10/06/2025 Thesis Defense

Data Driven Decoupling of Multivariate Functions

Nathan Bouquet
Supervisor: Philippe Dreesen
Examiner: Martijn Boussé

Problem definition

We aim to express a multivariate function f(u) as a linear combination of univariate functions g, each applied to combinations of the input variables using transformation matrices W and V.

Using Canonical Polyadic Decomposition (CPD), the function is represented in decoupled form as:

$$\mathbf{f}(\mathbf{u}) = \mathbf{W}\mathbf{g}(\mathbf{V}^T\mathbf{u})$$

Problem definition

Why?

- Simplifies modeling
- Useful in system identification, signal processing, control systems

My contribution?

Extension of the decoupling method into a **data driven** approach

Problem definition: decoupling pipeline

1) Computing the Jacobian matrix of the function F with respect to the input variables, arranging them in a tensor
2) Applying Canonical Polyadic Decomposition (CPD) to extract transformation matrices W, V and the univariate structure, using TensorLab

Research question

How can we accurately estimate the Jacobian tensor from data, and how does this affect the precision of the decoupling?

Sub questions:

- Given access to function evaluations, how accurately can the Jacobian be estimated using finite difference methods?
- How accurately can Jacobians be estimated from only input-output data using local regression techniques?
- How do the number of sample points and the number of nearest neighbors k affect the quality of the Jacobian estimation and the resulting CP decomposition?
- How robust and precise are these estimation strategies when applied to non-linear functions beyond multivariate polynomials?

Validating the Symbolic Pipeline: Finite Differentiation

Methods:

- Forward difference
- Central difference
- Complex-step method.

$$\frac{\partial f_j}{\partial u_k} \approx \frac{f_j(\mathbf{u} + h\mathbf{e}_k) - f_j(\mathbf{u})}{h}$$

Formula for forward

Parameters:

- Step size h
- Number of points N

$$\frac{\partial f_j}{\partial u_k} \approx \frac{\mathrm{Im}(f_j(\mathbf{u} + ih\mathbf{e}_k))}{h}$$

Formula for complex-step

Test Functions and Sampling

Functions:

$$F_1 = \begin{bmatrix} (x+y)^2 \\ (x-y)^3 \end{bmatrix}, \quad F_2 = \begin{bmatrix} \sin(x+y) + z^2 \\ xyz + \cos(z) \end{bmatrix}, \quad F_3 = \begin{bmatrix} \exp(xy) + \sin(z) \\ \log(1 + x^2 + y^2) \\ (x+y+z)^3 \end{bmatrix}$$

Sampling:

- F1 sampled from $[0, 1]^m$,
- F2, F3 sampled from $[-0.5, 0.5]^m$

Experiment setup:

- Step sizes > 10^{-5} to avoid numerical instabilities,
- 20 monte carlo trials per configuration,
- Error bars show ±1 standard deviation.

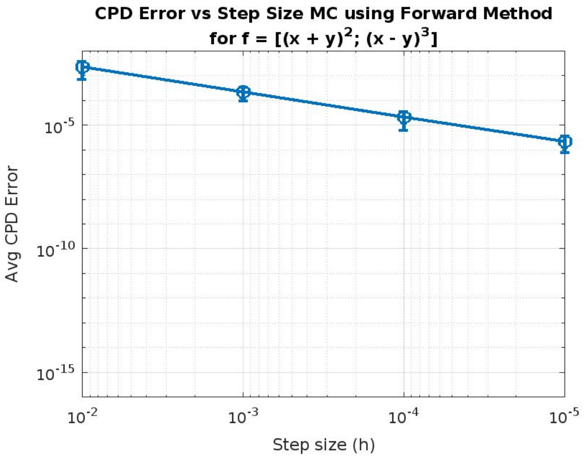
Finite Difference Experiments

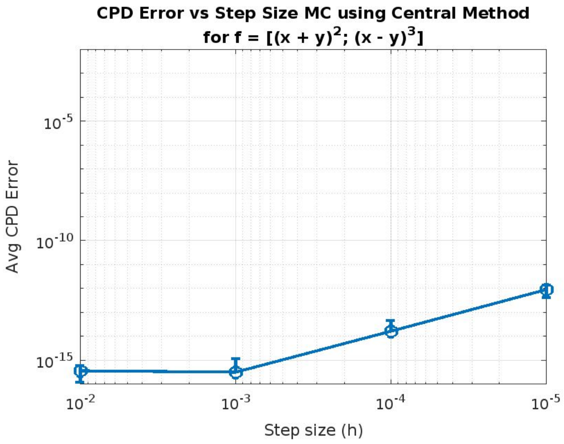
- CPD error vs. step size
- CPD error vs. number of sample points
- True vs. reconstructed function output

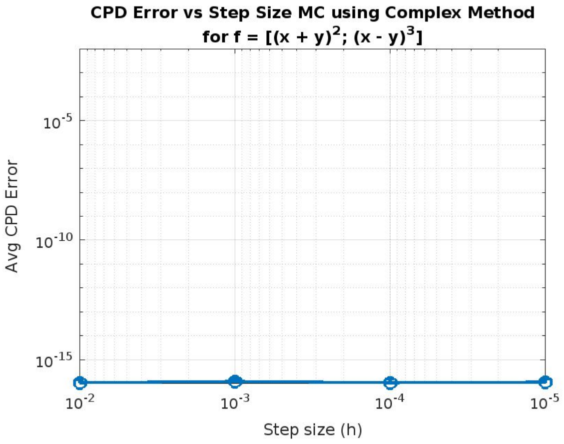
Other evaluations:

- Jacobian slice error vs. step size
- Factor matrices W, V vs. step size

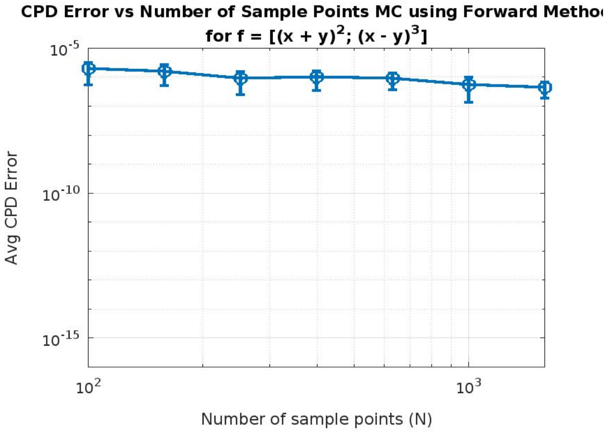
CPD Error vs. Step Size

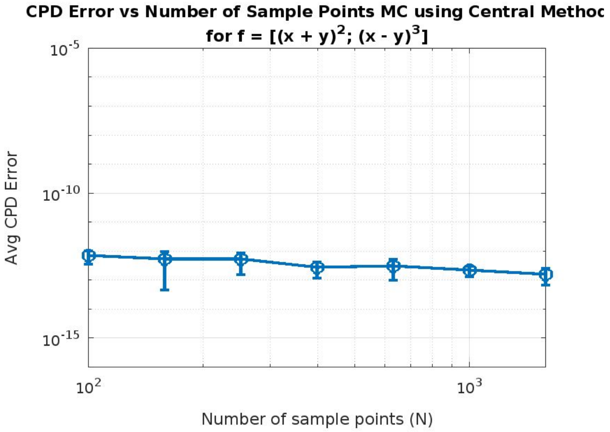


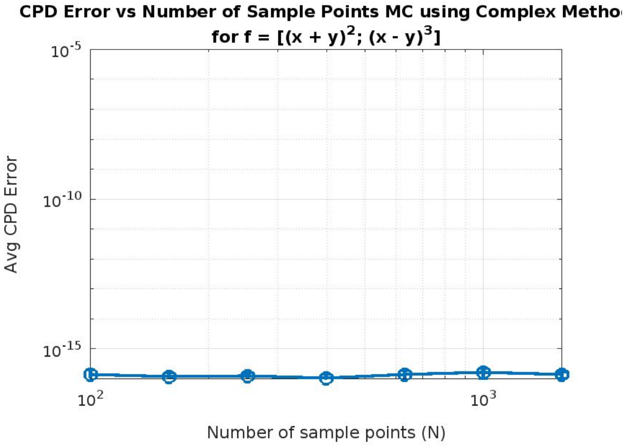




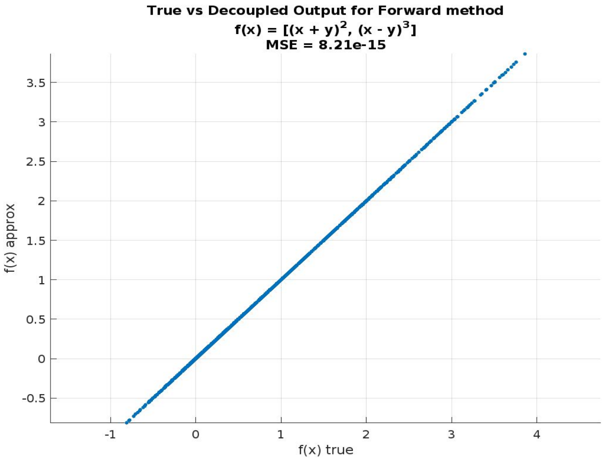
CPD error vs. number for sample points

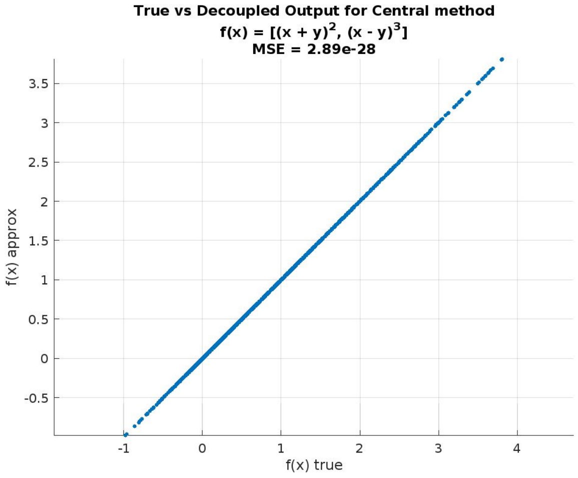


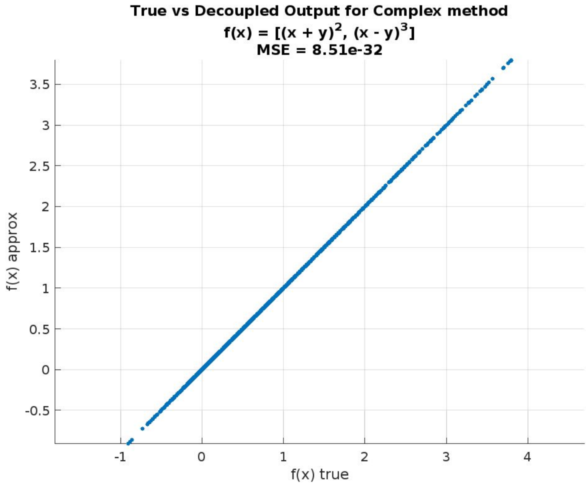




True vs. Reconstructed Output Accuracy







Diagonal alignment indicates accurate decoupling.

From Finite Differences to Data-Driven Jacobian Estimation

- Symbolic pipeline validated (CPD error of 10^{-16} with complex-step)
- But assumes access to function evaluations
- In practice: only input-output data

→ Estimate Jacobians from sampled data

Jacobian Estimation via Local Regression

- Local linear regression
- Regularized local linear regression
- Local polynomial regression

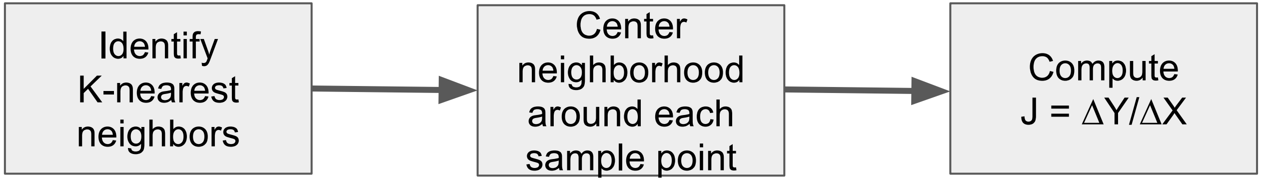
$$\mathbf{J}^{(i)} \approx \Delta\mathbf{Y}\,\Delta\mathbf{X}^{-1}$$

$$\mathbf{J}^{(i)} \approx \Delta\mathbf{Y}\,\Delta\mathbf{X}^{\top}(\Delta\mathbf{X}\,\Delta\mathbf{X}^{\top} + \lambda\mathbf{I})^{-1}$$

$$\Delta\mathbf{y}_l^{(i)} \approx \Phi^{(i)}\mathbf{c}_l^{(i}, \quad \text{for } l = 1, \ldots, n$$

Method parameters:

- Number of neighbors k
- Number of points N
- Regularization parameter λ



Jacobian Estimation via Global Polynomial Regression

- Fit a global multivariate polynomial $f(x)$ to input-output data
- Extract Jacobian tensor directly from the polynomial coefficients

Method parameters:

- Number of points N
- Polynomial degree d

Data-driven Jacobian Estimation Experiments

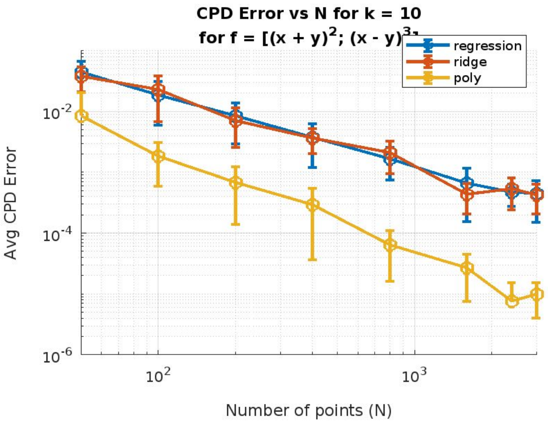
Local regressions:

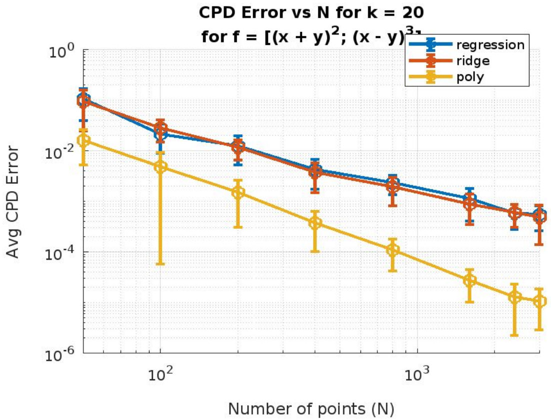
- CPD error vs. number of points and K-nearest neighbors
- CPD error vs. number of points, K-nearest neighbors and lambda;
- Local linear regression vs. ridge regression (with fixed λ)

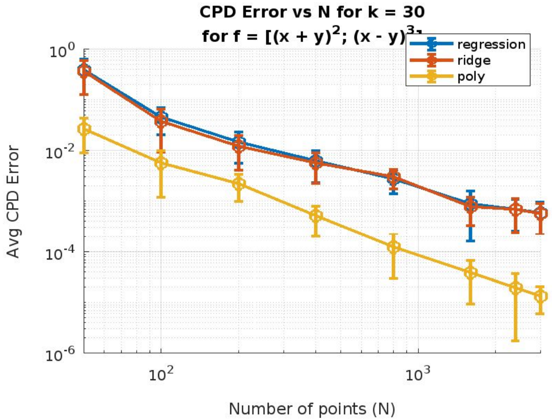
Global polynomial regression:

- CPD error vs. number of sample points

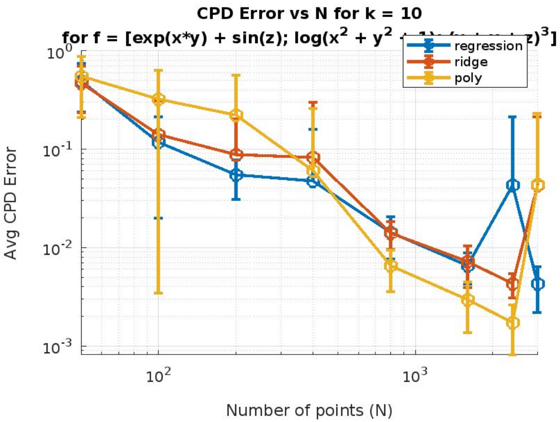
Local Regression: CPD Error vs. Number of Points and Neighbors (Function 1)

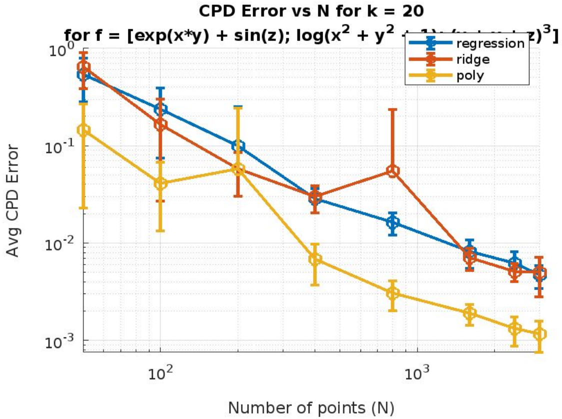


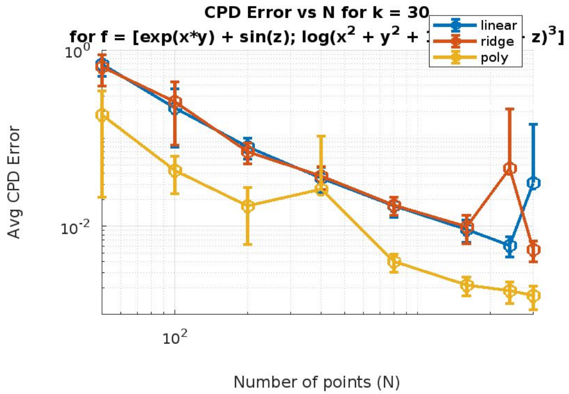




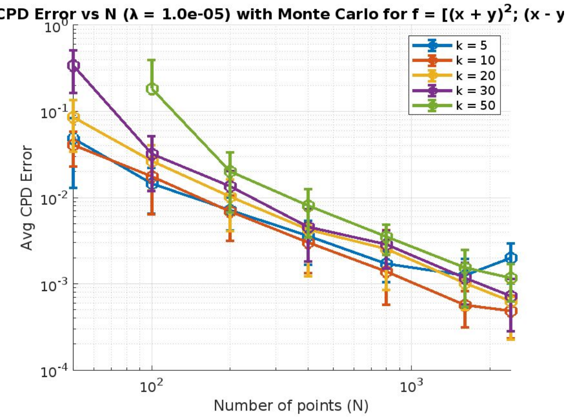
Local Regression: CPD Error vs. Number of Points and Neighbors (Function 3)

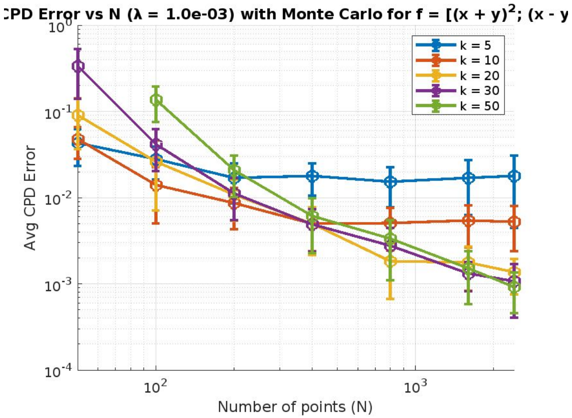


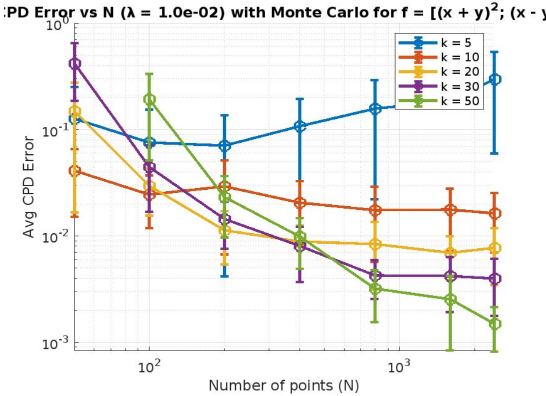




Regularized Local Linear Regression: CPD Error vs. Number of Points, Neighbors and Lambda







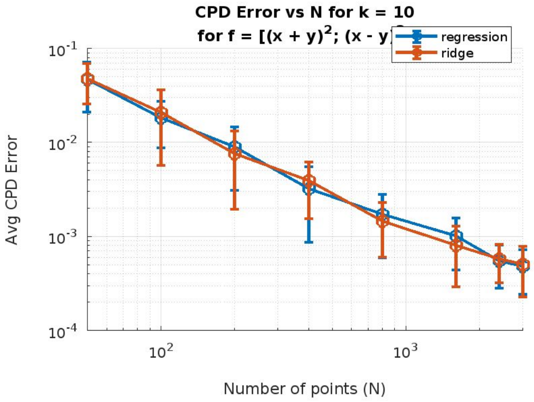
Local Linear Regression vs. Ridge Regression: CPD error comparison

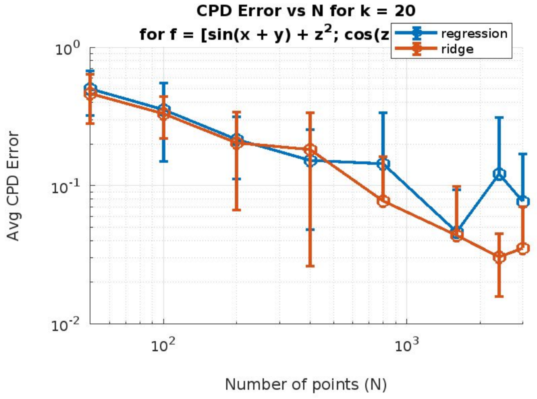
Best performing method per function:

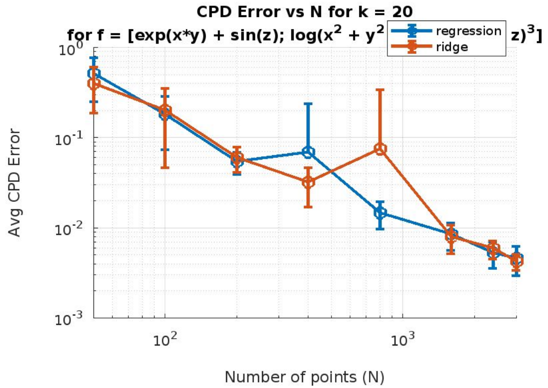
F1 → Linear Regression: CPD error = 3.92e-04

F2 → Ridge Regression: CPD error = 2.72e-02

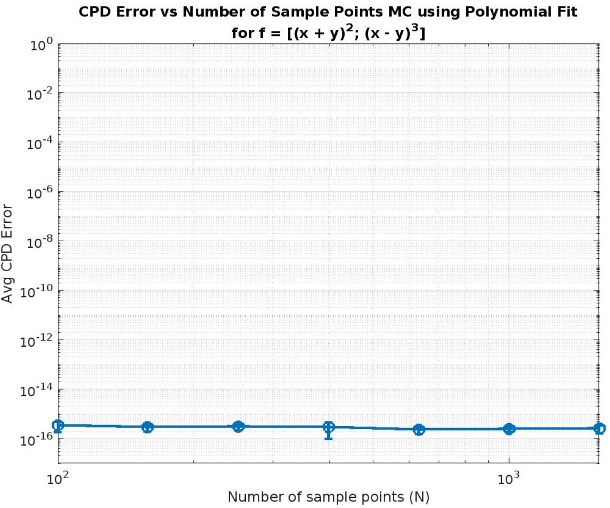
F3 → Ridge Regression: CPD error = 3.69e-03

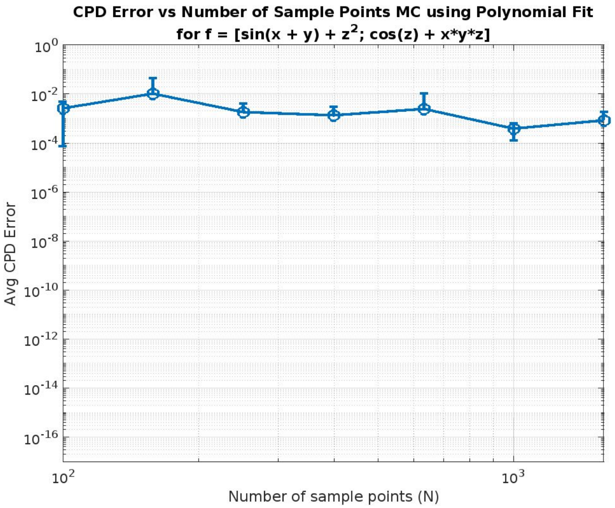


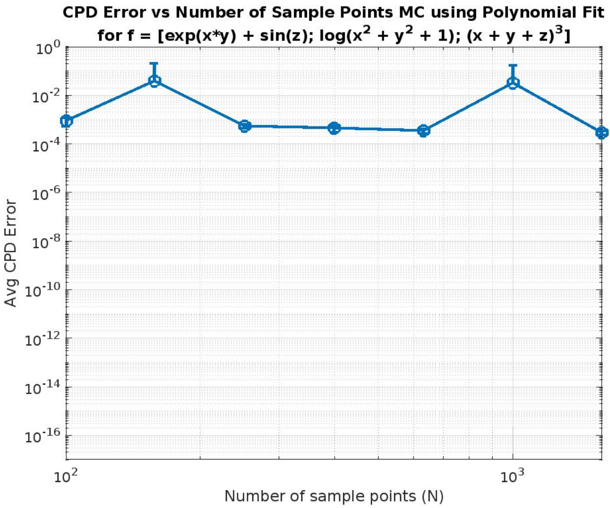




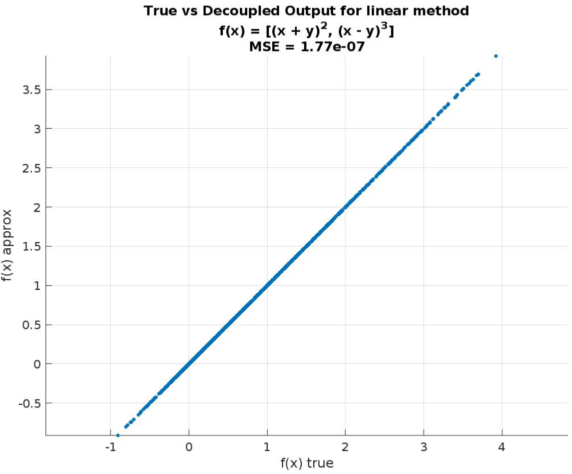
Global Polynomial Regression: CPD Error vs. Number of Points

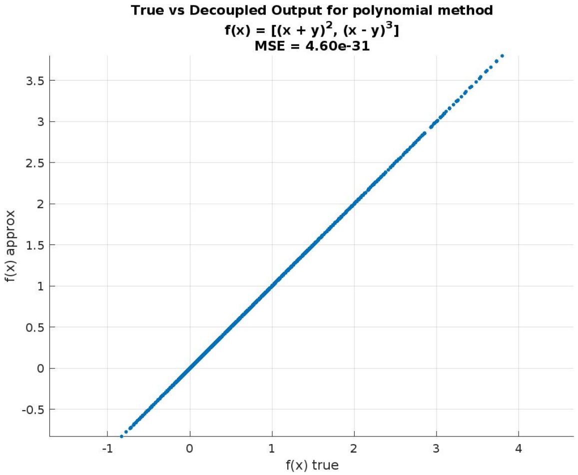


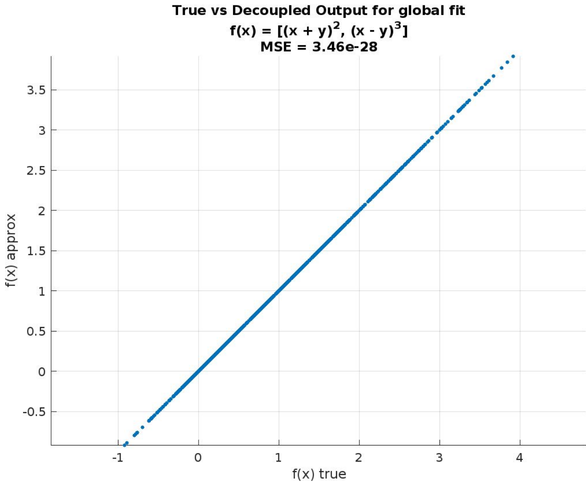




True vs. Reconstructed Output Accuracy







Diagonal alignment indicates accurate decoupling.

Summary

- **Goal:** Decouple multivariate functions using only input-output data
- **Approach:** Estimate Jacobian tensors from data and apply CPD
- **Methods:**
  - Local regression methods (linear, ridge, polynomial)
  - Global polynomial regression
- **Evaluation:**
  - Benchmarked against symbolic Jacobian
  - Tested across increasing function complexity

Conclusion

- **Complex-step differentiation**

  → Most accurate when function evaluations are available

- **Global polynomial regression:**

  → Most effective when function is polynomial

- **Local polynomial regression**

  → Best overall method for data-driven Jacobian estimation

- **Local Ridge regression**

  → Improves robustness in nonlinear or ill-conditioned regions

Opening Perspectives

- Investigate the impact of different sampling strategies:
    - Varying sampling density
    - Alternative input distribution
- Develop more robust Jacobian estimation methods
- Apply the methods to real-world datasets

Thank you for your attention

Questions?

Experimental Observations

- Local Polynomial regression achieves the best overall accuracy

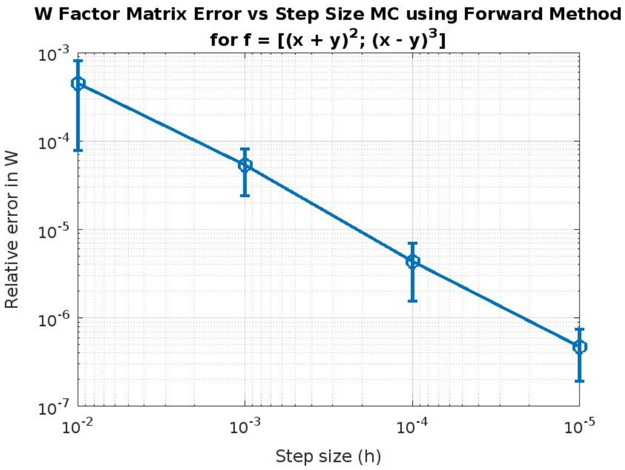
  → CPD error = 6.82×10^{-6} at N = 3000, k = 10.

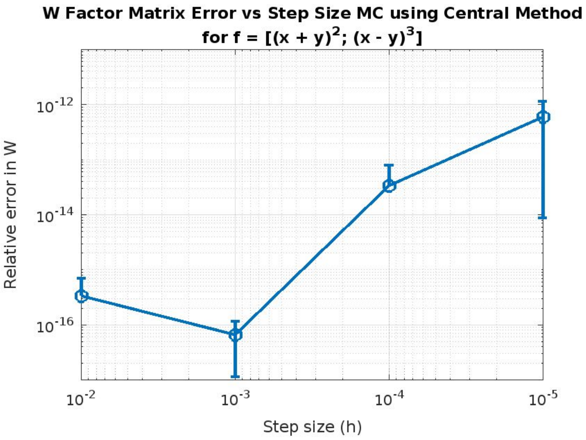
- Ridge regression outperforms local linear regression on non-linear functions
- Global polynomial regression is highly accurate for polynomial functions

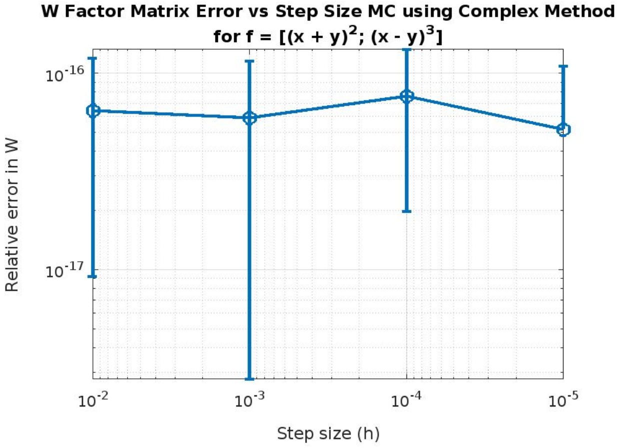
  → CPD error = 10^{-16}

  → But fails to generalize on non-polynomial functions (F2, F3)

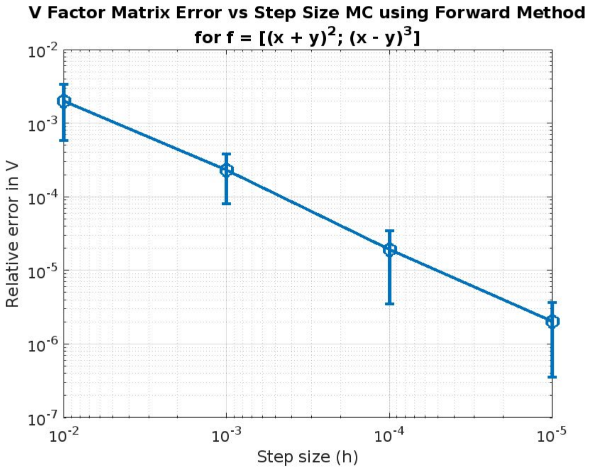
Factor matrices W vs. Step size

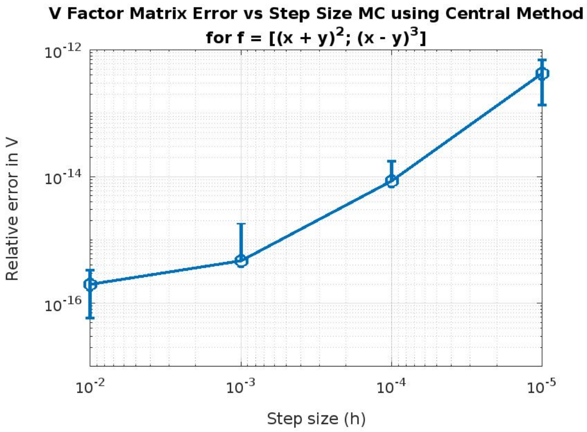


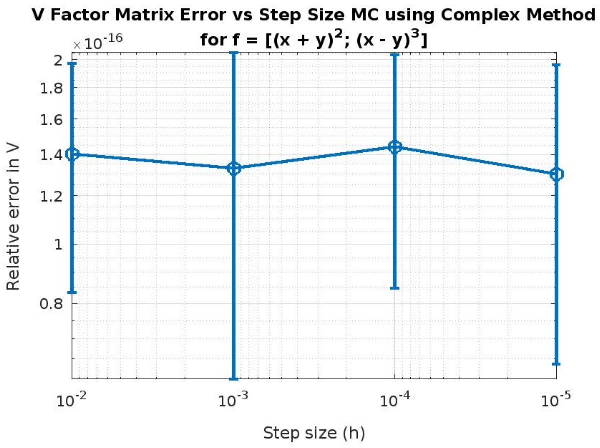




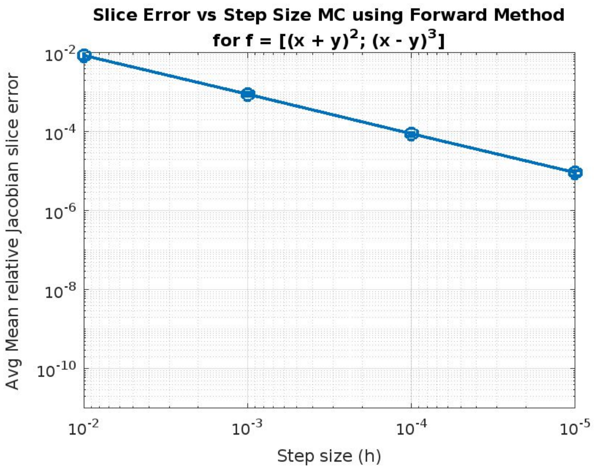
Factor matrices V vs. Step size

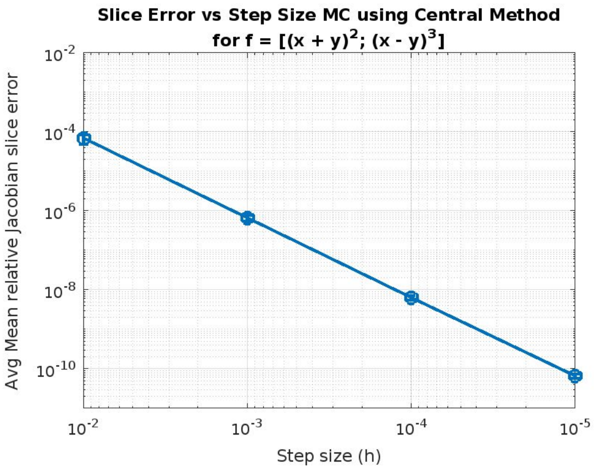


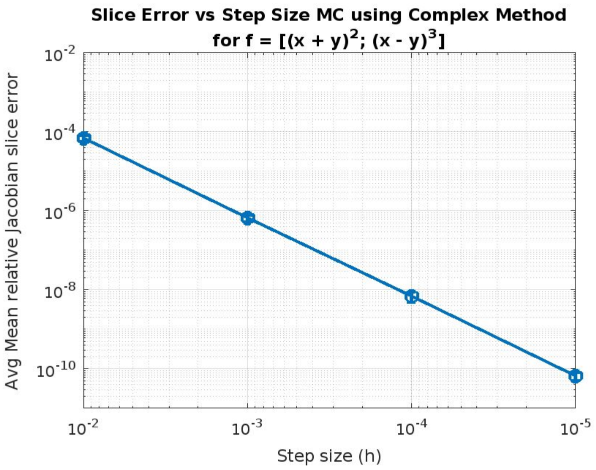




Jacobian Slice Error vs. Step Size







Methods Finite Differentiation: Backward