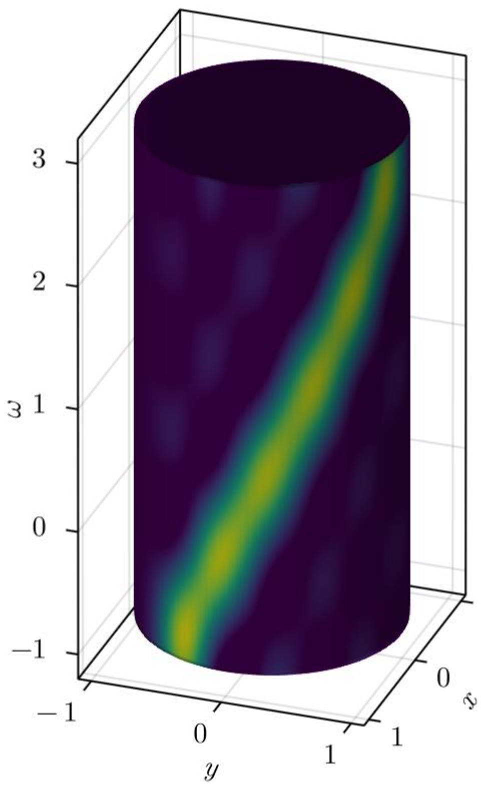
Prediction

- Density (Low-Rank Matrix)



Methods

- Tensorized Predictor
- Manifold-Aware



Fokker-Planck Prediction

Input

Given

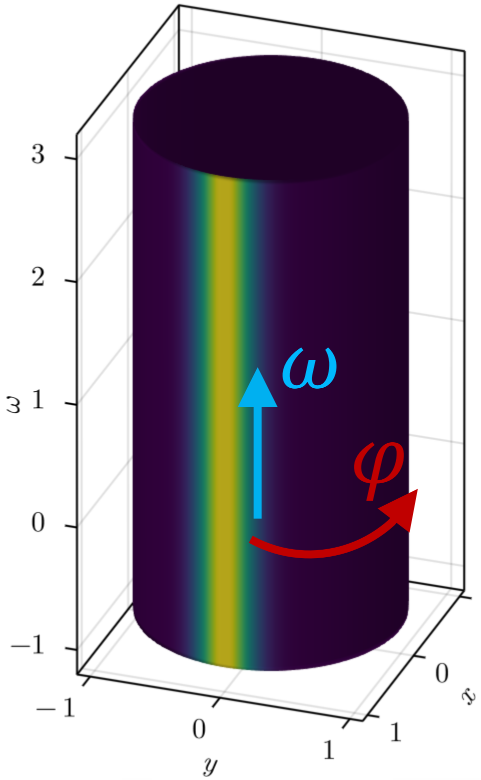
- System Model
- Density (Low-Rank Matrix)

Predictor

Output

Prediction

- Density (Low-Rank Matrix)

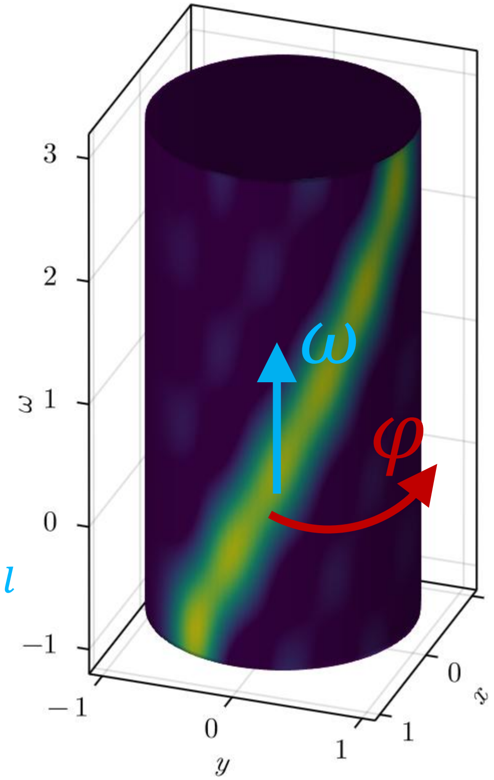


Methods

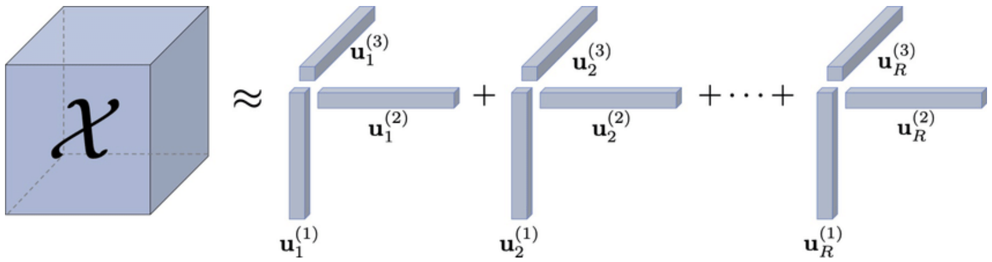
- Tensorized Predictor
- Manifold-Aware

$$\mathbf{P}(0) = \underline{\rho}_{\varphi} \cdot \underline{\rho}_{\omega}^{\top}$$

$$\mathbf{P}(1) = \sum_{l=1}^{L} \underline{\rho}_{\varphi,l} \cdot \underline{\rho}_{\omega,l}^{\top}$$



Alternative Names



Credits: Panagakis, Kossaifi, Chrysos, Tensor Methods in Computer Vision and Deep Learning

Many labels for the same thing!

- Tensor Decomposition
- Tensor rank decomposition
  - Matrix rank decomposition, e.g. SVD
- Kronecker Tensor Format
- CANDECOMP / PARAFAC (CP) Decomposition
  - canonical decomposition / parallel factor
- Canonical Polyadic Decomposition (CPD)

Singular Value Decomposition (SVD)

General Basics

$$\mathbf{A} = \sum_{i=1}^{D} \sigma_i \cdot \underline{u}_i \cdot \underline{v}_i^{\top}$$

size

singular vectors

singular values

Original

Singular Value Decomposition (SVD)

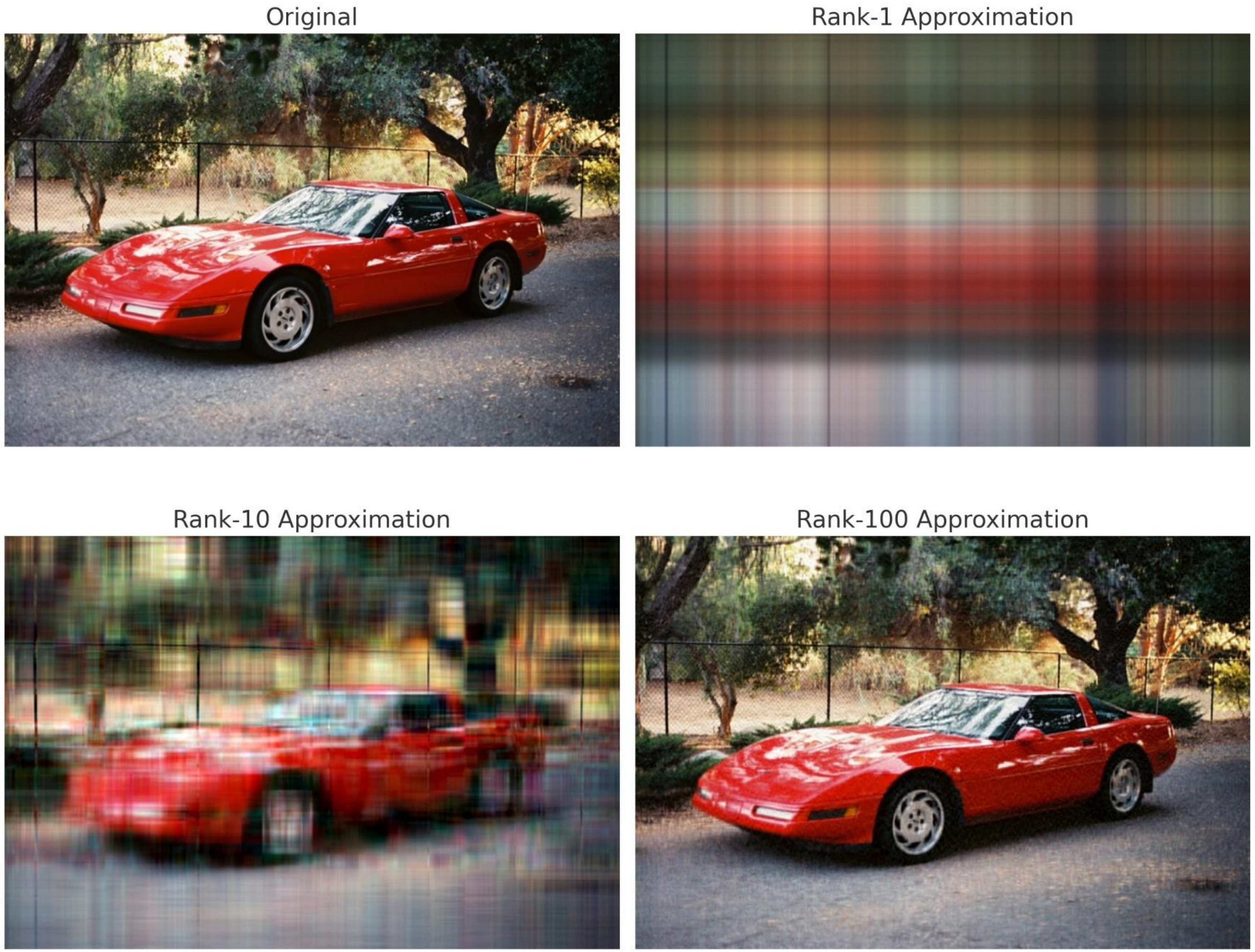
General Basics

size, singular vectors, singular values:

$$\mathbf{A} = \sum_{i=1}^{D} \sigma_i \cdot \underline{u}_i \cdot \underline{v}_i^{\top}$$

rank:

$$\mathbf{A} \approx \sum_{i=1}^{R} \sigma_i \cdot \underline{u}_i \cdot \underline{v}_i^{\top}$$

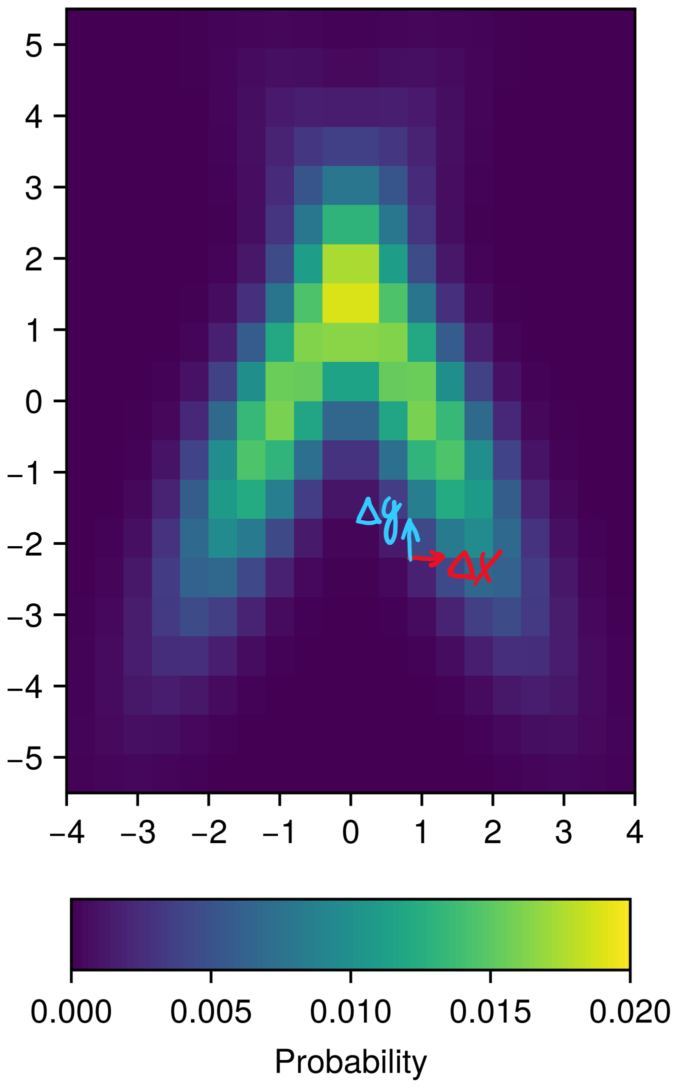


Low-Rank Density Representation

- Discretization
  - regular grid
  - spacing: Δx, Δy

- Rank Decomposition
  - SVD

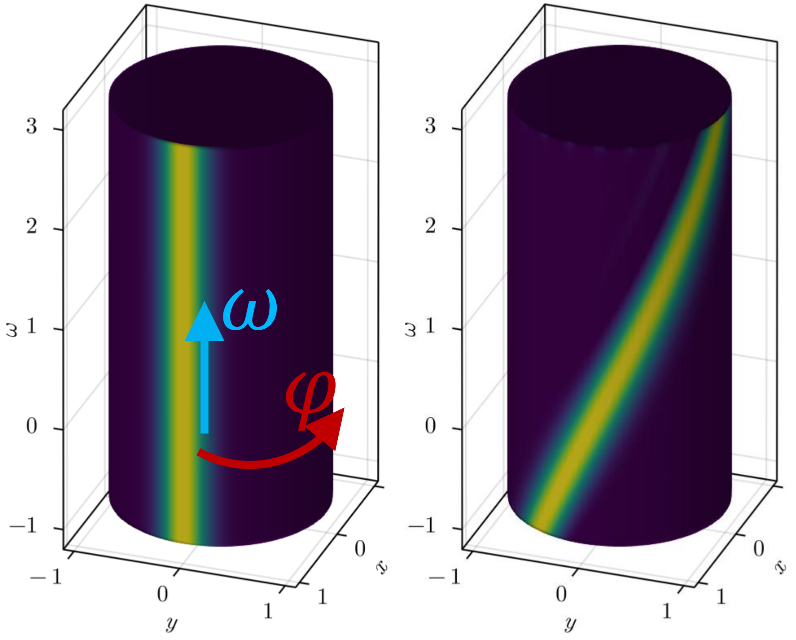
$$\mathbf{P} \approx \sum_{l=1}^{L} \underbrace{\underline{\rho}_{x,l} \cdot \underline{\rho}_{y,l}{}^{\top}}_{\text{loading vectors}}$$



- State Space $\underline{x} = \begin{bmatrix} \varphi \\ \omega \end{bmatrix} \in \mathbb{S} \times \mathbb{R}$

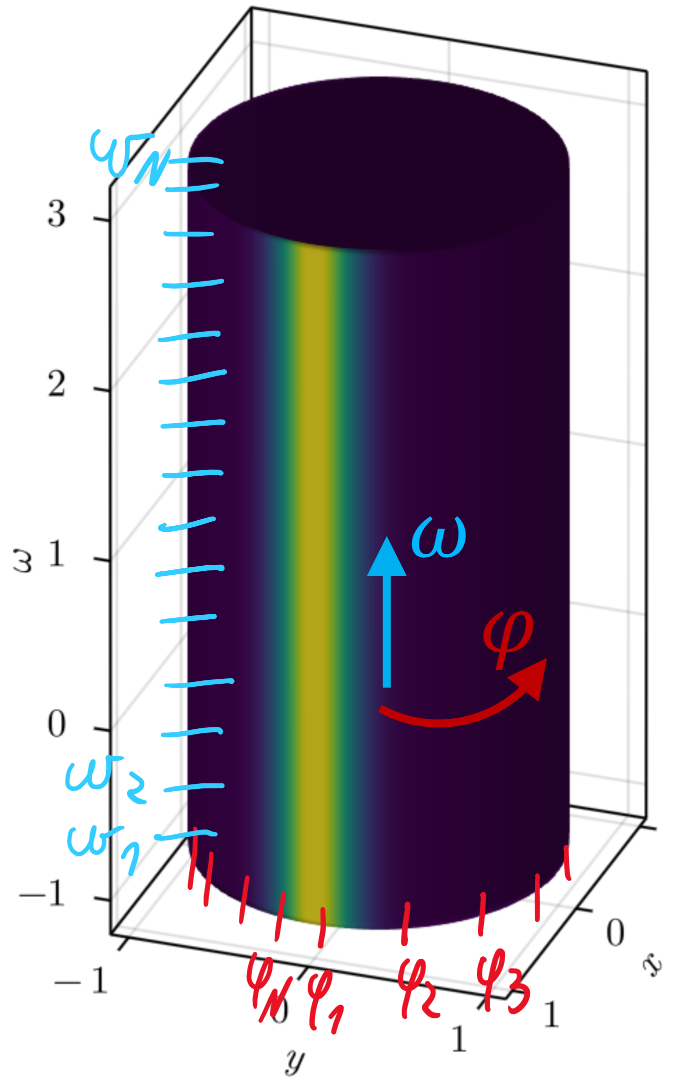
- Wiener Process $\mathrm{d}\underline{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \underline{x}\,\mathrm{d}t + q \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathrm{d}\omega$

- Fokker-Planck Equation $\frac{\partial p}{\partial t} = \underbrace{-\omega \frac{\partial p}{\partial \varphi}}_{\text{translation}} + \underbrace{\frac{q}{2} \frac{\partial^2 p}{\partial \omega^2}}_{\text{diffusion}}$



Discretization: State Space

- Grid φ, ω
- say, $N_\varphi = N_\omega = 100$



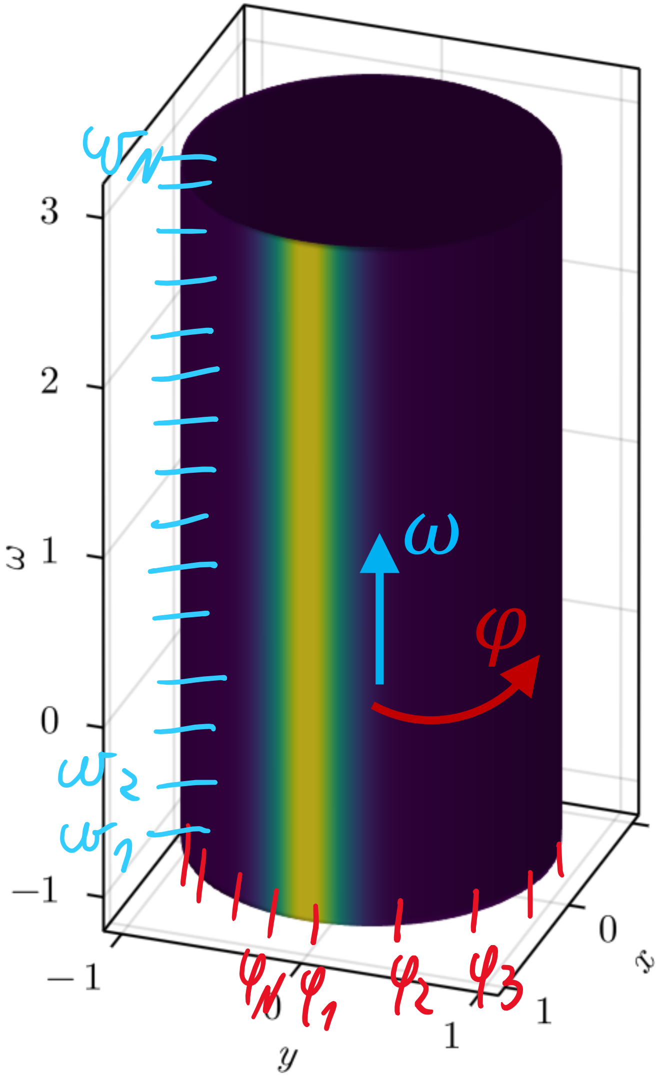
- Scalar field $p(\varphi, \omega)$

- Matrix $\mathbf{P} = \sum_{l=1}^{L} \underline{\rho}_{\varphi,l} \cdot \underline{\rho}_{\omega,l}^{\top}$

  [100 x 100]

- Vector $\underline{p} = \mathrm{vec}(\mathbf{P}) = \sum_{l=1}^{L} \underline{\rho}_{\omega,l} \otimes \underline{\rho}_{\varphi,l}$

  [10000 x 1]



Discretization: Differentials → Finite Differences

PDE

$$\frac{\partial p}{\partial t} = -\omega \frac{\partial p}{\partial \varphi} + \frac{q}{2} \frac{\partial^2 p}{\partial \omega^2}$$

[10000 x 1]

$$\frac{\partial \underline{p}}{\partial t} = \mathbb{L}_1 \underline{p} + \mathbb{L}_2 \underline{p}$$

ODE

[10000 x 10000]

$$\mathbf{D}_\varphi = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \mathbf{-1} \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ \mathbf{1} & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

[100 x 100]

$$\mathbf{D}_\omega = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

[100 x 100]

PDE → Fokker-Planck Operator

$$\frac{\partial p}{\partial t} = -\omega \frac{\partial p}{\partial \varphi} + \frac{q}{2} \frac{\partial^2 p}{\partial \omega^2}$$ PDE

[10000 x 1] $$\frac{\partial \underline{p}}{\partial t} = \underbrace{(\mathbb{L}_1 + \mathbb{L}_2)}_{\text{rank 2}} \underline{p}$$ ODE

$$\underbrace{\mathbb{L}_1}_{[10000 \times 10000]} = \underbrace{-\text{diag}(\underline{\omega})}_{[100 \times 100]} \otimes \underbrace{\mathbf{D}_\varphi}_{[100 \times 100]}$$

rank 1

$$\underbrace{\mathbb{L}_2}_{[10000 \times 10000]} = \underbrace{\mathbf{D}_\omega}_{[100 \times 100]} \otimes \underbrace{\left(\frac{q}{2}\mathbf{I}(N_\varphi)\right)}_{[100 \times 100]}$$

rank 1

Fokker-Planck Predictors

- Matrix Exponential
  - very slow: [10000 x 10000] eigen decomposition

$$\underline{p}(t-\tau) = \exp\{\tau(\mathbb{L}_1 + \mathbb{L}_2)\} \cdot \underline{p}(t)$$

- ODE Solver
  - slow due to [10000 x 10000] matrix multiplications

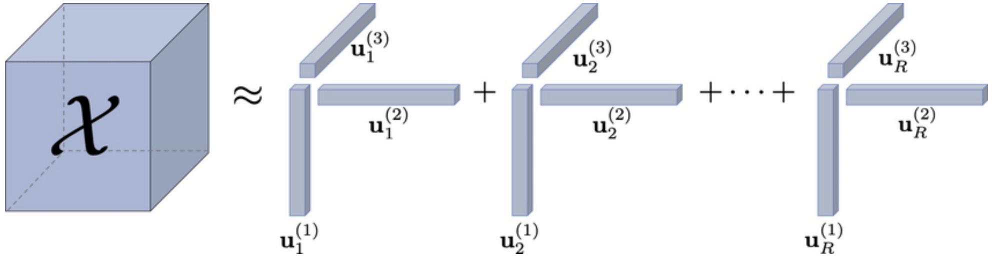
$$\frac{\partial \underline{p}}{\partial t} = \mathbb{L}_1 \underline{p} + \mathbb{L}_2 \underline{p}$$

- Tensorized Predictor
  - directly transforms loading vectors
  - purely via [100 x 100] matrices

… introduced in:

- Fusion 2016, Demissie, Khan, Govaers
- JAIF 2019, Govaers, Demissie, Khan, Ulmke, Koch
- →

Tensorized Predictor: Approximations



- Exploits & requires low-rank structure
  - of densities $\underline{p}$
  - of operator $\mathbb{L}$

- Approximations
  - Lie-Suzuki-Trotter $\exp\{\tau(\mathbb{L}_1 + \mathbb{L}_2)\} \approx \exp\{\tau\mathbb{L}_1\} \cdot \exp\{\tau\mathbb{L}_2\}$
  - Taylor series expansion $\exp\{\tau\mathbf{D}\} \approx \sum_{p=0}^{N_T} \frac{\tau^p}{p!} \cdot \mathbf{D}^p$
  - $\Rightarrow$ trade-off: step size τ vs N_T

Tensorized Predictor: Formula

prediction, $L \cdot N_T^2$ loading vectors — transformation — input, L loading vectors

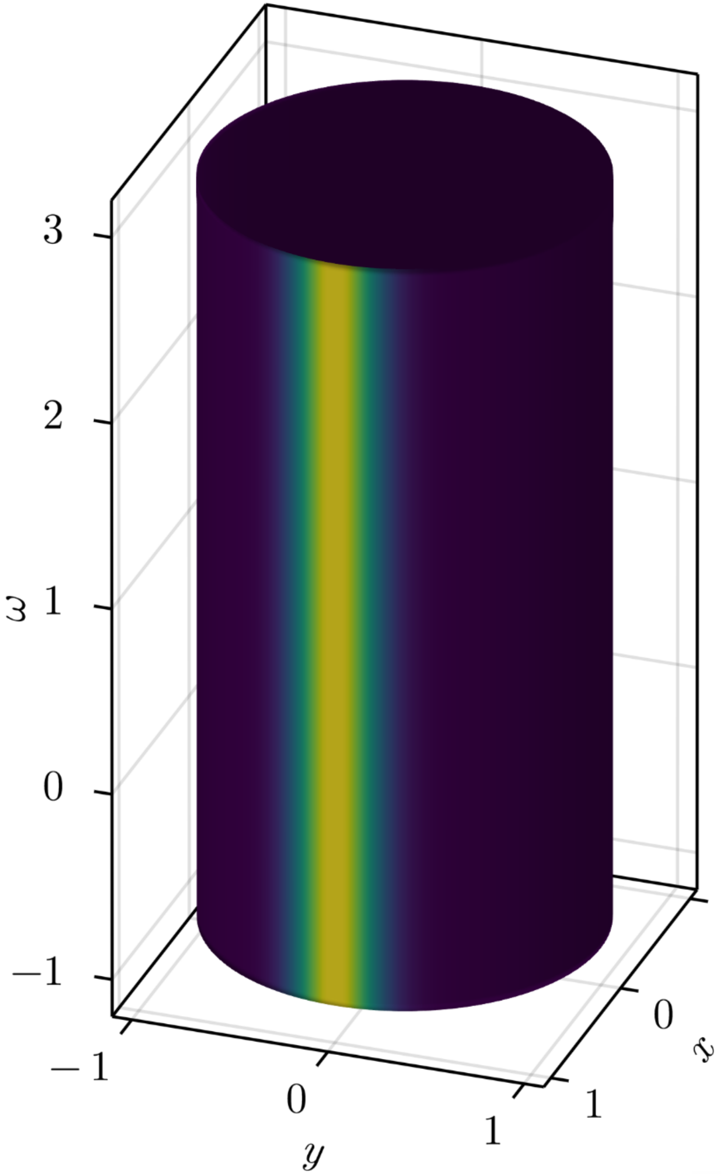
$$\underline{\rho}_{\varphi,l,p_1,p_2}(t+\tau) = \sqrt{\frac{\tau^{p_1+p_2}}{p_1! + p_2!}} \cdot V_{\varphi,1} D_{\varphi,1}^{p_1} V_{\varphi,1}^\top \cdot V_{\varphi,2} D_{\varphi,2}^{p_2} V_{\varphi,2}^\top \cdot \underline{\rho}_{\varphi,l}$$

[100 x 1] [100 x 100] [100 x 100] [100 x 1]

$$\underline{\rho}_{\omega,l,p_1,p_2}(t+\tau) = \sqrt{\frac{\tau^{p_1+p_2}}{p_1! + p_2!}} \cdot V_{\omega,1} D_{\omega,1}^{p_1} V_{\omega,1}^\top \cdot V_{\omega,2} D_{\omega,2}^{p_2} V_{\omega,2}^\top \cdot \underline{\rho}_{\omega,l}$$

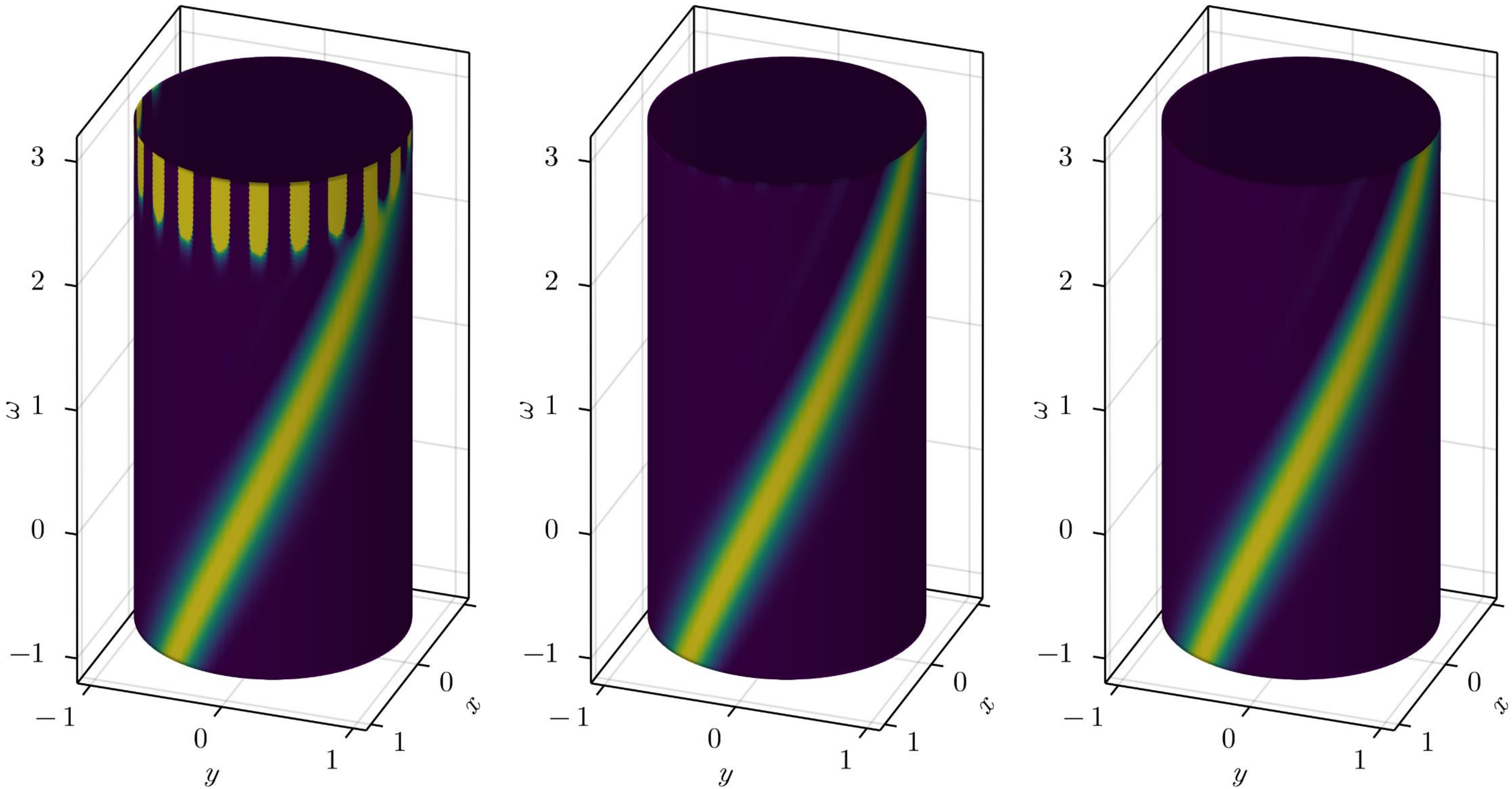
```
function fp_step_simple!(ρPred_φ, ρPred_ω, ρInit_φ, ρInit_ω, NTaylor, τ)
    for p1 = 0:NTaylor, p2 = 0:NTaylor
        fac = sqrt(τ^p1/factorial_list[p1+1]) * sqrt(τ^p2/factorial_list[p2+1])
        ρPred_φ[:,:,p1+1,p2+1] .= fac * (Vφ1 * (Dφ1^p1 * (Vφ1' * (Vφ2 * (Dφ2^p2 * (Vφ2' * ρInit_φ))))))
        ρPred_ω[:,:,p1+1,p2+1] .= fac * (Vω1 * (Dω1^p1 * (Vω1' * (Vω2 * (Dω2^p2 * (Vω2' * ρInit_ω))))))
    end
end
```

- Rotating shaft
- Encoder
- After 1st measurement →
  - φ: VMF
  - ω: uniform
  - $L = 1$



Evaluation: Single-Step

- Prediction
- $\tau = 0.5$
- $N_T = 40$
  - artifacts
- $N_T = 55$
  - enough
- ODE
  - slow



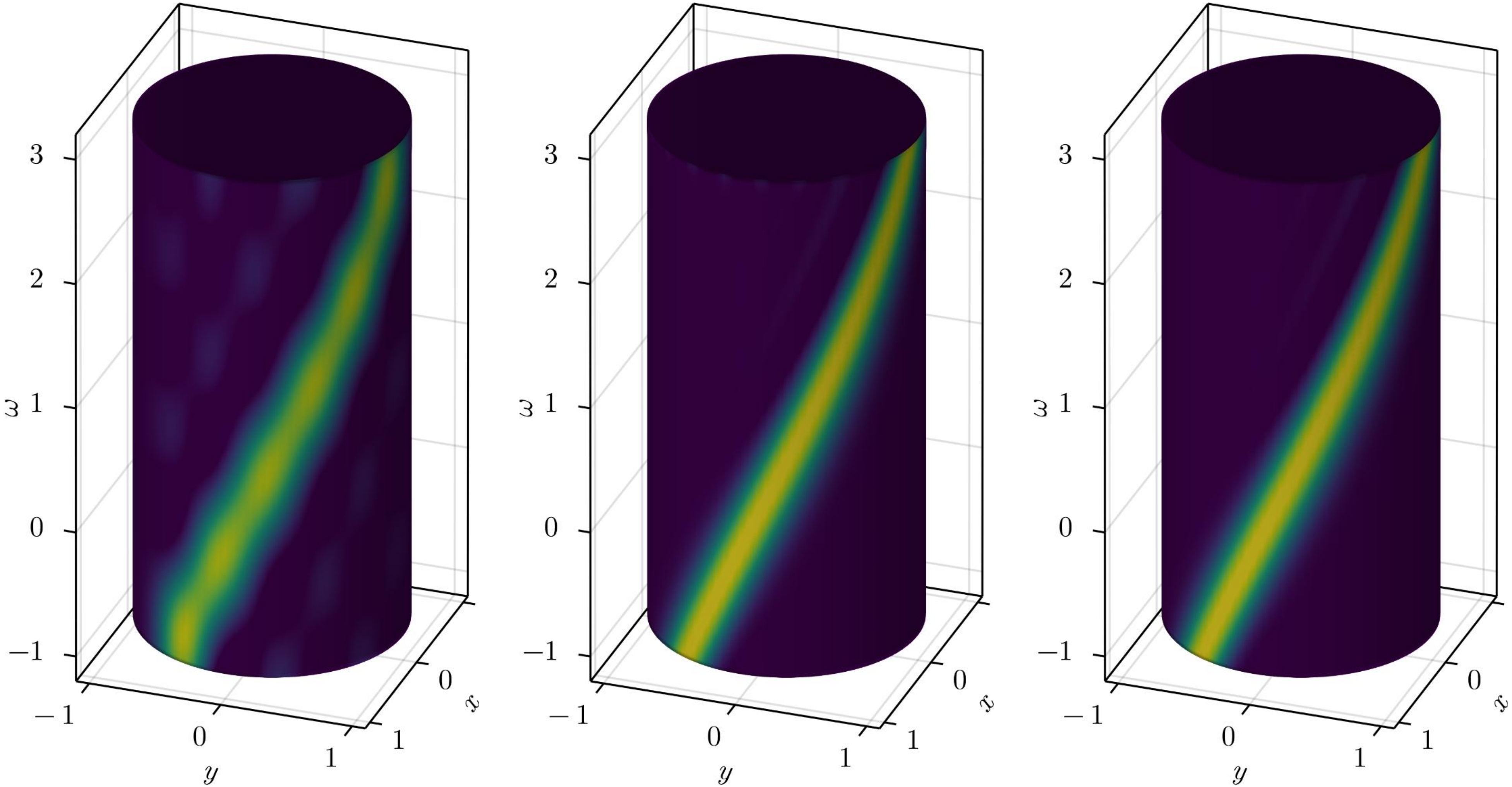
$N_T = 40$, 0.077s, $L = 40^2 = 1600$

$N_T = 55$, 0.15s, $L = 55^2 = 3025$

ground truth
ODE solver, 2.3s

Evaluation: SVD-Reduction

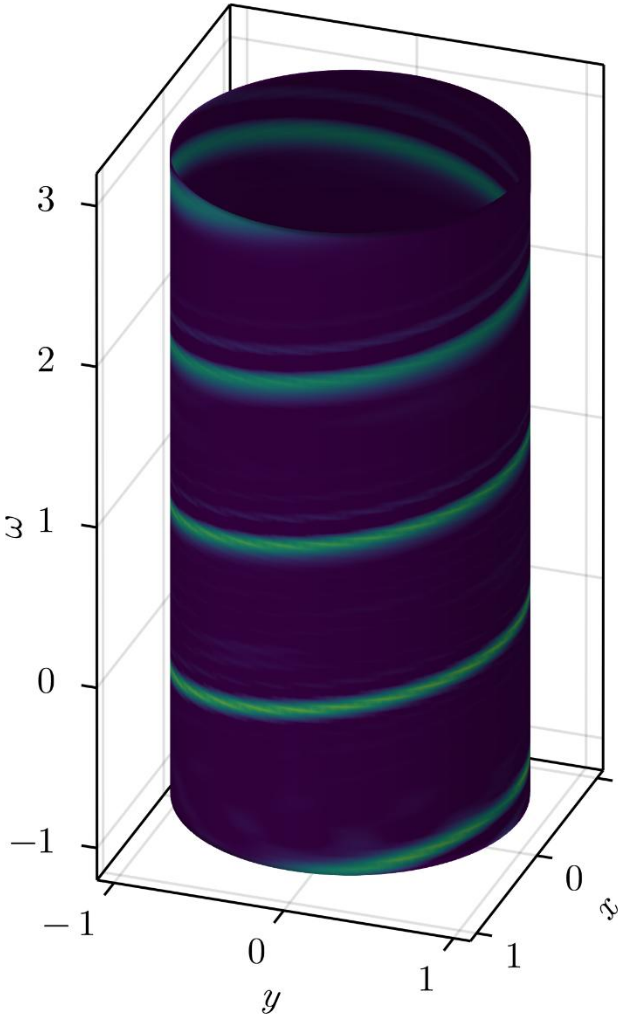
- Reduce L
- via SVD
- 3025→6
- 3025→15



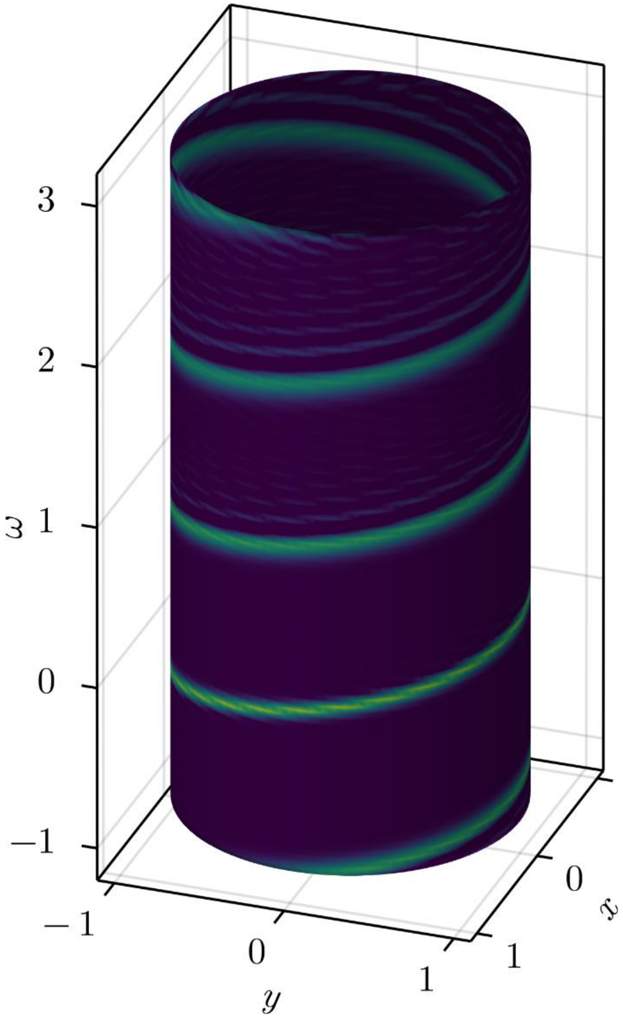
$L = 6$ | $L = 15$ | ground truth ODE solver

Evaluation: Recursive Multi-Step Predictor

- Arbitrary large steps: $\tau = 2\pi$
- Trade-off used
  - $\tau_k = 0.06$ step size
  - $N_T = 4$ Taylor terms
  - $L = 20$ rank reduction
  - 1.8s computation time



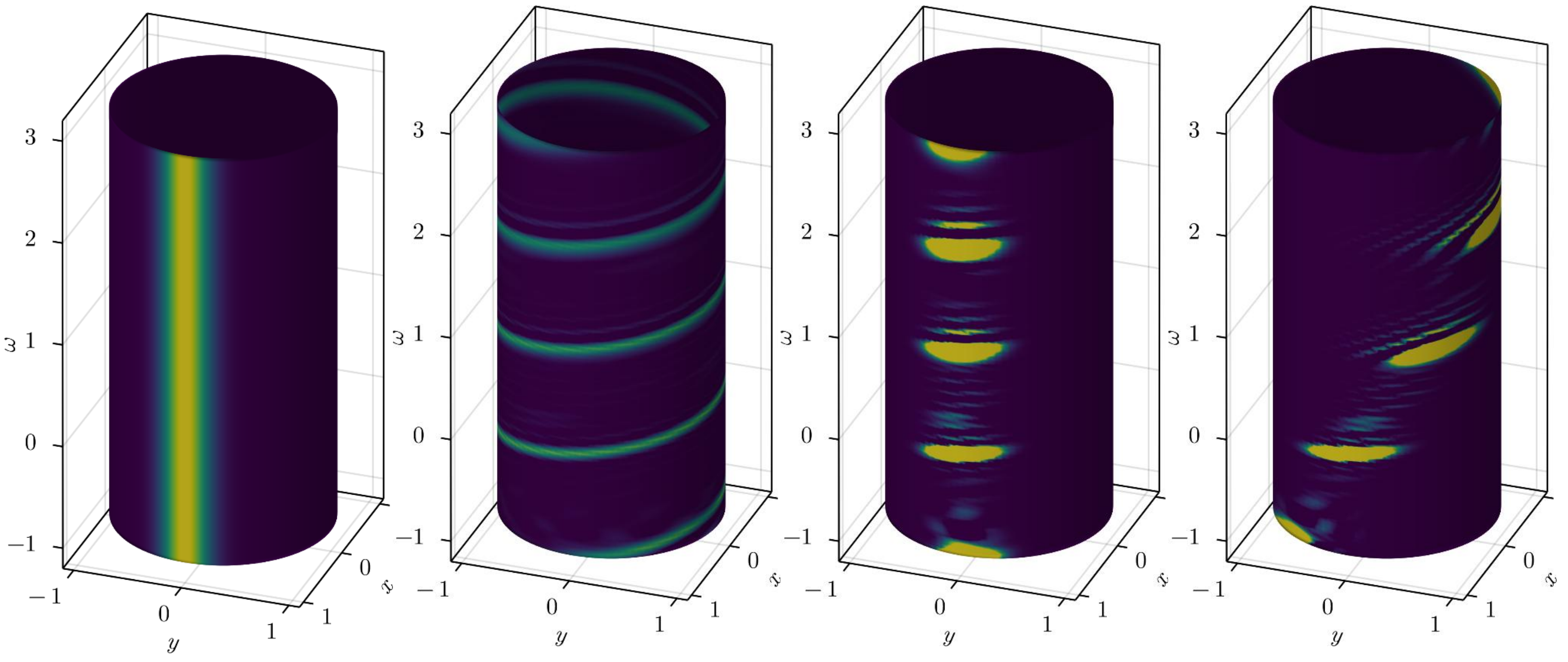
Multi-step FPE predictor
1.8s



ground truth
ODE solver, 32s

Evaluation: Tensorized Filter Step

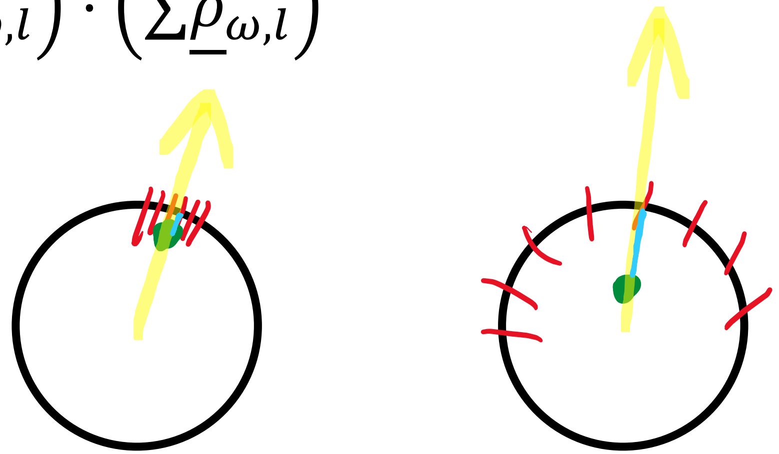
- Required:
  - low-rank likelihood
  - here: rank-1
- 2nd measurement
  - multi-modal result
- 3rd measurement
  - unique result



$p_e(t = 0)$ $p_p(t = 2\pi)$ $p_e(t = 2\pi)$ $p^p(t = 2.25\pi)$

Further Tensorized Computations

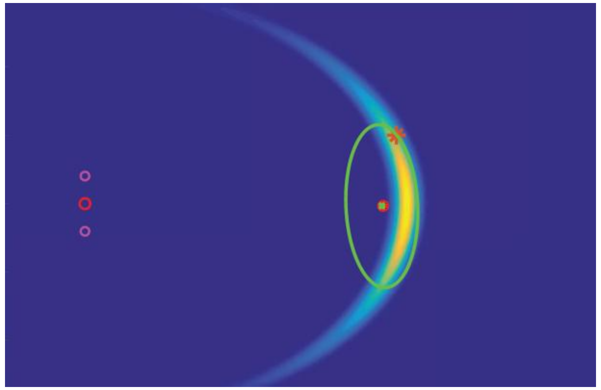
- Prediction step (see above)
- Filter step $\underline{\rho}^{e}_{\varphi,l,l'} = \underline{\rho}^{p}_{\varphi,l} \odot \underline{\rho}^{L}_{\varphi,l'}\,; \quad \underline{\rho}^{e}_{\omega,l,l'} = \underline{\rho}^{p}_{\omega,l} \odot \underline{\rho}^{L}_{\omega,l'}$
- Normalization constant $\mathrm{c} = \sum_{l=1}^{L} \left(\sum \underline{\rho}_{\varphi,l}\right) \cdot \left(\sum \underline{\rho}_{\omega,l}\right)$
- Mean of ω $\mu_\omega = \sum_{l=1}^{L} \left(\sum \underline{\rho}_{\varphi,l}\right) \cdot \left(\sum \underline{\omega} \odot \underline{\rho}_{\omega,l}\right)$
- Variance of ω $\sigma_\omega^2 = \sum_{l=1}^{L} \left(\sum \underline{\rho}_{\varphi,l}\right) \cdot \left(\sum \underline{\omega} \odot \underline{\omega} \odot \underline{\rho}_{\omega,l}\right) - \mu_\omega^2$
- Trigonometric moment $m_1 = \sum_{l=1}^{L} \left(\sum e^{i\underline{\varphi}} \odot \underline{\rho}_{\varphi,l}\right) \cdot \left(\sum \underline{\rho}_{\omega,l}\right)$
- Circular mean of φ $\mu_\varphi = \mathrm{ang}(m_1)$
- Circular variance of φ $\sigma_\varphi^2 = 1 - |m_1|$



Use Cases

Demonstrated Applications

- [Govaers 2019]
  - multitarget PHD
  - 16D SLAM
  - bistatic radar



- [Frisch 2025] (this)
  - rotating shaft

Nonlinear Filtering

- nonlinear periodic domains
- combined Euclidean and periodic domains
- robotic: joint angles & position

Posterior

- density on grid
- [directional] mean
- [directional] variance

Conclusion

Limitations (state of art)

- Linear domain (+ nonlinear estimation problem)
- Euclidean state space

Advantages (proposed)

- Nonlinear domain
  - always nonlinear estimation!
- Grid + periodic space
  - good match
- Periodic state spaces
  - Cylinder
  - Torus
  - SE(2)

Thank you for your attention

Intelligent

Sensor-Actuator-Systems