

Introdução a Decomposição Ortogonal Própria POD (*Proper Orthogonal Decomposition*)

Texto complementar à literatura do curso:
Tropea & Yarin & Foss, 2003., *“Handbook of Experimental Fluid
Mechanics – Part D Analysis and Post Processing of Data”*, Springer

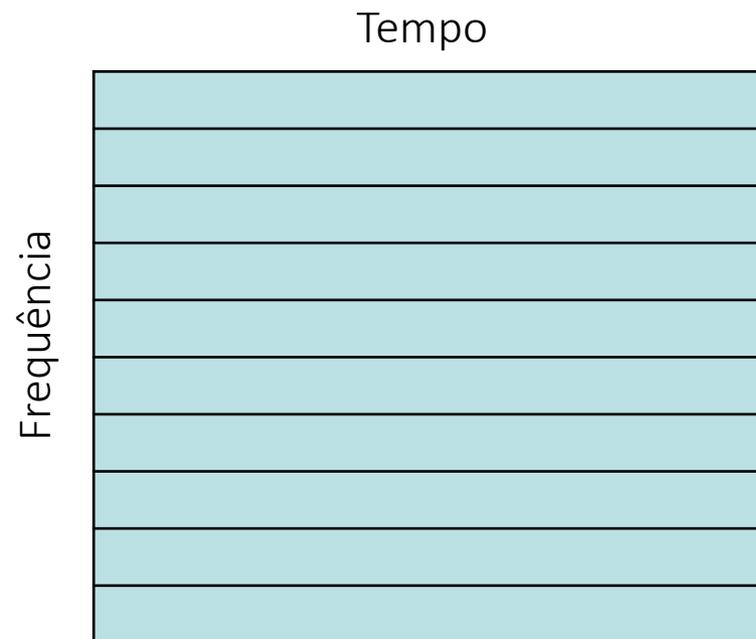
Revisão da análise no domínio da frequência

- Até aqui buscamos analisar o sinal através da decomposição do sinal em diferentes bases. No caso de Fourier, utilizamos senos e cossenos contínuos e de amplitude constante ao longo do tempo.

$$x(t) = \int X(\omega) e^{i\omega t} dt$$

Coeficientes Base

Representação no domínio da frequência - Fourier



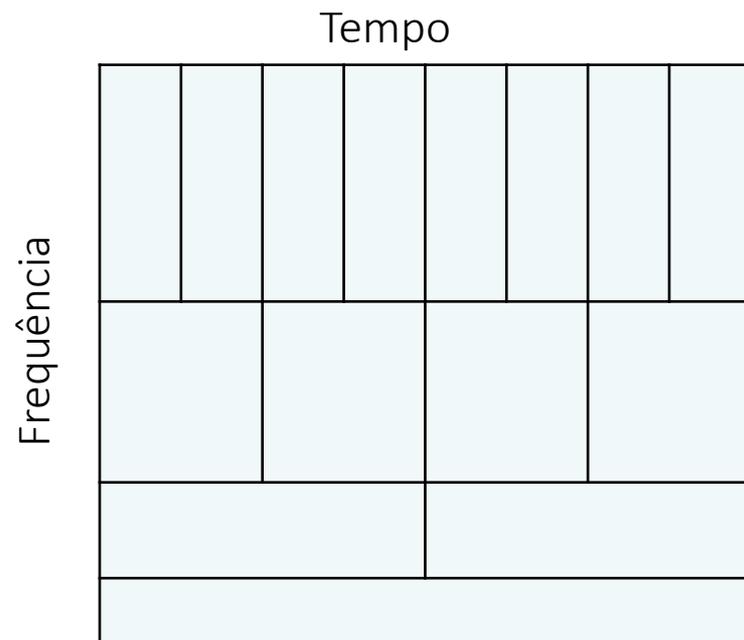
Revisão da análise no domínio da frequência

- No caso da transformada Wavelet, utilizamos uma base (função wavelet) que permitia localizar o sinal em um intervalo de tempo e de frequências.

Representação tempo-frequência Transformação Wavelet

$$x(t) = \frac{1}{C_\psi} \int \int W_{s,\tau} \psi_{s,\tau}(t) \frac{dsd\tau}{s^2}$$

Coeficientes Base



Introdução a decomposição ortogonal própria

- Até aqui, as bases utilizadas foram estabelecidas a priori para resolução do sinal no domínio das frequências e/ou tempo.
- Nenhuma informação sobre uma possível correlação espacial ou temporal entre os dados foi incluída na decomposição.
- A ideia com a decomposição ortogonal própria (POD) é buscar uma base de ótima, a partir dos dados.
- A definição de uma base ótima pode permitir uma representação simplificada de um problema físico de interesse.
- Isso é interessante para simplificar o objeto de estudo, mas também tem aplicações importantes na compactação de dados.

Introdução a decomposição ortogonal própria

- Na literatura essa técnica de decomposição é conhecida por diversos nomes: Karhunen-Loève, hotelling analysis, Principal Component Analysis e Single Value Decompositon.
- O nome POD é bastante utilizado na área de fluidos devido ao trabalho de Lumley (1967), que empregou a técnica no estudo de estruturas coerentes em escoamentos turbulentos
- A técnica tem aplicações em diversas áreas, dentre as quais destacam-se: a análise de imagens, a redução de ordem em sistemas dinâmicos, a compressão de dados, a implementação de controle ótimo, treinamento de redes neurais, modelagem de turbulência, etc.

Introdução a decomposição ortogonal própria

- A reconstrução de um sinal pode ser, de maneira geral, representada pelo somatório do produto entre uma base e um fator de escala, na forma.

$$U(x, t) = \sum_{k=1}^{\infty} a_k(t) \Phi_k(x)$$

Coeficientes Base

Onde $a_k(t)$ são os coeficientes da decomposição e $\Phi_k(x)$ os modos POD (ou base).

- A decomposição em modos próprios, busca maximizar a energia capturada pelos primeiros modos k .
- Para o número de modos $K \rightarrow \infty$ a representação tende a ser exata. Para uma sequencia finita, que contém N amostras de dados, o valor máximo de K é N .

Introdução a decomposição ortogonal própria

- Como o nome diz a decomposição é feita com modos ortonormais, de modo que

$$\int_x \Phi_{k1}(x)\Phi_{k2}(x)dx = \begin{cases} 1 & \text{se } k1 = k2 \\ 0 & \text{se } k1 \neq k2 \end{cases}$$

- A ortonormalidade implica que cada coeficientes de a_k depende somente do modo próprio Φ_k , logo

$$a_k(t) = \int_x U(x, t)\Phi_k(x)dx$$

Introdução a decomposição ortogonal própria

- A aproximação da função $u(x, t)$ a partir dos coeficientes de amplitude a_k e das bases Φ_k consiste em encontrar funções ortonormais $\{\Phi_k\}_{k=1}^K$ para $K < N$, de maneira que

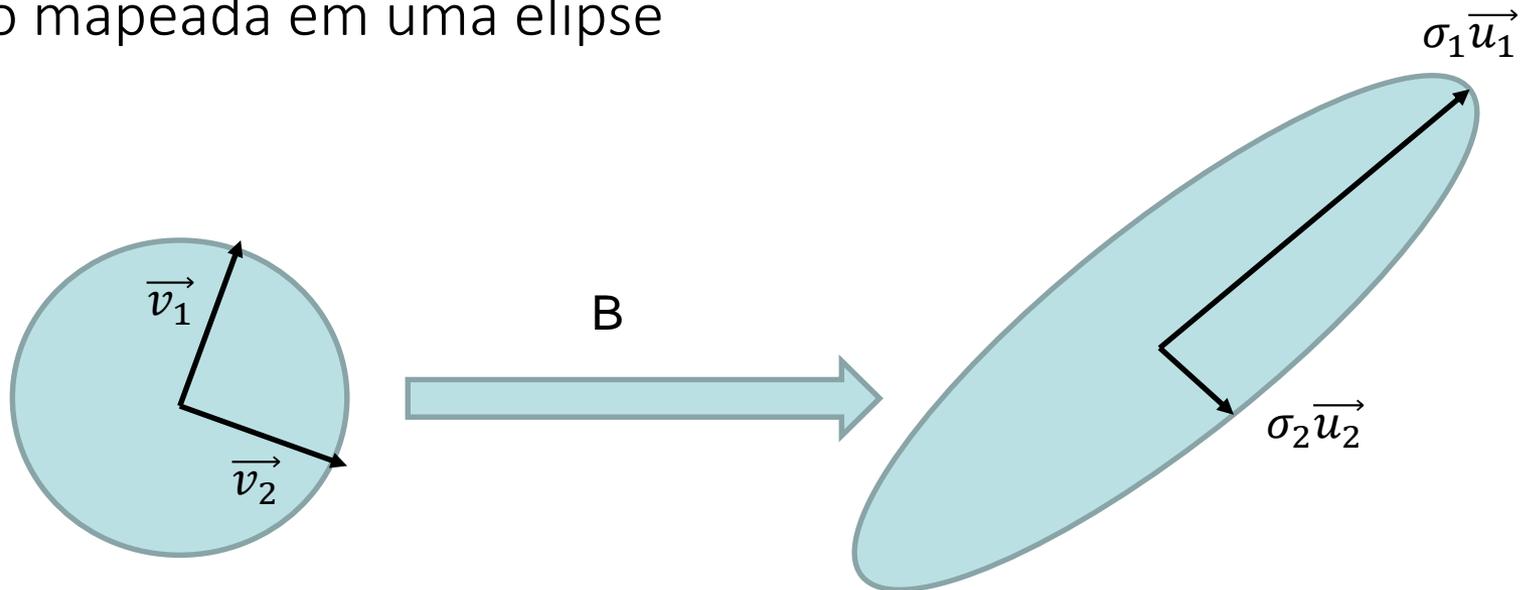
$$\min \sum_{i=1}^N \left\| U(x, t_i) - \sum_{k=1}^K a_k(t_i) \Phi_k(x) \right\|^2$$

- Na prática, a solução desse problema consiste em organizar os dados em uma matriz das amostras obtidas em tempos distintos e resolver o problema de minimização do erro
- Isso pode ser entendido como a maximização da descrição de $U(x, t)$ com o somatório de $a_k \Phi_k$.

Como resolver isso?

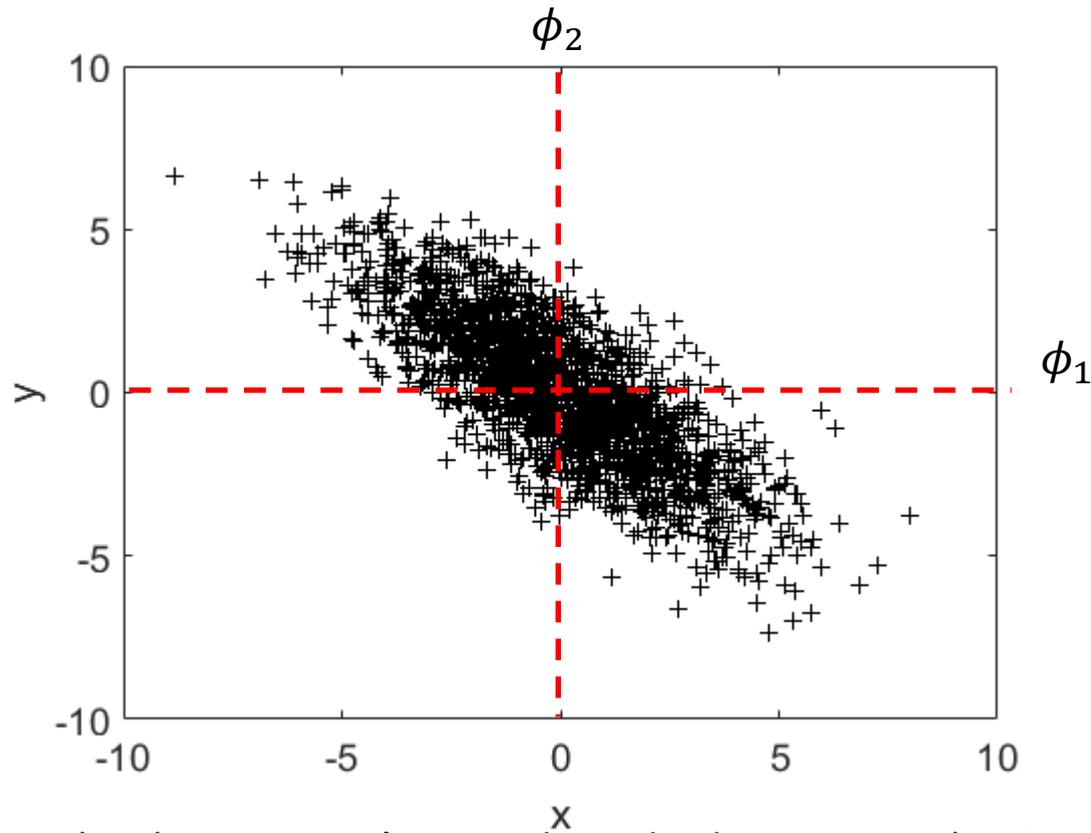
Revisão de álgebra linear

- É feita aqui uma breve revisão de álgebra linear (Mais detalhes na disciplina de Métodos Matemáticos II)
- Relembrando que a multiplicação de um vetor qualquer \vec{v} por uma matriz genérica B resulta em rotação e dilatação/compressão desse vetor. Um exemplo clássico, encontrado em muitos livros é o de uma esfera unitária sendo mapeada em uma elipse



Revisão de álgebra linear

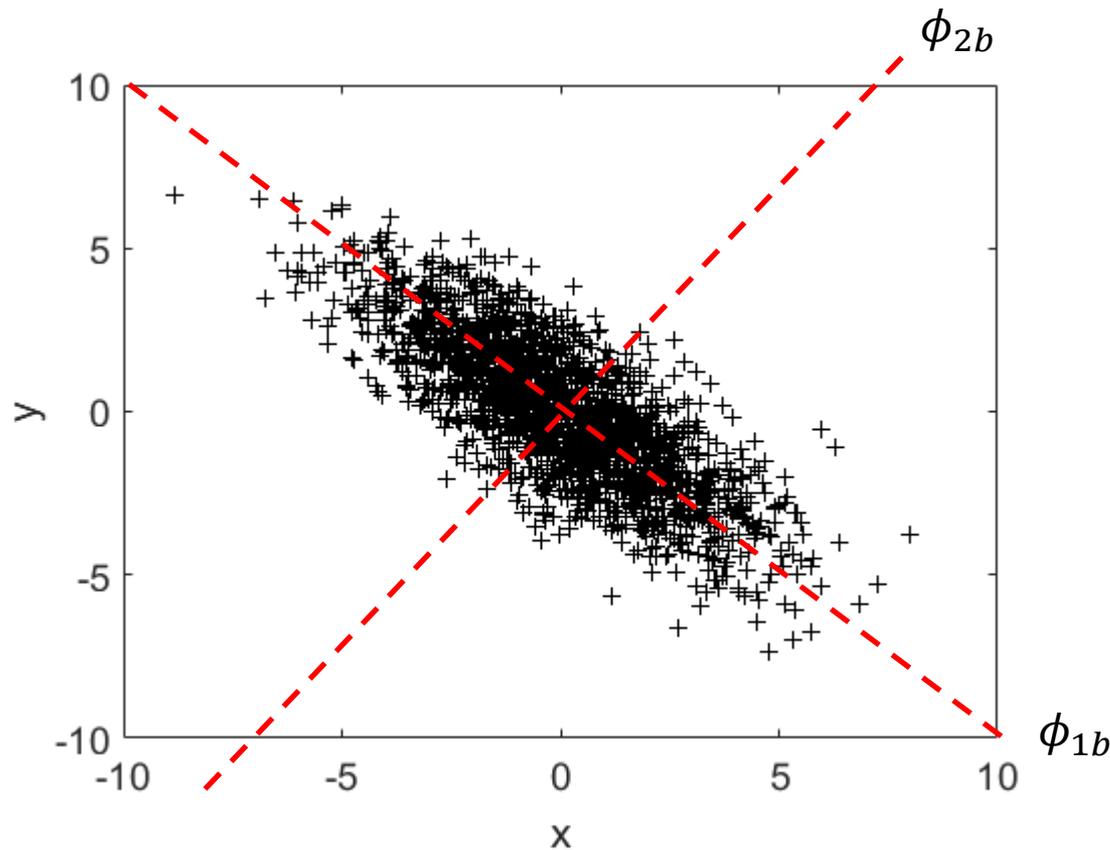
- Consideremos uma série de dados com remoção dos valores médios $U(x, t) = \tilde{U}(x, y) - \bar{U}$ conforme ilustrado na Figura abaixo.



- Pode-se calcular a variância dos dados em relação aos eixos ortogonais ϕ_1 e ϕ_2

Revisão de álgebra linear

- Se alterarmos os eixos ortogonais, de modo a maximizar a variância dos dados em um desses eixos



- Obtém-se o que se chama de eixos principais ou modos próprios ortogonais dos dados

Fundamentos de POD

- Logo para projetar os dados em um outro eixo precisamos somente efetuar o produto escalar do tipo

$$A = U\phi$$

- A variância nos eixos principais pode ser calculada através da relação abaixo (ver aula de matriz de covariância)

$$var_A = diag \left(\frac{1}{N-1} A^T A \right)$$

Onde o índice “ T ” se refere a transposta de “ A ”

- Como encontrar as direções dos eixos principais?

Isso pode ser obtido através dos autovetores da matriz de covariância dos dados ($C = \frac{1}{N-1} U^T U$)

Fundamentos de POD

- Sabe-se que na base própria ortogonal a matriz de covariâncias de A deve ser diagonal (eixo principal contém as variâncias e demais posições as covariâncias), pois na base própria a covariância entre eixos deve ser nula.
- Logo, a base ótima corresponde aos autovetores da matriz de covariâncias.
- O problema de autovalor pode ser escrito como:
$$C\phi = \phi\Lambda$$
- O problema pode ser facilmente resolvido em MATLAB com o comando: `[phi, Lambda]=eig(C)`

Fundamentos de POD

- Os dados na base própria são obtidos pela projeção dos dados nos autovetores

$$A = U\phi \Rightarrow U = A\phi^{-1} = A\phi^T$$

- Com essa relação é possível reconstruir U para um número truncado de modos k .
- Resumindo a técnica POD clássica consiste em:
 1. Organizar os dados com snapshots ao longo das linhas
 2. Calcular a matriz de covariância dos dados
 3. Resolver o problema de autovalor
 4. Ordenar os dados do maior autovalor para o menor
 5. Projetar os dados na base principal
 6. Reconstruir o modo desejado

Fundamentos de POD

- Organização da matriz de realizações (ou retratos – do inglês snapshots)

$$U = \begin{pmatrix} u'(x_1, t_1) & u'(x_2, t_1) & \dots & u'(x_M, t_1) \\ u'(x_1, t_2) & u'(x_2, t_2) & \dots & u'(x_M, t_2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ u'(x_1, t_N) & u'(x_2, t_N) & \dots & u'(x_M, t_N) \end{pmatrix}$$

Onde u' é uma amostra de u subtraída da sua média temporal



Fundamentos de POD

```
%*****  
%           Script for Demonstration of  
%           Proper Orthogonal Decomposition  
%*****  
% Signal processing course  
% Last Review 24- Jun 2019, de Paula I.B. DEM PUC  
%*****  
% INPUT DATA  
% Y       - snapshots of data Y(i,j,time)  
% n_mode  - number of POD mode shown  
%%*****  
Y_mean = mean(Y,3);  
Desv_Y=(Y-Y_mean); % Remove mean data from files  
n_snaps=length(desv_y(1,1,:)); % number of snapshots  
n_lines=length(desv_y(:,1,1));  
n_columns=length(desv_y(1,(:,1)));
```

Exemplo

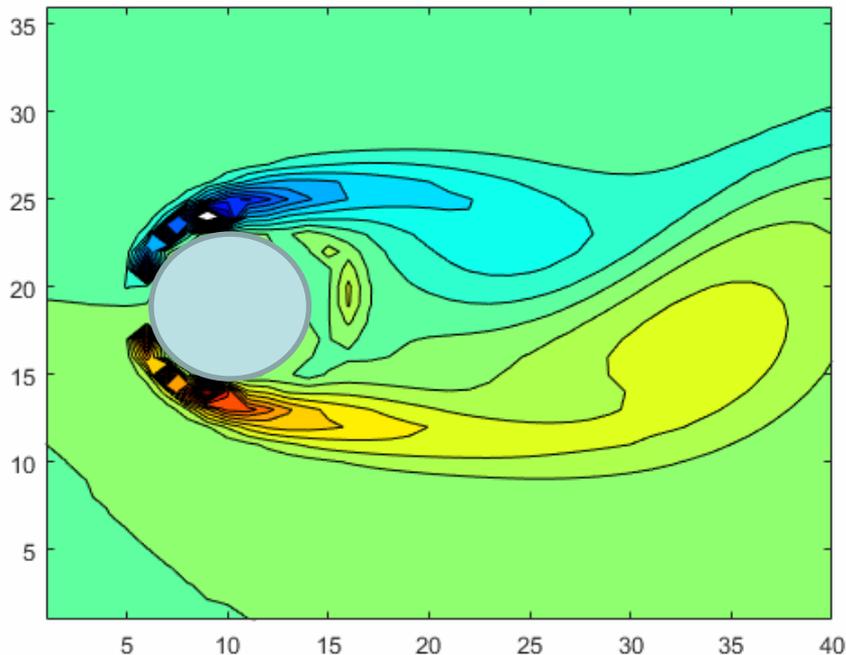
```
%%***** CLASSICAL POD *****  
% solving equation  $[A*V=V*L]$   
Y_modif=reshape(permute(Desv_Y,[3 2 1]),n_snaps,[]); % reorganizing  
% matrix of data lines are snapshots and columns are spatial data(i,j)  
C=(Y_modif'*Y_modif)/(n_snaps-1); %Matrix of covariances (Matrix A)  
[V,L] = eig(C,'vector'); % computing eigenvalues  
[Lambda,Pos_lambda]=sort(L,'descend'); % Sorting L in descend order  
V=V(:,Pos_lambda); % Adjust V to follow the same sorting of Lambda  
A=Y_modif*V; % Project data matrix y_modif onto each eigenvector V  
  
% reconstruction of 1st snapshot (for reconstruction using several  
% POD modes is necessary to sum n_reconstrued modes)  
% applying the inverse procedure adopted in reorganization of y_modif  
tmp=A(:,n_mode)*V(:,n_mode)';  
POD_mode=reshape(tmp(1,:),n_columns,[]); clear tmp
```

Exemplo

- escoamento na esteira de um cilindro. Campos de vorticidade

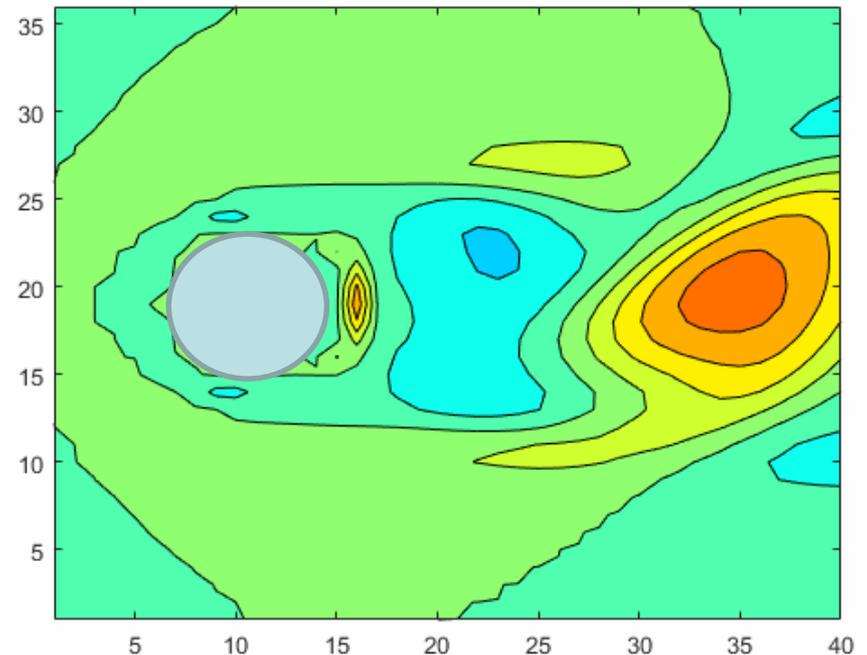
Matriz de dados: 36x40x100 double

Vorticidade com a média



Caxis([-12 12])

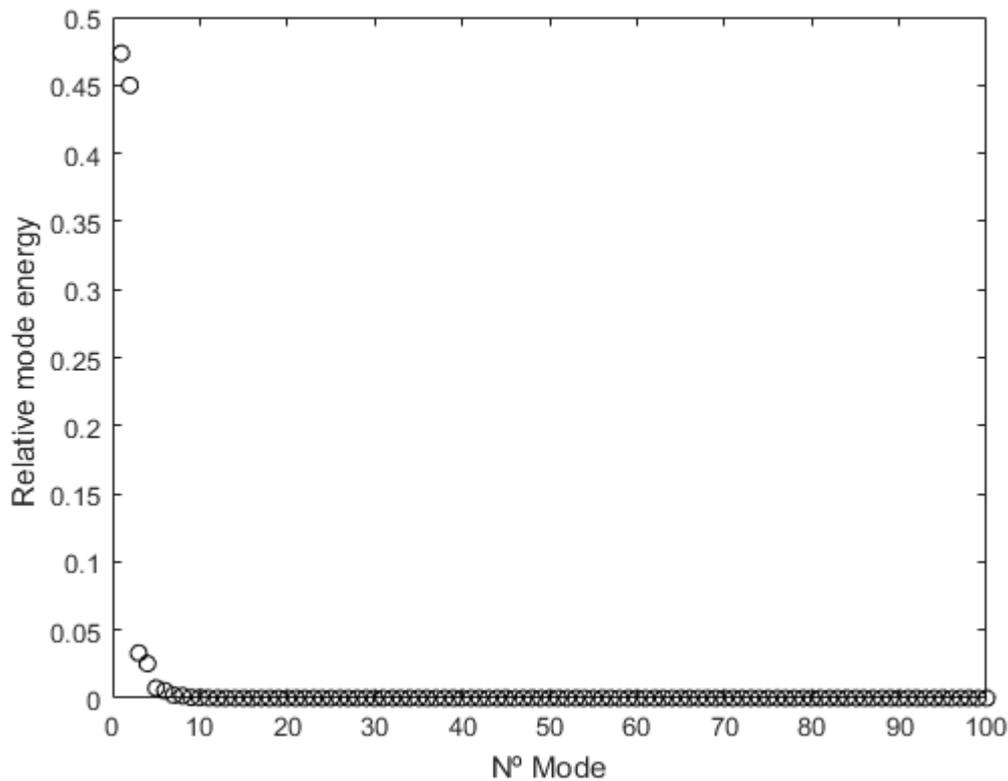
Vorticidade sem a média



Caxis([-4 4])

Exemplo

➤ Análise POD - Autovalores



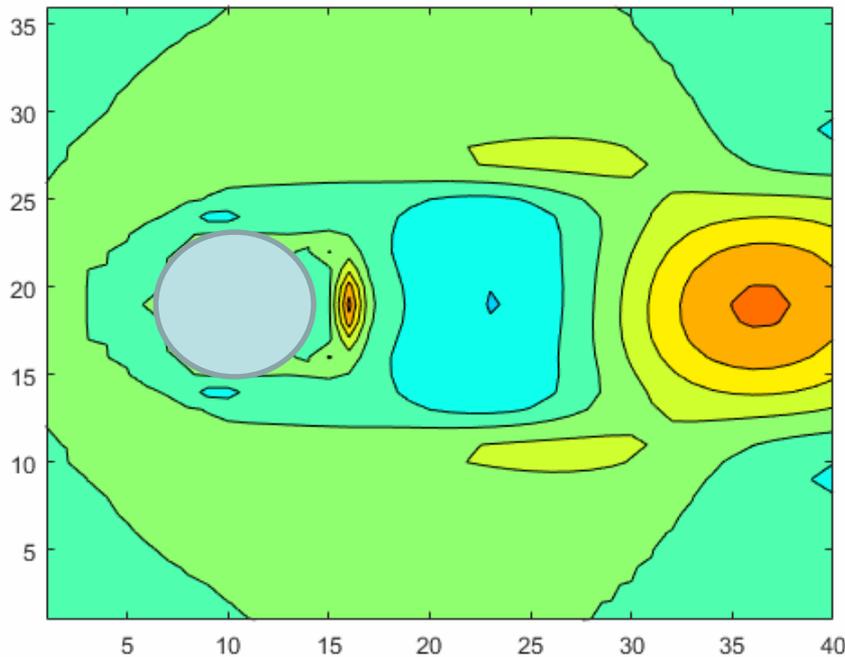
Quase 100% da energia está contida em apenas dois modos.

Exemplo

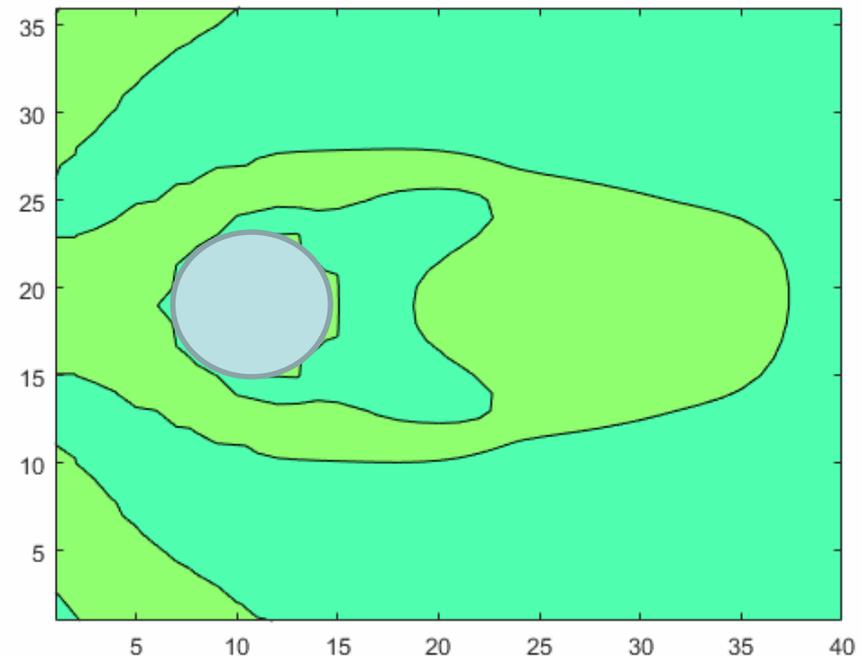
- Análise POD - Autovetores

Observa-se estrutura espacial parecida defasamento no tempo (ortogonais).
Na energia relativa, esses modos aparecem pares com energia similar.
Porquê??

Modo 1



Modo 2



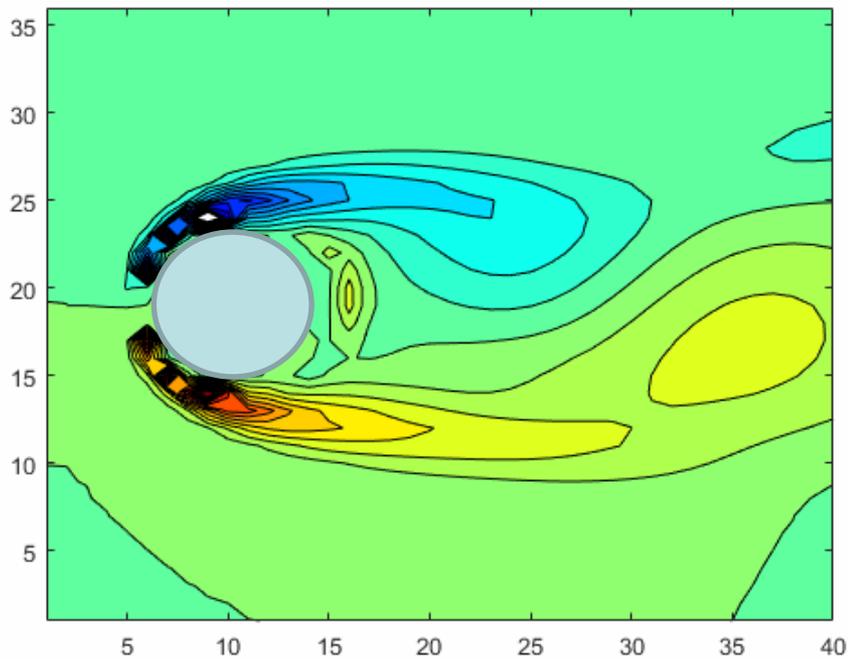
Caxis([-4 4])

Exemplo

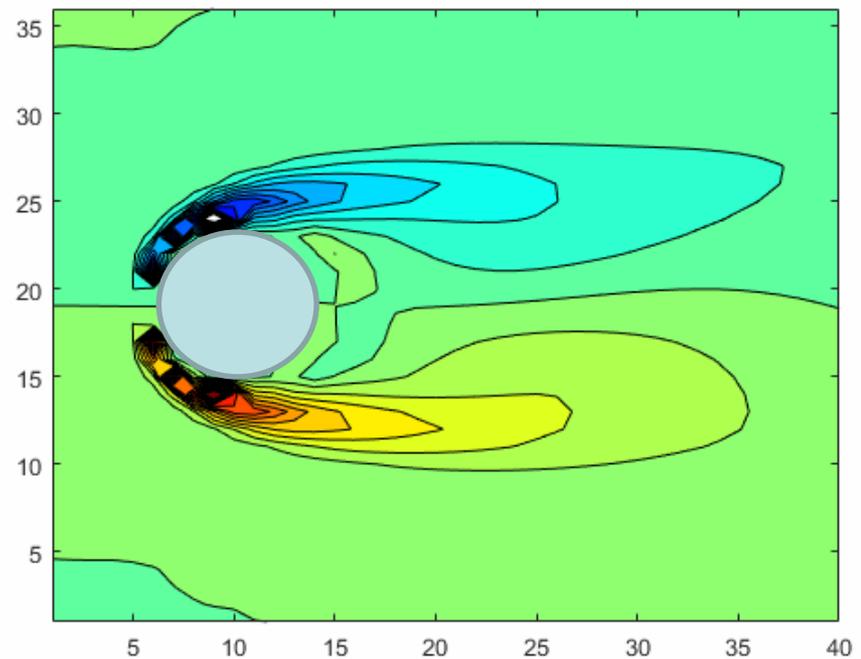
➤ Análise POD – Reconstrução

$$U = \sum_{k=1}^N U_k$$

Média + Modo 1



Média + Modo 2



Caxis([-12 12])

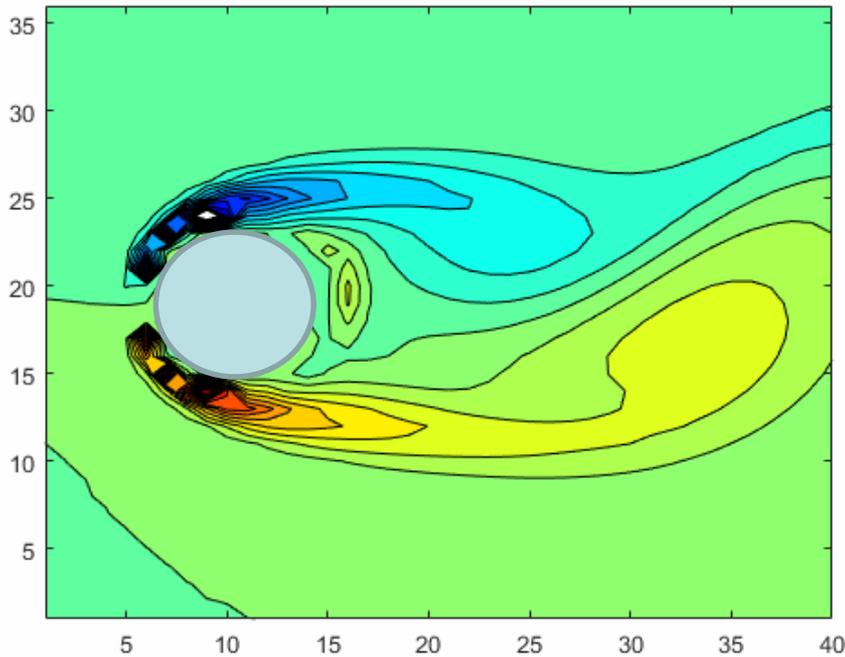
Exemplo

Representação com poucos modos se aproxima bem do campo original

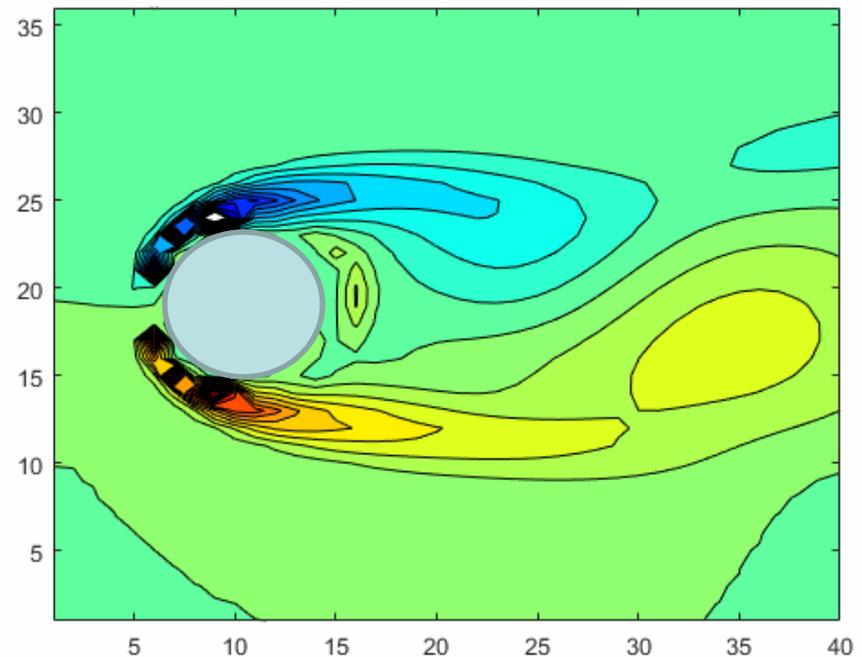
➤ Análise POD – Reconstrução

$$U = \sum_{k=1} U_k$$

Sinal original



Média + Modos 1 e 2



Caxis([-12 12])

Análise dos resultados

- É importante lembrar que os autovalores e autovetores são obtidos a partir dos dados amostrados.
- Os autovalores obtidos a partir de um número grande de amostras devem ser iguais àqueles obtidos com poucas amostras?
- É de esperar que o problema apresente uma convergência para a partir de um número de amostras. (ver Hekmati et al. (2011). *About the convergence of POD and EPOD modes computed from CFD simulation*, *Comp & Fluids*, v.50)
- Infelizmente, não é possível definir um número que seja suficiente para todos os casos.
- Isso depende do nível de ruído espúrio e da energia dos modos

Análise dos resultados

- Um outro ponto importante é que no cálculo da matriz de covariâncias cria-se uma matriz quadrada M_x por M_y elementos (no caso de um campo de velocidades).
- Assim, pode-se facilmente atingir alguns milhões de elementos dependendo do tamanho das matrizes de dados.
- Para contornar esse problema Sirovich (1987) introduziu uma modificação na técnica.
- A modificação proposta por Sirovich foi no cálculo da matriz de covariâncias. Ao invés de utilizar correlações espaciais ele sugeriu utilizar uma correlação na dimensão temporal.

Análise dos resultados

- Com isso o cálculo de C fica: $C_s = \frac{1}{N-1} UU^T$, de modo que a matriz se torna NxN, onde N é o número de amostras no tempo
- Os autovetores e autovalores de Cs podem ser calculados com o comando `[As, Lambdas]=eig(Cs)`
- A decomposição é feita na dimensão do tempo N.
- Para a obtenção dos coeficientes no espaço, é necessário projetar os dados nos modos temporais através de

$$\phi s = U^T A s$$

- A matrix ϕs contém N coeficientes espaciais (equivalentes aos modos do método clássico)

Exemplo

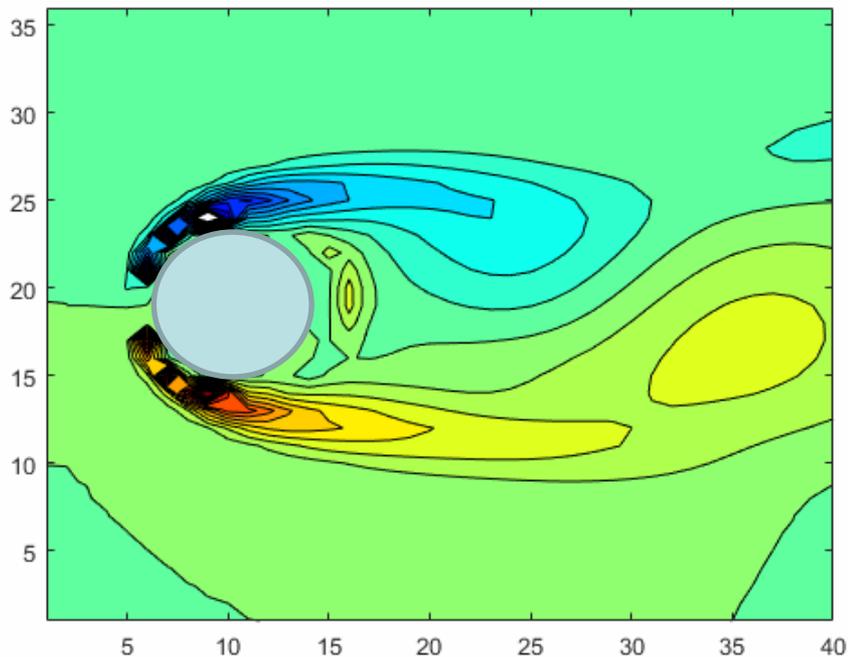
```
%%***** SNAPSHOT POD *****  
% solving equation  $[\Phi * A_s = A_s * L]$   
Y_modif=reshape(permute(Desv_Y,[3 2 1]),n_snaps,[]); % reorganizing  
% matrix of data lines are snapshots and columns are spatial data(i,j)  
Cs=(Y_modif'*Y_modif)/(n_snaps-1); %Matrix of covariances (Matrix A)  
[As,L] = eig(Cs,'vector'); % computing eigenvalues  
[Lambda,Pos_lambda]=sort(L,'descend'); % Sorting L in descend order  
As=As(:,Pos_lambda); % Adjust V to follow the same sorting of Lambda  
Phi=Y_modif*As; % Project data matrix y_modif onto each eigenvector As  
  
% reconstruction of 1st snapshot (for reconstruction using several  
% POD modes is necessary to sum n_reconstructed modes)  
% applying the inverse procedure adopted in reorganization of y_modif  
tmp=As(:,n_mode)*Phi(:,n_mode)';  
POD_mode=reshape(tmp(1,:),n_columns,[]); clear tmp
```

Exemplo

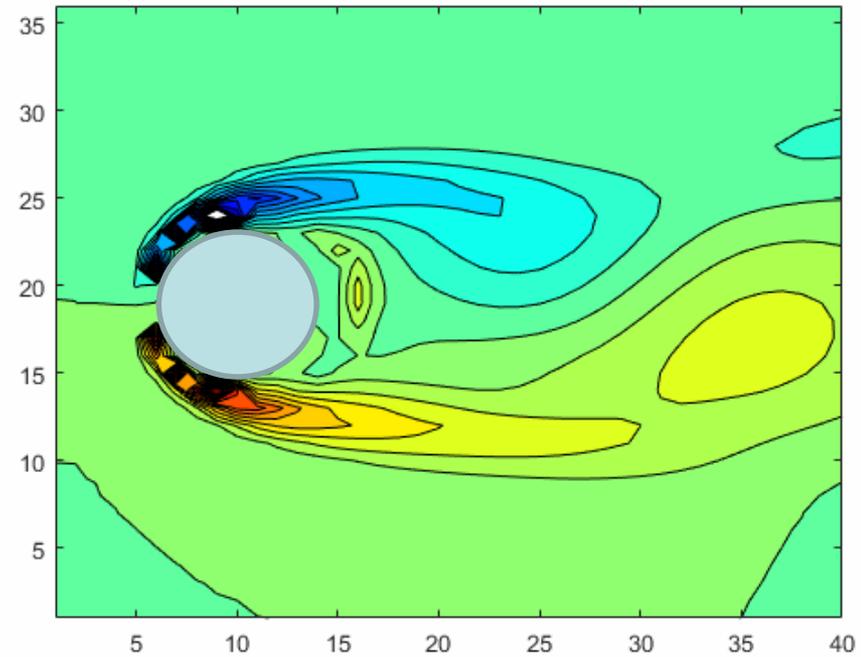
- Análise POD – Reconstrução

$$U = \sum_{k=1}^N U_k$$

Média + Modo 1 – POD CLASSICO



Média + Modo 1 – SNAPSHOT POD



Caxis([-12 12])

Decomposição em valores singulares

- A técnica POD pode ser implementada de uma terceira forma, utilizando decomposição em valores singulares

$$U = L\hat{\Sigma}R^T$$

- Em essência a decomposição com autovalores só se aplica a matrizes quadradas, enquanto que SVD permite diagonalizar qualquer matriz retangular.
- A conexão entre SVD e POD pode ser observada se avaliarmos as matrizes C e C_s , substituindo U de acordo com a relação da SVD

$$C = \frac{1}{N-1} (U^T U) = \frac{1}{N-1} \left[(L\hat{\Sigma}R^T)^T (L\hat{\Sigma}R^T) \right]$$
$$C = \frac{1}{N-1} \left[R\hat{\Sigma}L^T L\hat{\Sigma}R^T \right] = \frac{1}{N-1} \left[R\hat{\Sigma}^2 R^T \right]$$

Decomposição em valores singulares

$$C_S = \frac{1}{N-1} (UU^T) = \frac{1}{N-1} \left[(L\hat{\Sigma}R^T)(L\hat{\Sigma}R^T)^T \right]$$
$$C_S = \frac{1}{N-1} \left[L\hat{\Sigma}R^T R\hat{\Sigma}L^T \right] = \frac{1}{N-1} \left[L\hat{\Sigma}^2 L^T \right]$$

➤ Nota-se que as abordagens apresentam estreita relação. Na metodologia clássica os modos POD são dados pelo vetores singulares R e no caso do snapshot são dados pelos vetores singulares L.

➤ A implementação em MATLAB fica:

```
%*****
```

```
y_modif=reshape(desv_y,n_lines*n_columns,[]);
```

```
[L,S,R] = svd(y_modif,'econ'); % L are POD modes
```

```
tmp=L(:,n_mode,1)*S(n_mode,n_mode,1)*R(:,n_mode,1)';
```

```
pod_mode=reshape(tmp(:,1),n_lines,[]);
```

Aplicação na compactação de dados

- Conforme foi dito na introdução, a técnica é muito utilizada na compactação de dados e na elaboração de modelos de ordem reduzida.
- Exemplo imagem de palhaço (MATLAB)

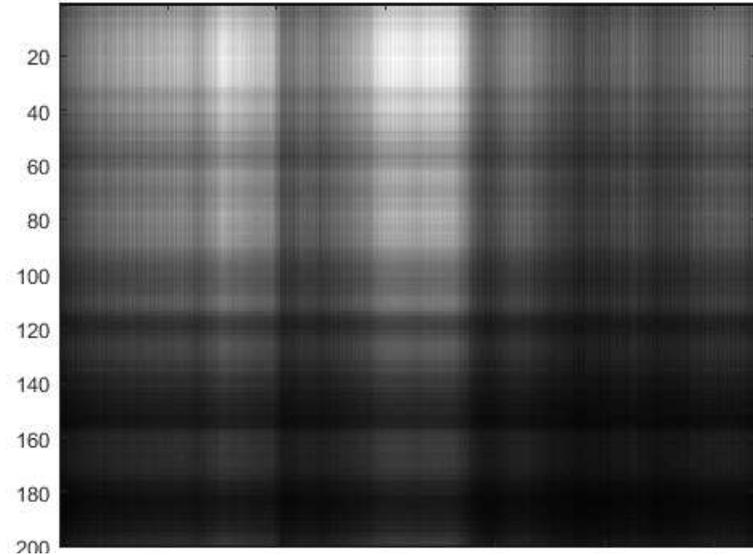
```
a=load('clown')
X=a.X; figure; imagesc(X); colormap('gray'); title('Original Image');
% Single Value Decomposition
[L,S,R] = svd(X,'econ');% Reconstruction using n_modes
nmodes_rec =1; % cases with 1 ,15 , 30
img1=L(:,1:nmodes_rec)*S(1:nmodes_rec,1:nmodes_rec)*R(:,1:nmodes_rec)';
figure
imagesc(img1); colormap('gray'); title(['Nº modes used: ' num2str(nmodes_rec)]);
```

Aplicação na compactação de dados

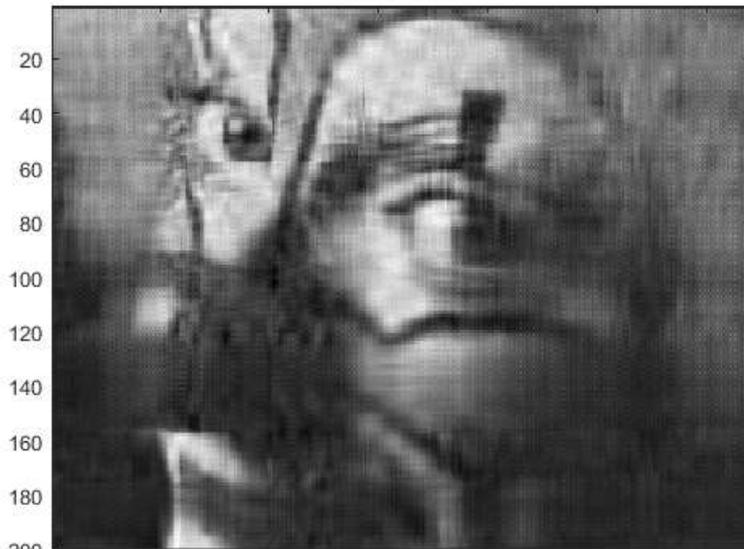
Original Image



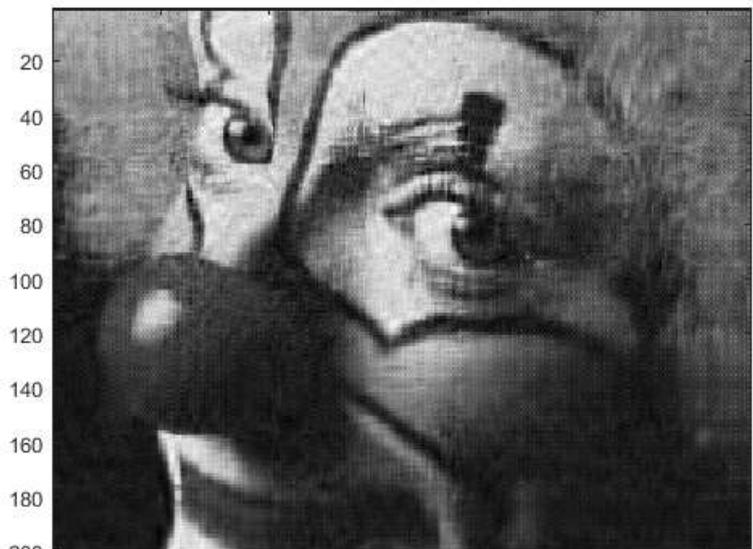
Nº modes used: 1



Nº modes used: 15



Nº modes used: 30



Aplicação na compactação de dados

➤ Armazenamento

Original = 200×320

= 64000

POD = $(200+320+1) * \text{nmodes}$

para $\text{nmodes}=30$

= 15630

Nesse exemplo simples, a reconstrução com POD requer menos que 25% do espaço de armazenamento da imagem original!!