Chapter 8

Experiments with a Wavelet-Based APOD

In this chapter we present another aspect of wavelet-based compression. We will use wavelet packet compression to reduce the complexity of the Proper Orthogonal Decomposition (POD) and obtain an Approximate POD (APOD). The performance of this APOD is evaluated by using it to represent and analyze the behavior of a dynamical system. We will show that only very few eigenmodes are needed to allow a good representation of the dynamical system, thanks to the decorrelating properties of a wavelet packet transform.

8.1 Introduction

In many domains of applied sciences it is important to analyze the behavior of large-scale dynamical systems, described by partial differential equations. These dynamical systems are processes in the real world, like e.g. chemical reactions. Because of the high-dimensionality of such a system, it is necessary to first reduce the complexity of the system, while retaining its intrinsic properties, to allow for an efficient analysis.

The Proper Orthogonal Decomposition (POD) or Karhunen-Loève Transform (KLT) is a powerful tool to obtain low-dimensional models for large-scale dynamical systems. Starting from a set of solutions (obtained by experiment or computation), called snapshots, the method computes an "optimal" basis of eigenmodes for the snapshots, which can be used to construct a low-dimensional model of the dynamical system. Unfortunately the construction of a POD and also the projection of the original problem onto the eigenmodes are very expensive operations.

We have done experiments with an Approximate Proper Orthogonal Decomposition, based on data compression using symmetric biorthogonal wavelets. A wavelet packet transform allows to decorrelate and compress the snapshots, after which we compute the POD of the compressed data set of much lower dimension, while we still obtain good approximations for the eigenmodes of the original system.

We will present results for the computation of an Approximate POD for a system of partial differential equations (the one-dimensional Brusselator reaction-diffusion model), using symmetric wavelets (Cohen-Daubechies-Feauveau) [53].

8.2 The Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD), also known as Karhunen-Loève Transform (KLT), is a tool to analyze a given set of data by decomposing the data into an "ideal" orthonormal basis.

With "ideal" basis we mean an orthonormal basis that gives the best (least squares) approximation using as few basis vectors as possible. The basis vectors are ordered such that the best approximation using only i vectors is obtained by using the first i vectors. Thus they are ordered by decreasing "importance".

Suppose we have S linear independent snapshots X_i of size N (i.e. linear independent vectors with N elements) with $S \ll N$. We assume that the average of the snapshots

$$\overline{X} = \frac{1}{S}\sum_{i=0}^{S-1} X_i$$

is zero. If this would not be the case, we can always subtract the average $\overline{X}$ from all snapshots:

$$X_i \leftarrow X_i - \overline{X}, \qquad i = 0,\dots,S-1.$$

One can show that the eigenvectors E_k of the $N \times N$ covariance matrix C

$$C = \frac{1}{S}\sum_{i=0}^{S-1} X_i X_i^T,$$

satisfying

$$CE_k = \lambda_k E_k, \qquad k = 0,\dots,S-1$$

with

$$\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{S-1} > 0$$

form the best basis we are looking for (the $N-S$ remaining eigenvalues of C are zero.). Since C is symmetric, the eigenvalues λ_k are real and the eigenvectors E_k are orthogonal. For every eigenvector E_k, the corresponding eigenvalue λ_k indicates the relative importance of that eigenvector in the best basis. The eigenvectors are also called the "eigenmodes" of the original snapshots.

Although this approach results in the "best basis" and is ideal in terms of decorrelation, it is not ideal in terms of computing time. Constructing the covariance matrix C requires $O(SN^2)$ operations.

Fortunately, one can use an alternative method, called the "method of the snapshots" [45]. Here one uses the property that the eigenvectors E_k are unique linear combinations of the linearly independent snapshots X_i and thus can be written as

$$E_k = \sum_{j=0}^{S-1} a_{k,j} X_j, \qquad k = 0, \ldots, S-1.$$

Now the eigenvalue equation becomes:

$$\begin{aligned} CE_k &= \lambda_k E_k, \\ \frac{1}{S}\sum_{i=0}^{S-1} X_i X_i^T \sum_{j=0}^{S-1} a_{k,j} X_j &= \lambda_k \sum_{j=0}^{S-1} a_{k,j} X_j, \\ \frac{1}{S}\sum_{i=0}^{S-1} X_i \left(\sum_{j=0}^{S-1} X_i^T X_j a_{k,j}\right) &= \sum_{i=0}^{S-1} \lambda_k a_{k,i} X_i. \end{aligned}$$

Because the X_i form a basis, we can drop the summations on both sides of the last equation and we get

$$\sum_{j=0}^{S-1} X_i^T X_j a_{k,j} = \lambda_k^* a_{k,i}, \qquad \left\{ \begin{array}{rcl} i &=& 0, \ldots, S-1, \\ k &=& 0, \ldots, S-1, \\ \lambda_k^* &=& S\lambda_k. \end{array} \right.$$

Define the matrix C^* as

$$C^* = \left[c_{i,j}^*\right] = \left[X_i^T X_j\right], \qquad i, j = 0, \ldots, S-1,$$

and let the vector $A_k = [a_{k,i}]\,, i = 0, \ldots, S-1$, denote the coefficients of the eigenvector E_k in the basis of the snapshots, then the original eigenvalue problem is equivalent to the eigenvalue problem

$$C^* A_k = \lambda_k^* A_k, \qquad \text{with } \lambda_k^* = S\lambda_k.$$

The construction of the matrix C^* requires $O(NS^2)$ operations, which is much cheaper than the construction of the covariance matrix C, since $C \ll N$.

If $S > N$, the snapshots X_i are not independent and thus they no longer form a basis. As a consequence, the matrix C^* will have rank $r \leq N$. However, the method of the snapshots can still be used, because we have for $k = r, \ldots, S-1$:

- the additional eigenvalues λ_k will all be zero because of the rank deficiency of C^*,

- the additional eigenvectors A_k will be unit vectors (up to a constant) because the mean of the snapshots is zero, and the resulting additional eigenmodes E_k will be zero. Note that in a strict sense the zero vector cannot be an eigenmode, but here we are considering it in a more general framework.

Note:

We can "normalize" the snapshots

$$X_i \leftarrow \frac{X_i}{||X_i||\sqrt{S}} \tag{8.1}$$

such that

$$\sum_{k=0}^{S-1} \lambda_k^* = 1.$$

Indeed, the sum of the eigenvalues of a matrix is equal to the sum of the diagonal elements of that matrix. In that case eigenmode E_k contains $100 \times \lambda_k^*\%$ of the total energy.

8.3 The Approximate Proper Orthogonal Decomposition

The main idea behind the Approximate Proper Orthogonal Decomposition (APOD) is to decorrelate the original snapshots using wavelets and perform the normal Proper Orthogonal Decomposition (POD) on the compressed snapshots, in the hope that we will get a reasonable approximation for the original eigenmodes, with less computational effort.

8.3.1 Theory of Operation

To calculate the APOD, we perform the following operations:

1. Transform the snapshots X_i using an appropriate transform W (we will use a wavelet packet transform, as will be mentioned in Section 8.3.2) that decorrelates the original data:

$$\widehat{X}_i = W(X_i), \qquad i = 0, \ldots, S-1.$$

2. Compress the vectors $\widehat{X}_i$ by removing from every vector a number of entries:

$$\widehat{X}_i \triangleright \widehat{X}_i' \in \mathbf{R}^n, \qquad i = 0, \ldots, S-1,$$

with $n \ll N$.

Removing $m\%$ entries gives a compression rate of $\frac{100}{100-m}$: 1. The strategy to decide which entries to remove can be adapted to the set of snapshots, i.e. based on statistical properties of the set, or can be fixed, i.e. based on a previous statistical analysis.

3. Calculate the POD of the compressed snapshots $\widehat{X}_i'$. This gives the eigenmodes $\widehat{E}_i' \in \mathbf{R}^n$, while the eigenvalues λ_i indicate the energy distribution over the eigenmodes.

4. Decompress the eigenmodes $\widehat{E}_i'$ by putting zeros at the places where a vector entry was removed in step 2:
$$\widehat{E}_i' \triangleleft \widehat{E}_i, \qquad i = 0, \ldots, S-1.$$

5. Perform the inverse transform W^{-1} on the eigenmodes $\widehat{E}_i$ to get the approximate eigenmodes E_i^*, which are an approximation to the eigenmodes E_i of the snapshots X_i:
$$E_i^* = W^{-1}(\widehat{E}_i) \approx E_i, \qquad i = 0, \ldots, S-1.$$
If the transform W is not orthogonal but biorthogonal, we not only have eigenmodes E_i^*, but also dual eigenmodes $\widetilde{E}_i^*$.

A schematic overview of these operations can be found in Figure 8.1.

The APOD can be used instead of the genuine POD, for two possible goals:

- to reduce the computation time needed to construct (an approximation of) the eigenmodes E_i,
- to use the APOD to construct a "low-dimensional" model, based on the eigenmodes $\widehat{E}_i'$.

8.3.2 Decorrelating Transforms

We have chosen to use a wavelet packet transform as the decorrelating transform (see Section 2.6). Our motivation is that a wavelet packet transform is closer to a POD than an ordinary wavelet transform due to the additional degrees of freedom in the choice of the basis functions.

If we transform a square image $X = [x_{ij}]$ with $n \times n$ elements, the two-dimensional wavelet packet decomposition tree $Y = [y_{ijk}]$ has $n \times n \times \log n$ elements, with a topology of $\log n$ "levels" of $n \times n$ elements. To compress the representation in the best basis, we remove the $m\%$ coefficients that are the smallest in magnitude.

Since we want to find the best basis not for a single snapshot, but for a whole *set* of snapshots, we must choose the *same* wavelet packet basis for every snapshot. This basis must be based on the statistical properties of the set of snapshots.

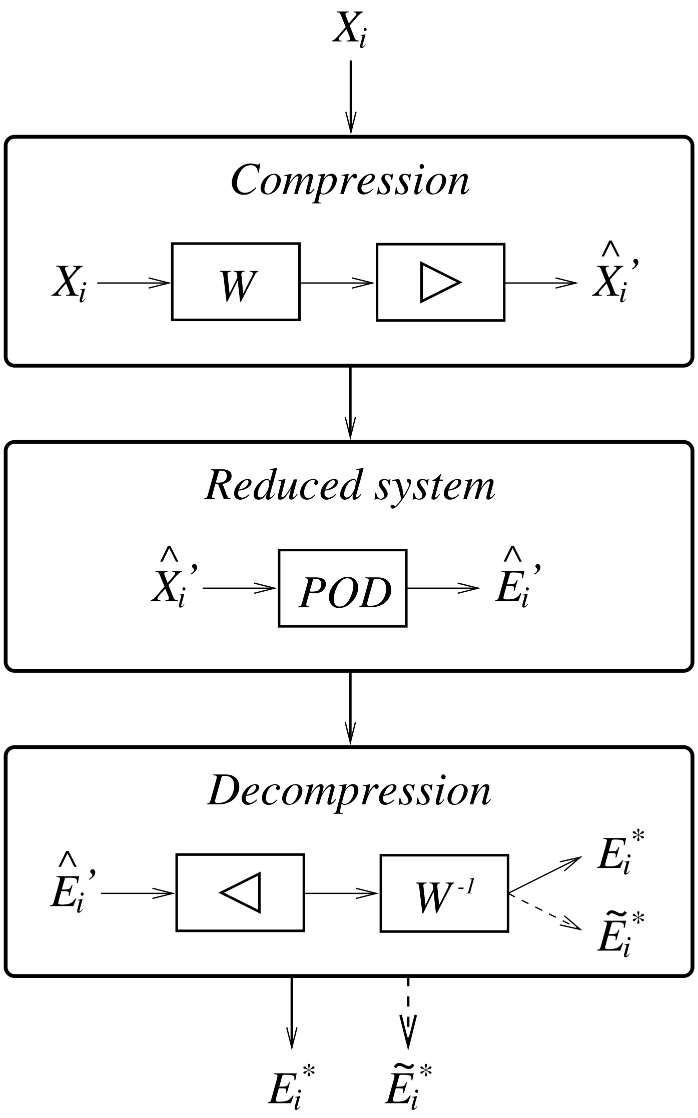


Figure 8.1: Schematic overview of the Approximate Proper Orthogonal Decomposition. The dual eigenmodes $\widetilde{E}_i^*$ are only relevant if W (and W^{-1}) are biorthogonal transforms.

To select this basis, we calculate the wavelet packet decomposition tree Y_l of every snapshot X_l

$$X_l = [x_{i,l}] \xrightarrow{\text{WPT}} Y_l = [y_{ij,l}],$$

square all entries in the Y_l and sum the corresponding entries of Y_l to obtain a "sum-squared" wavelet packet decomposition tree Y^*:

$$Y^* = [y^*_{ij}] \qquad \text{with} \quad y^*_{ij} = \sum_l y^2_{ij,l}.$$

Then we apply the best basis selection algorithm to the "sum-squared" wavelet packet decomposition tree Y^*.

To compress the snapshots in their best basis representation, we remove the coefficients from the Y_l that correspond to the $m\%$ smallest coefficients (in absolute value) in the tree Y^*.

Depending on the wavelet filter, the wavelet packet transform can be an orthogonal or a biorthogonal transform.

Orthogonal Transforms

If the transform W and its inverse W^{-1} are orthogonal transforms, the APOD will yield the same results as the POD if there is no compression. Indeed, because the POD is an *orthogonal* decomposition, applying an orthogonal transform before and its inverse after the POD has no influence on the result.

In the case of compression, we have the following: if we decompose a function in an orthonormal basis $\varphi_i(x)$,

$$f(x) = \sum_k a_k \varphi_k(x),$$

Parseval's identity holds:

$$\sum_k a_k^2 = \|f\|^2.$$

Neglecting the smallest coefficients a_k will result in only a small error after reconstruction. In addition the orthogonality guarantees a good approximation.

Of course the results also depend on the criterion used in step 2.

Biorthogonal Transforms

If the transform W and its inverse W^{-1} are not orthogonal but only biorthogonal transforms, we cannot guarantee a good approximation of the real POD in all circumstances. In fact, if we do not compress the transformed snapshots, we will not even get the same eigenmodes as with the real POD.

However, if the transform is based on biorthogonal wavelets, the fact that the resulting basis is a Riesz basis will guarantee no longer a perfect but still a good approximation. In a Riesz basis (Equation 2.5), the coefficients of the decomposition show the following property:

$$A\|f\|^2 \leq \sum_k a_k^2 \leq B\|f\|^2,$$

with A and B positive constants. Neglecting the smallest coefficients a_k will still result in only a small controllable error after reconstruction.

Note that in the biorthogonal case we not only have primal eigenmodes E_i^*, but also dual eigenmodes $\widetilde{E}_i^*$, which are biorthogonal with respect to the primal eigenmodes, as shown below.

8.3.3 Decomposition into the Basis of Approximate Eigenmodes

If we want to write an arbitrary snapshot as a linear combination of the normalized approximate eigenmodes

$$X \approx \sum_i a_i E_i^*,$$

the unknown coefficients a_i can be obtained in either the original or the wavelet domain.

In the original domain, we can calculate inner products with the eigenmodes, or with the dual eigenmodes in the biorthogonal case:

- In case an orthonormal transform is used:

$$a_i = \langle X, E_i^* \rangle.$$

- In case a biorthonormal transform is used:

$$a_i = \langle X, \widetilde{E}_i^* \rangle.$$

However, we can also obtain the coefficients a_i by calculating them in the wavelet domain. Let $\widehat{X}'$ be the result of the wavelet transform and compression of the input image X:

$$\begin{aligned} \widehat{X} &= W(X), \\ \widehat{X} &\triangleright \widehat{X}'. \end{aligned}$$

Due to the (bi)orthonormality of the wavelet packet transform and the orthogonality of the eigenmodes $\widehat{E}_i'$ of the transformed and compressed snapshots, the unknown coefficients a_i are given by

$$a_i = \langle \widehat{X}', \widehat{E}_i' \rangle.$$

Indeed, we have in the biorthonormal case:

$$\begin{aligned} a_i &= \langle X, \widetilde{E}_i^* \rangle, \\ &= \langle \sum_j b_j \Psi_j, \sum_k c_{ik} \widetilde{\Psi}_k \rangle, \\ &= \sum_{j,k} b_j c_{ik} \langle \Psi_j, \widetilde{\Psi}_k \rangle, \\ &= \sum_j b_j c_{ij}, \\ &= \langle B, C_i \rangle, \\ &= \langle \widehat{X}', \widehat{E}_i' \rangle, \end{aligned}$$

with Ψ_i and $\widetilde{\Psi}_i$ the biorthonormal basis functions in the wavelet domain, and $B = [b_i]$ and $C = [c_{i,j}]$.

Reconstruction using the wavelet domain is straightforward: first we reconstruct $\widehat{X}'$ in the wavelet domain, followed by a decompression step and the inverse transform W^{-1}:

$$\begin{aligned} \widehat{X}' &= \sum_i a_i \widehat{E}_i', \\ \widehat{X}' &\triangleleft \widehat{X}, \\ X &\approx W^{-1}(\widehat{X}). \end{aligned}$$

Which approach for the calculation of the coefficients a_i is most efficient depends on a number of factors. If we denote the number of retained eigenmodes by s and we assume that the images are square and count $\sqrt{N} \times \sqrt{N}$ pixels, we have:

- in the original domain:
  - s inner products in $\mathbf{R}^N$: $O(sN)$,
- in the wavelet domain:
  - S wavelet packet transforms: $O(SN \log \sqrt{N})$,
  - s inner products in $\mathbf{R}^n$: $O(sn)$.

8.4 Wavelet Transforms

Let us denote the primal filter pair that is characteristic for a specific wavelet transform by (g, h) and the dual filter pair by $(\widetilde{g}, \widetilde{h})$.

We recall that while the periodic extension of a discrete (sampled) signal is straightforward and unique, there are four possible ways to extend a discrete signal symmetrically (Figure 3.7). However, to ensure to get a perfect reconstruction, one has to apply the correct extension in every step of the decomposition and reconstruction, as indicated in Figure 3.8 [4], cfr. Section 3.5.1.

All wavelet transform operations on a vector can be denoted by matrix multiplications. We use the following notations:

W Transformation matrix for the forward wavelet transform algorithm, using the filter pair (g,h),

$\mathbf{W^{-1}}$ Transformation matrix for the inverse wavelet transform algorithm, using the filter pair (g,h),

$\mathbf{W^{T}}$ Transposition (adjoint) of the transformation matrix for the forward wavelet transform algorithm, using the filter pair (g,h),

$\mathbf{W^{-T}}$ Transposition (adjoint) of the transformation matrix for the inverse wavelet transform algorithm, using the filter pair (g,h).

The transform matrices for the corresponding dual transforms using the dual filter pair $(\tilde{g},\tilde{h})$ are denoted by adding a tilde, e.g. $\widetilde{W}$.

The properties of these transformation matrices depend on the type of wavelet transform (orthogonal vs. biorthogonal) and the type of signal extension (periodic vs. symmetric). We can distinguish the following cases:

Case 1

This is the generic case, i.e. for biorthogonal wavelets in combination with symmetric extension:

Forward/Inverse FWT	**Orthogonality**
$W^{-1}\cdot\widetilde{W}=I$	$W^{-T}\perp\widetilde{W}$
$\widetilde{W}\cdot W^{-1}=I$	$\widetilde{W}^{T}\perp W^{-1}$
$\widetilde{W}^{-1}\cdot W=I$	$\widetilde{W}^{-T}\perp W$
$W\cdot\widetilde{W}^{-1}=I$	$W^{T}\perp\widetilde{W}^{-1}$

Case 2

If we use biorthogonal wavelets in combination with periodic extension, we can make the following simplifications:

$$\begin{aligned}
W^{T} &\equiv W^{-1},\\
W^{-T} &\equiv W,\\
\widetilde{W}^{T} &\equiv \widetilde{W}^{-1},\\
\widetilde{W}^{-T} &\equiv \widetilde{W},
\end{aligned}$$

and we get:

Forward/Inverse FWT	Orthogonality
$W^{-1} \cdot \widetilde{W} = I$ $\widetilde{W} \cdot W^{-1} = I$ $\widetilde{W}^{-1} \cdot W = I$ $W \cdot \widetilde{W}^{-1} = I$	$W \perp \widetilde{W}$ $\widetilde{W}^{-1} \perp W^{-1}$

Case 3

If we use orthogonal wavelets, we can simplify even further:

$$\begin{aligned} \widetilde{W} &\equiv W, \\ \widetilde{W}^{-1} &\equiv W^{-1}, \end{aligned}$$

and we get:

Forward/Inverse FWT	Orthogonality
$W^{-1} \cdot W = I$ $W \cdot W^{-1} = I$	$W \perp W$ $W^{-1} \perp W^{-1}$

In most wavelet transform implementations, one has to care only about the *forward/inverse FWT* properties. However, for the APOD we also have to care about the *orthogonality* properties. Indeed, if we use a biorthogonal wavelet transform or wavelet packet transform, we will end up with *primal* and *dual* eigenmodes (E_i^* and $\widetilde{E}_i^*$) that need to be biorthogonal to be able to decompose a random snapshot into a precalculated basis of eigenmodes.

8.5 Results

We have evaluated the use of the Approximate Proper Orthogonal Decomposition to represent and to analyze the dynamical behavior of the discretization of a system of partial differential equations (PDEs), namely a reaction-diffusion system.

8.5.1 Wavelets

Because of their symmetry, we have chosen for biorthogonal wavelets. In our test cases, we used biorthogonal Cohen-Daubechies-Feauveau wavelets [9], in particular the member with 2 vanishing moments for both the primal and the dual wavelet (Figure 2.4), with symmetric signal extension.

The cost criterion used for choosing the “best basis” among all possible wavelet packet bases is based on the logarithm of the energy

$$\text{cost} = \sum_i \log x_i^2,$$

which is equivalent with

$$\text{cost} = \sum_i \log |x_i|.$$

8.5.2 The Dynamical System

The dynamical system we want to analyze is described by a system of partial differential equations. We evaluated the performance of the APOD using the one-dimensional Brusselator reaction-diffusion system [22]:

$$\begin{aligned} \frac{\partial X}{\partial t} &= \frac{D_X}{L^2}\frac{\partial^2 X}{\partial z^2} + X^2 Y - (B+1)X + A, \\ \frac{\partial Y}{\partial t} &= \frac{D_Y}{L^2}\frac{\partial^2 Y}{\partial z^2} - X^2 Y + BX, \end{aligned}$$

with Dirichlet boundary conditions

$$\begin{aligned} X(t,z=0) &= X(t,z=1) = A, \\ Y(t,z=0) &= Y(t,z=1) = B/A. \end{aligned}$$

The reactor length L is used as a free parameter (continuation parameter), while we keep the other parameters fixed:

$$\begin{aligned} A &= 2.0, \\ B &= 5.45, \\ D_X &= 0.008, \\ D_Y &= 0.004. \end{aligned}$$

Branches of periodic solutions bifurcate from the trivial steady state branch ($X = A, Y = B/A$), as can be seen in the bifurcation diagram in Figure 8.2 [32].

8.5.3 Snapshots

To create a low-dimensional model of our system, we started with four sets of snapshots, calculated for four values of the continuation parameter L, using time integration with a time step $\Delta t = 0.2$, as shown in Table 8.1.

Each snapshot is a vector with 256 elements, containing the concentration X of the first reagent. Figure 8.3 gives a graphical representation of the concentration X for the different parameter values. White indicates a high concentration, black a low concentration. The z (spatial) axis runs horizontally, while the t (time) axis runs vertically (time runs from the top to the bottom).

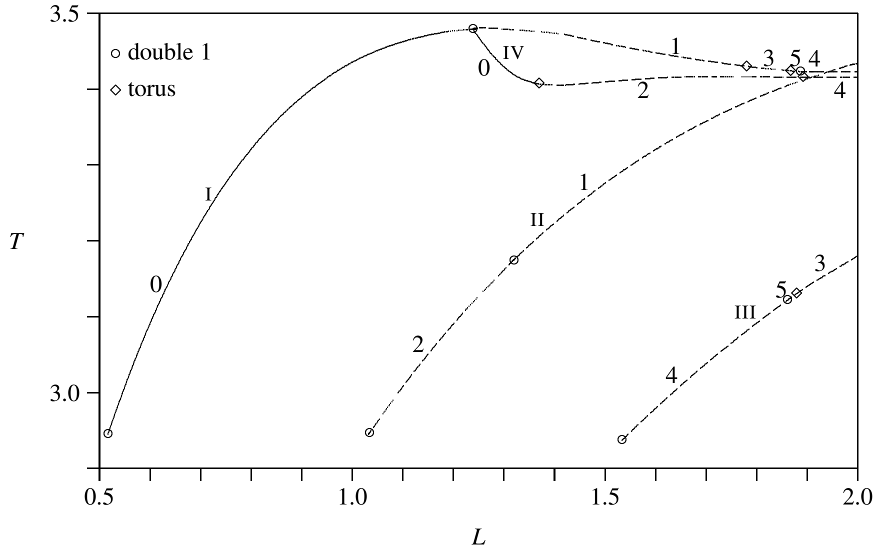


Figure 8.2: Bifurcation diagram of periodic (branches I–IV) and quasi-periodic (branch V) solutions for the Brusselator model, period T versus the reactor length L [32]. Note that this diagram is calculated using a 31-point discretization and that it is not complete.

Set	L	Branch	# snapshots	
A	0.75	I	20	Periodic solution
B	1.25	I	20	Periodic solution (before period-doubling)
C	1.43	V	192	Quasi-periodic solution (torus)
D	1.50	IV	20	Periodic solution
Total			**252**	

Table 8.1: The snapshots used for testing and their bifurcation parameter values.

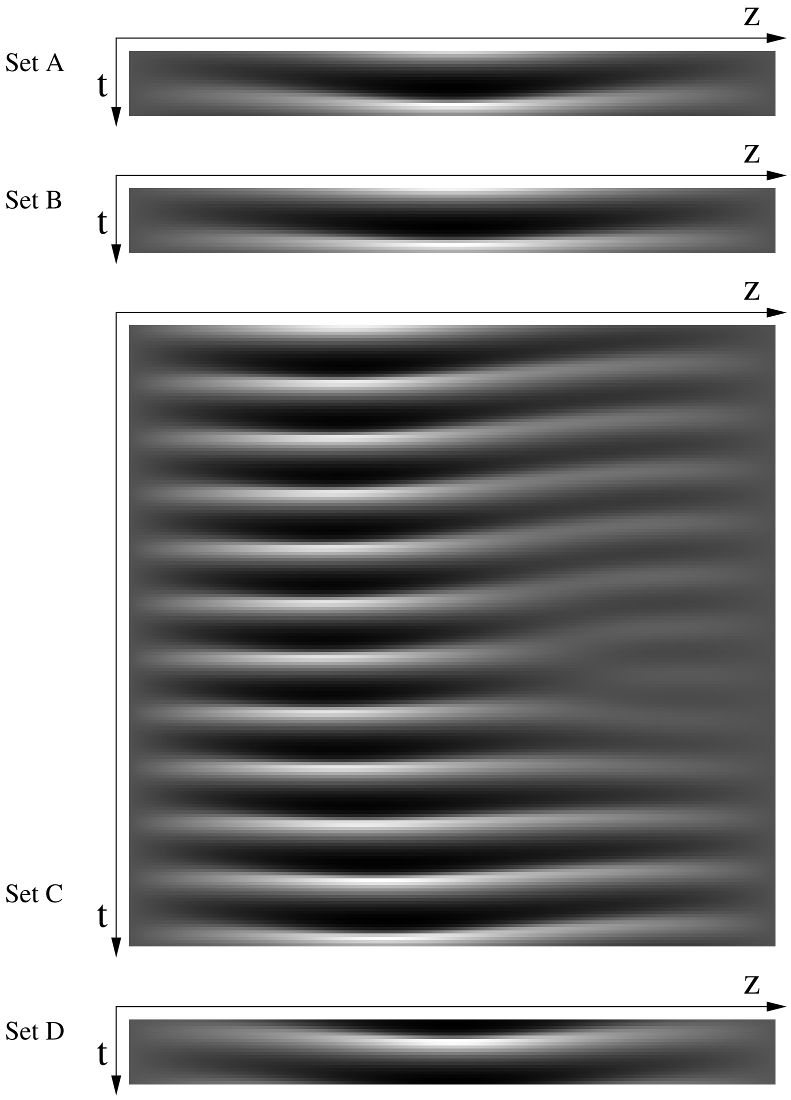


Figure 8.3: The four sets of input snapshots calculated for four values of the continuation parameter L (cfr. Table 8.1), using time integration. Each snapshot is a vector with 256 elements, containing the concentration X of the first reagent. White indicates a high concentration, black a low concentration.

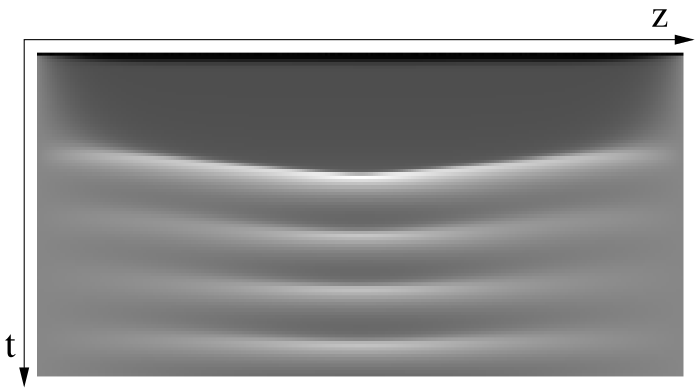


Figure 8.4: The set of 100 “strange” input snapshots, calculated for $L = 1.40$ using time integration. Each snapshot is a vector with 256 elements, containing the concentration X of the first reagent. White indicates a high concentration, black a low concentration.

8.5.4 “Strange” Snapshots

To test how well this low-dimensional model can represent the dynamical behavior of the parameter dependent Brusselator model, we decomposed a set of “strange” snapshots — snapshots not belonging to the set of snapshots we used to construct the APOD model — in the APOD basis. The set consists of 100 snapshots for $L = 1.40$, obtained by time integration with a time step $\Delta t = 0.2$ starting from the perturbed trivial solution. The “strange” snapshots show a periodic solution including the transient behavior.

8.5.5 Eigenvalue Distribution

Since the snapshots were normalized by subtracting their mean and by rescaling them (see 8.1), the eigenvalues are equal to the fraction of captured energy.

To approximate the snapshots by using only a few eigenmodes, we need to keep the eigenmodes which capture the most energy and hence have the largest eigenvalues.

In Figure 8.5 the sum of the largest i eigenvalues is plotted for different compression rates. The plot reveals that e.g. the 4 most important eigenmodes capture more than 99% of the energy present in the set of snapshots.

Observe that for a fixed number of eigenmodes the captured energy is increasing in function of the compression ratio. Due to the lossy compression some small non-correlated information is lost, and the resulting data is thus more correlated, resulting in the effect that the same number of eigenvalues capture more energy.

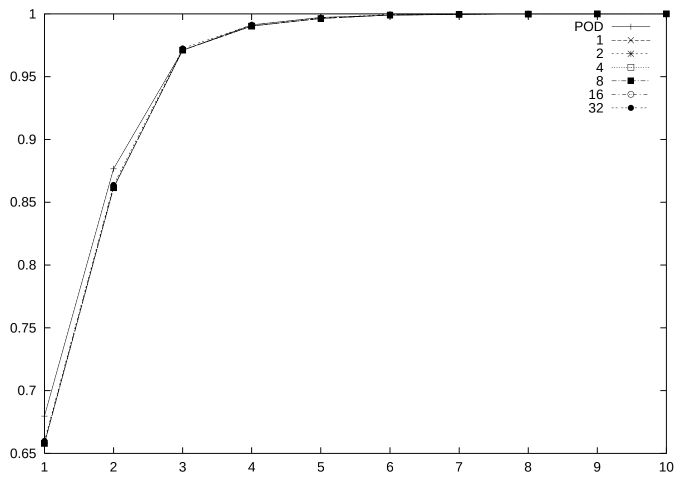


Figure 8.5: Sum of the eigenvalues vs. number of eigenvalues, both for the POD and the APOD with different compression rates.

Note that in the case of compression rate 32, we retain only 8 eigenmodes because the dimension of the compressed system can not be higher due to the large compression ratio $(256/32 = 8)$.

The distribution of the eigenvalues is a measure for the decorrelation achieved by the (approximate) POD. If many eigenvalues are very small, the original set of snapshots was highly correlated and can be represented well by only a few eigenmodes. In contrast, if all eigenvalues have about the same magnitude, the set of snapshots was not correlated at all.

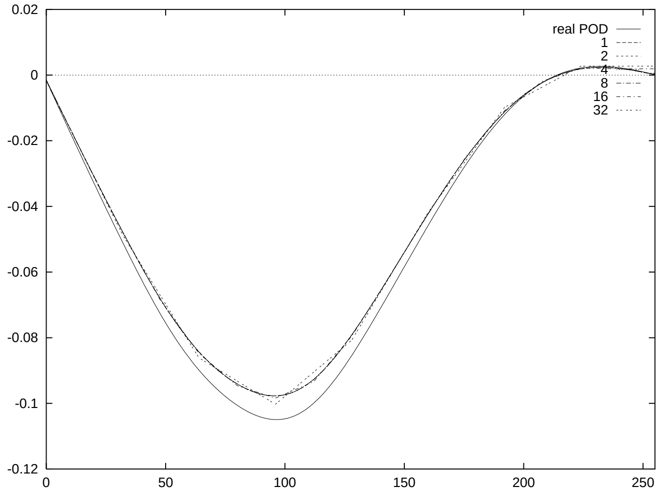
8.5.6 Eigenmodes

The 4 most important eigenmodes, both for the POD and the Approximate POD with different compression rates, are shown in Figures 8.6 and 8.7.

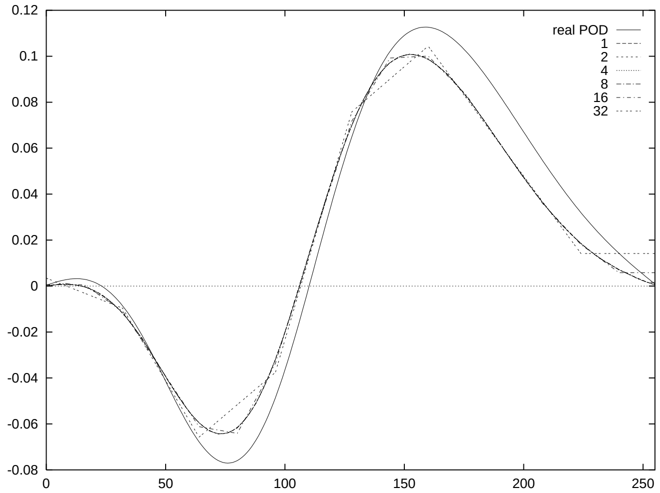
The difference between the results for the POD and the Approximate POD using compression rate 1 (i.e. no compression) is due to the biorthogonality of the wavelet packet basis. If we would have used an orthogonal wavelet packet basis, there would be no difference.

8.5.7 Reconstruction of the Snapshots

If we reconstruct the snapshots (both the set used for the construction of the APOD model and the "strange" set) using only the most important eigenmodes, we get

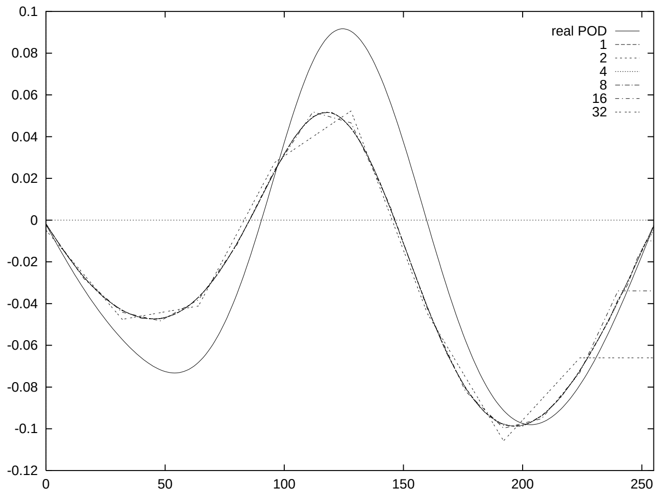


Eigenmode 1

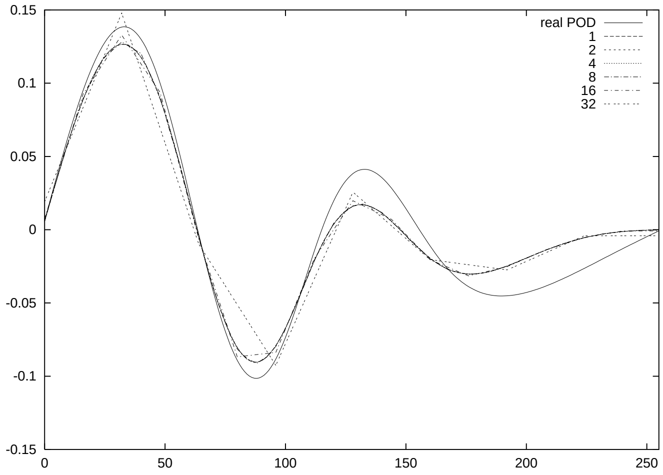


Eigenmode 2

Figure 8.6: The 4 most important eigenmodes for the POD and for the APOD with different compression rates.



Eigenmode 3



Eigenmode 4

Figure 8.7: The 4 most important eigenmodes for the POD and for the APOD with different compression rates. (cont'd)

Number of eigenmodes	POD	APOD with compression rate					
		1	2	4	8	16	32
1	67.97	68.11	68.11	68.11	68.11	68.10	67.97
2	87.66	87.69	87.69	87.69	87.69	87.64	87.04
3	97.10	97.17	97.17	97.17	97.17	97.17	96.83
4	99.13	99.10	99.10	99.10	99.09	99.07	98.50
5	99.72	99.68	99.68	99.68	99.68	99.65	99.04
6	99.89	99.90	99.90	99.90	99.89	99.86	99.23
7	99.95	99.94	99.94	99.94	99.93	99.90	99.24
8	99.99	99.98	99.98	99.98	99.97	99.93	99.25
9	99.99	99.99	99.99	99.99	99.98	99.94	
10	100.00	99.99	99.99	99.99	99.99	99.95	

Table 8.2: Mean percentage of energy after reconstruction of the snapshots vs. number of used eigenmodes for different compression rates.

the following results. Tables 8.2 and 8.3 show the mean percentage of captured energy (over all reconstructed snapshots) after reconstruction by using the most important eigenmodes for different compression ratios.

8.5.8 Coefficients of the Snapshots in the APOD Basis

We can analyze the behavior of the dynamical system by analyzing the coefficients of the projection of the snapshots onto the APOD basis. Figure 8.8 shows these coefficients for the two most important eigenmodes. The first 20 snapshots are from set A, the next 20 from set B and so on (cfr. Table 8.1). The periodicity of the solutions is clearly visible. There is no visible difference among the coefficients for the APOD bases with different compression rates.

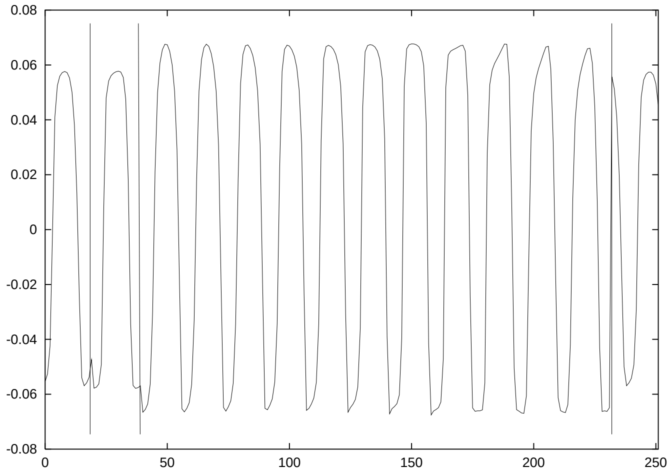
Figure 8.9 shows how the maximum error in the coefficients of the 10 most important eigenmodes depends on the compression rate.

The same results for the decomposition of the "strange" snapshots in the APOD basis are shown in the Figures 8.10 and 8.11.

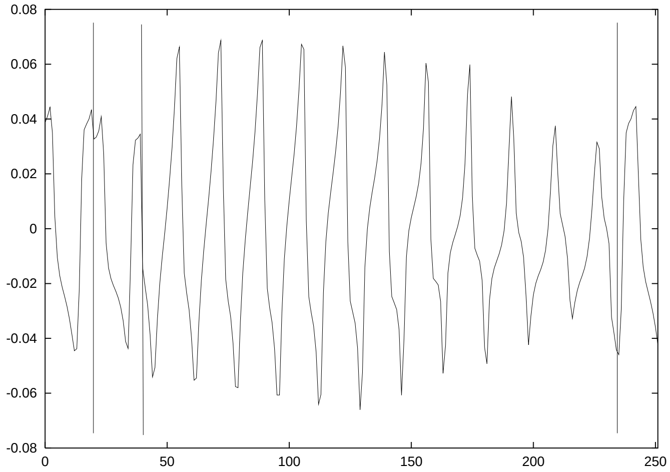
8.5.9 Timings

All timings were obtained on one node of an IBM SP2 cluster of RS/6000 POWER2 thin nodes, with an implementation in C++ using LAPACK++.

The normalization of the snapshots took 0.32 seconds. The decomposition timings for the APOD are shown in Table 8.4. As a comparison, the real POD required 17.40 seconds. Note that the wavelet packet transform routines used — taking ca. 25 seconds — were very generic and were not optimized at all. A



Coefficient 1



Coefficient 2

Figure 8.8: First two coefficients of the decomposition of the snapshots into the best basis.

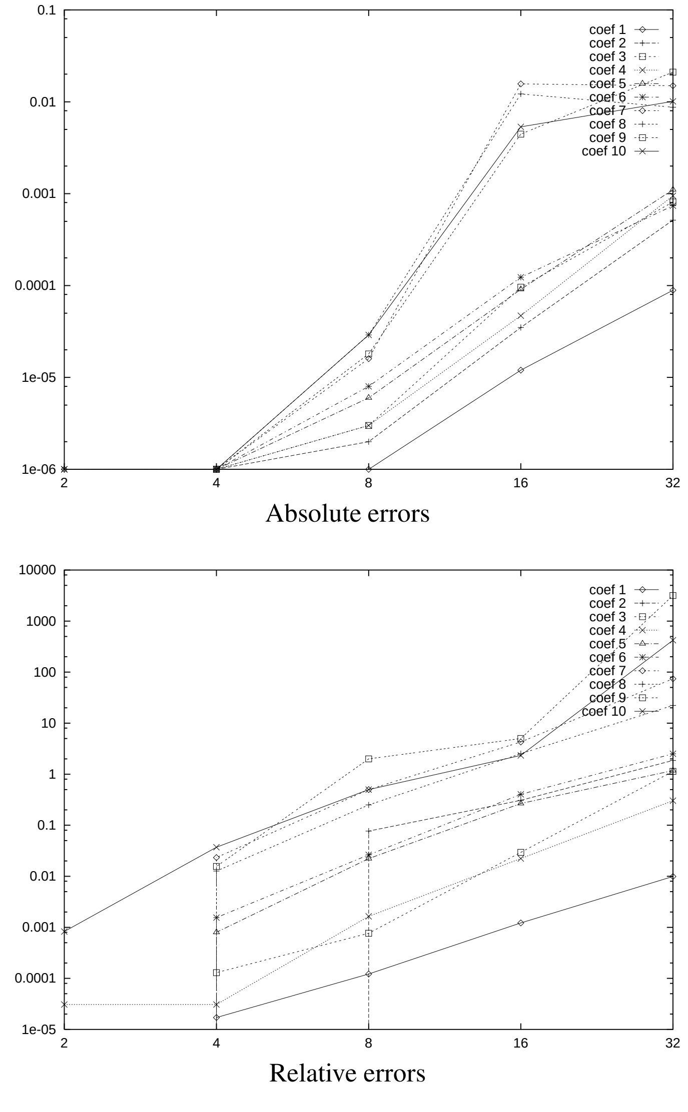
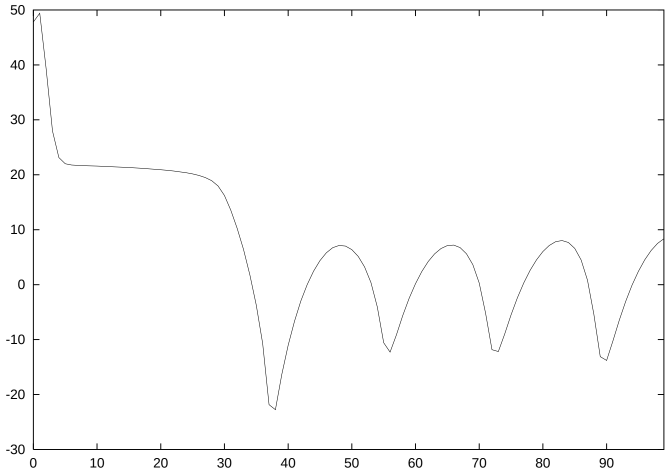
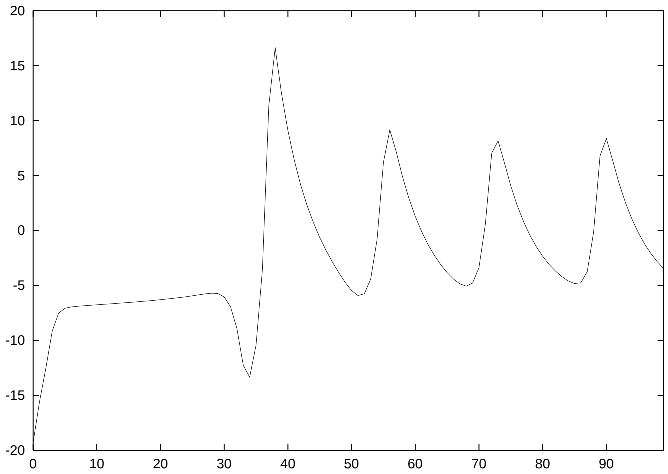


Figure 8.9: Maximum error in the coefficients (compared to the case without compression) for different compression rates.



Coefficient 1



Coefficient 2

Figure 8.10: First two coefficients of the decomposition of the "strange" snapshots into the best basis.

Number of eigenmodes	APOD with compression rate					
	1	2	4	8	16	32
1	52.87	52.87	52.87	52.87	52.87	52.81
2	71.62	71.62	71.62	71.62	71.60	71.34
3	91.75	91.75	91.75	91.74	91.79	91.90
4	96.87	96.87	96.87	96.87	96.90	96.84
5	97.40	97.40	97.40	97.39	97.43	97.52
6	98.22	98.22	98.22	98.21	98.23	98.32
7	99.12	99.12	99.12	99.10	99.16	99.90
8	99.09	99.09	99.09	99.07	99.11	99.88
9	99.32	99.32	99.32	99.30	99.37	
10	99.43	99.43	99.43	99.40	99.47	

Table 8.3: Mean percentage of energy after reconstruction of the "strange" snapshots vs. number of used eigenmodes for different compression rates.

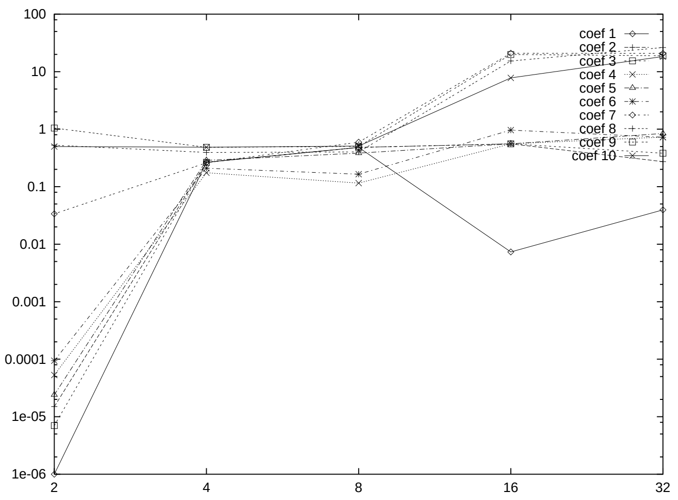


Figure 8.11: Maximum absolute error in the coefficients (compared to the case without compression) for different compression rates.

Compression rate	Calculation time	
1	45.30	$\approx$ 25+17.4
2	37.30	$\approx$ 25+9
4	31.15	
8	27.45	
16	25.62	
32	25.40	$\approx$ 25+0.5

Table 8.4: Computation times (in seconds) for the Approximate Proper Orthogonal Decomposition for the various compression ratios. The real POD required ca. 17.40 seconds. The wavelet packet transform took ca. 25 seconds. The third column shows that the total cost of the APOD is approximately equal to the sum of the cost of the wavelet packet transform (ca. 25 seconds) and the cost of the construction of the matrix C^* for the POD of the reduced system (proportional to the reciprocal of the compression ratio).

Method	Complexity
POD	$O(NS^2)+O(S^3)$
WPT	$O(SN\log N)$
APOD	$O(SN\log N)+O(S^2N/C)+O(S^3)$

Table 8.5: Comparison of the complexity of the various methods. S is the number of snapshots, N is the size of a snapshot, and C is the compression rate.

cautious implementation using the lifting scheme algorithm will speed it up an order of magnitude. But at the time this research was done, the lifting scheme had just emerged and was not mature yet.

A comparison of the complexity of the various methods is shown in Table 8.5. However, the factor $\log N$ is only valid for the one-dimensional case we considered here. For higher dimensions it becomes

$$\log \sqrt[d]{N},$$

with d the dimension of the snapshots and thus $\sqrt[d]{N}$ the number of data points per dimension. This factor decreases quickly for increasing dimensions, so for higher dimensions we eventually win using the APOD.

The selected wavelet packet basis can also be fixed in advance, instead of being determined adaptively by a best basis search. In that case the factor $\log N$ must be replaced by a number between 1 and $\log N$, depending on the chosen basis. E.g. for an ordinary wavelet transform, that factor is 1.

8.6 Conclusions

The Approximate Proper Orthogonal Decomposition using biorthogonal wavelet packets allows a good reconstruction of the original data, as is shown by the amount of captured energy. We only need very few eigenmodes (basis functions), while methods based on other approximations like the Discrete Cosine Transform (DCT) and the Fast Wavelet Transform (FWT) require much more basis functions. One can analyze the behavior of a dynamical system by interpreting the most important coefficients of the decomposition in the APOD basis.

The decomposition can be calculated in both the original domain and in the wavelet packet domain. The eigenmodes are not localized, but the underlying wavelet packets are, and thus the calculation of inner products in the wavelet packet domain is a cheaper operation.

This is useful for the construction of low-dimensional approximate models for PDEs (reaction-diffusion, Navier-Stokes, ...). When assuming the solution of a PDE can be written as

$$u(x,t) = \sum_{i=0}^{K-1} a_i(t) E_i^*(x),$$

the PDE can be transformed into the form

$$\begin{aligned} \frac{da_i(t)}{dt} &= f(a_0, \cdots, a_{K-1}) \\ &= c_i + c_{ij} a_j + c_{ijk} a_j a_k + \cdots \end{aligned}$$

by means of a Galerkin projection. The Galerkin projection requires $O(K^3)$ inner products of vectors of size N/C, which can be sped up by using the properties of wavelet packets (localization, biorthogonality).

Another application is the compressed storage of related data. In [45] Sirovich et al. suggest to use the POD to store a library of pictures of human faces by storing the most important POD eigenmodes and, for each face (snapshot), the coefficients in this basis. The APOD allows a further storage reduction by storing the compressed eigenmodes.