POD Analysis of Turbulent Pipe Flow

M. Raba

Created: 2025-09-10 Wed 03:38

1. Code Execution and Layout

1.1. Layout

1. b7.m
2. initSpectral.m
    - reads in binary files, takes eg m-fft
3. $\hookrightarrow$ initEigs.m
    - forms corrMat, finds eigenvalues

1.2. Layout 2

1. $\hookrightarrow$ initPod.m
    - carries out POD calculations (quadrature, multiplication ggf betwen $\alpha\Phi$) according to Papers (Citriniti George 2000 for Classic POD, Hellstrom Smits 2017 for Snapshot POD)
2. $\hookrightarrow$ timeReconstructFlow.m
    - performs 2d reconstruction + plotSkmr (generates 1d radial graph)

1.3. Important Switches

pipe = Pipe(); creates a Pipe Class. As the functions (above) are called, data is stored in sub-structs:

1. obj.CaseId - stores properties like Re, rotation number S, experimental flags such as quadrature (simpson/trapezoidal), number of gridpoints, frequently called vectors (rMat $r = 1, \ldots, 0.5$)
2. obj.pod - eigen data, used for calculating POD
3. obj.solution - computed POD modes
4. obj.plt - plot configuration

2. Equations Used in Code Procedure

2.1. Classic POD Equations

The following equations are used in the above code.

$$\int_{r'} \mathbf{S}\left(k;m;r,r'\right)\Phi^{(n)}\left(k;m;r'\right)r'\mathrm{d}r' = \lambda^{(n)}(k;m)\Phi^{(n)}(k;m;r)$$

$$\mathbf{S}\left(k;m;r,r'\right) = \lim_{\tau\to\infty}\frac{1}{\tau}\int_0^{\tau}\mathbf{u}(k;m;r,t)\mathbf{u}^*\left(k;m;r',t\right)\mathrm{d}t$$

$$\alpha^{(n)}(k;m;t) = \int_r \mathbf{u}(k;m;r,t)\Phi^{(n)^*}(k;m;r)r\,\mathrm{d}r$$

2.2. Classic POD Equations (Fixed)

$$\int_{r'} \underbrace{r^{1/2} S_{i,j}\left(r, r'; m; f\right) r'^{1/2}}_{W_{i,j}(r,r';m;f)} \underbrace{\phi_j^{*(n)}\left(r'; m; f\right) r'^{1/2}}_{\hat{\phi}_j^{\psi(i)}(r';m;f)} \mathrm{d}r'$$

$$= \underbrace{\lambda^{(n)}(m, f)}_{\hat{\lambda}^{(n)}(m;f)} \underbrace{r^{1/2} \phi_i^{(n)}(r; m; f)}_{\hat{\phi}_i^{(n)}(r,m;f)}$$

$$\alpha_n(m;t) = \int_r \mathbf{u}(m; r, t) r^{1/2} \Phi_n^*(m; r) dr$$

2.3. Snapshot POD Equations

$$\lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau \mathbf{u}_\mathrm{T}(k;m;r,t)\alpha^{(n)^*}(k;m;t)\mathrm{d}t$$
$$=\Phi_\mathrm{T}^{(n)}(k;m;r)\lambda^{(n)}(k;m)$$
$$\mathbf{R}\left(k;m;t,t'\right)=\int_r \mathbf{u}(k;m;r,t)\mathbf{u}^*\left(k;m;r,t'\right)r\,\mathrm{d}r$$
$$\lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau \mathbf{u}_\mathbf{T}(k;m;r,t)\alpha^{(n)*}(k;m;t)\,\mathrm{d}t$$
$$=\Phi_\mathbf{T}^{(n)}(k;m;r)\lambda^{(n)}(k;m).$$

2.4. Reconstruction

The reconstruction is given by

$$q(\xi,t) - \bar{q}(\xi) \approx \sum_{j=1}^{r} a_j(t)\varphi_j(\xi) \Rightarrow$$

$$q(r,\theta,t;x) = \bar{q}(r,\theta,t;x) + \sum_{n=1}\sum_{m=0}\alpha^{(n)}(m;t)\Phi^{(n)}(r;m;x)$$

Since the snapshot pod implementation is not error-free, the reconstruction can only be recovered by writing for factor $\gg 0$.

$$q(r,\theta,t;x) = \bar{q}(r,\theta,t;x) + (\text{factor } \gamma)\sum_{n=1}\sum_{m=0}\alpha^{(n)}(m;t)\Phi^{(n)}(r;m;x)$$

2.5. Reconstruction

In order to reconstruct in code, caseId.fluctuation = ’off’. This is incorrect. The necessary use of (factor γ) is incorrect

3. Derivation

To derive the questioned equation, consider the integral:

$$\frac{1}{\tau}\int_0^\tau \mathbf{u}_\mathrm{T}(k;m;r,t)\alpha^{(n)^*}(k;m;t)dt.$$

Substitute $\mathbf{u}_\mathrm{T}$ with its expansion:

$$\frac{1}{\tau}\int_0^\tau \left(\sum_l \Phi_\mathrm{T}^{(l)}(k;m;r)\alpha^{(l)}(k;m;t)\right)\alpha^{(n)^*}(k;m;t)dt.$$

3.1. 4 Derivation

Exchange the order of summation and integration, and apply orthogonality,

$$\sum_{l} \Phi_{\mathrm{T}}^{(l)}(k;m;r)\left(\frac{1}{\tau}\int_{0}^{\tau}\alpha^{(l)}(k;m;t)\alpha^{(n)^{*}}(k;m;t)dt\right).$$

Due to the orthogonality, namely that $\alpha^{(n)}$ and $\alpha^{(p)}$ are uncorrelated

$$\langle a^{(n)}\alpha^{(p)}\rangle = \lambda^{(n)}\delta_{np}$$

all terms where $l \neq n$ will vanish, and there remains only the $l = n$ term,

$$\Phi_{\mathrm{T}}^{(n)}(k;m;r)\left(\frac{1}{\tau}\int_{0}^{\tau}\alpha^{(n)}(k;m;t)\alpha^{(n)^{*}}(k;m;t)dt\right).$$

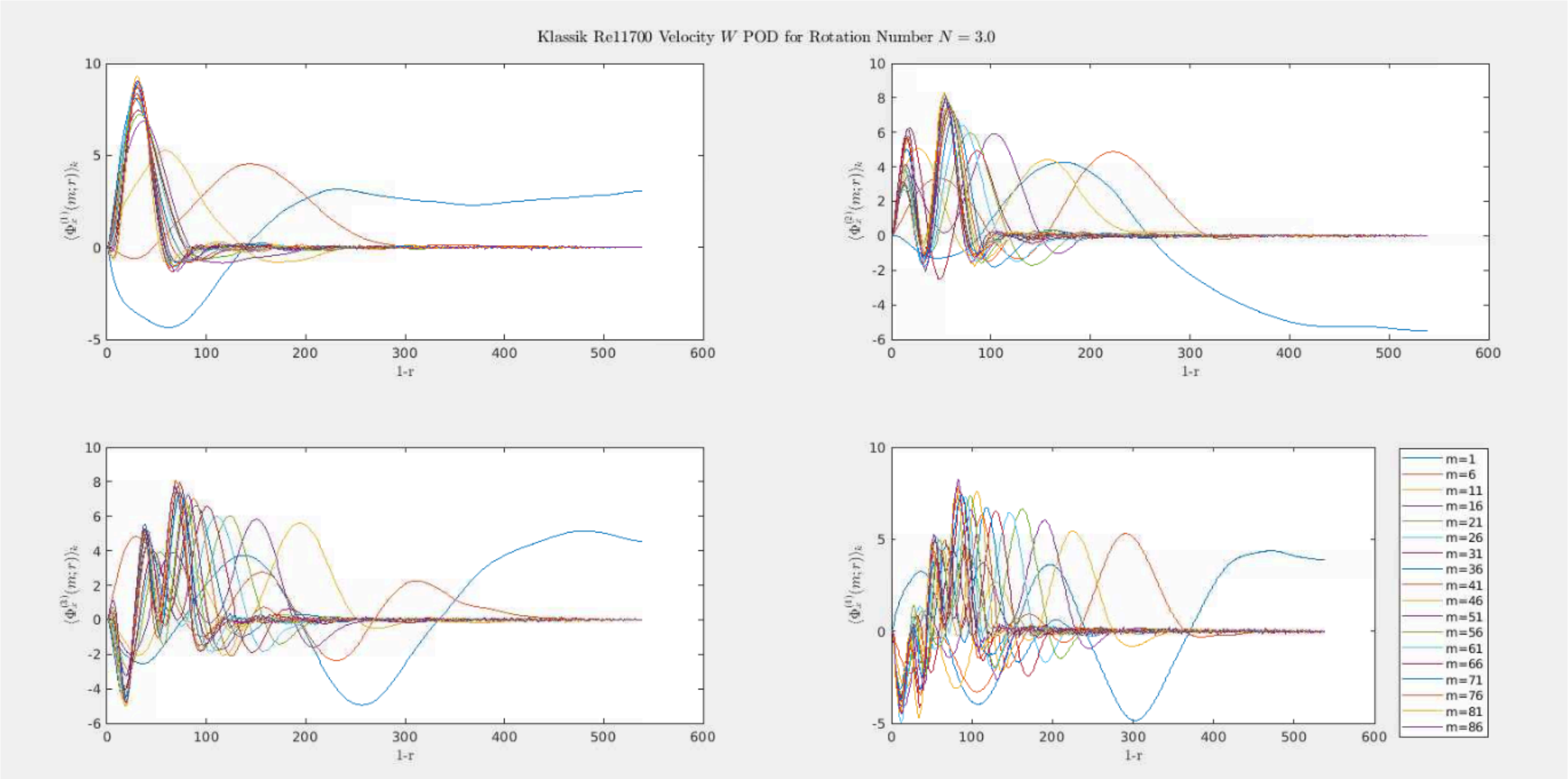
This derivation assumes the normalization of modes and their orthogonality, along with the eigenvalue relationship to simplify the original integral into a form that reveals the spatial structure ($\Phi_{\mathrm{T}}^{(n)}$) of each mode scaled by its significance $\left(\lambda^{(n)}\right)$.

3.2. 6 Derivation

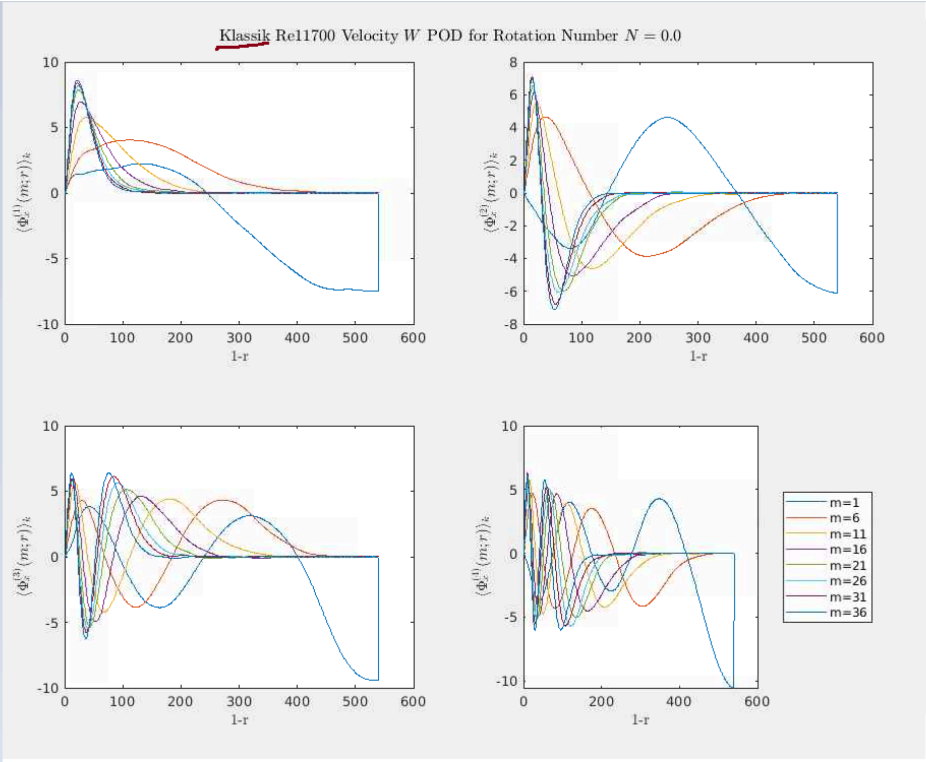
The cross-correlation tensor $\mathbf{R}$ is defined as $\mathbf{R}\left(k;m;t,t'\right)=\int_r \mathbf{u}(k;m;r,t)\mathbf{u}^*\left(k;m;r,t'\right) r \; \mathrm{d}r$. This tensor is now transformed from $[3r \times 3r']$ to a $[t \times t']$ tensor. The n POD modes are then constructed as,

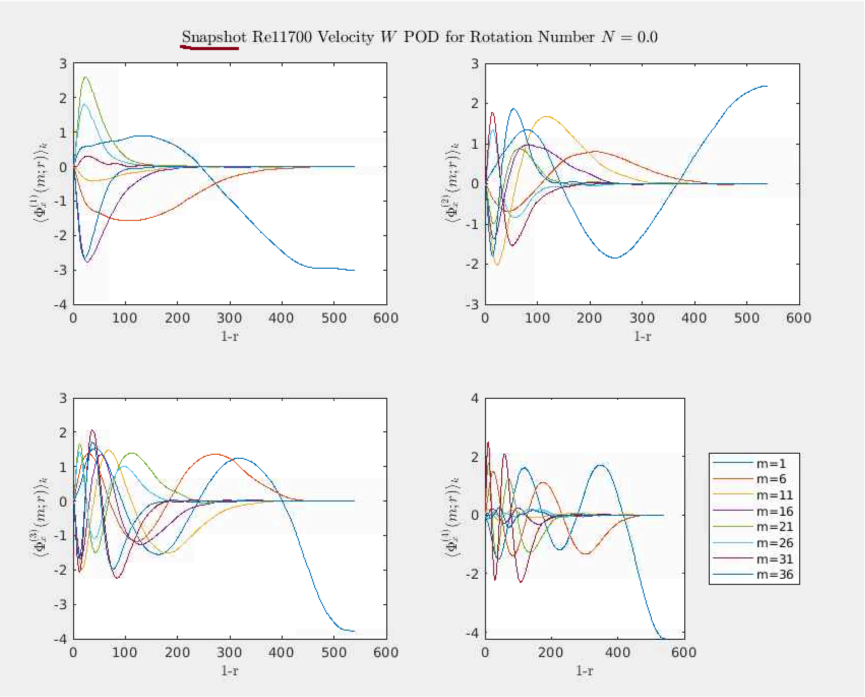
$$\lim_{\tau\to\infty}\frac{1}{\tau}\int_0^{\tau}\mathbf{u}_{\mathrm{T}}(k;m;r,t)\alpha^{(n)^*}(k;m;t)\mathrm{d}t=\Phi_{\mathrm{T}}^{(n)}(k;m;r)\lambda^{(n)}(k;m).$$

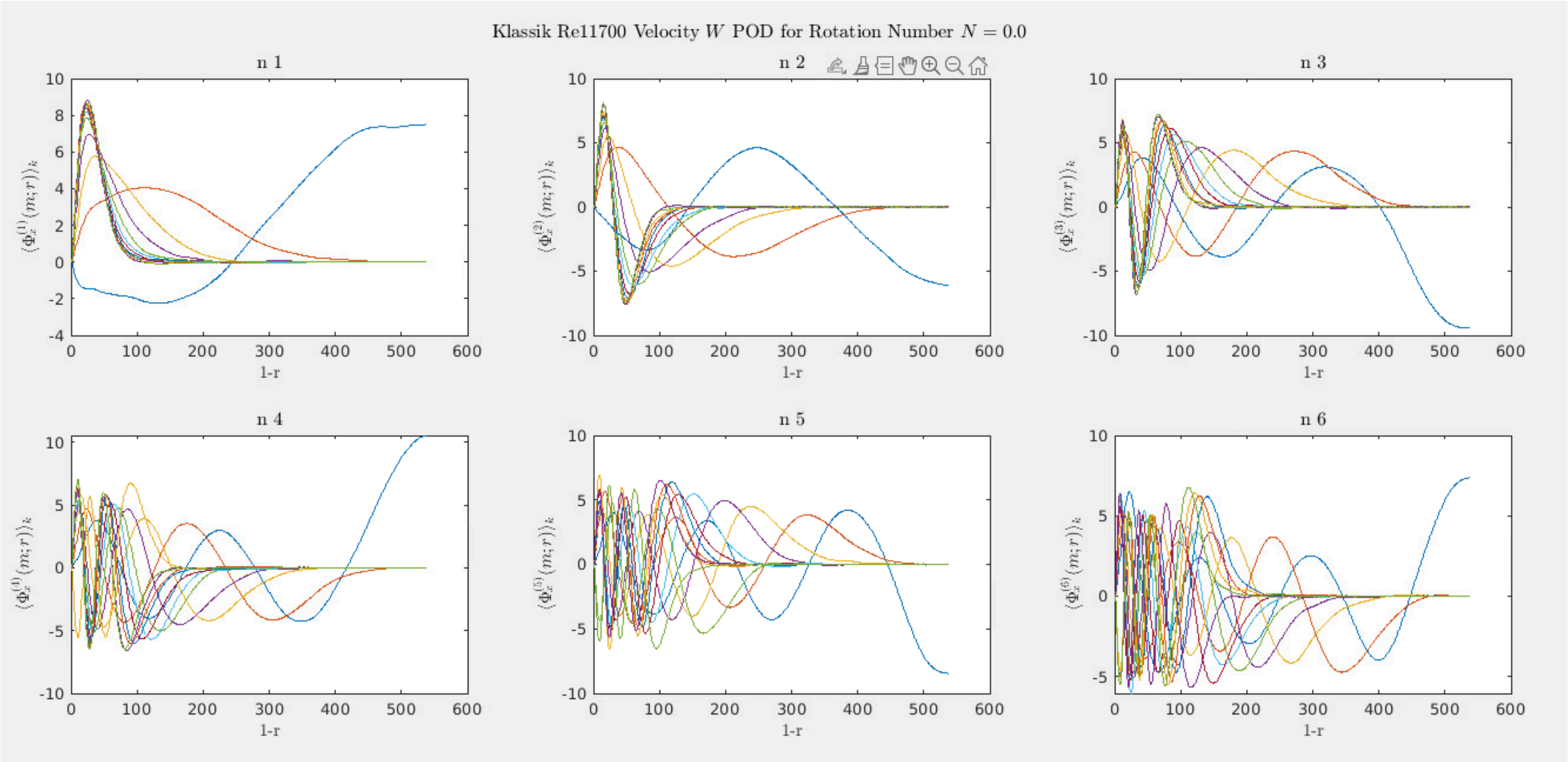
4. Result Comparison Classic/Snapshot



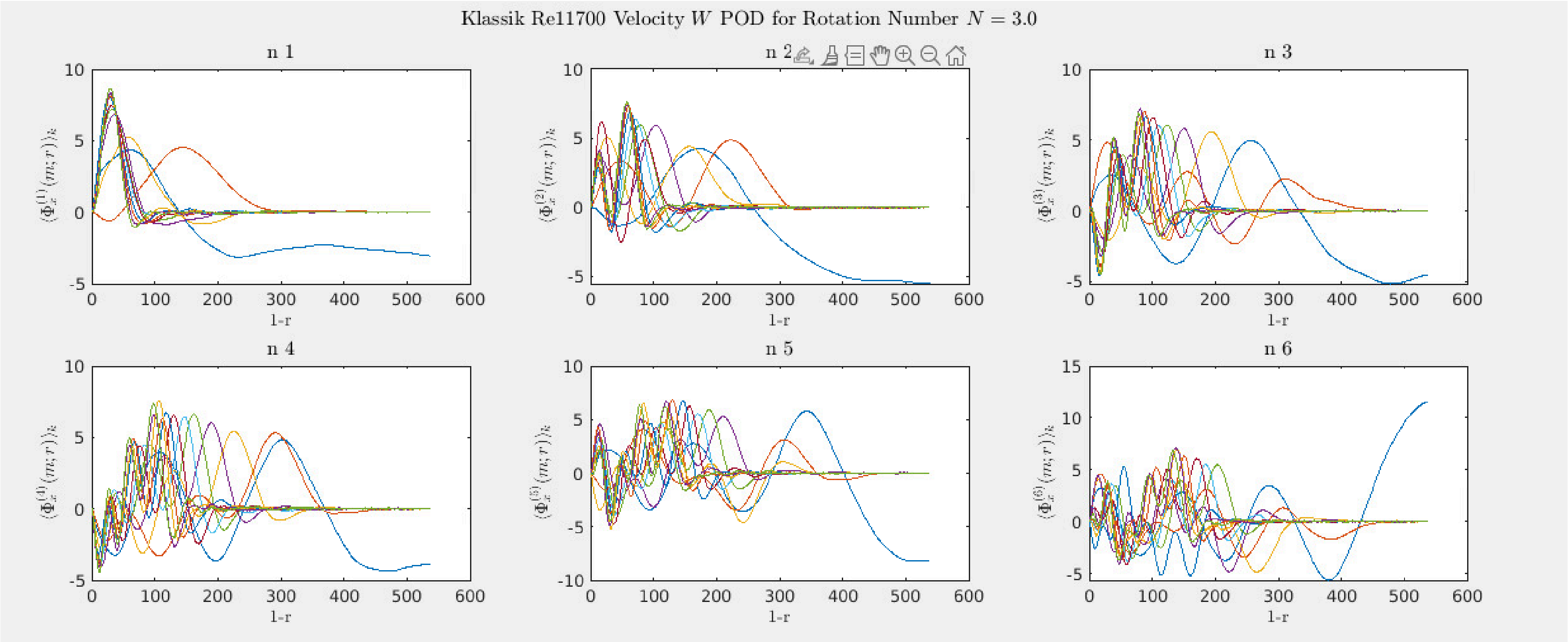
Klassik Re11700 Velocity W POD for Rotation Number $N = 3.0$
1-r
m=1
m=6
m=11
m=16
m=21
m=26
m=31
m=36
m=41
m=46
m=51
m=56
m=61
m=66
m=71
m=76
m=81
m=86


Klassik Re11700 Velocity W POD for Rotation Number $N = 0.0$
1-r
m=1
m=6
m=11
m=16
m=21
m=26
m=31
m=36


Snapshot Re11700 Velocity W POD for Rotation Number $N = 0.0$
1-r
m=1
m=6
m=11
m=16
m=21
m=26
m=31
m=36

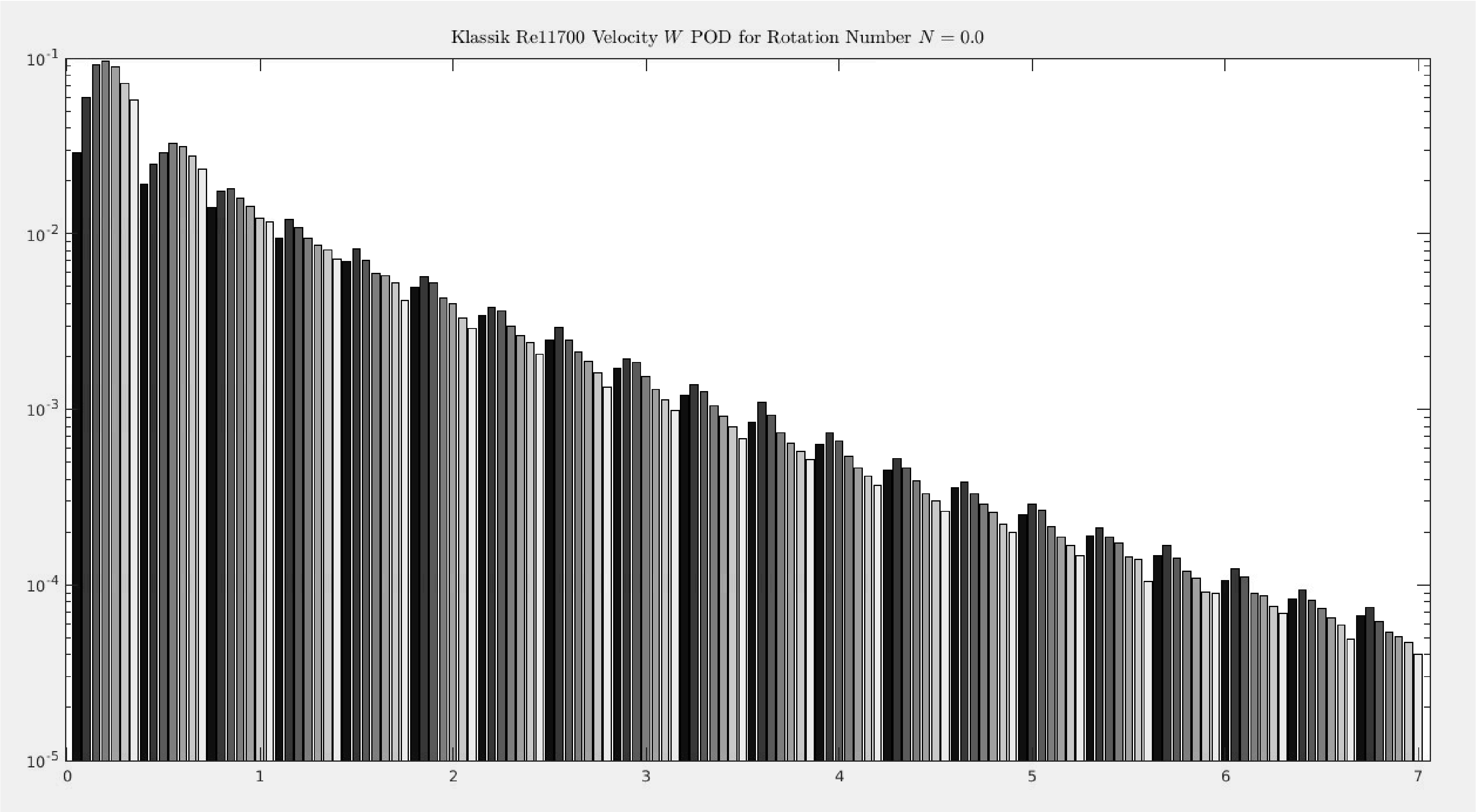


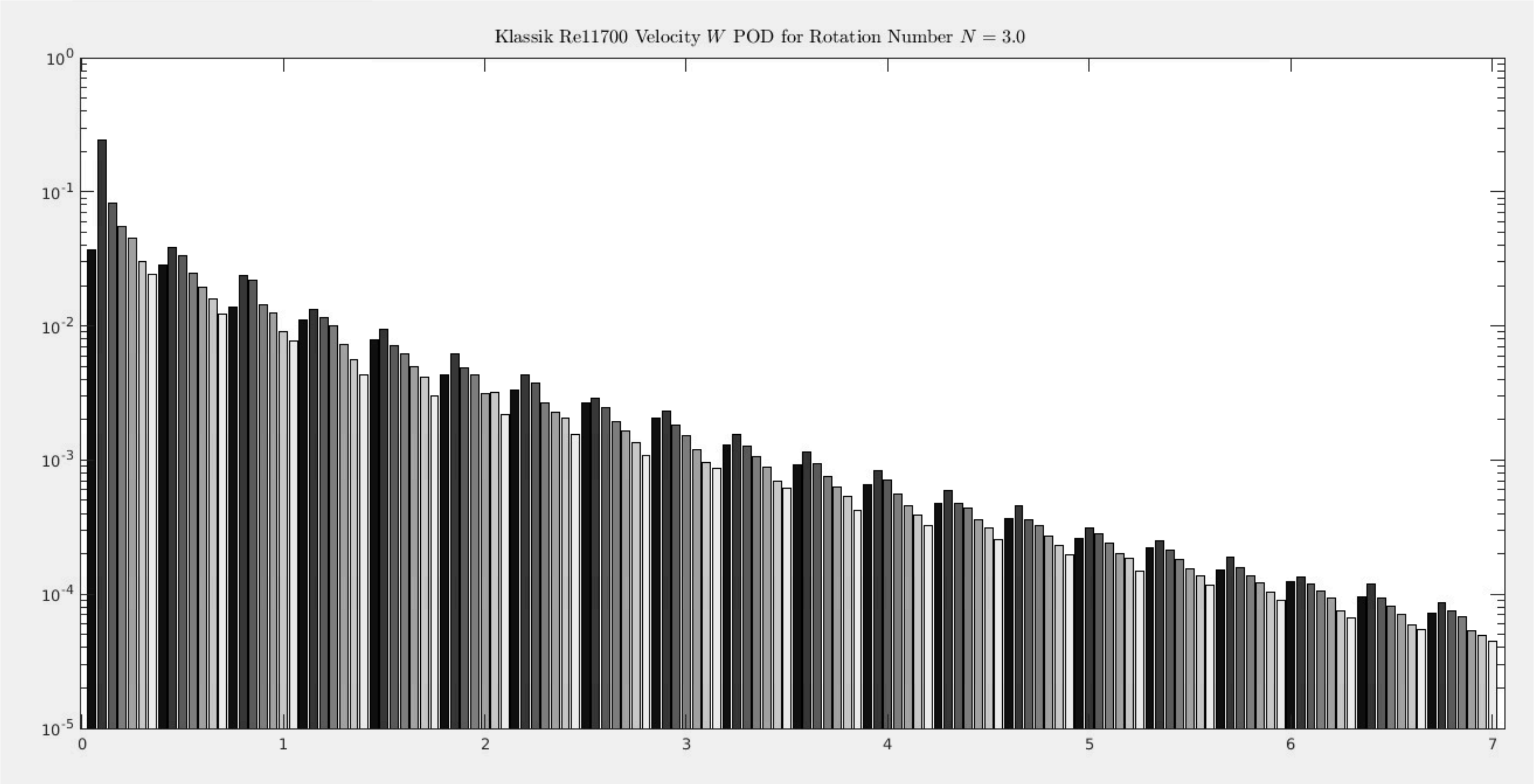
Klassik Re11700 Velocity W POD for Rotation Number $N = 0.0$
n 1
n 2
n 3
n 4
n 5
n 6
$\langle\Phi_x^{(1)}(m;r)\rangle_k$
$\langle\Phi_x^{(2)}(m;r)\rangle_k$
$\langle\Phi_x^{(3)}(m;r)\rangle_k$
$\langle\Phi_x^{(4)}(m;r)\rangle_k$
$\langle\Phi_x^{(5)}(m;r)\rangle_k$
$\langle\Phi_x^{(6)}(m;r)\rangle_k$
1-r
0
100
200
300
400
500
600
10
8
6
4
2
0
-2
-4
5
-5
-10



Klassik Re11700 Velocity W POD for Rotation Number $N = 3.0$
n 1
n 2
n 3
n 4
n 5
n 6
1-r

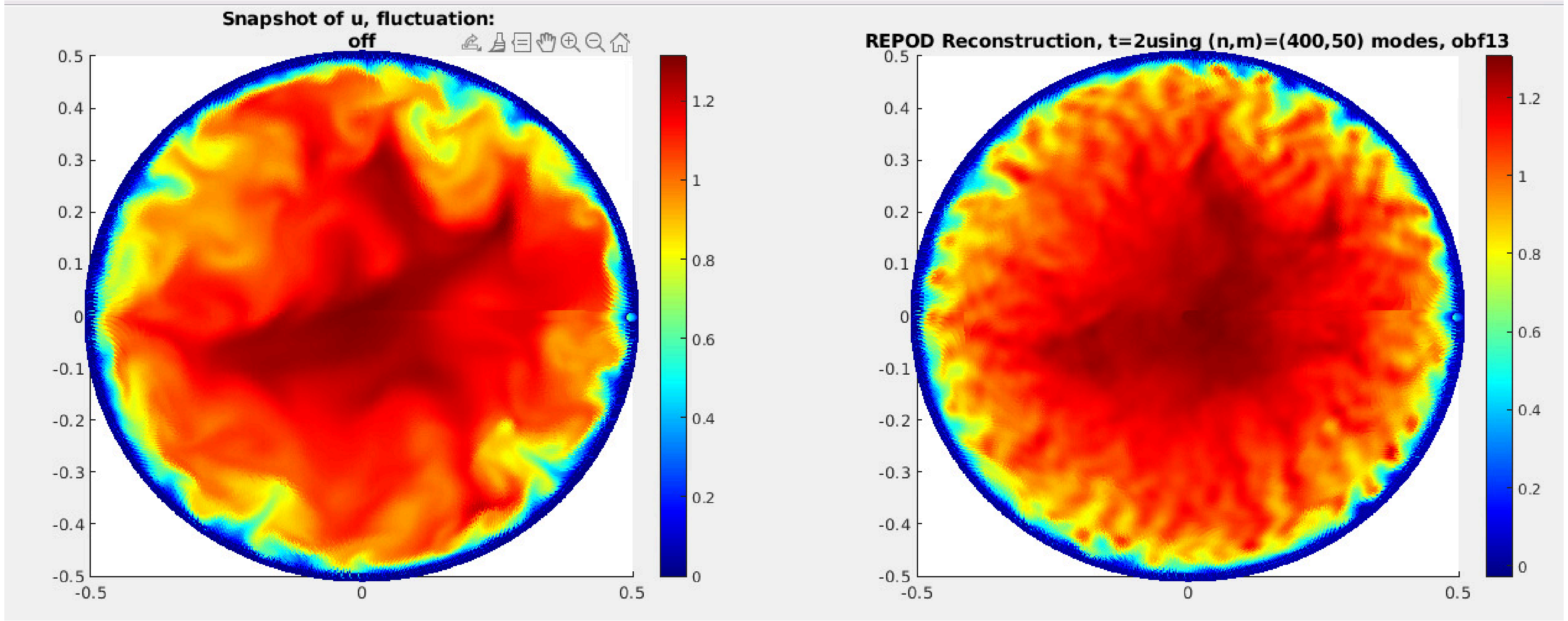
5. Energy n=0 Classic




Klassik Re11700 Velocity W POD for Rotation Number $N = 3.0$
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0
1
2
3
4
5
6
7

5.2. Analysis

6. Reconstruction


Snapshot of u, fluctuation:
off
REPOD Reconstruction, t=2using (n,m)=(400,50) modes, obf13
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.5
0
0.5
1.2
1
0.8
0.6
0.4
0.2
0