Learning Dynamical Models through Data Science

Physics-informed ML vs DMD

Jose M. Moreu, Dakota Wenberg, Richa Naik, Vineet J Nair
Mentor: Zhong Yi Wan

2.168 Final Project Presentation
Wednesday, 29 April 2020

Summary

Goals: Extract PDE of a system from dynamical data using PDE-FIND, improve its robustness to noise & compare with DMD.

1. Use DMD to make future state predictions + interpret results
2. Improving robustness of PDE-FIND to noisy data
    a. Different sparsity techniques
    b. Hyperparameter optimization
    c. Denoising methods
        i. Error plots
        ii. PDE-FIND coefficients
        iii. PDE-FIND simulated results

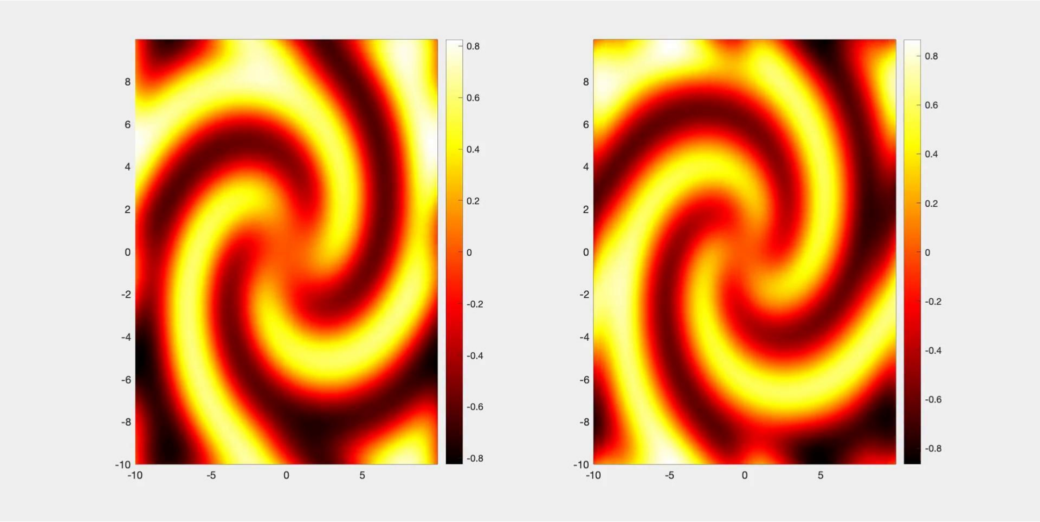
System Studied: Reaction-Diffusion λ-Ω model

$$\frac{\partial u}{\partial t} = lam(A)u - ome(A)v + d_1\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0$$

$$\frac{\partial v}{\partial t} = ome(A)u + lam(A)v + d_2\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = 0$$

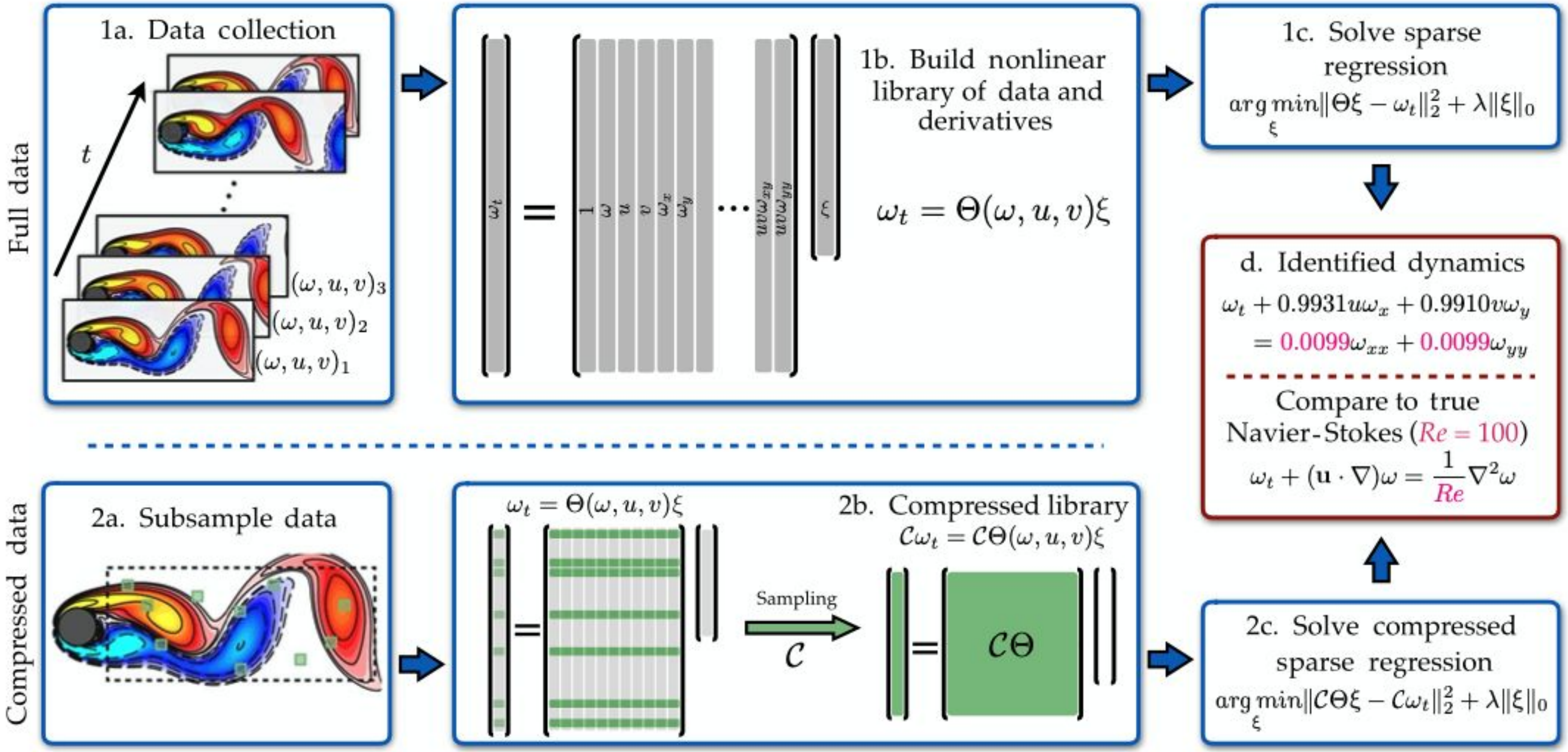
$$A^2 = u^2 + v^2$$
$$lam(A) = 1 - A^2$$
$$ome(A) = -\beta * A^2$$
$$d_1 = 0.5; \quad d_2 = 0.8; \quad \beta = 2.0;$$

u(x,y,t) v(x,y,t)



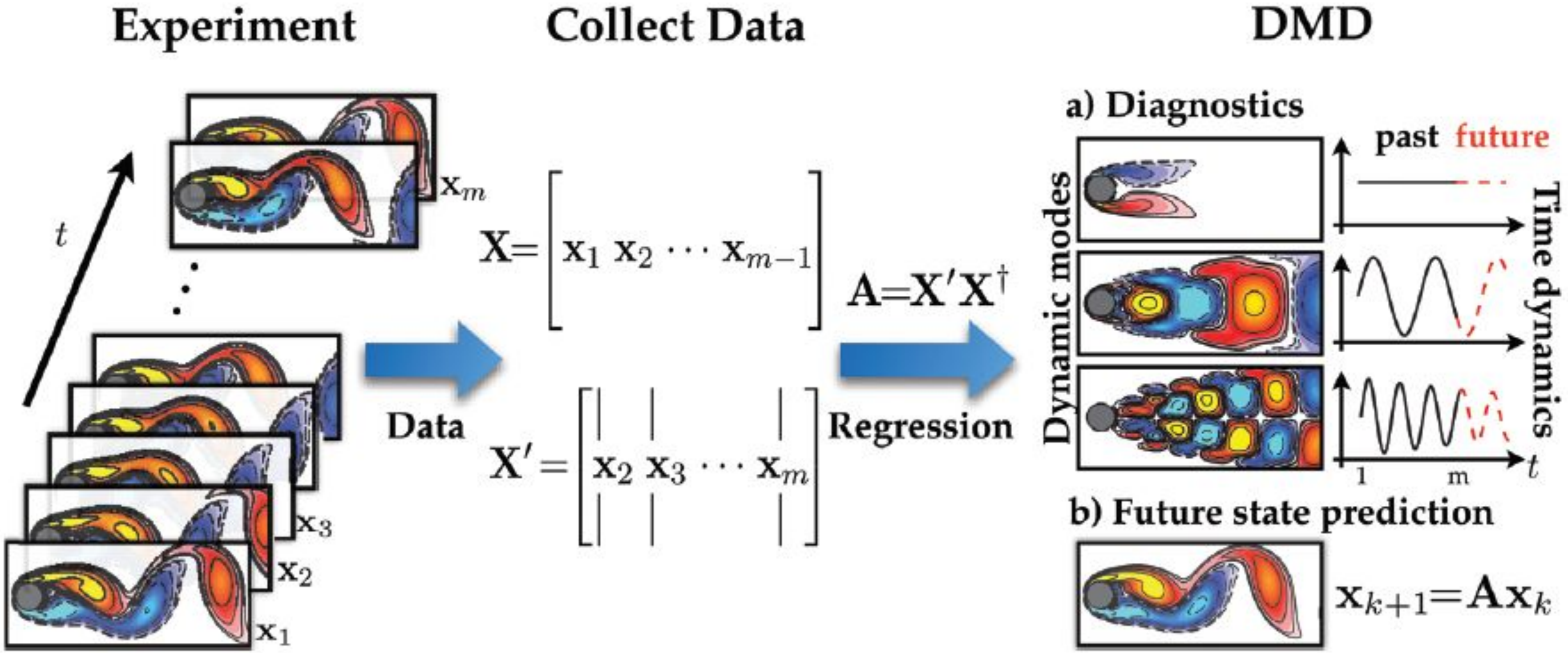
$$u_t = d_1 u_{xx} + d_1 u_{yy} + u - u^3 - uv^2 + \beta u^2 v + \beta v^3$$
$$v_t = d_2 v_{xx} + d_2 v_{yy} - \beta u^3 - \beta v^2 u + v - vu^2 - v^3$$

What is PDE FIND?



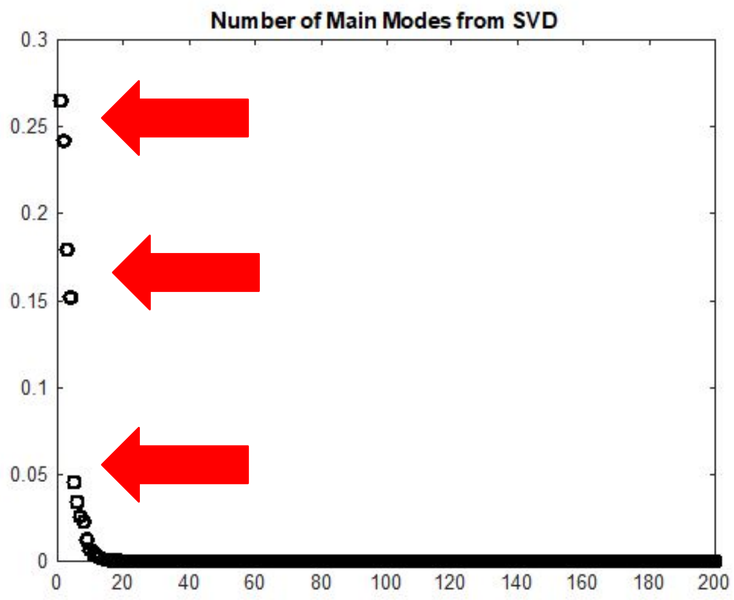
Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. Science Advances, 3(4), e1602614.

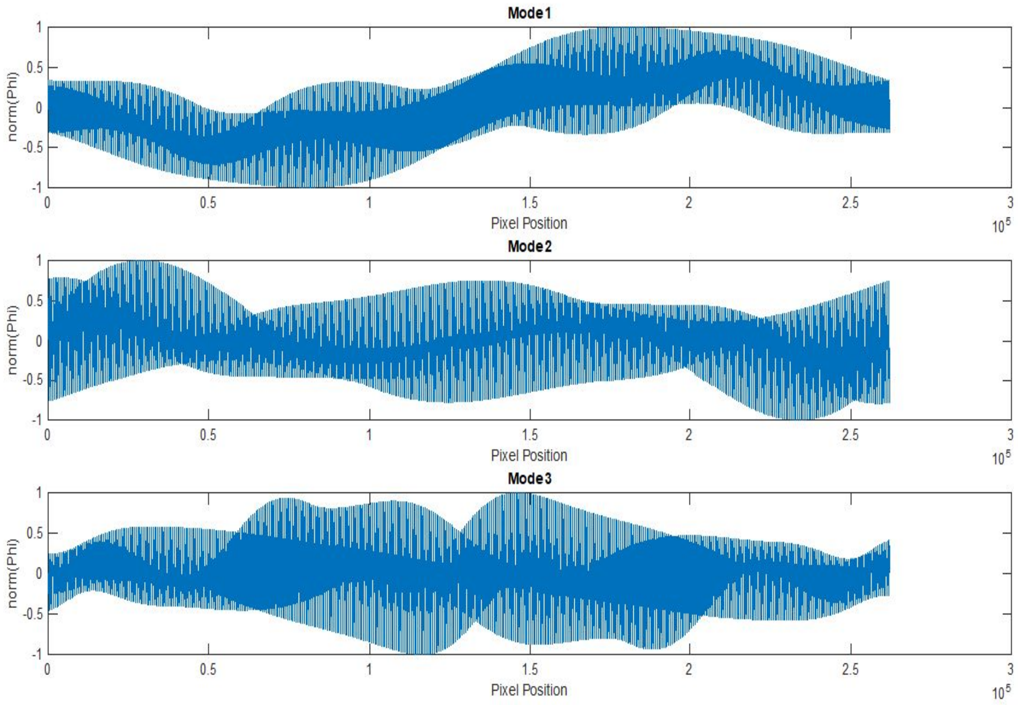
What is DMD?



Kutz, J. N., Brunton, S. L., Brunton, B. W., & Proctor, J. L. (2016). Dynamic mode decomposition: data-driven modeling of complex systems. Society for Industrial and Applied Mathematics.

DMD Modes


Number of Main Modes from SVD
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120
140
160
180
200


Mode 1
Mode 2
Mode 3
norm(Phi)
Pixel Position
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
2.5
3
10^5

DMD Future State Prediction

$$\omega = \frac{\ln(\lambda)}{\Delta t}$$

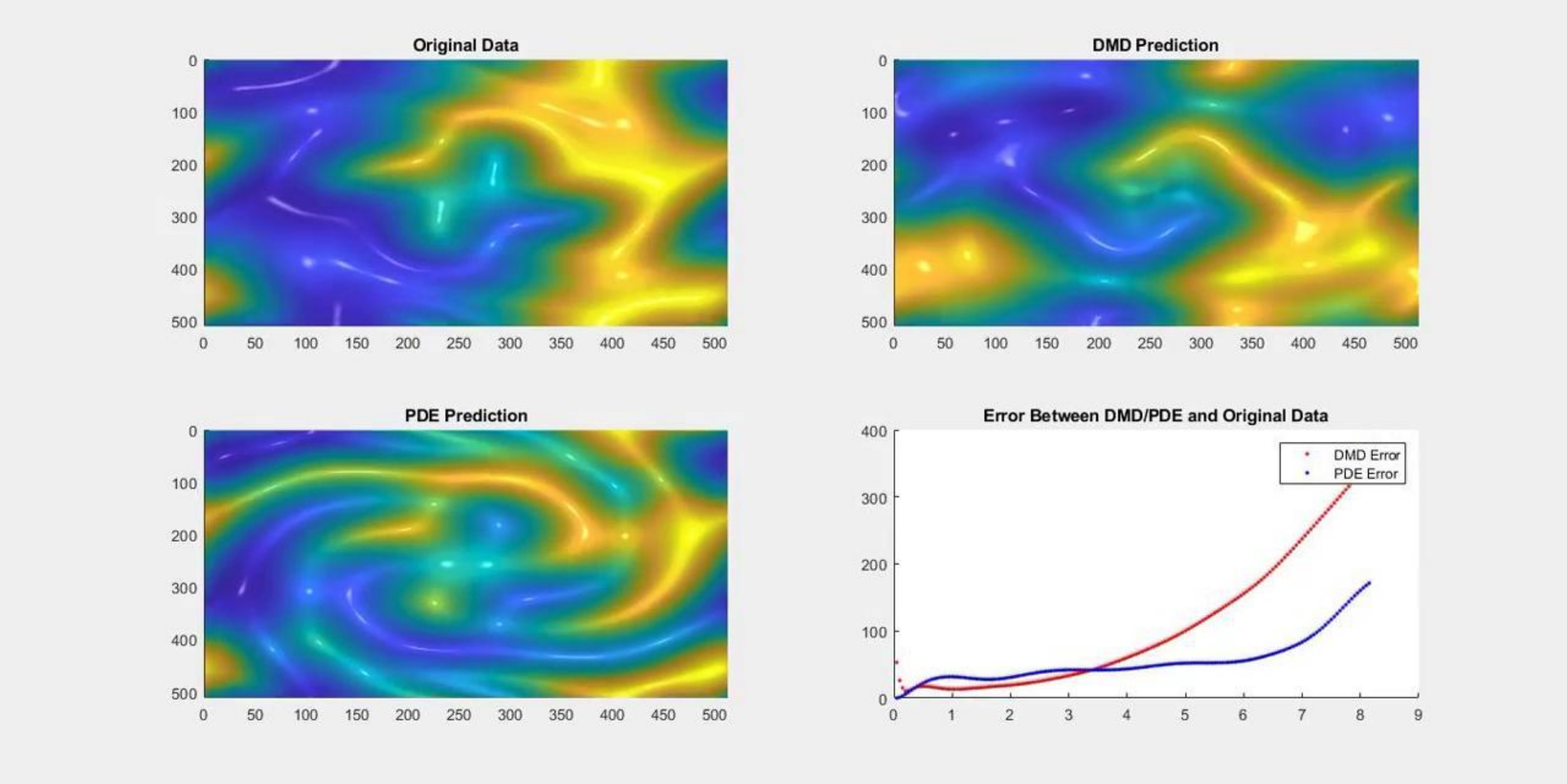
$$x_{DMD}(t) = \phi * diag(e^{\omega t}) * b$$

λ: Discrete time DMD eigenvalues

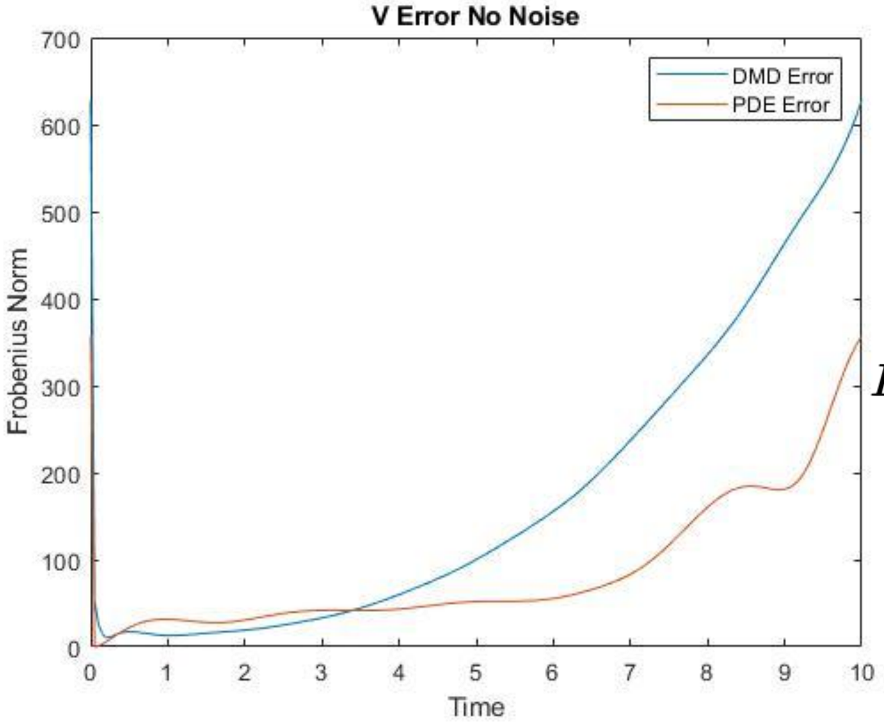
ϕ: DMD modes

b: vector with magnitude of modes

DMD Prediction Results: V(x,y,t)

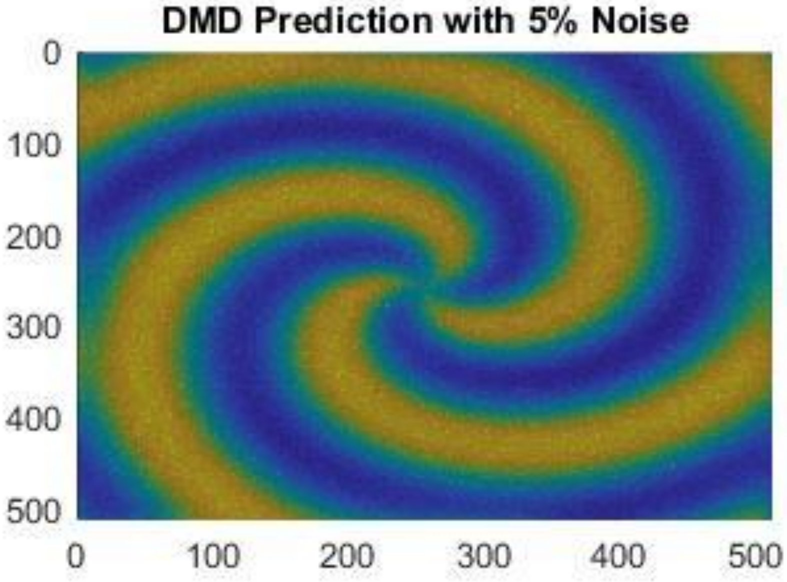


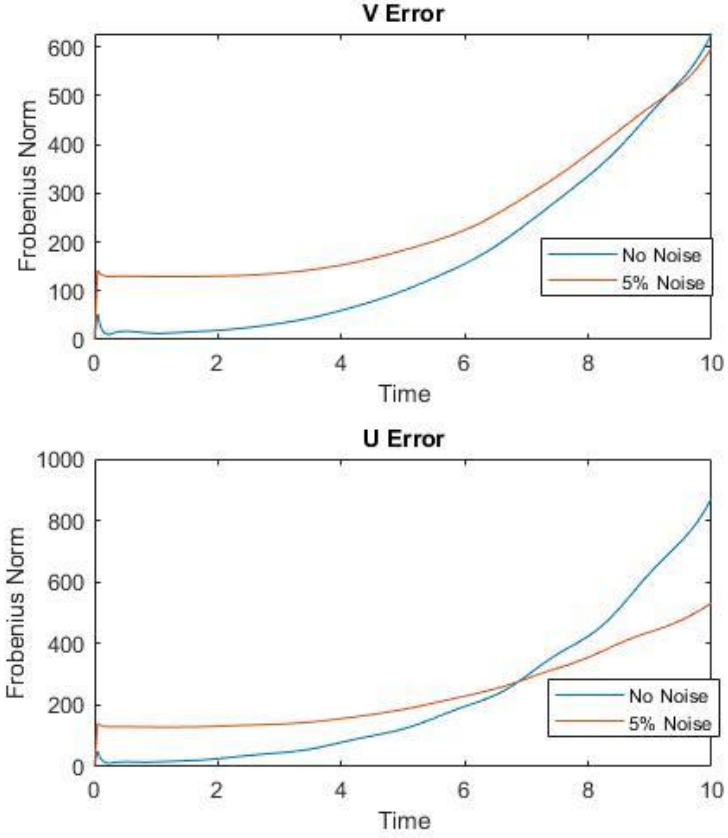
Result



$$Error_v(t) = ||v(:,:,t) - v_{predicted}(:,:,t)||_{fro}$$

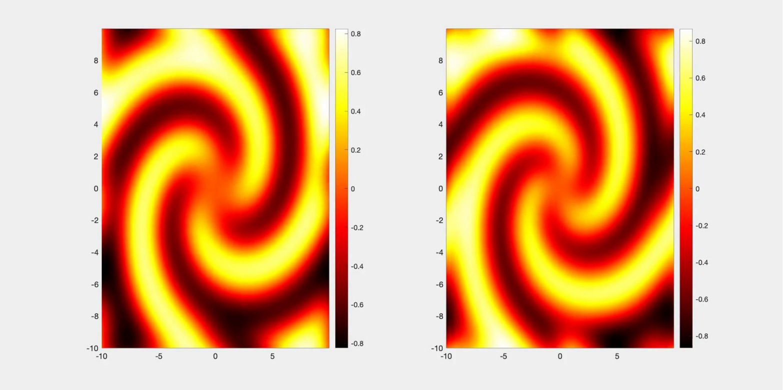
DMD Error Plots





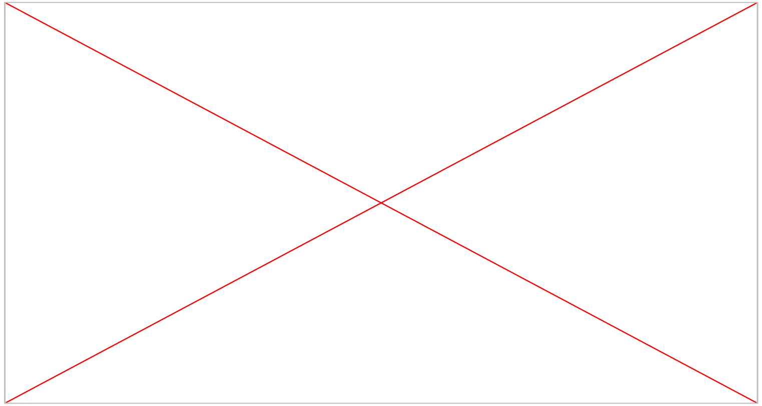
PDE-FIND Results: No Noise

		u_{xx}	u_{yy}	v_{xx}	v_{yy}	u	v	uv^2	u^2v	u^3	v^3
u_t	Actual	0.5	0.5			1		-1	2	-1	2
	Noise Free	0.5	0.5			1		-1	2	-1	2
v_t	Actual			0.8	0.8		1	-2	-1	-2	-1
	Noise Free			0.8	0.8		1	-2	-1	-2	-1



PDE-FIND Results: Noisy Data 0.5%

		u_{xx}	u_{yy}	v_{xx}	v_{yy}	u	v	uv^2	u^2v	u^3	v^3
u_t	Actual	0.5	0.5			1		-1	2	-1	2
	Noisy data					0.18	0.41		1.32	-0.22	1.28
v_t	Actual			0.8	0.8		1	-2	-1	-2	-1
	Noise data					-0.43	-0.21	-1.16		-1.16	0.35



Sparse Regression Methods

- Lasso Regression: Convex relaxation of Subset Selection problem using L-0 norm is L-1 Norm

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \lambda \|w\|_1,$$

- Elastic Net Regression

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \lambda \left[\alpha \|w\|_1 + \tfrac{1-\alpha}{2} \|w\|_2^2\right],$$

- Sequential Threshold Ridge Regression (STRidge)

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \frac{1}{2\gamma} \|w\|_2^2 \text{ s.t. } \|w\|_0 \leqslant k,$$

- Forward Backward Greedy Algorithm

Sparse Regression Methods: Results

		u_{xx}	u_{yy}	v_{xx}	v_{yy}	u	v	uv^2	u^2v	u^3	v^3
v_t	Actual			0.8	0.8		1	-2	-1	-2	-1
	Lasso	0.043	0.031	0.659	0.642	0.040	0.651	-1.965	0.000	-1.964	-0.636
	Elastic Net	0.000	0.000	0.799	0.799	0.000	1.000	-1.999	-1.000	-1.999	-1.000
	STRidge			0.800	0.800		1.000	-1.999	-1.000	-1.999	-1.000
	Greedy			0.799	0.800	0.000	0.999	-1.999	-1.000	-1.999	-1.000

Thus, the STRidge method has the best empirical performance for PDE-FIND of any sparse regression algorithm tested in the work.

Hyperparameter Optimization

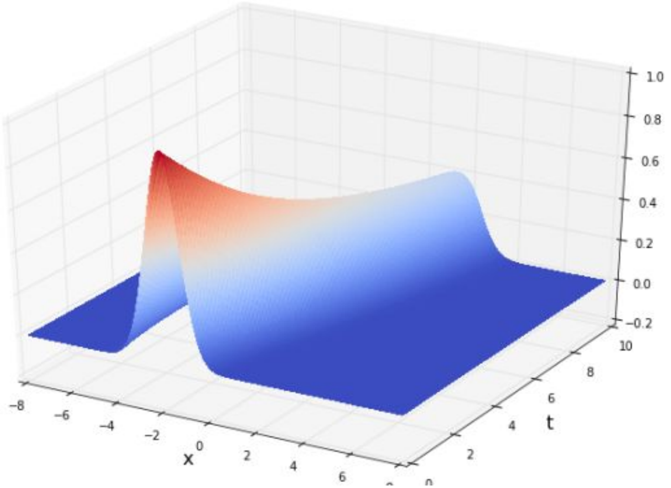
- We look at the STRidge algorithm – lambda and tolerance

$$\text{STRidge}(\mathbf{\Theta}, \mathbf{U}_t, \lambda, tol, \text{iters})$$

- Burger's Equation: Derived from the Navier Stokes equations for the velocity field by dropping the pressure gradient term

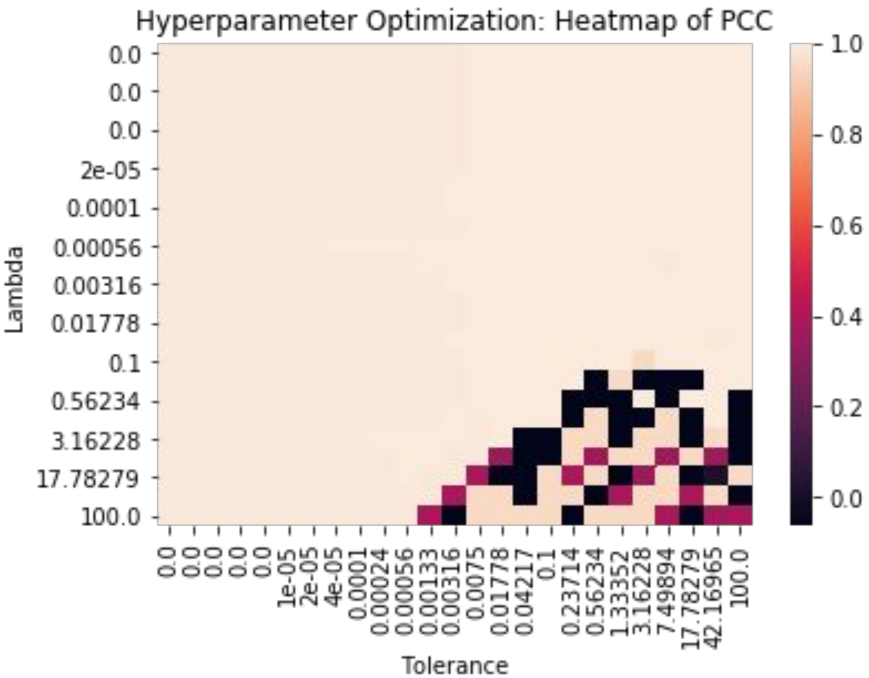
$$u_t + uu_x - \epsilon u_{xx} = 0$$

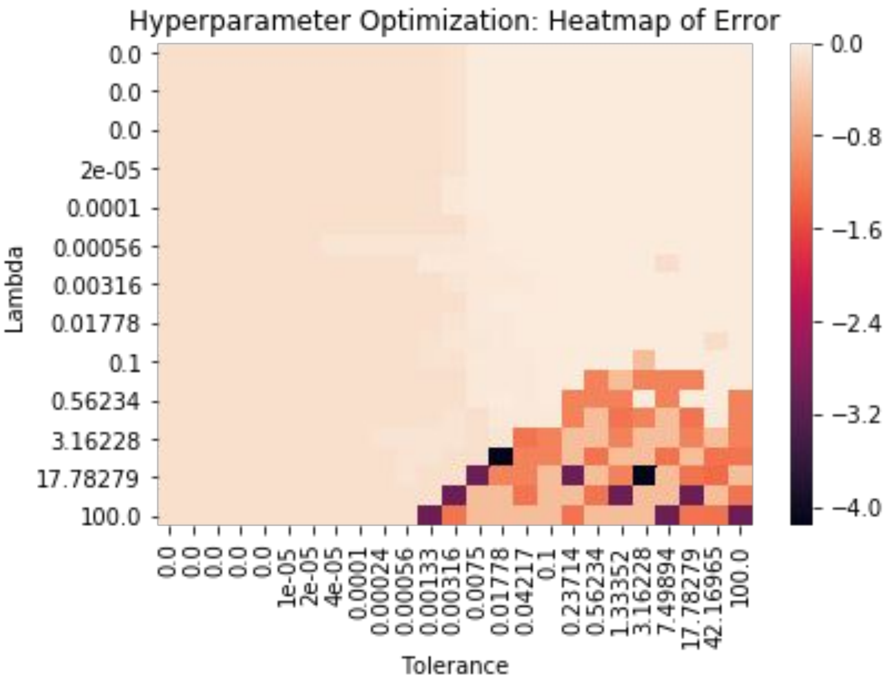
- Evaluate the performance using:
  - Pearson Correlation Coefficient
  - L2 norm of error



Burger's Equation Data

Hyperparameter Optimization: Results





Denoising Methods for PDE-FIND

- Truncated Singular Value Decomposition (TSVD)

$$X = U\Sigma V^T \longrightarrow \tilde{X} = U_r \Sigma_r V_r^T$$

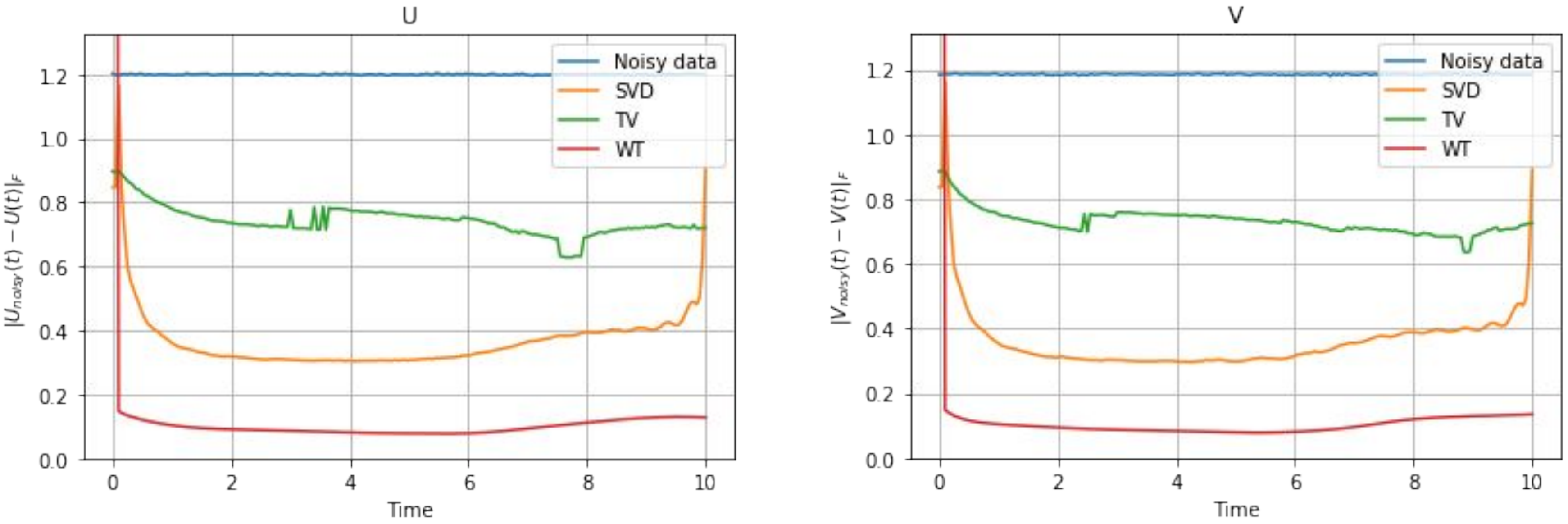
- Total-Variation based method

$$\underset{x}{\operatorname{argmin}} \left\{ ||\mathbf{X} - \mathbf{B}||^2 + 2\lambda \mathbf{TV}(\mathbf{X}) \right\}$$

- Wiener-Tikhonov

$$G(f) = \frac{H^*(f)S(f)}{|H(f)|^2 S(f) + N(f)}$$

Denoising comparison



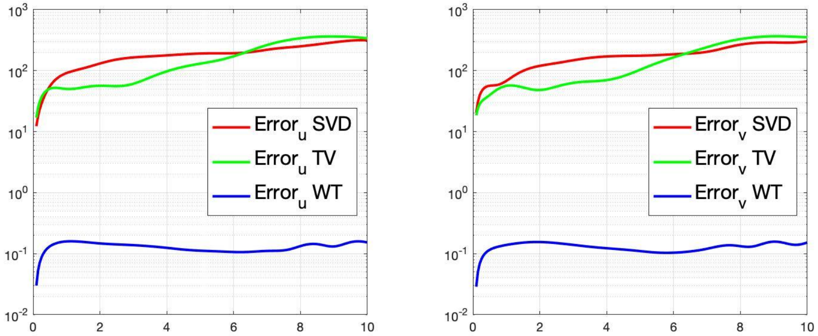
Parameter prediction with 0.5% noise

		u_{xx}	u_{yy}	v_{xx}	v_{yy}	u	v	uv^2	u^2v	u^3	v^3
u_t	Actual	0.5	0.5			1		-1	2	-1	2
	SVD	0.220	0.221			0.561	0.226	-0.493	1.626	-0.586	
	TV	0.054	0.051			0.248	0.364		1.396	-0.279	1.364
	WT	0.500	0.500			1.000		-1.000	2.001	-1.000	2.001
v_t	Actual			0.8	0.8		1	-2	-1	-2	-1
	SVD			0.287	0288	-0.281	0.121	-1.455		-1.451	
	TV			0.064	0.079	-0.392	-0.124	-1.231		-1.241	0.254
	WT			0.800	0.800		1.000	-2.001	-1.000	-2.001	-1.000

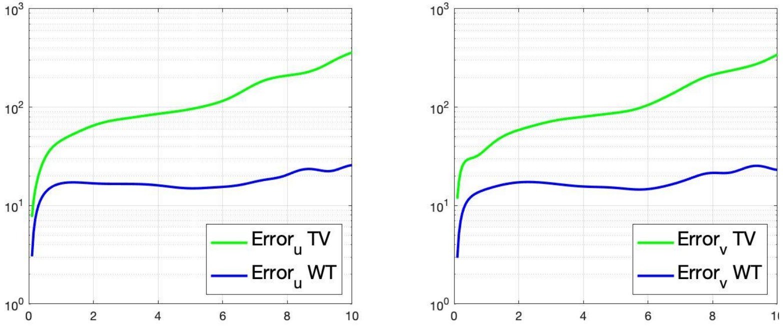
Simulation errors

$$Error_u(t) = ||u(:,:,t) - u_{predicted}(:,:,t)||_{fro}$$
$$Error_v(t) = ||v(:,:,t) - v_{predicted}(:,:,t)||_{fro}$$

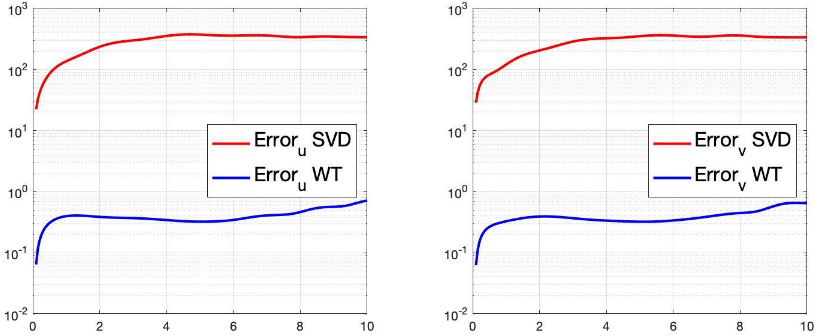
Noise = 0.5%



Noise = 5%



Noise = 1%



Denoising: DMD

- Forward-Backward DMD (fbDMD)

$$\tilde{A} = (\tilde{A}_f \tilde{A}_b)^{1/2}$$

- Optimized DMD
  - Use total least squares (TLS) regression to reduce reconstruction error

$$||Y - X_{reconstructed}||_F$$

$$A = argmin\ ||Y - AX||_F$$

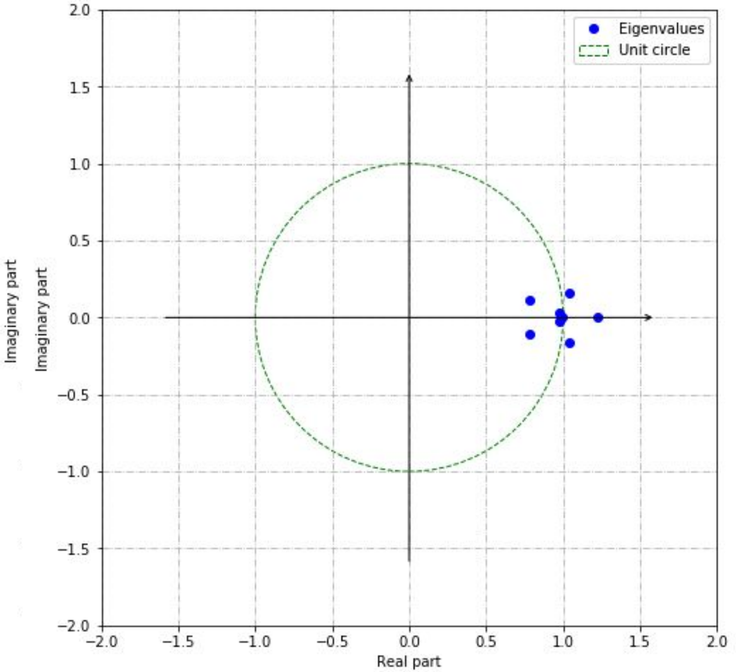
- Higher order DMD (hoDMD)
  - Useful when # of snapshots < dimension of each snapshot

$$\mathbf{x}_{k+1} = A\mathbf{x}_k \longrightarrow \mathbf{x}_{k+2} = A_1\mathbf{x}_k + A_2\mathbf{x}_{k+1}$$

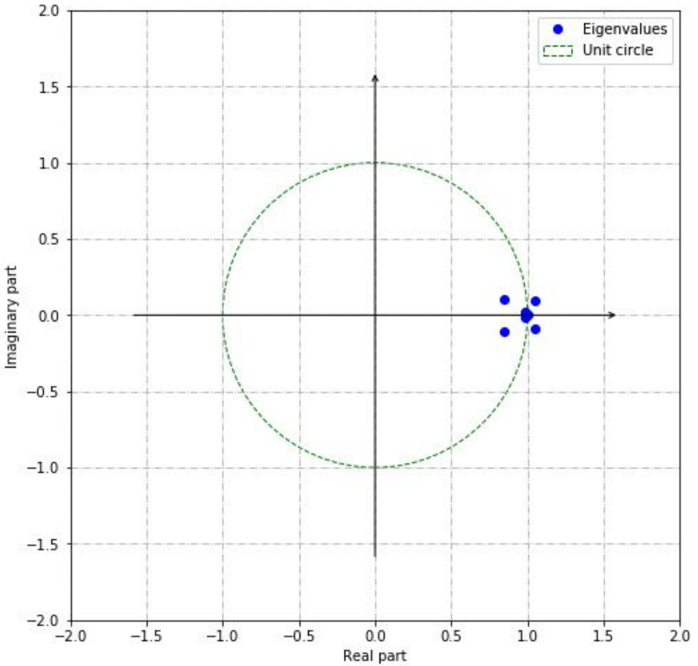
Stability considerations

Discrete-time eigenvalues with fbDMD

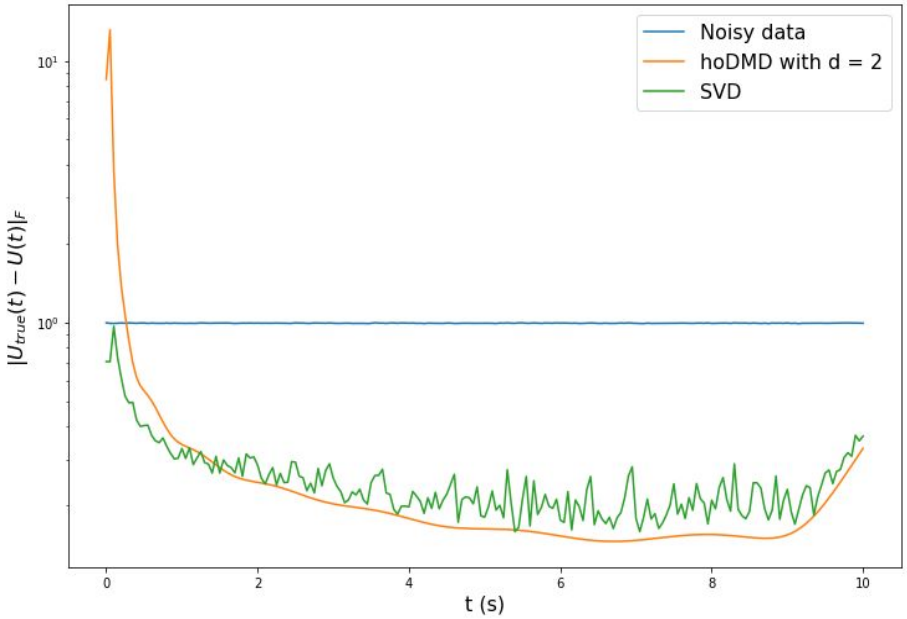
λ_u

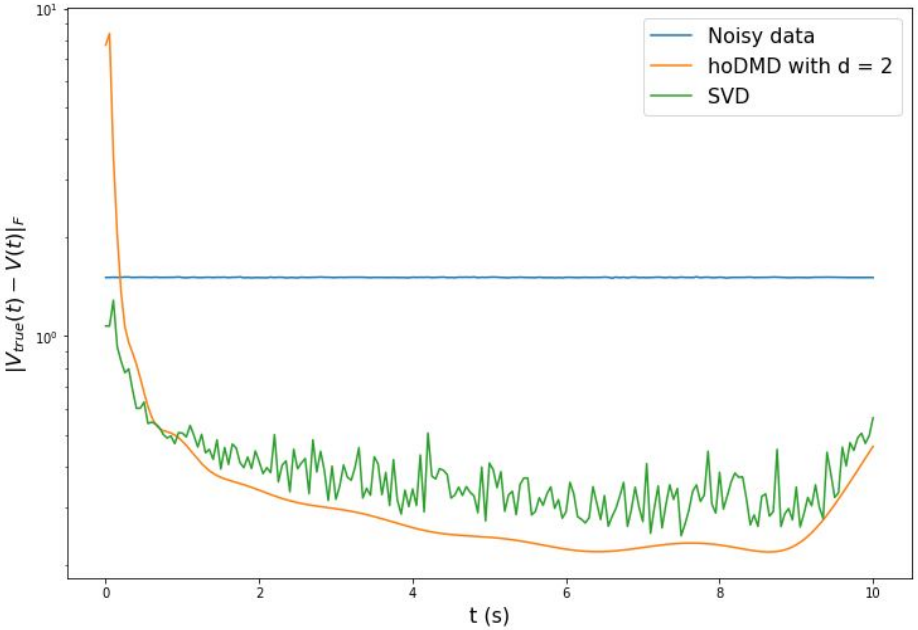


λ_v



DMD Denoising results





Comparison of PDE-FIND predictions

	Mean relative error (%)	Standard deviation (%)
0.5% noise		
Denoising with TSVD	1.085	0.939
Denoising with hoDMD	0.515	0.406
1% noise		
Denoising with TSVD	4.384	3.806
Denoising with hoDMD	2.347	1.961
5% noise		
Denoising with TSVD	67.001	56.809
Denoising with hoDMD	39.497	38.154

Conclusions

- PDE-FIND works properly without noise
- DMD helps interpret modes & make predictions few timesteps into future
- Effects of sparsity techniques
  - STRidge works best for PDE-FIND
- Effects of hyperparameters
  - PDE-FIND insensitive to exact values of regularization parameter and tolerance as long as both are reasonably small
- In presence of noise: PDE-FIND very sensitive → Incorrect results
  - If we know noise characteristics: Wiener Tikhonov
  - If we don't know noise statistics: Higher order DMD with TSVD and TLS
- Future work
  - Dealing with higher noise levels?
  - If possible, use physical information from the system e.g can add regularization penalties to enforce conservation laws ($A^2 = u^2 + v^2$)

Citations

Rudy SH, Brunton SL, Proctor JL, Kutz JN. Data-driven discovery of partial differential equations. Science Advances. 2017.

E. Kammers. Using the Dynamic Mode Decomposition (DMD) to Rotate Long-Short Exposure Between Stock Market Sectors. Quantoisseur. 2019.

N. Kutz, S. Brunton, B. Brunton, and J. Proctor, Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. 2016.

Hemati, Rowley, Deem, Cattafesta. De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets. Theoretical and Computational Fluid Dynamics, 2017.

Dawson, Hemati, Williams, Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. Experiments in Fluids, 2016.

Le Clainche, Vega. Higher Order Dynamic Mode Decomposition. Journal on Applied Dynamical Systems, 2017.

Thank You!

Appendix

Gray Scott Model

Another 2D reaction diffusion model but produces a diverse array of patterns, depending on initial conditions and parameters

$$
\begin{aligned}
U + 2V &\rightarrow 3V \\
V &\rightarrow P
\end{aligned}
\qquad\qquad
\begin{aligned}
u_t &= r_u \nabla^2 u - uv^2 + f(1-u) \\
v_t &= r_v \nabla^2 v + uv^2 - (f+k)v
\end{aligned}
$$

r_u , r_v = Diffusion rates of the two species

k = Rate of conversion of V to P

f = feed rate of U and kill rate of U, V and P

Gray Scott: PDE-FIND parameter predictions

		1	u_{xx}	u_{yy}	v_{xx}	v_{yy}	u	v	uv^2
u_t	Actual	0.025	0.01	0.01	-	-	-0.025	-	-1
	Noise Free	0.025	0.01006	0.01002	-	-	-0.02499	-	-0.999834
	0.5% noise	0.025002	0.009815	0.009846	-	-	-0.025709	-	-0.996995
	1 % noise	0.25022	0.009270	0.009344	-	-	-0.988428	-	-0.027894
v_t	Actual	-	-	-	0.005	0.005	-	-0.085	1
	Noise Free	-	-	-	0.005002	0.005006	-	-0.084988	0.999980
	0.5% noise	-	-	-	0.004919	0.004930	-	-0.085084	0.998909
	1% noise	-	-	-	0.004697	0.004706	-	-0.085394	0.996312

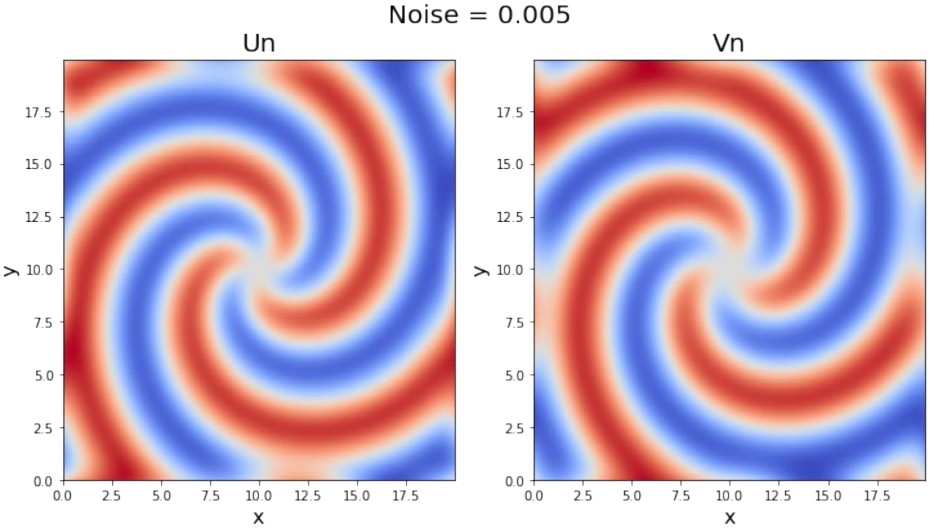
PDE-FIND: Noisy Data 2

```
Un = U + noise*std(U)*np.random.randn(n,n,steps)
```

$$A^2 = u^2 + v^2$$

```
Vn = np.sqrt(A-U**2)*np.sign(V)
```

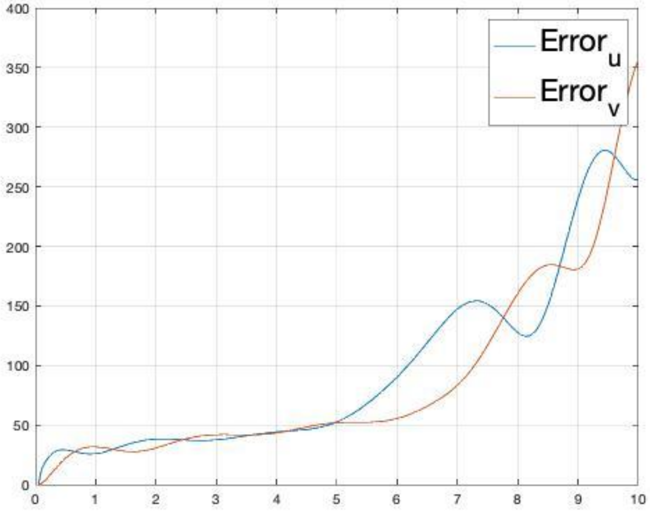
	u_t	û_t	v_t	^v_t
u_xx	0.5	0.220	0	0
u_yy	0.5	0.221	0	0
v_xx	0	0	0.8	0.798
v_yy	0	0	0.8	0.798
u	1	0.561	0	0
v	0	0.226	-1	0.99
uv^2	-1	-0.493	-2	-2.00
u^2v	2	1.626	1	-1
u^3	-1	-0.586	-2	-1.999
v^3	2	1.601	1	-0.991



Error vs. Simulation time

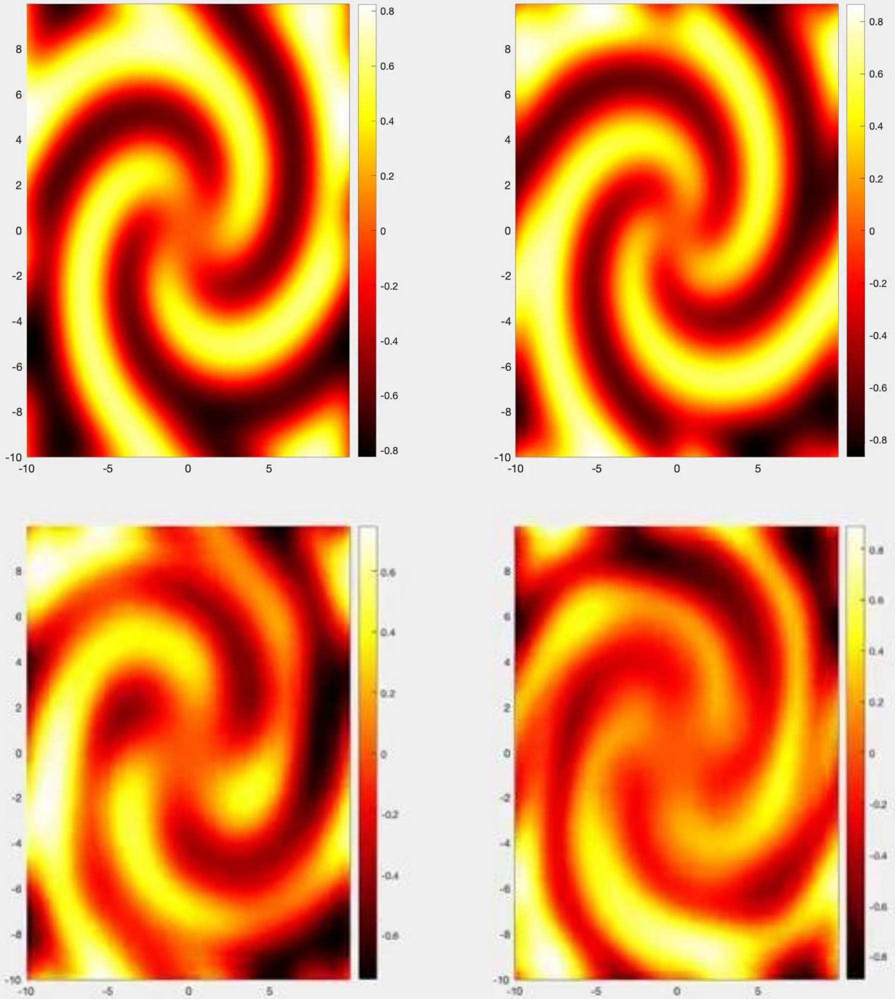
$$Error_u(t) = ||u(:,:,t) - u_{predicted}(:,:,t)||_{fro}$$

$$Error_v(t) = ||v(:,:,t) - v_{predicted}(:,:,t)||_{fro}$$



Original

Predicted



Sparse Regression Methods: Results

		u_{xx}	u_{yy}	v_{xx}	v_{yy}	u	v	uv^2	u^2v	u^3	v^3
u_t	Actual	0.5	0.5			1		-1	2	-1	2
	Lasso	0.499	0.499	0.000	0.000	1.000	0.000	-1.000	1.999	-1.000	1.999
	Elastic Net	0.500	0.499	0.000	0.000	1.000	0.000	-1.000	1.999	-1.000	1.999
	STRidge	0.500	0.500			1.000		-1.000	1.999	-1.000	1.999
	Greedy	0.500	0.500			1.000	0.000	-1.000	1.999	-1.000	1.999

Sparse Regression Methods

- Subset Selection: Combinatorial brute-force search across all possible term combinations optimized by cardinality-penalized estimators: L-0 Norm (NP-hard)

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \lambda \|w\|_0,$$

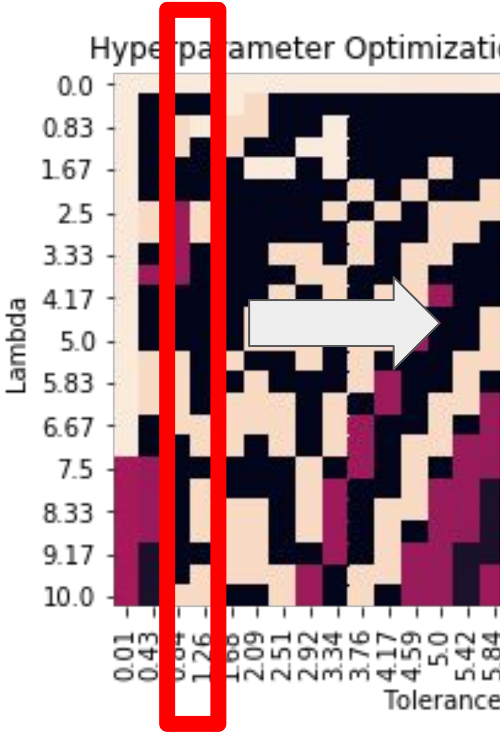
- Lasso Regression: Convex relaxation of L-0 optimization problem is L-1 Norm

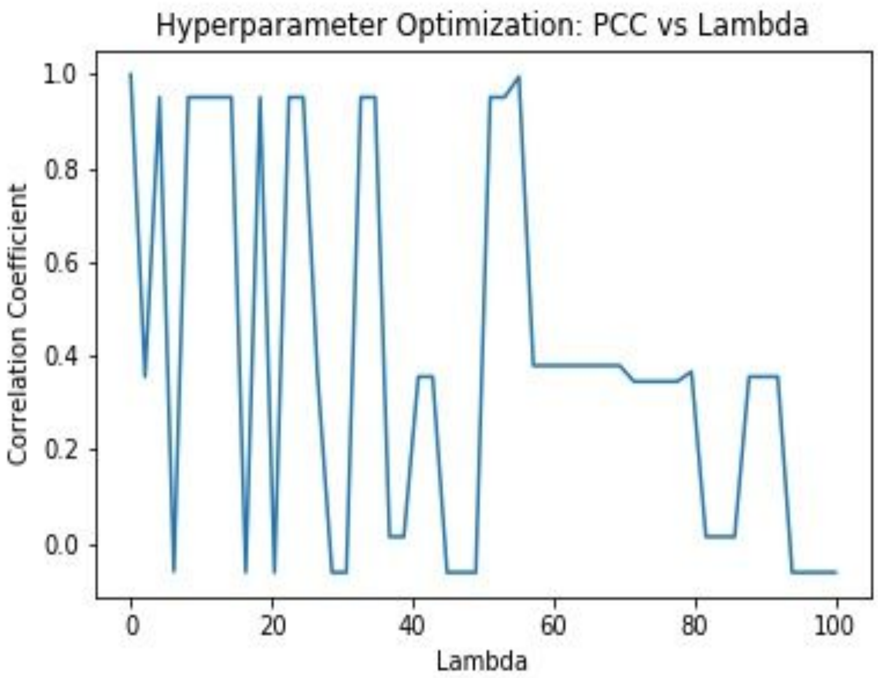
$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \lambda \|w\|_1,$$

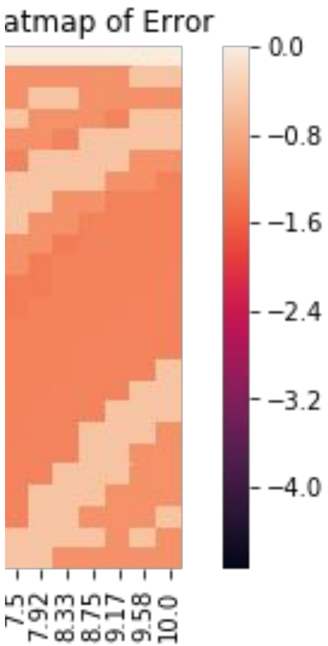
- Elastic Net Regression:

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) + \lambda \left[\alpha \|w\|_1 + \frac{1-\alpha}{2} \|w\|_2^2\right],$$

Hyperparameter Optimization

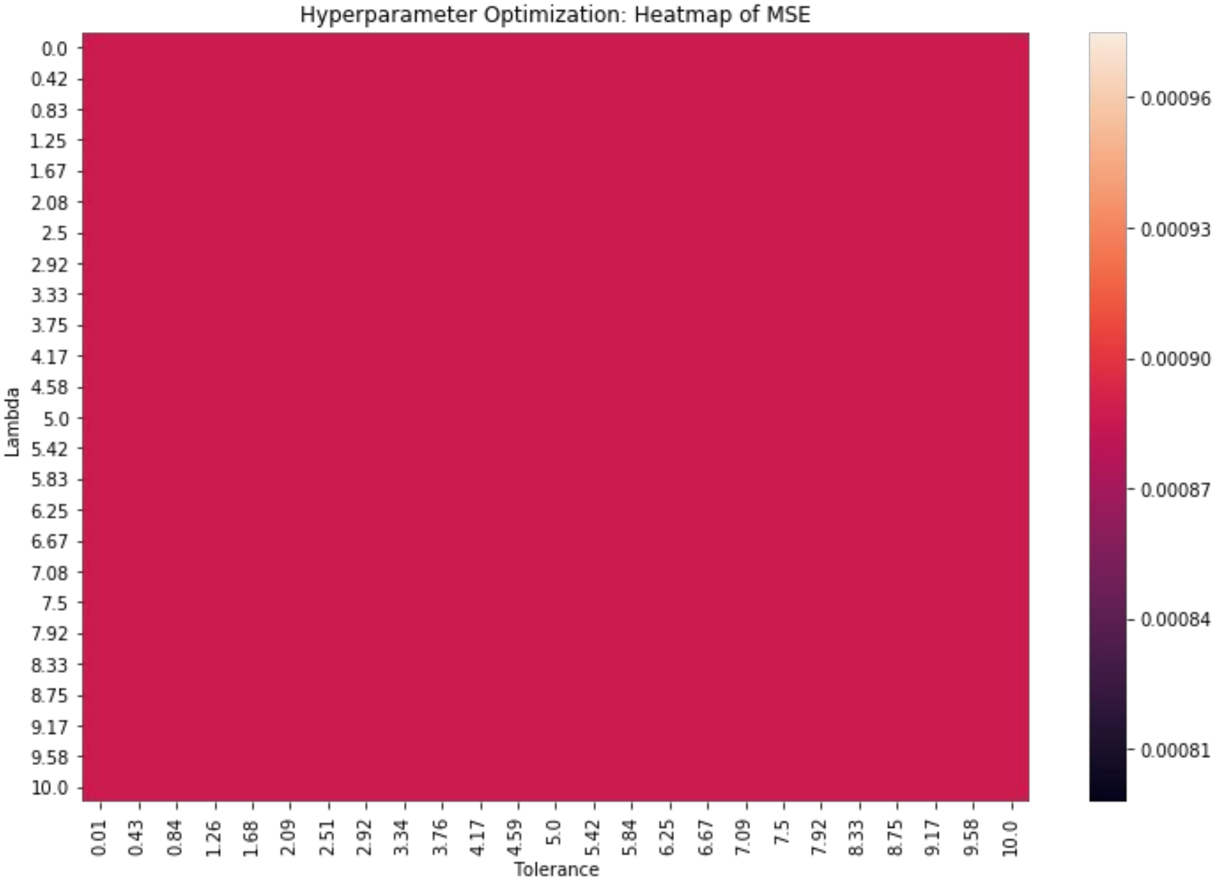






Varying regularization parameter (lambda) and tolerance

Hyperparameter Optimization:MSE



Denoising: SVD

- Truncated SVD (or POD)
- Already implemented in PDE-FIND
- Select 1st *r* dominant singular values to create low-rank approximation for data matrix X

$$X = U\Sigma V^T \longrightarrow \tilde{X} = U_r \Sigma_r V_r^T$$

- Decide on optimal rank truncation *r* to minimize error:
    - Plot all singular values
    - Some trial and error needed

Denoising: Weiner Tikhonov Regularization

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right]G(u,v)$$

$$= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v) \qquad (5.8\text{-}2)$$

$$= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right]G(u,v)$$

$H(u,v)$ = degradation function

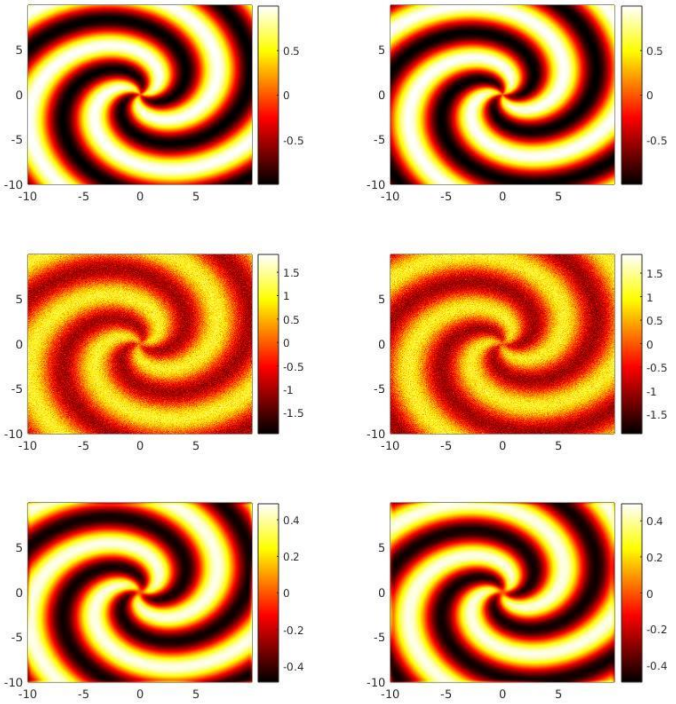
$H^*(u,v)$ = complex conjugate of $H(u,v)$

$|H(u,v)|^2 = H^*(u,v)H(u,v)$

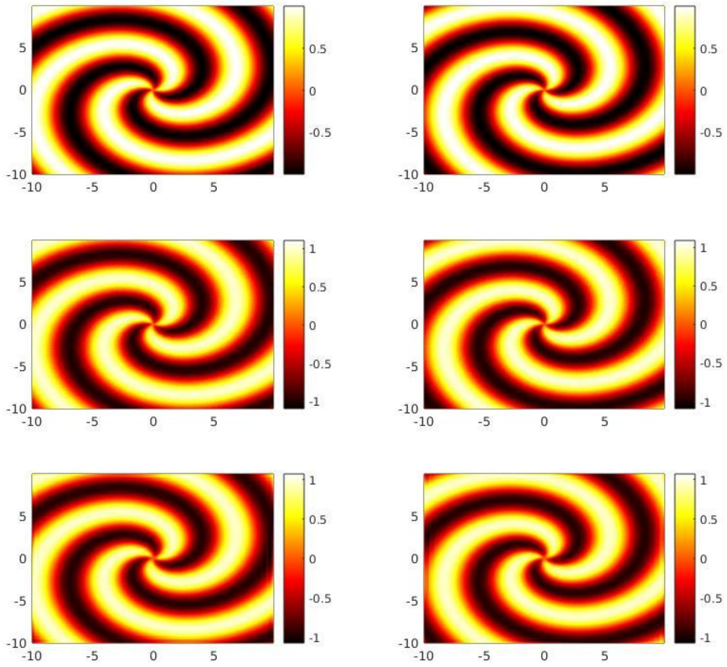
$S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise [see Eq. (4.2-20)]

$S_f(u,v) = |F(u,v)|^2$ = power spectrum of the undegraded image.

Denoising: Weiner Tikhonov Regularization



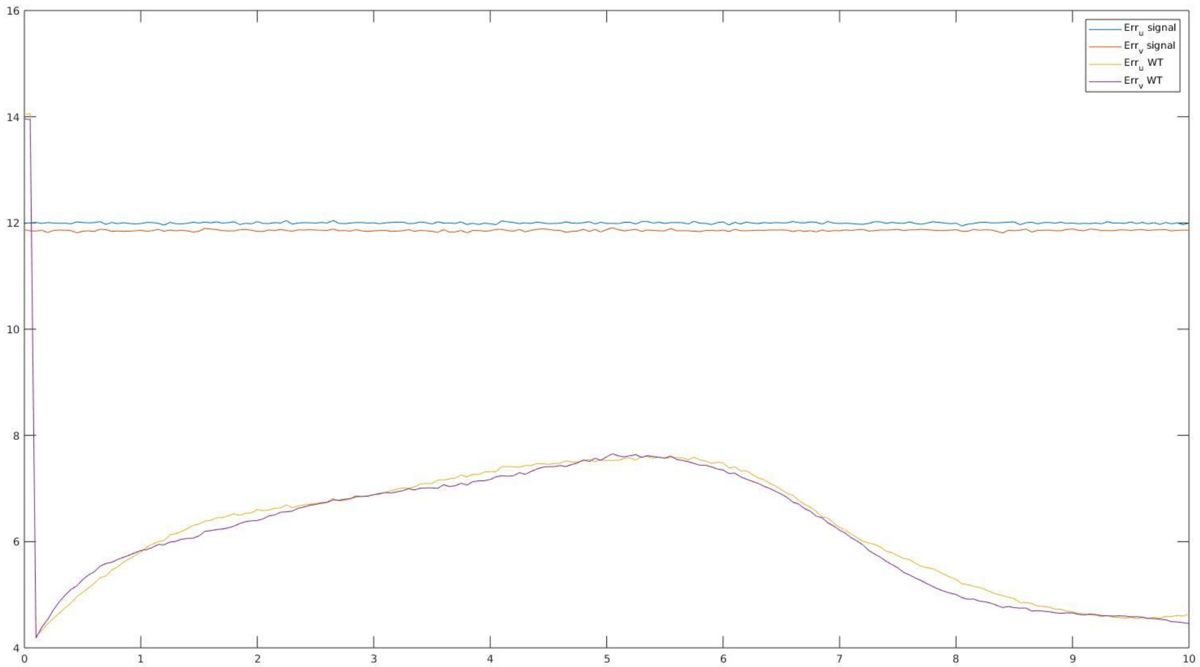
50% Noise



5% Noise

Denoising: Weiner Tikhonov Regularization

Error Plot - 5% Noise



Denoising: Total Variation

$$\underset{x}{\operatorname{argmin}} \left\{ ||\mathbf{X} - \mathbf{B}||^2 + 2\lambda \mathrm{TV}(\mathbf{X}) \right\}$$

- **B** is the noisy sample.
- TV(**X**) is:

$$\begin{aligned} \mathbf{X} \in \mathbb{R}^{m \times n}, \quad \mathrm{TV}_{l_1}(\mathbf{X}) \;&=\; \textstyle\sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \{|X_{i,j} - X_{i+1,j}| + |X_{i,j} - X_{i,j+1}|\} \\ &\quad + \textstyle\sum_{i=1}^{m-1} |X_{i,n} - X_{i+1,n}| + \sum_{j=1}^{n-1} |X_{m,j} - X_{m,j+1}|. \end{aligned}$$