Case study on SVD multiresolution analysis

Masaaki Yoshikawa* Yan Gong†
Ryuichi Ashino*‡ Rémi Vaillancourt†

CRM-3179

January 2005

*Division of Mathematical Sciences, Osaka Kyoiky University, Kashiwara, Osaka 582-8582, Japan

†Department of Mathematics and Statistics, University of Ottawa, 585 King Edward Ave., Ottawa ON, Canada K1N 6N5, remi@uottawa.ca

‡ashino@cc.osaka-kyoiku.ac.jp

Abstract

Basic results on the singular value decomposition of a matrix are recalled. The Kakarala and Ogunbona form of a multiresolution analysis for singular value decomposition is presented. Numerical case studies on a multiresolution analysis method based on the singular value decomposition include the 2001 Nikkei Stock Exchange, the 1940 El Centro earthquake wave, the 1995 Kobe earthquake wave and the MATLAB `leleccum` sample data. The MATLAB code used in this paper is listed in the appendix.

Keywords. SVD, singular value decomposition, MRA, multiresolution analysis

To appear in Scientific Proceedings of Riga Technical University, Series 5, Computer Science

Résumé

On rappelle les éléments de la décomposition d'une matrice selon les valeurs singulières et l'analyse multi-résolution de Kakarala et Ogunbona de cette décomposition. On applique cette méthode à l'étude numérique du Nikkei Stock Exchange pour 2001, des tremblements de terre d'El Centro en 1940 et de Kobe en 1995 et du signal `leleccum` de MATLAB. Les codes MATLAB sont en appendice.

1 INTRODUCTION

The singular value decomposition (SVD) of a real $M \times N$ matrix A is

$$A = U\Sigma V^T,$$

where U is an orthogonal $M \times M$ matrix whose columns (called the left singular vectors) are the eigenvectors of AA^T, V is an orthogonal $N \times N$ matrix whose columns (called the right singular vectors) are the eigenvectors of A^TA, and Σ is an $M \times N$ diagonal matrix whose diagonal entries are the singular values of A. More on the singular decomposition theorem can be found in [1], [2], and [3]. The history of the singular value decomposition is reviewed in [4].

R. Kakarala and P. Ogunbona [5] proposed a multiresolution analysis based on singular value decomposition. They let $\hat{A} = U^TA = \Sigma V^T$, and write the SVD as $A = U\hat{A}$. This second form reveals a useful connection with recent research in signal-adapted filterbanks: U is essentially the decorrelating matrix obtained from an input signal measured in second-order statistics, and $\hat{A}$ contains the subband decomposition of the signal.

The main purpose of this paper is to show, by numerical experiences, how SVD multiresolution analysis works, that is, how, by recursively resampling and decomposing the largest rank-one matrix, the SVD may be developed into a multiresolution SVD, thereby providing useful information for analyzing and comparing signals.

This paper is organized as follows. Notation is introduced in section 2. Basic properties of SVD are recalled in section 3. The multiresolution form of the SVD is presented in section 4. Section 5 contain the numerical results obtained from four case studies. The MATLAB codes used in section 5 are listed in appendix A.

2 NOTATION AND CONVENTIONS

Throughout this paper, the following notation and conventions are used. An N dimensional row vector is indexed as $[x(1) \cdots x(N)]$ and similarly an $M \times N$ matrix X is indexed as

$$X = \begin{bmatrix} x(1,1) & x(1,2) & \cdots & x(1,N) \\ x(2,1) & x(2,2) & \cdots & x(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ x(M,1) & x(M,2) & \cdots & x(M,N) \end{bmatrix} \tag{1}$$

The kth row and kth column of X are denoted by $X(k,\cdot)$ and $X(\cdot,k)$, respectively. The methods described in this paper are for real-valued vectors, but they extend to complex-valued vectors by replacing every instance of transpose with conjugate-transpose.

3 BASIC RESULTS ON SVD

Henceforth, a matrix denoted by S (possibly with subscripts) represents a diagonal matrix of singular values. The singular values are written $s(k)$ for $1 \le k \le M$, or sometimes $s(X;k)$ when the matrix needs to be identified. *Singular values are always assumed to be arranged in decreasing order so that* $s(1) \ge s(2) \ge \cdots \ge s(M) \ge 0$. Note that the SVD of X, $X = USV^T$, may be written as a sum of outer products

$$X = \sum_{k=1}^{M} s(k)U(\cdot,k)V(\cdot,k)^T. \tag{2}$$

Each outer product $U(\cdot,k)V(\cdot,k)^T$ is a rank-one matrix, and the partial sum

$$X^{(q)} = \sum_{k=1}^{q} s(k)U(\cdot,k)V(\cdot,k)^T \tag{3}$$

has rank q for q smaller than or equal to the rank of X. This partial sum has an important approximation property [6] stated in Lemma 1 in terms of the Frobenius norm.

Definition 1 *The Frobenius norm $||\cdot||_F$ of X is defined as*

$$||X||_F = \sqrt{\sum_{k=1}^{M}\sum_{l=1}^{N}|x(k,l)|^2}. \tag{4}$$

Note that

$$||X||_F = \sqrt{s(X;1)^2 + \cdots + s(X;M)^2}. \tag{5}$$

Lemma 1 *For any matrix Y with rank q smaller than or equal to the rank of X, we have the inequality*

$$||X - Y||_F \geq ||X - X^{(q)}||_F = \sqrt{s(X;q+1)^2 + \cdots + s(X;M)^2}. \tag{6}$$

In this sense, $X^{(q)}$ provides the best rank-q approximation to X.

Definition 2 *The matrix*

$$H_N := I_N - (1/N)\, e_N e_N^T \tag{7}$$

is called the row centering matrix, where I_N is the $N \times N$ identity, and e_N is the $N \times 1$ vector with all components equal to 1.

Lemma 2 *The $N \times N$ row centering matrix H_N is symmetric and idempotent.*

Proof. Let $H_N = I_N - (1/N)e_N e_N^T$ and consider H_N^T:

$$\begin{aligned} H_N^T &= \{I_N - (1/N)e_N e_N^T\}^T \\ &= I_N^T - (1/N)(e_N^T)^T e_N^T \\ &= I_N - (1/N)e_N e_N^T = H_N. \end{aligned}$$

Thus H_N is symmetric. Since $H_N H_N = H_N$, then H_N is idempotent.

Lemma 3 *The singular values of H_N are*

$$s(1) = \cdots = s(N-1) = 1, \qquad s(N) = 0.$$

Suppose now that $\overline{X} := XH_N$ is the mean corrected matrix for some $M \times N$ matrix X, with $M \leq N$. Let the SVD of $\overline{X}$ be denoted $\overline{X} = AS(\overline{X})B^T$, with A the matrix of eigenvectors of $\overline{X}\overline{X}^T$, $S(\overline{X})$ the matrix of singular values, and B the $N \times M$ matrix of eigenvectors of $\overline{X}^T\overline{X}$. As in $X^{(q)} = \sum_{k=1}^{q} s(k)U(\cdot,k)V(\cdot,k)^T$, let

$$\overline{X}^{(q)} = \sum_{k=1}^{q} s(\overline{X};k)A(\cdot,k)B(\cdot,k)^T. \tag{8}$$

Then, by (6) we have the following lemma.

Lemma 4 *The optimum rank-q approximation matrix to* $\overline{X}$ *is* $\overline{X}^{(q)}$.

Suppose now that the SVD of X is $X = USV^T$ and that $X^{(q)}$ is obtained as in (3). Using results in [2], on can establish some important facts about the SVDs of $\overline{X}$ and X.

Lemma 5 *For* $k = 1, \ldots, M$, *we have*

$$s(\overline{X}; k) \leq s(X; k). \tag{9}$$

Lemma 6 *From* (9), *it follows that mean correction generally reduces the error in rank-q approximations,*

$$||\overline{X} - \overline{X}^{(q)}||_F \leq ||X - X^{(q)}||_F. \tag{10}$$

Definition 3 *The matrix* $\Upsilon := (1/N) X e_N e_N^T$ *is called the mean matrix (constant along rows).*

Lemma 7 *Adding the mean to the rank-q approximation to* $\overline{X}$, *i.e., forming* $\overline{X}^{(q)} + \Upsilon$, *gives a better approximation to the original matrix* X *than simply using* $X^{(q)}$.

Proof.

$$\begin{aligned} ||X - (\overline{X}^{(q)} + \Upsilon)||_F &= ||X - (1/N) X e_N e_N^T - \overline{X}^{(q)}||_F \\ &= ||X(I_N - (1/N)\, e_N e_N^T) - \overline{X}^{(q)}||_F \\ &= ||XH - \overline{X}^{(q)}||_F = ||\overline{X} - \overline{X}^{(q)}||_F \\ &\leq ||X - X^{(q)}||_F. \end{aligned}$$

Hence, from the approximation point of view, it is better to remove the mean, form the rank q approximation to the corrected matrix, and add the mean back at the end. In [7], it is shown that the columns of $\overline{X}^{(q)} + \Upsilon$ are the optimum q-dimensional subspace approximation to the corresponding columns of X.

4 MULTIRESOLUTION FORM OF THE SVD

This section describes how the multiresolution SVD is constructed. Recall the following dyadic wavelet transform.

Method 1 (Wavelet Transform) *The signal is filtered separately by the lowpass and highpass filters, and both outputs are decimated by a factor of two. This procedure is recursively repeated on the decimated lowpass output, until the desired level of decomposition is achieved.*

The basic idea behind the multiresolution SVD is to replace filtering with SVD at each level of approximation. This idea is now described, initially for one-dimensional (1-D) signals, and for the dyadic case. It would be easy to extend to higher dimensions and to p-adic decompositions.

Let $X = [x(1), \ldots, x(N)]$ represent a finite-extent 1-D signal. Assume that N is divisible by 2^L for some $L \geq 1$. Let the data matrix at the first level, denoted X_1, be constructed so that its top row contains the odd-numbered samples and the bottom row contains the even-numbered samples:

$$X_1 = \begin{bmatrix} x(1) & x(3) & \cdots & x(N-1) \\ x(2) & x(4) & \cdots & x(N) \end{bmatrix}. \tag{11}$$

The corresponding centered matrix is $\overline{X}_1 = X_1 H_{N/2}$. Let U_1 be the matrix of eigenvectors which diagonalizes the scatter matrix $T_1 = \overline{X}_1 \overline{X}_1^t$:

$$U_1^t T_1 U_1 = S_1^2,$$

where $S_1^2 = \mathrm{diag}\{s_1(1)^2, s_1(2)^2\}$ contains the squares of the two singular values, with $s_1(1) \geq s_1(2)$. Now let

$$\hat{X}_1 = U_1^t \overline{X}_1,$$

so that

$$\overline{X}_1 = U_1 \hat{X}_1.$$

Lemma 8 *The two rows of X satisfy the following inequality:*

$$||\hat{X}_1(2,\cdot)||_F \leq ||\hat{X}_1(1,\cdot)||_F.$$

Proof. The top row of $\hat{X}_1$, namely $\hat{X}_1(1,\cdot)$ corresponds to the largest eigenvalue and the bottom row corresponds to the smallest eigenvalue.

It is considered that natural images contain very rare singularities which correspond to edges and so on, and the main parts are continuous. This means that the energy of singularities, which can be measured by the Frobenius norm, is very small and the continuous parts should be included in $\hat{X}_1(1,\cdot)$ when $\hat{X}_1(1,\cdot)$ is much bigger than $\hat{X}_1(2,\cdot)$. Therefore, $\hat{X}_1(1,\cdot)$ contains the "principal" component and $\hat{X}_1(2,\cdot)$ contains the "detail" component.

Definition 4 *The top row of $\hat{X}_1$ is called the smooth or approximation component of X and the bottom row of $\hat{X}_1$ is called the detail component of X. The elements of the approximation component are called approximation coefficients and the elements of the detail component are called detail coefficients.*

Let $\Phi_1 = \hat{X}_1(1,\cdot)$ and $\Psi_1 = \hat{X}_1(2,\cdot)$ represent the smooth and detail components of X, respectively. Note that Φ_1 and Ψ_1 are uncorrelated since the rows of $\hat{X}_1$ have zero mean, and moreover

$$\hat{X}_1 \hat{X}_1^T = S_1^T. \tag{12}$$

Hence the vector has been decomposed into uncorrelated smooth and detail components.

The next level of the multiresolution SVD repeats the procedure described above, but now using the smooth component Φ_1 in place of X. This procedure is repeated recursively L times.

Method 2 (SVD Multiresolution Analysis) *Let $\Phi_0(1,\cdot) = X$, so that the initial smooth component is the original vector. For $\ell = 1, \ldots, L-1$, set*

$$X_\ell = \begin{bmatrix} \phi_{\ell-1}(1) & \phi_{\ell-1}(3) & \cdots & \phi_{\ell-1}(2N_\ell - 1) \\ \phi_{\ell-1}(2) & \phi_{\ell-1}(4) & \cdots & \phi_{\ell-1}(2N_\ell) \end{bmatrix},$$
$$\overline{X}_\ell = X_\ell H_{N_\ell}.$$

Diagonalize the symmetric matrix $T_\ell := \overline{X}_\ell \overline{X}_\ell^T$ as

$$T_\ell = \overline{X}_\ell \overline{X}_\ell^T = U_\ell S_\ell^2 U_\ell^T.$$

Define

$$\hat{X}_\ell = U_\ell^T \overline{X}_\ell,$$
$$\Phi_\ell = \hat{x}_\ell(1,\cdot), \qquad \Psi_\ell = \hat{x}_\ell(2,\cdot).$$

Remark 1 *Note that, for each level ℓ, the vector*

$$\Phi_\ell = [\phi_\ell(1), \ldots, \phi_\ell(N_\ell)]$$

has $N_\ell = N/2^\ell$ elements and that we require the singular values in T_ℓ to be arranged so that $s_\ell(1) \geq s_\ell(2)$.

Definition 5 *The transformation:*

$$X \longrightarrow \{\Phi_L, \{\Psi_\ell\}_{\ell=1}^L, \{U_\ell\}_{\ell=1}^L, \{\mu_\ell\}_{\ell=1}^L\} \tag{13}$$

is called the L-level dyadic SVD of X, where $\mu_\ell = (1/N_\ell)X_\ell e_{N_\ell}$ are the mean vectors.

Theorem 1 *To fully specify X, it is sufficient to store the lowest resolution smooth component Φ_ℓ, and the detail components Ψ_ℓ, for $\ell = 1, \ldots, L$. In addition, the matrices U_ℓ and the mean vectors μ_ℓ must also be stored.*

Proof. It is easy to see how X can be reconstructed from the right-hand side of (13) since each of the steps in Definition 5 is reversible.

Remark 2 *The same procedure may be applied without mean removal. Although mean removal improves the approximation (see section 3), reasons not to remove the mean include a reduction in computation. But without mean removal, the components of $\hat{X}_\ell$ are not necessarily uncorrelated; however they are still orthogonal.*

The following lemma gives a useful inequality for singular values at different resolution levels.

Lemma 9 *The singular values at levels ℓ and $\ell + 1$ satisfy the following inequality:*

$$s_{\ell+1}(1)^2 + s_{\ell+1}(2)^2 \leq s_\ell(1)^2.$$

If the mean is not removed, then equality is obtained.

Proof. This inequality follows from (5), (9) and (12).

5 FOUR CASE STUDIES

We give four typical experiments of fifth level dyadic multiresolution SVD. We define the energy of a signal by its Frobenius norm. We give the energies of the approximation and the detail in each fifth level of Φ and Ψ in Method 2. The ratio of the energy of approximation to the energy of detail explains the accuracy of the decomposition. The bigger the ratio is, the more accurate the decomposition is.

In the following experiments, the matrix *Energy* and the vector *Energy_Ratio* are:

$$\begin{bmatrix} ||\Phi_1||_F & ||\Psi_1||_F \\ ||\Phi_2||_F & ||\Psi_2||_F \\ ||\Phi_3||_F & ||\Psi_3||_F \\ ||\Phi_4||_F & ||\Psi_4||_F \\ ||\Phi_5||_F & ||\Psi_5||_F \end{bmatrix}, \qquad \begin{bmatrix} ||\Phi_1||_F/||\Psi_1||_F \\ ||\Phi_2||_F/||\Psi_2||_F \\ \vdots \\ ||\Phi_5||_F/||\Psi_5||_F \end{bmatrix}.$$

This vector gives the energy ratios at each level.

Table 1: Left:Nikkei Stock Average in 2001; right: El Centro earthquake wave.

Energy × 1.0e-05		Energy Ratio	Energy × 1.0e-04		Energy Ratio
1.9382	0.0179	108.5363	2.2617	0.6785	3.3332
1.9381	0.0204	95.0042	2.0590	0.9358	2.2002
1.9379	0.0274	70.6873	1.6518	1.2290	1.3440
1.9376	0.0337	57.4113	1.2616	1.0659	1.1836
1.9373	0.0298	65.0026	1.1197	0.5808	1.9277

Experiment 1 (2001 Nikkei Stock Average)

The *Nikkei Stock Average* (Nikkei Average) is Japan's most widely watched stock index. It has roots going back to 1950. Nihon Keizai Shimbun, Inc., has calculated and announced the average since 1971. Since October 1, 1985, the index has been calculated every minute during trading hours. The current calculation method, based on the Dow Jones method, was first used in September 1950. The Tokyo Stock Exchange calculated the average at that time and retroactively calculated it back to May 1949, when the exchange reopened. This is the longest-running stock price index in Japan's history. The Nikkei Stock Average is an average stock price adjusted by the Dow Jones method, which is suitable for monitoring the level of the market and its changes. The components of the Nikkei Stock Average are 225 actively traded issues of the TSE first section. Taken together, the 225 issues reflect up-to-the-moment market trends. Since October 1991, components have been checked every year and those of relatively low liquidity have been replaced by issues of high liquidity. Therefore, the index corresponds to changes in the market environment while maintaining consistency. The Tokyo Stock Exchange average for year 2001 is listed in the left part of Table 1. Figure 1 shows five levels of the dyadic multiresolution SVD of the Nikkei Stock Average of 2001.

Experiment 2 (1940 El Centro earthquake wave)

El Centro is located in California and is famous for the 1940 earthquake. The signal listed in the right part of Table 1 is a record of a typical strong ground motion. Figure 2 shows five levels of the dyadic multiresolution SVD of the 1940 El Centro earthquake wave.

Experiment 3 (1995 Kobe earthquake wave)

The 1995 Kobe earthquake was as follows. On Tuesday, January 17, 1995, at 5:46AM local time, an earthquake of magnitude 7.2 struck the region of Kobe and Osaka in South-central Japan. This region is the second most populated and industrialized area in Japan, after Tokyo, with a total population of about 10 million. The shock occurred at a shallow depth on a fault running from Awaji Island through the city of Kobe, which in itself has a population of about 1.5 million. Strong ground shaking lasted for about 20 seconds and caused severe damage over a large area. Nearly 5,500 deaths have been confirmed, with the number of injured people reaching about 35,000. Nearly 180,000 buildings were badly damaged or destroyed, and officials estimate that more than 300,000 people were homeless on the night of the earthquake. The earthquake wave data is listed in the left part of Table 2. Figure 3 shows five levels of the dyadic multiresolution SVD of the Kobe earthquake wave.

Experiment 4 (leleccum)

Table 2: Left:Kobe earthquake wave; right: leleccum data.

Energy × 1.0e-03		Energy Ratio	Energy × 1.0e-04		Energy Ratio
5.0628	0.7507	6.7442	1.6554	0.0150	110.1318
4.8784	1.3539	3.6032	1.6553	0.0160	103.6214
4.2838	2.3341	1.8353	1.6551	0.0207	79.8380
3.4449	2.5464	1.3529	1.6548	0.0350	47.2716
3.0398	1.6208	1.8755	1.6486	0.0576	28.6077

The MATLAB sample data *leleccum.dat* involves a real-world signal electrical consumption measured over the course of three days. This signal is particularly interesting because of noise introduced when a defect developed in the monitoring equipment as the measurements were being made. The SVD multiresolution analysis effectively removes the noise. The leleccum data is listed in the right part of Table 2. Figure 4 shows five levels of the dyadic multiresolution SVD of the leleccum.

Observation

Here we shall focus our attention on two main aims of SVD multiresolution analysis, namely, de-noising and data compression.

De-noising

Using SVD multiresolution analysis to remove noise from a signal requires identifying which component or components contain the noise and then reconstructing the signal without these components. In the above experiments, we note that successive approximations become less and less noisy as more and more high-frequency information is filtered out of the signal. The level 3 approximation, Φ_3, is quite clean as compared with the original signal.

Of course, in discarding all the high-frequency information, we loose many of the sharpest features of the original signal. Optimal de-noising requires a more subtle approach called thresholding. This involves discarding only the portion of the details that exceeds a certain limit.

Data Compression

The compression features of the SVD multiresolution analysis are primarily linked to the relative scarceness of the SVD decomposition of the signal. The notion behind compression is based on the concept that the regular component of the signal can be accurately approximated using the following elements: a small number of approximation coefficients (at a suitably chosen level) and only some of the detail coefficients.

Like de-noising, the compression procedure contains three steps:

Step 1. **Decompose**.

Step 2. **Threshold detail coefficients**. For each level from 1 to N, a threshold is selected and hard thresholding is applied to the detail coefficients.

Step 3. **Reconstruct**.

The difference with the de-noising procedure is in step 2. There are two compression approaches available. The first consists in taking the SVD decomposition of the signal and keeping the largest absolute value coefficients. In this case we can set a global threshold, a compression performance,

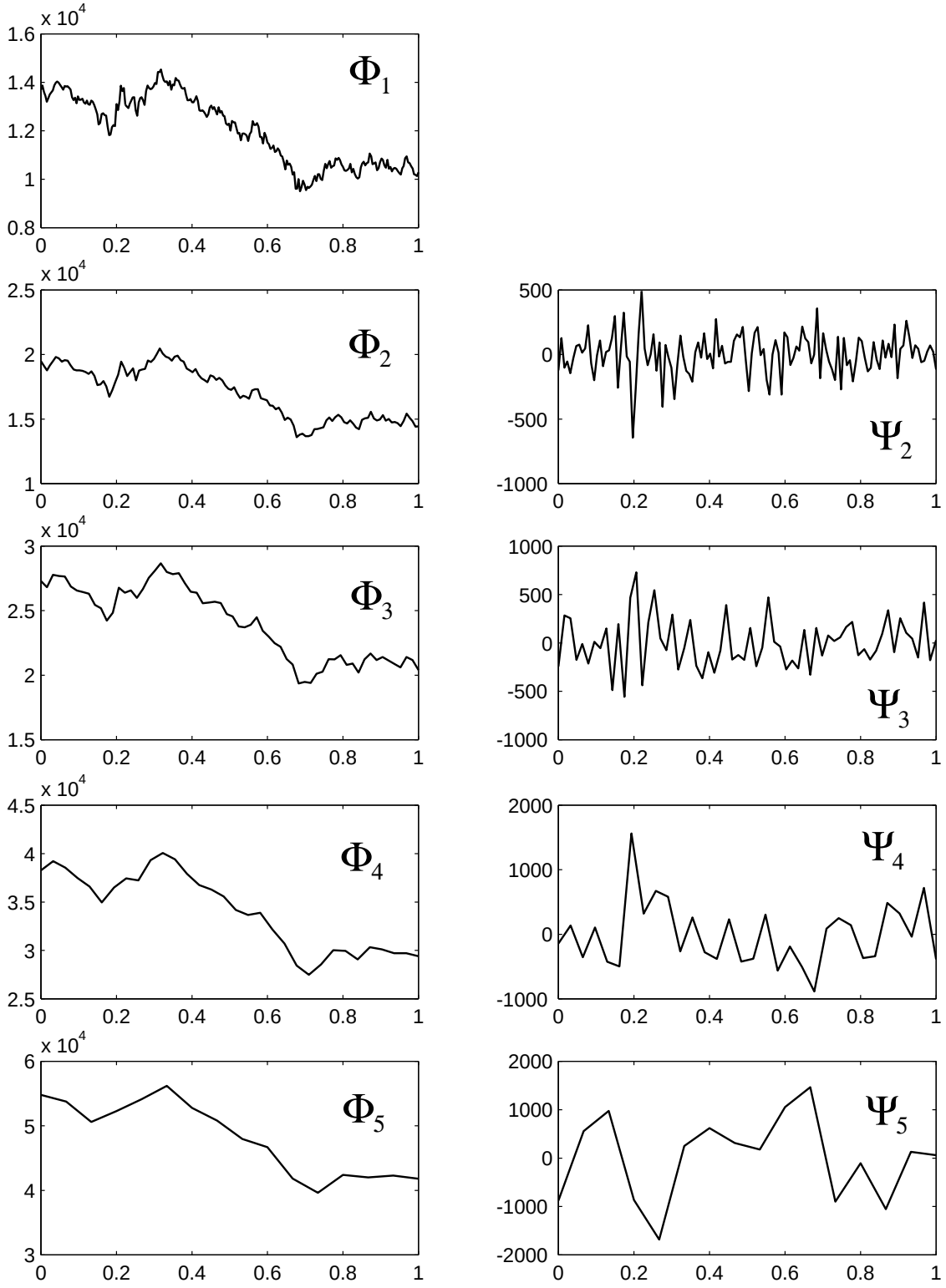

Figure 1: Five levels of the dyadic multiresolution SVD of the 2001 Nikkei Stock Average.

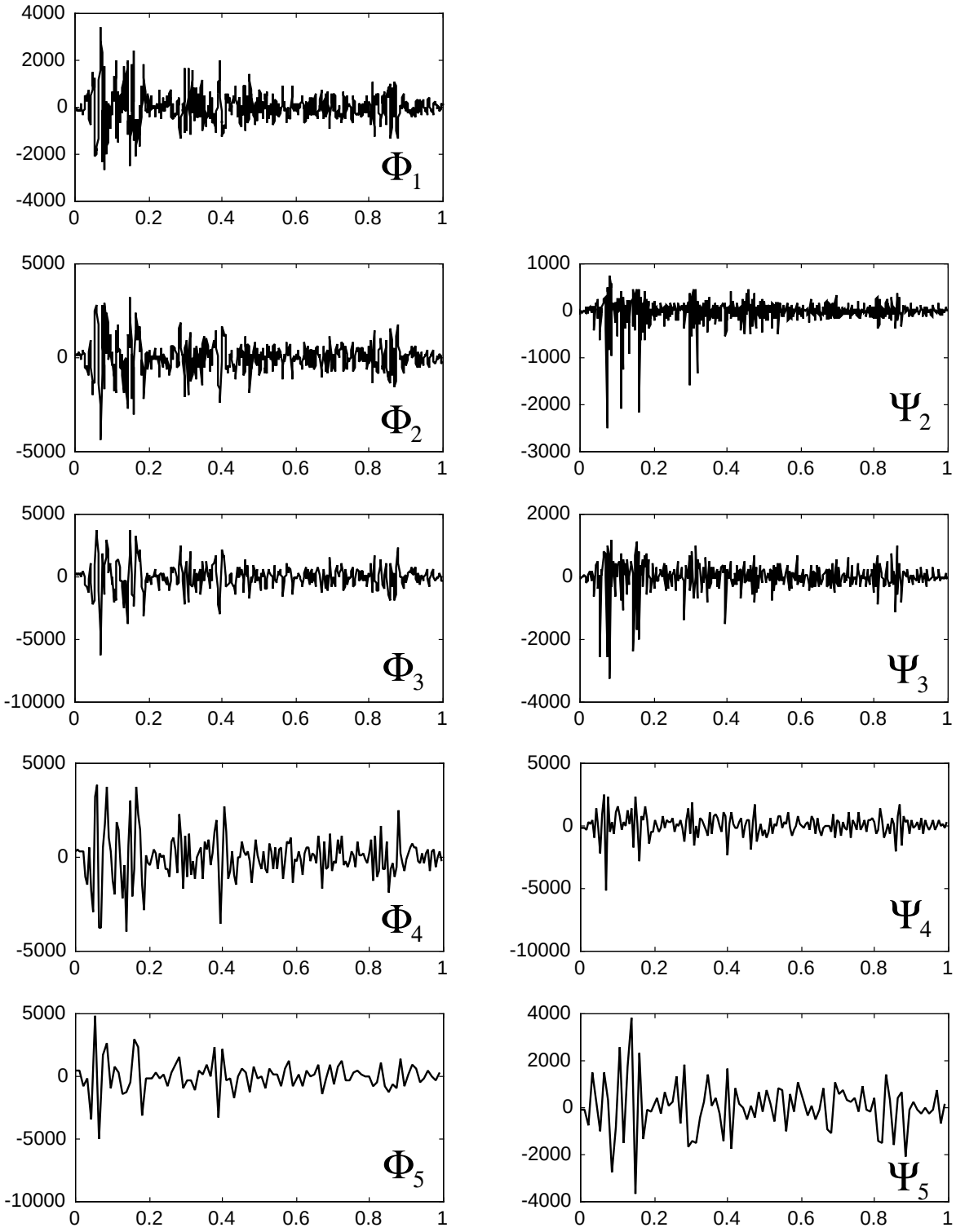

Figure 2: Five levels of the dyadic multiresolution SVD of the El Centro earthquake wave.

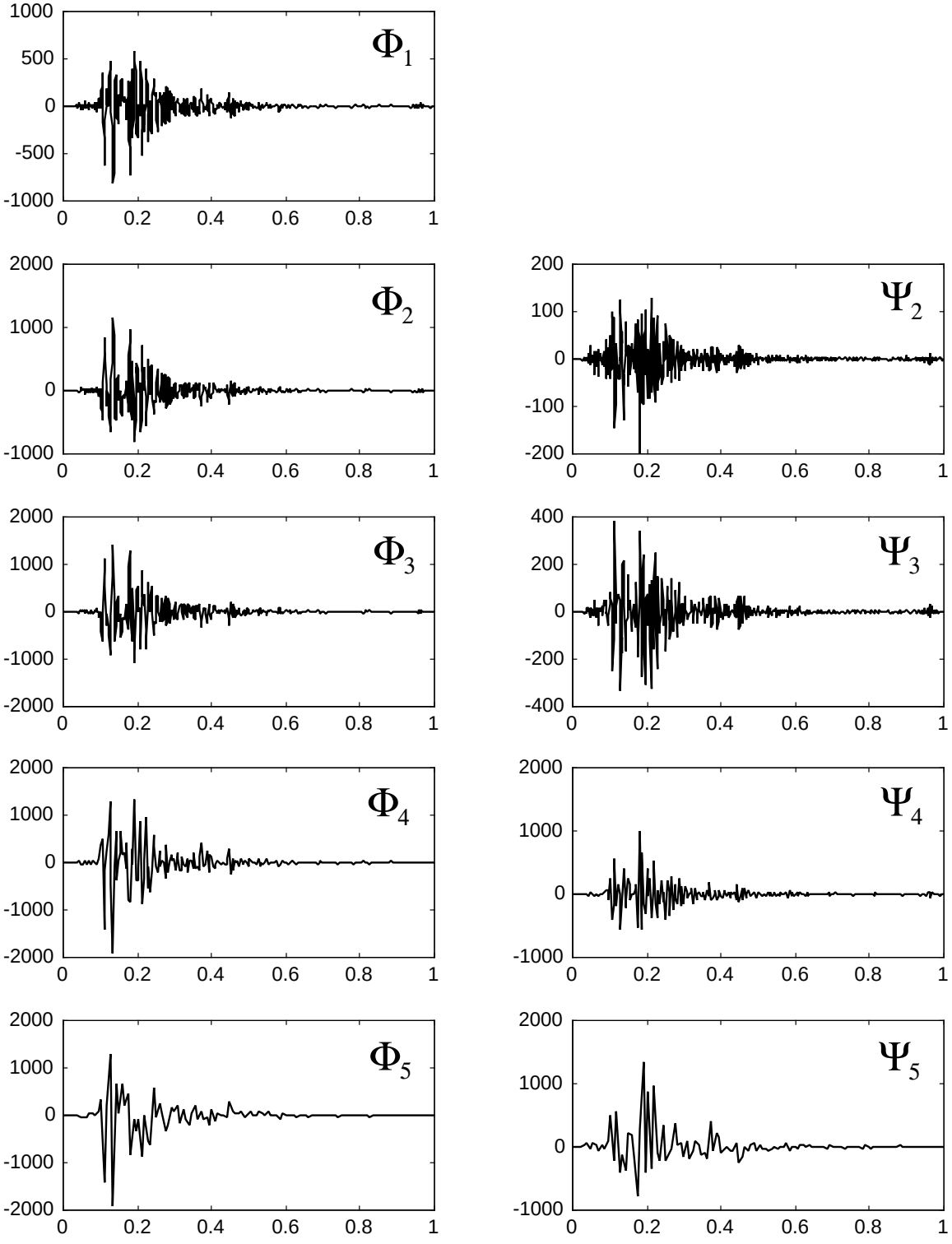

Figure 3: Five levels of the dyadic multiresolution SVD of the Kobe earthquake wave.

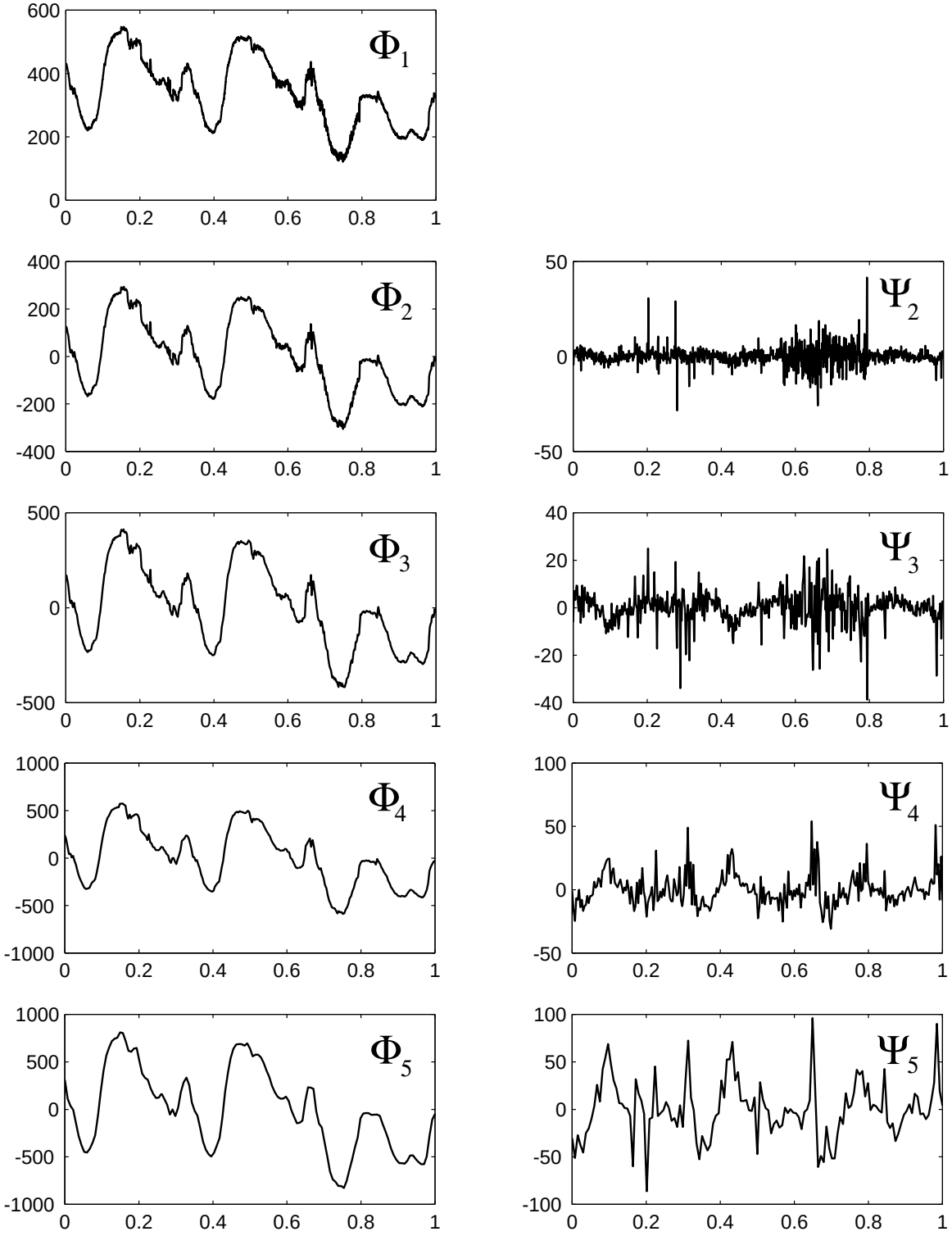

Figure 4: Five levels of the dyadic multiresolution SVD of leleccum.

or a relative square norm recovery performance. Thus only one parameter needs to be selected. The second approach consists in applying visually determined level-dependent thresholds.

Now we investigate our four experiments.

(i) **De-noising**. First, note that we assume that the energy of the noise is very small compared to that of the signal. Experiments 1 and 4 satisfy this condition. In fact, the figures for Experiment 1 show that de-noising does works fine at level 3 and the figures for Experiment 4 show that de-noising does works fine at level 5. For Experiments 2 and 3, nothing can be said about de-noising.

(ii) **Data Compression**. First, note that, at each level, the number of elements of the approximation Φ_ℓ and detail Ψ_ℓ is a half the number of elements of the approximation $\Phi_{\ell-1}$ at the previous level. Therefore, if the energy of Φ_ℓ is almost the same as the energy of $\Phi_{\ell-1}$, then we may throw the detail Ψ_ℓ away. We can use Φ_ℓ as an approximation to $\Phi_{\ell-1}$ with half the number of elements. In fact, looking at Experiments 1 and 4, the energy of Φ_ℓ is almost the same as the energy of $\Phi_{\ell-1}$, for $\ell = 2, 3, 4, 5$, and the figures show that data compression works well at each level. But looking at Experiments 2 and 3, the energy of Φ_ℓ differs from the energy of $\Phi_{\ell-1}$, for $\ell = 2, 3, 4, 5$, and the figures show that data compression does not work well at these levels.

Thus, if the energy of approximations is almost conserved, then we can say that de-noising and data compression perform well.

A THE MATLAB CODES

MATLAB **code 1** *This is a function M-file for the SVD decomposition of a given signal.*

```
function [PHI,PSI,U,MU,Energy] = svd_mra(signal);
% SVD decomposition of a given signal
% Input: signal
% Outputs: PHI, PSI, U, MU, Energy
% By Masaaki YOSHIKAWA, 2001/11
l = length(signal);
signal = reshape(signal,1,l);
i1 = 1:2:(l-1);
i2 = 2:2:l;
approximation = signal(i1);
detail = signal(i2);
sum(approximation)
sum(detail)
X = [approximation; detail];
H = eye(l/2) - (2/l)*ones(l/2);
X = X*H;
[U,S,V] = svd(X);
HX1 = U.'*X;
PHI = HX1(1,:);
PSI = HX1(2,:);
Energy = [norm(PHI,'fro'), norm(PSI,'fro')];
EN = ones(l/2,1);
MU = 2/l*X*EN;
```

MATLAB **code 2** *This is a script M-file to load a signal named* `nikkei.dat` *and to execute the SVD decomposition of the signal.*

```
% Script File: svd_dec.m
%
% SVD decomposition
%
% By Masaaki YOSHIKAWA, 2001/11
%
clear all; close all;
load nikkei.dat;
signal=nikkei;
l = length(signal);
[PHI1,PSI1,U1,MU1,Energy1] = svd_mra(signal);
[PHI2,PSI2,U2,MU2,Energy2] = svd_mra(PHI1);
[PHI3,PSI3,U3,MU3,Energy3] = svd_mra(PHI2);
[PHI4,PSI4,U4,MU4,Energy4] = svd_mra(PHI3);
[PHI5,PSI5,U5,MU5,Energy5] = svd_mra(PHI4);
figure(1)
subplot(3,2,1); plot(linspace(0,1,l),signal);
subplot(3,2,3); plot(linspace(0,1,l/2),PHI1);
subplot(3,2,4); plot(linspace(0,1,l/2),PSI1);
subplot(3,2,5); plot(linspace(0,1,l/4),PHI2);
subplot(3,2,6); plot(linspace(0,1,l/4),PSI2);
figure(2)
subplot(3,2,1); plot(linspace(0,1,l/4),PHI2);
subplot(3,2,3); plot(linspace(0,1,l/8),PHI3);
subplot(3,2,4); plot(linspace(0,1,l/8),PSI3);
subplot(3,2,5); plot(linspace(0,1,l/16),PHI4);
subplot(3,2,6); plot(linspace(0,1,l/16),PSI4);
figure(3)
hold on;
plot(linspace(0,1,l),signal,'y');
plot(linspace(0,1,l/2),PHI1,'m-');
plot(linspace(0,1,l/2),PSI1,'m--');
plot(linspace(0,1,l/4),PHI2,'c-');
plot(linspace(0,1,l/4),PSI2,'c--');
plot(linspace(0,1,l/8),PHI3,'r-');
plot(linspace(0,1,l/8),PSI3,'r--');
plot(linspace(0,1,l/16),PHI4,'g-');
plot(linspace(0,1,l/16),PSI4,'g--');
hold off;
figure(4)
hold on;
plot(linspace(0,1,l),signal,'y');
plot(linspace(0,1,l/2),PHI1,'m-');
% plot(linspace(0,1,l/2),PSI1,'m--');
plot(linspace(0,1,l/4),PHI2,'c-');
% plot(linspace(0,1,l/4),PSI2,'c--');
plot(linspace(0,1,l/8),PHI3,'r-');
```

```
% plot(linspace(0,1,l/8),PSI3,'r--');
plot(linspace(0,1,l/16),PHI4,'g-');
% plot(linspace(0,1,l/16),PSI4,'g--');
hold off;
Energy = [Energy1; Energy2; Energy3; Energy4; Energy5];
Energy_square = Energy.^2;
Energy_square(:,1)+Energy_square(:,2);
plot(Energy)
Ratio_of_Energy = Energy(:,1)./Energy(:,2)
```

ACKNOWLEDGMENT

This research was partially supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (C) 15540170 (2003–2004), the Natural Sciences and Engineering Research Council of Canada and the Centre de recherches mathématiques of the Université de Montréal.

References

[1] B. N. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole, Pacific Grove CA, 1995.

[2] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore and London, 1996.

[4] G. W. Stewart, *On the early history of the singular value decomposition*, Siam Review, Vol. 35, No. 4, pp. 551–566, December 1993.

[5] R. Kakarala and P. O. Ogunbona, *Signal analysis using a multiresolution form of the singular value decomposition*, IEEE Transactions on Image Processing, Vol. 10, No. 5, May 2001.

[6] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, Vol. 1 (1936) 211–218.

[7] C. R. Rao, *Matrix approximations and reduction of dimensionality in multivariate statistical analysis*, in Multivariate Analysis, V. P. R. Krishnaiah, ed., North-Holland, Amsterdam, 1980.