



UPPSALA  
UNIVERSITET

# Dimension Reduction for the Black-Scholes Equation

Alleviating the Curse of Dimensionality

---

Erik Ekedahl, Eric Hansander and Erik Lehto

**Report in Scientific Computing, Advanced Course**

June 2007

PROJECT REPORT





### **Abstract**

An important area in computational finance concerns the pricing of options. When options depend on several underlying stocks, the complexity of the problem makes it difficult to solve using conventional finite difference methods. Instead, stochastic approaches are employed despite the extremely slow convergence of these methods.

The objective of this report is to determine if a dimension reduction technique, namely principal component analysis, could be utilized in combination with a finite difference method for these problems. A number of numerical experiments were designed to examine the efficiency under different conditions. In each case the result was compared to a reference solution from a Monte Carlo method.

The results show that the proposed approach performs very well when the correlation between the underlying stocks is sufficiently high. An example with an option on the OMXS30-index indicates that this criterium is most likely fulfilled in real-world cases.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Option pricing . . . . .	3
1.2	The curse of dimensionality . . . . .	3
1.3	A different approach – principal component analysis . . . . .	4
<b>2</b>	<b>Theory</b>	<b>4</b>
2.1	Options and basket options . . . . .	4
2.2	The Black-Scholes equation . . . . .	5
2.3	Principal component analysis . . . . .	6
2.3.1	Mathematical derivation . . . . .	7
2.4	Asymptotic expansion . . . . .	8
2.5	Monte Carlo methods . . . . .	9
<b>3</b>	<b>Implementation</b>	<b>9</b>
3.1	Solution algorithm . . . . .	9
3.2	Finite difference method . . . . .	10
<b>4</b>	<b>Results</b>	<b>11</b>
4.1	2-dimensional option with highly correlated assets . . . . .	11
4.2	5-dimensional option with highly correlated assets . . . . .	13
4.3	5-dimensional option with weakly correlated assets . . . . .	14
4.4	30-dimensional stock option (OMXS30) . . . . .	14
<b>5</b>	<b>Discussion</b>	<b>15</b>
5.1	Performance . . . . .	15
5.2	Possible improvements . . . . .	16
5.3	Other features . . . . .	17
<b>6</b>	<b>Acknowledgements</b>	<b>17</b>
<b>A</b>	<b>Initial data for the numerical experiments</b>	<b>19</b>
A.1	30-dimensional stock option (OMXS30) . . . . .	19

# 1 Introduction

In financial markets of today, where prices fluctuate by the minute, an increasingly common strategy to keep up with the competition is to use computational models to gain some insight into what the future holds. As the technology advances, these methods are expected to perform even better and are increasingly well trusted by market analysts, further increasing the demands on accuracy and performance.

The focus of this report is to investigate a numerical method for estimating the proper price of basket options using a technique which is quite different from the standard methods.

## 1.1 Option pricing

We begin with an overview of the terminology. The holder of an option has the right, but not the obligation, to perform a certain act. In finance, a European style option is a contract which grants the holder the right to buy or sell an asset at a given price (the strike price) at a given date in the future (the exercise time). An option to buy an asset is commonly known as a call option and an option to sell is called a put option.

The main reason for trading options is to manage risks in a portfolio (a collection of investments). The value of an option depends on at least five factors. These are the expiry date, the strike price, the interest rate, the price of the underlying asset and the volatility of this asset. The volatility is the standard deviation of the change in value of an asset within a given time period. This is in some sense a measure of the risk associated with the option, since a high volatility indicates a higher uncertainty of the value of the asset at the expiry date.

A model for the pricing of European style options was introduced in 1973 by Fischer Black and Myron Scholes, known as the Black-Scholes equation for option pricing [2]. The Black-Scholes equation is a parabolic partial differential equation, which in combination with the known option value at the expiry date gives rise to a final value problem. This problem will be discussed in detail later on.

## 1.2 The curse of dimensionality

In this paper a certain kind of European style option called a basket option will be studied. A basket option has two or more underlying assets and the holder of the option has the right to purchase a specified amount of these at the expiry date. When determining the value of such an option using the Black-Scholes model, the dimensionality of the problem grows linearly with the number of underlying assets.

The conventional method for solving partial differential equations numerically is to discretize the continuous variables in space and time and solve the equation in discrete form, using e.g. finite difference methods [4]. Since every dimension must be discretized, the number of discrete points where a solution has to be calculated will increase exponentially with the number of dimensions. This is known as “the curse of dimensionality”.

A common approach to deal with this problem is to use a Monte Carlo method<sup>1</sup>. Monte Carlo methods have the advantage of handling a growing number of dimensions very well – the amount of computational work grows only linearly with the number of

---

<sup>1</sup>Monte Carlo methods are described further in section 2.2.

dimensions. The problem is that these methods generally offer extremely slow convergence, requiring huge numbers of simulations to yield a sufficiently accurate result.

### 1.3 A different approach – principal component analysis

The approach that will be tried and evaluated in this project is called principal component analysis (PCA) in combination with a finite difference method, which is an entirely different strategy compared to the Monte Carlo methods. Since the assets are correlated, the idea is to reduce the number of dimensions by finding a low-dimensional structure that captures the overall behavior of the high-dimensional problem, disregarding smaller variations. If the contribution to the solution from these variations is negligible, “the curse of dimensionality” can be alleviated and a finite difference method may be used.

## 2 Theory

First, we will take a quick look at the economic theory concerning options in section 2.1. In dealing with the pricing of basket options, it is essential to understand the Black-Scholes equation and the final value problem it yields. This equation will be briefly discussed in section 2.2.

The main theoretical concern in this report is principal component analysis (PCA), which is described in section 2.3. This is the method we will implement for reducing the dimension of a partial differential equation in order to make it possible to solve with a finite difference method.

As a reference solver of the same equations, a Monte Carlo method was used, and therefore a very brief overview of the theory behind Monte Carlo simulations is provided in section 2.5.

### 2.1 Options and basket options

Since the theory for the pricing of put options is identical to that of call options, apart from the final value, only the latter will be addressed in the following sections.

The European style call option is a contract which gives the buyer the right to buy a specified number of assets at a specific date. At the expiry time  $T$  the value  $u(S, t)$  of this option is given by

$$u(S, T) = \max(S - K, 0), \quad (1)$$

where  $K$  is the strike price and  $S$  denotes the price of the underlying asset. Since the holder of the option is not obliged to exercise it, the value is never less than zero. The corresponding equation for a call option on a  $d$ -dimensional basket (of  $d$  assets) is

$$u(S_1, S_2, \dots, S_d, T) = \max\left(\sum_{i=1}^d \mu_i S_i - K, 0\right), \quad (2)$$

where  $S_i$  is the price of underlying asset  $i$  and  $\mu_i$  is the weight factor of asset  $i$ . A thorough introduction to options can be found in [10].

Since the price of the underlying asset at expiry is unknown today (i.e. at time  $t = 0$ ), a method to estimate the value of the option at this time is required. A famous model for calculating the value of European style options is the Black-Scholes equation [10].

## 2.2 The Black-Scholes equation

The Black-Scholes partial differential equation describes the evolution of the price of an option and is given by

$$\frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 u}{\partial S^2} + rS \frac{\partial u}{\partial S} - ru = 0, \quad (3)$$

where  $r$  is the interest rate and  $\sigma$  denotes the volatility of the underlying asset. This equation is easy to solve analytically, when  $r$  and  $\sigma$  are constant. In Figure 1,  $u$  is plotted as a function of  $S$ , here with the strike price  $K = 40$ . The dashed line is the analytical solution for the price of the option according to the Black-Scholes equation and the solid line represents the value at the time of expiry given by (1).

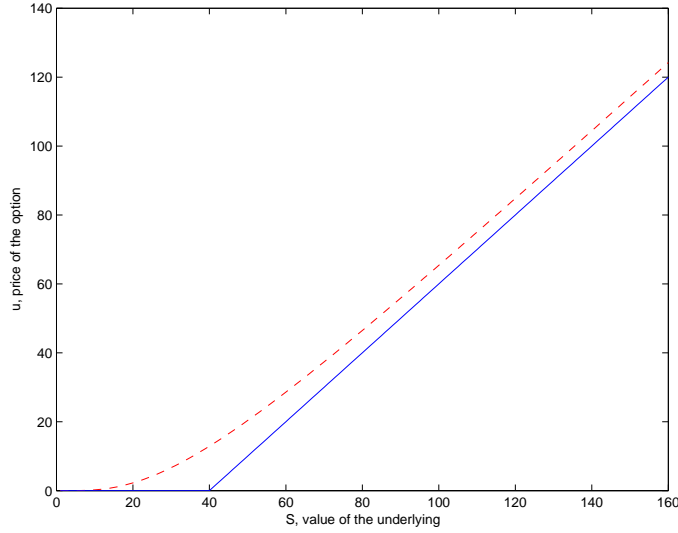


Figure 1: Call option. Analytical solution for the value today (dashed line) and the value at time of expiry (solid line).

With multi-asset basket options, equation (3) generalizes to

$$\frac{\partial u}{\partial t} + \frac{1}{2} \sum_{i,j=1}^d \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 u}{\partial S_i \partial S_j} + r \sum_{i=1}^d S_i \frac{\partial u}{\partial S_i} - ru = 0, \quad (4)$$

where  $\rho_{ij}$  symbolizes the correlation of asset  $i$  and  $j$ . Equation (4) in combination with (2) encompasses a final value problem for the option price  $u$ . Whenever the option depends on more than one asset, which is the case for basket options, the dimensionality of the problem is equal to the number of underlying assets and the complexity grows quickly.

This final value problem is usually transformed into a well-posed initial boundary value problem by a reversion of the time axis  $\tau \rightarrow T - t$  and an introduction of boundary conditions on  $S_i$ . A typical choice of boundary conditions on the asset price for the one-dimensional equation is homogenous Dirichlet condition at  $S = 0$  and either a Dirichlet condition  $u(S, t) = S$  or assuming linearity, e.g.  $\partial^2 u / \partial S^2 = 0$  at  $S \rightarrow \infty$  [10].

### 2.3 Principal component analysis

The main idea of PCA is to reduce the dimensionality of a data set while keeping as much of the variation as possible, thereby minimizing the loss of information contained in the data, in some sense. This method was introduced by Pearson in 1901 [7] and has been widely used in statistics for decades. One strength of principal component analysis is that it gives a set of uncorrelated orthogonal variables, which greatly simplifies the transformation of the partial differential equation.

One can get an intuitive idea of the concept by considering the following simple example. Suppose we have a set of data, distributed as in Figure 2. It is then easy to see that the data is strongly correlated, as the points are clearly distributed around a line in the plane. If the axes were rotated so that the x-axis would lie along this line, the scattering of points would now mainly be in the x-direction, with a much smaller variation in the y-direction.

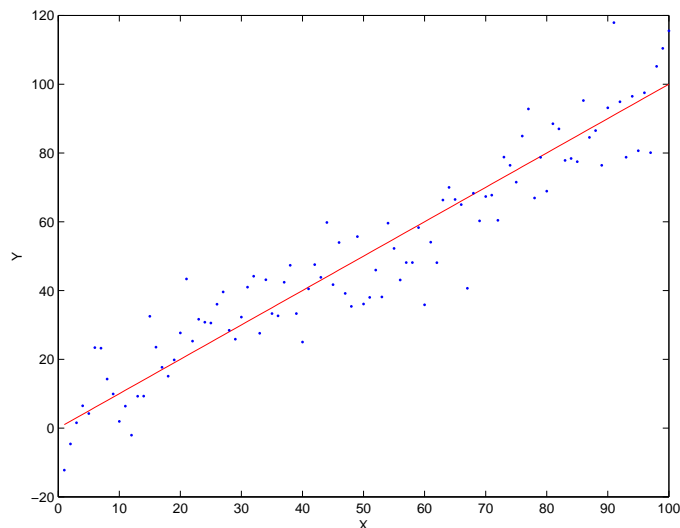


Figure 2: Correlated data set. Direction of largest variation given by the solid line.

If the variables are well correlated, as in the example, then the variation in the x-direction will be so much larger than in the y-direction that neglecting the y-axis entirely will introduce only a small approximation error. This way, we have managed to reduce the dimensionality of the data set from two to one dimension, and still we have kept most of the information in the set.

If the data is scattered in a way implicating that the correlation of the variables is low, the dimension reduction will introduce a larger error. This is because the variation is large in both directions and in this case neglecting one of them would result in a greater loss of information.

The idea can be generalized to higher dimensions by an equivalent rotation of the coordinate system and projecting the data onto one or several of the new directions, disregarding as many of the least significant dimensions as is deemed acceptable.



### 2.3.1 Mathematical derivation

A thorough derivation of the principal components for a set of variables can be found in [5] and only a brief outline will be given in this section. Here,  $\bar{S}$  denotes the original set of variables and  $\bar{x}$  is the new set acquired through PCA.

To maximize the variance of each component of the new set of variables, first find the coefficients  $\alpha_{1j}$  for the linear combination

$$\bar{\alpha}_1^T \bar{S} = \alpha_{11}S_1 + \alpha_{12}S_2 + \cdots + \alpha_{1d}S_d = \sum_{j=1}^d \alpha_{1j}S_j \quad (5)$$

which maximizes

$$\text{var}[\bar{\alpha}_1^T \bar{S}] = \bar{\alpha}_1^T \Sigma \bar{\alpha}_1 \quad (6)$$

under some constraint on  $\bar{\alpha}_1$ , e.g.  $\bar{\alpha}_1^T \bar{\alpha}_1 = 1$ . Here  $\Sigma$  denotes the covariance matrix of the original variables. This optimization problem has the solution

$$\Sigma \bar{\alpha}_1 = \lambda \bar{\alpha}_1. \quad (7)$$

Thus  $\bar{\alpha}_1$  is an eigenvector of  $\Sigma$  with eigenvalue  $\lambda$  and since

$$\bar{\alpha}_1^T \Sigma \bar{\alpha}_1 = \lambda, \quad (8)$$

which is the quantity to be maximized,  $\bar{\alpha}_1$  is the eigenvector corresponding to the largest eigenvalue. The direction given by  $\bar{\alpha}_1$  is called the principal axis. Note that a large eigenvalue indicates a large variation in this direction. Using the same approach for a linear combination  $\bar{\alpha}_2^T \bar{S}$  and imposing the additional constraint that  $\bar{\alpha}_1^T \bar{S}$  and  $\bar{\alpha}_2^T \bar{S}$  should be uncorrelated will lead to  $\bar{\alpha}_2$  corresponding to the eigenvector with second largest eigenvalue. This procedure is repeated for  $\bar{\alpha}_3, \bar{\alpha}_4, \dots, \bar{\alpha}_d$  giving the new set of variables  $x_k = \bar{\alpha}_k^T \bar{S}$ .

Let  $\mathbf{Q}$  denote the matrix with the eigenvectors of the covariance matrix, i.e.  $\mathbf{Q} = (\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_d)$ . The transformation to the principal components can then be written as

$$\bar{x} = \mathbf{Q}^T \bar{S}. \quad (9)$$

Unfortunately, using this transformation for the Black-Scholes equation would lead to a very unwieldy equation. Instead a change to logarithmic coordinates, a reversion of time and a translation to remove the drift is combined with the rotation in equation (9) to give the following transformation

$$\bar{x} = \mathbf{Q}^T \ln \bar{S} + \bar{b}\tau, \quad (10)$$

where  $\tau = T - t$  and  $b_i = \sum_{j=1}^d q_{ji} (r - \sigma_j/2)$ . For a full derivation of this transformation, see [8]. Applying the change of variables given by (10) to the Black-Scholes equation (4) results in

$$\frac{\partial u}{\partial \tau} = \frac{1}{2} \sum_{i=1}^d \lambda_i \frac{\partial^2 u}{\partial x_i^2} - ru, \quad (\bar{x}, \tau) \in \mathbb{R}^d \times (0, T), \quad (11)$$

where  $\lambda_i$  is eigenvalue number  $i$  of the covariance matrix. Transforming the final condition (2) gives

$$u(\bar{x}, 0) = \max \left( \sum_{i=1}^d \mu_i e^{\sum_{j=1}^d q_{ij} x_j}, 0 \right), \quad \bar{x} \in \mathbb{R}^d. \quad (12)$$

As visible in equation (11), the transformation leads to a diagonal diffusion equation with constant coefficients, and this is the main advantage of this change of variables. Consequently the sum in equation (11) may be truncated and the partial differential equation is solved only for the first  $n$  principal components.

A disadvantage of this transformation is that the domain is now unbounded in every direction in space and when solving the equation numerically, a reasonable domain must be specified and numerical boundary conditions have to be applied. Due to the logarithmic coordinates, linearity along the boundary may no longer be considered accurate. The equation is luckily tolerant regarding boundary conditions and the solution far from the boundary is not affected significantly by sensible choices of boundary conditions.

## 2.4 Asymptotic expansion

Since equation (11) may be truncated to any number of dimensions, it is tempting to try with just one or two dimensions seeing that this problem is easy to discretize and can be solved quickly using a finite difference method. It will be shown however, that large errors can be introduced by disregarding variables with small variations, even in cases with high correlation. This was also noted in [9].

A naive way of reducing the error would be to neglect fewer dimensions and solve a higher dimensional problem. However, the covariance matrix for a correlated problem of this kind will have one eigenvalue much larger than the rest and the size of the following eigenvalues decline slowly. Therefore the small contribution to the solution from each of the non-principal dimensions will have similar magnitude.

To account for all dimensions without solving the original high-dimensional problem, each of the non-principal dimensions can be approximated by a linear asymptotic expansion, similar to a Taylor expansion. This asymptotic expansion is given by

$$u = u^{(1)} + \sum_{j=2}^d \lambda_j \left. \frac{\partial u}{\partial \lambda_j} \right|_{\bar{\lambda}=\bar{\lambda}^{(1)}} + \mathcal{O}(\|\bar{\lambda} - \bar{\lambda}^{(1)}\|^2), \quad (13)$$

where  $u^{(1)}$  is the solution in the principal axis,  $\bar{\lambda}$  is a parameter vector of eigenvalues and  $\bar{\lambda}^{(1)} = (\lambda_1, 0, \dots, 0)$  is the parameter vector when truncating the equation to one dimension. Using a finite difference approach, the terms in the sum may be approximated by

$$\left. \frac{\partial u}{\partial \lambda_j} \right|_{\bar{\lambda}=\bar{\lambda}^{(1)}} = \frac{u^{(1,j)} - u^{(1)}}{\lambda_j} + \mathcal{O}(\lambda_j^2), \quad (14)$$

where  $u^{(1,j)}$  corresponds to the solution of the 2D problem on the plane spanned by the principal axis and variable  $j$ .

All together, the PCA method with asymptotic expansion leads to the following solution algorithm. An  $n$ -dimensional problem is first reduced to one dimension and solved for example with a finite difference method giving the solution  $u^{(1)}$ . Then, the two dimensional problem in the plane spanned by the principal axis and the axis corresponding to the second largest eigenvalue is solved, yielding the solution  $u^{(1,2)}$ . This solution is used to correct the principal solution  $u^{(1)}$  according to (13) and (14). This is then repeated once for every remaining dimension  $d = 3, \dots, n$ , correcting the principal solution with  $u^{(1,d)}$ .

For example, for a 30 dimensional problem, one 1D problem (the principal problem) and 29 different 2D problems must be solved.

## 2.5 Monte Carlo methods

We chose a Monte Carlo method for generating the reference solution since it should be relatively unaffected by modelling and approximation errors (e.g. from boundary conditions), and handles high dimensional problems well. It is assumed that the asset price adheres to the stochastic differential equation

$$dS_i = rS_i dt + \sum_{j=1}^d \sigma_{ij} S_i dW_j, \quad i = 1, \dots, d \quad (15)$$

$$(16)$$

where  $\sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$  is the volatility matrix and  $dW_j$  is a Wiener process and can be expressed as  $dW_j = \sqrt{dt} g_j$ , with  $g_j$  being independent normally distributed random numbers [3]. The Monte Carlo method we use only takes one step in time, instead of doing a random walk. This does not lead to lower accuracy in this case and is therefore used for the higher efficiency. We solved the Black-Scholes equation written in the following form:

$$S_i(T) = S_i(0) e^{rT - \sum_{j=1}^d \sigma_{ij} \sqrt{T} g_j} \quad (17)$$

To account for the correlations of the underlying assets a Cholesky decomposition of the volatility matrix  $\sigma_{ij}$  was used.

This simulation is repeated a large number of times, and in the end of every simulation the payoff is computed as

$$\max \left( \sum_{i=1}^d \mu_i S_i - K, 0 \right) \quad (18)$$

Finally, when all simulations are done, the sum of all payoffs is divided by the number of simulations, to give an approximation of  $u(S_i, 0)$ .

## 3 Implementation

The solution algorithm was implemented in MATLAB and an explanation of the basic steps is found in section 3.1. The finite difference method used is discussed in section 3.2.

### 3.1 Solution algorithm

For a given high-dimensional problem, the following steps are performed to obtain a solution.

- Step 1.** Calculate the eigenvalues and eigenvectors of the covariance matrix. Sort the eigenvalues (and the corresponding eigenvectors) by size in descending order.
- Step 2.** Find the initial value  $\bar{x}^0$ , given by the transformation  $\bar{x}^0 = \mathbf{Q}^T \ln \bar{S}^0 + \bar{b}T$  of the current value of the underlying assets  $\bar{S}^0$ , as in (10).
- Step 3.** Create an equidistant grid for the principal axis around  $x_1^0$  and solve the 1D heat equation on this grid using e.g. a finite difference method. This solution corresponds to  $u^{(1)}$  in (13).

**Step 4.** Create an equidistant grid for axis  $d$ , where  $d = 2, \dots, n$ , and solve the 2D heat equation in the plane spanned by the principal axis and axis  $d$ . This solution corresponds to  $u^{(1,d)}$  in (14). Repeat for each dimension  $d$ .

**Step 5.** Correct the solution according to the asymptotic expansion given in equations (13) and (14). From each 2D solution, the solution on the line with  $x_1 = x_1^0$  is used in the correction.

This means that the implementation takes the prices of the assets today as input, and returns the approximation of the option value for every point along the intersection of the 2D grids (including the point  $\bar{x}^0$ ), i.e. for all grid points in the 1D grid. In the numerical experiments in section 4, only the solution at  $\bar{x}^0$  is examined.

### 3.2 Finite difference method

In order to solve the initial-boundary value problem given by (11) and (12), a finite difference method was implemented in MATLAB. The scheme chosen was BDF-2, which is unconditionally stable and second order accurate in both space and time. This implicit scheme admits the use of large time steps, which is desirable since it reduces the number of computations, and will not deteriorate the solution significantly as the total error will be dominated by the dimension reduction approximation.

Since BDF-2 is a two step method, a small disadvantage is the higher memory requirement and the need for another scheme to perform the initial time step. Here, the implicit Euler method was used for the first iteration.

The BDF-2 scheme is given by

$$a_0 u_i^n = \Delta t P(u_i^n) - a_1 u_i^{n-1} - a_2 u_i^{n-2}, \quad (19)$$

with constants  $a_0 = 1.5$ ,  $a_1 = -2$  and  $a_2 = -0.5$ . The spatial difference operator  $P$  is in the 1D case given by

$$P(u_i^n) = \frac{1}{2} \lambda \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\Delta x^2} - r u_i^n, \quad (20)$$

and in the 2D case,

$$P(u_i^n) = \frac{1}{2} \lambda_1 \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\Delta x_1^2} + \frac{1}{2} \lambda_d \frac{u_{i-N_p}^n - 2u_i^n + u_{i+N_p}^n}{\Delta x_d^2} - r u_i^n, \quad (21)$$

with the solution vector  $u$  lexicographically ordered and  $N_p$  corresponding to the number of points in the principal axis.

As noted earlier, the domain must be specified and since the equation behaves nicely far from the strike price, it is customary to set a far-field boundary condition at e.g.  $S = 4K$  in 1D. Similar boundaries in the logarithmic domain were rather arbitrarily chosen as  $x_1 \in [0, 2x_1^0]$  for the principal axis and  $x_d \in [x_d^0 - 3, x_d^0 + 3]$  for the other axes. Here,  $\bar{x}^0$  corresponds to the transformation of  $\bar{S}^0$ , the current price of the underlying assets. Other boundary placements were tried but these boundaries proved to work well in numerical experiments. An improvement would be to have boundary conditions independent of the scaling of initial values.

The boundary conditions used were

$$u = 0, \quad \text{at } x_1 = 0, \quad (22)$$

$$u = \sum_{i=1}^d \mu_i e^{\sum_{j=1}^n q_{ij}(x_j - b_j \tau)} - K e^{-r\tau}, \quad \text{at } x_1 = 2x_1^0, \quad (23)$$

$$\frac{\partial^2 u}{\partial x_d^2} = 0, \quad \text{at } x_d = x_d^0 - 3, x_d^0 + 3, \quad (24)$$

with the linear boundary conditions discretized as

$$u_i^n = 2u_{i+N_p}^n - u_{i+2N_p}^n, \quad i = 1, \dots, N_p, \quad (25)$$

$$u_i^n = 2u_{i-N_p}^n - u_{i-2N_p}^n, \quad i = 1 + (N_d - 1)N_p, \dots, N_d N_p, \quad (26)$$

where  $N_d$  is the number of points in the secondary axis.

These conditions were chosen since they gave a reasonably accurate solution near the boundary when solving a 2D problem without dimension truncation.

## 4 Results

To evaluate the dimension reduction method, a number of numerical tests were performed. These experiments were devised to analyze when the method is applicable and determine the accuracy of the solution in these cases.

### 4.1 2-dimensional option with highly correlated assets

This experiment was selected to show that the finite difference method converges to the exact solution, as this problem can be solved in 2D without truncation. The reference solution is given by the Monte Carlo simulation with  $n = 10^8$  iterations. Another incentive for this particular test was to make sure that the finite difference scheme is implemented correctly, i.e. that the convergence is second order in both space and time. The initial values and parameters used are presented in Table 1, and the convergence results are plotted in Figures 3 through 5. See Table 2 for the results.

$r$	$T$	$K$	$\bar{\mu}$	$\bar{S}^0$	$\bar{\sigma}$	$\rho_{ij}$	$\bar{\lambda}$
0.05	1.0	50	(1 1)	$\begin{pmatrix} 25 \\ 25 \end{pmatrix}$	$\begin{pmatrix} 0.24 \\ 0.25 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.95 \\ 0.95 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 0.177 \\ 0.003 \end{pmatrix}$

Table 1: Parameters for the 2D example.

Method	$u$	Abs. error	Rel. error
1D PCA	5.9919	0.0233	$3.9 \cdot 10^{-3}$
2D	6.0156	$4 \cdot 10^{-4}$	$6.2 \cdot 10^{-5}$
MC	6.0152	–	–

Table 2: Comparison of different methods for the 2D option. For the FDM, 500 time steps and  $N_p = N_d = 161$  were used. The 1D PCA solution corresponds to  $u^{(1)}$ , i.e. the solution when truncating the original 2D problem to one dimension. The 2D entry is the solution  $u^{(1,2)}$ , i.e. the solution to the original problem transformed to the new coordinate system.

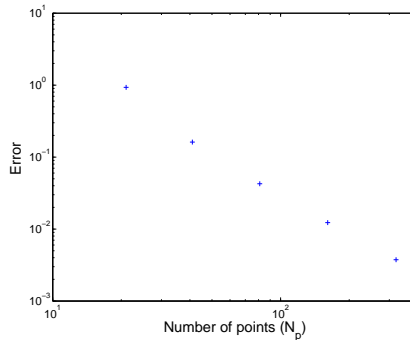


Figure 3: Log-log plot showing convergence in space with 500 time steps, principal axis solution. Estimated slope is  $-1.99$ .

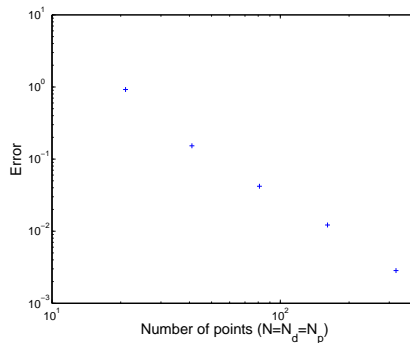


Figure 4: Log-log plot showing convergence in space with 500 time steps, 2D solution. Estimated slope is  $-2.07$ .

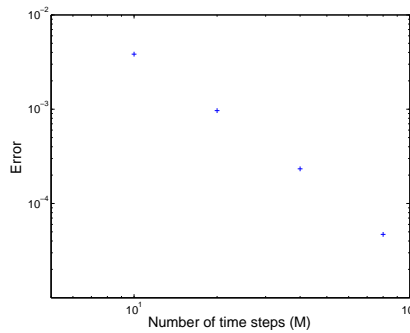


Figure 5: Log-log plot showing convergence in time with  $N_p = N_d = 161$ , 2D solution. Estimated slope is  $-2.11$ .

## 4.2 5-dimensional option with highly correlated assets

In this example a 5D basket option was considered where the assets had high correlation. The error when truncating to a low-dimensional problem was evaluated and compared to the error when performing the asymptotic expansion, see Table 4. Again, a Monte Carlo simulation with  $n = 10^8$  simulations is used as a reference. An eigenvalue spectrum for this test problem is available in Figure 6. The parameters were chosen as  $r = 0.05$ ,  $T = 2.0$  and  $K = 125$ , and  $\bar{\mu}$ ,  $\bar{S}^0$ ,  $\bar{\sigma}$  and  $\rho_{ij}$  as in Table 3.

$\bar{\mu}$	$\bar{S}^0$	$\bar{\sigma}$	$\rho_{ij}$
$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 25 \\ 25 \\ 25 \\ 25 \\ 25 \end{pmatrix}$	$\begin{pmatrix} 0.518 \\ 0.648 \\ 0.623 \\ 0.570 \\ 0.530 \end{pmatrix}$	$\begin{pmatrix} 1.0 & 0.79 & 0.82 & 0.91 & 0.84 \\ 0.79 & 1.0 & 0.73 & 0.80 & 0.76 \\ 0.82 & 0.73 & 1.0 & 0.77 & 0.72 \\ 0.91 & 0.80 & 0.77 & 1.0 & 0.90 \\ 0.84 & 0.76 & 0.72 & 0.90 & 1.0 \end{pmatrix}$

Table 3: Parameters for the 5D example.

Method	$u$	Abs. error	Rel. error
1D PCA	36.42	4.60	11.2%
2D PCA	38.30	2.72	6.63%
Asymptotic Exp.	40.96	0.0614	0.15%
MC	41.02	–	–

Table 4: Comparison of different methods for the 5D option with high correlation. For the FDM, 80 time steps and  $N_p = N_d = 161$  were used. The 1D PCA solution corresponds to  $u^{(1)}$  and the 2D PCA entry is the solution  $u^{(1,2)}$ , i.e. the solution to the original 5D problem truncated to two dimensions.

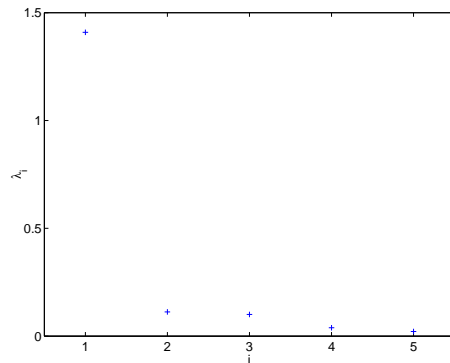


Figure 6: Ordered eigenvalue spectrum of the 5D example with high correlation.

### 4.3 5-dimensional option with weakly correlated assets

In order to study the irreducible error introduced by the principal components analysis, a basket option with assets having low correlation was analyzed. The only difference between this example and the one in section 4.2 is that the correlation matrix is altered. Here, the matrix is given by dividing the off-diagonal elements of the correlation matrix in Table 3 by a factor of 5. For the eigenvalue spectrum for this experiment, see Figure 7. Table 5 shows the results, and a comparison of the different methods.

Method	$u$	Abs. error	Rel. error
1D PCA	13.56	16.0	54.1%
2D PCA	18.21	11.4	38.4%
Asymptotic Exp.	27.19	2.37	8.04%
MC	29.57	–	–

Table 5: Comparison of different methods for the 5D option with low correlation. For the FDM, 80 time steps and  $N_p = N_d = 161$  were used. The 1D PCA solution corresponds to  $u^{(1)}$  and the 2D PCA entry is the solution  $u^{(1,2)}$ , i.e. the solution to the original 5D problem truncated to two dimensions.

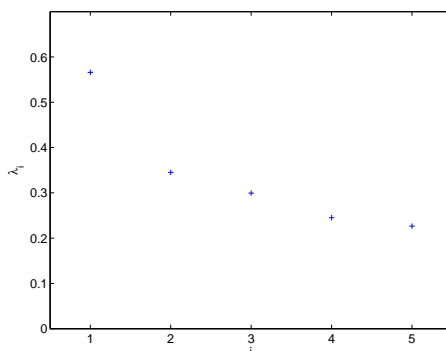


Figure 7: Ordered eigenvalue spectrum of the 5D example with low correlation.

### 4.4 30-dimensional stock option (OMXS30)

OMXS30 is an index on the 30 most traded stocks<sup>2</sup> on the Stockholm stock exchange. A basket option was created on these stocks, with weights according to the total stock market value. In order to get a comparison with actual call options on the OMXS30-index, asset prices were scaled according to the current index value. Volatilities and the correlation matrix were estimated from daily returns for one year, using the estimate

$$\sigma = \sqrt{\frac{1}{(n-1)dt} \sum_j (R_j - \bar{m})^2}, \quad (27)$$

for the volatility [1], where  $\bar{m}$  is the mean of the daily returns  $R_j$ . When estimating volatility for an option with expiry date 7 months from now, the 120 latest daily returns

<sup>2</sup>For a list of the included stocks, see <http://www.omxgroup.com/nordicexchange/> cited on May 28, 2007.



were used. See section A.1 in the Appendix for the estimates and initial values. Eigenvalues are plotted in Figure 8. For results and comparison of the different methods, see Table 6. In this table, the actual option value was the asking price<sup>3</sup> at the OMX Nordic Exchange.

Method	$u$	Abs. error	Rel. error
1D PCA	96.948	10.5	9.78%
Asymptotic Exp.	107.31	0.1502	0.14%
MC	107.48	–	–
Actual option	101.50	–	–

Table 6: Comparison of different methods for the 30D option on the OMXS30. For the FDM, 20 time steps and  $N_p = N_d = 161$  were used. The 1D PCA solution corresponds to  $u^{(1)}$ .

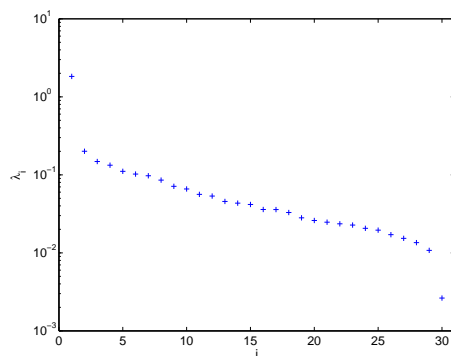


Figure 8: Ordered eigenvalue spectrum of the 30D stock option. The particularly small eigenvalue is presumably the result of the index containing two stocks from the same company (Atlas Copco A and Atlas Copco B) which are very highly correlated.

## 5 Discussion

First we will review the performance of this method, compare it to other methods, and say something about the sources of errors. Then, potential improvements will be proposed. Finally, non-obvious features of this method are mentioned.

### 5.1 Performance

From our results, it can be concluded that the PCA combined with asymptotic expansion is an efficient method of pricing high-dimensional options with high correlation. In cases where the largest eigenvalue of the correlation matrix is about ten times greater

<sup>3</sup>Data collected from the price information pages of <http://www.omxgroup.com/nordicexchange/Themarket/> on May 28, 2007.

than the second largest, the irreducible error is very small and the gain in numerical efficiency can be considerable. With the finite difference approach, it is also possible to get an approximate solution quickly by using a low number of discrete points.

We have refrained from giving quantitative performance estimates as our implementation of the finite difference method was written in MATLAB while the Monte Carlo method was implemented in Fortran90, resulting in a huge performance advantage. The main focus has been on evaluating the dimension reduction technique and assess when it is in fact applicable. Solving the heat equation in 1D and 2D can be done a lot more efficiently than in our simple finite difference implementation.

It must be noted that the error is quite large when dealing with weakly correlated assets. In reality, as shown by the OMXS30 example, assets generally have a rather high correlation even when stocks from different market sectors are considered. By studying the eigenvalue spectrum in Figure 8 we conclude that the method should certainly be applicable for practical problems.

The poor correspondence between the calculated values and the market value for the OMXS30-index option most likely originates from the data we used for estimating volatilities and correlations. This data certainly differs from the up-to-date numbers used by market analysts. It is also uncertain if the Black-Scholes model is employed to determine the market value of these options. Furthermore, advanced models for estimating the volatility are presumably used by the financial institutions. The error in the FDM solution is still small when compared to the Monte Carlo method. We consider the MC solution to be a more reasonable reference as it is certainly based on the same model and initial data.

## 5.2 Possible improvements

To see the true potential of the method, the finite difference method should be implemented in a more efficient fashion, for instance in Fortran. A way of improving it further would be to use an adaptive grid instead of an equidistant one. Due to the logarithmic coordinates, a lot of points end up where the solution is very close to zero. An adaptive grid should thus decrease the number of discrete points needed for a given error. The algorithm should not be difficult to parallelize either, given that the 2D subproblems are completely independent and can be solved simultaneously.

Again, since we are dealing with the simple heat equation in at most two dimensions, there are other methods which could be used to obtain a numerical solution. A finite element method would certainly be possible. A method using interpolation with radial basis functions (RBF) could also be an interesting approach [6]. These functions only depend on the distance between nodes, making them very easy to implement when data points are distributed unevenly in space. RBF methods also possess very good convergence properties.

The boundary conditions used in our implementation could also be improved as can be seen from the convergence results, where the solution does display second order convergence when the number of points is large enough. When using few points in space however, the error is rather large and mostly due to the boundary conditions. Given more time we would have tried to find optimal boundary conditions for the transformation used here.

Specifying a sensible domain is also challenging. Optimally the boundary conditions should be independent of the initial data of the problem. Applying a different transformation or scaling the variables in some way are techniques which may prove fruitful in dealing with this problem.

### **5.3 Other features**

Another advantage of dimension reduction in combination with finite difference methods is the possibility of calculating “the Greeks”. These are the sensitivities of the option price with respect to different variables, for example asset price. Since the option price is available at a number of points, difference operators may be applied to calculate these sensitivities. When trading with options, “the Greeks” are used to further analyze the risk of a given option. Monte Carlo methods are inherently weak in this aspect since the option value is only calculated for a single initial value and many runs must be performed to be able to approximate sensitivities.

## **6 Acknowledgements**

We would like to thank our supervisors Lina von Sydow and Per Lötstedt at the Department of Information Technology, Uppsala University, for their much appreciated support and helpful advice. We would also like to thank Johan Tysk at the Department of Mathematics for valuable insight into specifying boundary conditions for the Black-Scholes equation.

## References

- [1] Peter. A. Abkin and Saikat Nandi. Options and volatility. *Economic Review, Federal Reserve Bank of Atlanta*, pages 21–35, December 1996.
- [2] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973.
- [3] Jesper Carlsson, Erik von Schwerin, and Anders Szepessy. Lecture notes: Stochastic and partial differential equations with adapted numerics. URL: <http://www.math.kth.se/~szepessy/sdepde.pdf> (date of citation: May 15, 2007), 2006.
- [4] Michael T. Heath. *Scientific Computing, An Introductory Survey*. McGraw-Hill, New York, 2002.
- [5] Ian T. Jolliffe. *Principal Component Analysis*. Springer-Verlag New York Inc., 1986.
- [6] Elisabeth Larsson, Krister Åhlander, and Andreas Hall. Multi-dimensional option pricing using radial basis functions and the generalized fourier transform. Technical Report 2006-037, Department of Information Technology, Uppsala University, 2006.
- [7] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.
- [8] Christoph Reisinger. *Numerische Methoden für hochdimensionale parabolische Gleichungen am Beispiel von Optionspreisaufgaben*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 2004.
- [9] Christoph Reisinger and Gabriel Wittum. Efficient hierarchical approximation of high-dimensional option pricing problems. *SIAM J. Scientific Computing*, 29(1):440–458, 2007.
- [10] Paul Wilmott, Sam Howison, and Jeff Dewynne. *The Mathematics of Financial Derivatives, A Student Introduction*. Cambridge University Press, 1995.

## A Initial data for the numerical experiments

### A.1 30-dimensional stock option (OMXS30)

In this problem, in order to be able to compare with a real option on the OMXS30-index, parameters were chosen as  $r = 0.05$ ,  $T = 7/12$  and  $K = 1260$ . Estimates for the volatilities are given in (28), along with the weights and the unscaled initial values for  $\bar{S}^0$ .

$$\bar{\sigma} = \begin{pmatrix} 0.3213 \\ 0.4041 \\ 0.2563 \\ 0.2153 \\ 0.3524 \\ 0.3692 \\ 0.4830 \\ 0.2176 \\ 0.4316 \\ 0.2555 \\ 0.2713 \\ 0.3011 \\ 0.2644 \\ 0.2701 \\ 0.2618 \\ 0.3633 \\ 0.2596 \\ 0.3553 \\ 0.3652 \\ 0.3006 \\ 0.3549 \\ 0.3685 \\ 0.2499 \\ 0.2655 \\ 0.4042 \\ 0.2751 \\ 0.4037 \\ 0.4716 \\ 0.4676 \\ 0.2744 \end{pmatrix}, \quad \bar{\mu} = \begin{pmatrix} 0.0186 \\ 0.0122 \\ 0.0024 \\ 0.0182 \\ 0.034 \\ 0.0164 \\ 0.0458 \\ 0.018 \\ 0.0145 \\ 0.0058 \\ 0.1446 \\ 0.0891 \\ 0.027 \\ 0.0057 \\ 0.0965 \\ 0.0416 \\ 0.0171 \\ 0.013 \\ 0.0509 \\ 0.0189 \\ 0.0027 \\ 0.0249 \\ 0.0465 \\ 0.0181 \\ 0.0452 \\ 0.0126 \\ 0.0143 \\ 0.0891 \\ 0.0483 \\ 0.008 \end{pmatrix}, \quad \bar{S}^0 = \begin{pmatrix} 136 \\ 420 \\ 154 \\ 364.5 \\ 264.5 \\ 249.5 \\ 154 \\ 383.5 \\ 169.5 \\ 26 \\ 88.75 \\ 171.5 \\ 433.5 \\ 177.75 \\ 117 \\ 130.75 \\ 116.5 \\ 670 \\ 234.5 \\ 97.5 \\ 153.5 \\ 144.5 \\ 129.25 \\ 202.5 \\ 253.5 \\ 123.5 \\ 108 \\ 0.5 \\ 139.5 \\ 393.5 \end{pmatrix} \quad (28)$$

The correlation matrix looks as follows:

```
corrmat =
[1, 0.707, 0.643, 0.348, 0.696, 0.703, 0.707, 0.591, 0.530, 0.549,
0.431, 0.759, 0.575, 0.592, 0.688, 0.687, 0.547, 0.399, 0.663, 0.628,
0.690, 0.719, 0.553, 0.647, 0.618, 0.377, 0.472, 0.346, 0.567, 0.708;
0.707, 1, 0.637, 0.263, 0.698, 0.700, 0.633, 0.558, 0.504, 0.456,
0.319, 0.629, 0.474, 0.486, 0.608, 0.689, 0.438, 0.337, 0.609, 0.560,
0.596, 0.714, 0.444, 0.605, 0.537, 0.327, 0.468, 0.331, 0.552, 0.625;
0.643, 0.637, 1, 0.359, 0.691, 0.691, 0.581, 0.551, 0.446, 0.483,
0.364, 0.673, 0.556, 0.536, 0.657, 0.653, 0.496, 0.336, 0.616, 0.604,
```

0.620,0.642,0.466,0.638,0.530,0.319,0.489,0.393,0.522,0.622;  
0.348,0.263,0.359,1,0.339,0.345,0.325,0.425,0.311,0.342,  
0.187,0.456,0.296,0.320,0.407,0.294,0.388,0.202,0.394,0.318,  
0.318,0.315,0.336,0.293,0.287,0.253,0.317,0.287,0.218,0.389;  
0.696,0.698,0.691,0.339,1,0.977,0.639,0.547,0.521,0.513,  
0.334,0.706,0.573,0.527,0.651,0.803,0.473,0.403,0.657,0.542,  
0.576,0.845,0.526,0.581,0.512,0.281,0.412,0.362,0.593,0.654;  
0.703,0.700,0.691,0.345,0.977,1,0.657,0.559,0.520,0.512,  
0.352,0.705,0.584,0.536,0.652,0.808,0.489,0.380,0.662,0.553,  
0.578,0.838,0.534,0.579,0.514,0.288,0.422,0.387,0.614,0.673;  
0.707,0.633,0.581,0.325,0.639,0.657,1,0.511,0.497,0.453,  
0.370,0.654,0.476,0.459,0.591,0.612,0.503,0.342,0.598,0.562,  
0.595,0.676,0.478,0.616,0.530,0.349,0.455,0.366,0.561,0.749;  
0.591,0.558,0.551,0.425,0.547,0.559,0.511,1,0.432,0.468,  
0.380,0.619,0.431,0.507,0.633,0.537,0.477,0.288,0.552,0.530,  
0.572,0.532,0.434,0.586,0.556,0.351,0.336,0.314,0.425,0.585;  
0.530,0.504,0.446,0.311,0.521,0.520,0.497,0.432,1,0.498,  
0.201,0.584,0.397,0.441,0.498,0.488,0.363,0.291,0.519,0.384,  
0.502,0.510,0.315,0.469,0.468,0.224,0.365,0.259,0.406,0.452;  
0.549,0.456,0.483,0.342,0.513,0.512,0.453,0.468,0.498,1,  
0.297,0.658,0.469,0.551,0.602,0.462,0.460,0.264,0.536,0.465,  
0.522,0.537,0.378,0.499,0.449,0.294,0.428,0.323,0.419,0.533;  
0.431,0.319,0.364,0.187,0.334,0.352,0.370,0.380,0.201,0.297,  
1,0.433,0.264,0.320,0.454,0.274,0.409,0.304,0.441,0.391,  
0.417,0.362,0.407,0.421,0.471,0.302,0.254,0.224,0.394,0.398;  
0.759,0.629,0.673,0.456,0.706,0.705,0.654,0.619,0.584,0.658,  
0.433,1,0.614,0.623,0.740,0.662,0.575,0.489,0.755,0.630,  
0.713,0.741,0.580,0.693,0.674,0.406,0.538,0.426,0.585,0.725;  
0.575,0.474,0.556,0.296,0.573,0.584,0.476,0.431,0.397,0.469,  
0.264,0.614,1,0.535,0.559,0.546,0.438,0.312,0.604,0.468,  
0.606,0.573,0.455,0.554,0.522,0.339,0.434,0.420,0.508,0.497;  
0.592,0.486,0.536,0.320,0.527,0.536,0.459,0.507,0.441,0.551,  
0.320,0.623,0.535,1,0.583,0.506,0.461,0.252,0.571,0.467,  
0.532,0.528,0.504,0.548,0.515,0.275,0.422,0.287,0.504,0.512;  
0.688,0.608,0.657,0.407,0.651,0.652,0.591,0.633,0.498,0.602,  
0.454,0.740,0.559,0.583,1,0.596,0.612,0.384,0.797,0.577,  
0.673,0.648,0.596,0.662,0.686,0.390,0.494,0.419,0.611,0.651;  
0.687,0.689,0.653,0.294,0.803,0.808,0.612,0.537,0.488,0.462,  
0.274,0.662,0.546,0.506,0.596,1,0.490,0.319,0.592,0.559,  
0.596,0.830,0.491,0.570,0.497,0.320,0.446,0.372,0.634,0.601;  
0.547,0.438,0.496,0.388,0.473,0.489,0.503,0.477,0.363,0.460,  
0.409,0.575,0.438,0.461,0.612,0.490,1,0.266,0.621,0.537,  
0.509,0.512,0.680,0.521,0.551,0.319,0.524,0.420,0.496,0.576;  
0.399,0.337,0.336,0.202,0.403,0.380,0.342,0.288,0.291,0.264,  
0.304,0.489,0.312,0.252,0.384,0.319,0.266,1,0.371,0.273,  
0.404,0.402,0.280,0.275,0.360,0.200,0.242,0.145,0.398,0.348;  
0.663,0.609,0.616,0.394,0.657,0.662,0.598,0.552,0.519,0.536,  
0.441,0.755,0.604,0.571,0.797,0.592,0.621,0.371,1,0.584,  
0.697,0.645,0.575,0.714,0.780,0.343,0.569,0.449,0.558,0.638;  
0.628,0.560,0.604,0.318,0.542,0.553,0.562,0.530,0.384,0.465,

0.391,0.630,0.468,0.467,0.577,0.559,0.537,0.273,0.584,1,  
0.537,0.594,0.470,0.610,0.502,0.323,0.477,0.419,0.449,0.594;  
0.690,0.596,0.620,0.318,0.576,0.578,0.595,0.572,0.502,0.522,  
0.417,0.713,0.606,0.532,0.673,0.596,0.509,0.404,0.697,0.537,  
1,0.647,0.499,0.635,0.595,0.384,0.458,0.366,0.535,0.637;  
0.719,0.714,0.642,0.315,0.845,0.838,0.676,0.532,0.510,0.537,  
0.362,0.741,0.573,0.528,0.648,0.830,0.512,0.402,0.645,0.594,  
0.647,1,0.524,0.644,0.548,0.341,0.484,0.437,0.653,0.689;  
0.553,0.444,0.466,0.336,0.526,0.534,0.478,0.434,0.315,0.378,  
0.407,0.580,0.455,0.504,0.596,0.491,0.680,0.280,0.575,0.470,  
0.499,0.524,1,0.496,0.502,0.315,0.390,0.309,0.557,0.512;  
0.647,0.605,0.638,0.293,0.581,0.579,0.616,0.586,0.469,0.499,  
0.421,0.693,0.554,0.548,0.662,0.570,0.521,0.275,0.714,0.610,  
0.635,0.644,0.496,1,0.672,0.349,0.485,0.483,0.464,0.608;  
0.618,0.537,0.530,0.287,0.512,0.514,0.530,0.556,0.468,0.449,  
0.471,0.674,0.522,0.515,0.686,0.497,0.551,0.360,0.780,0.502,  
0.595,0.548,0.502,0.672,1,0.352,0.467,0.387,0.536,0.562;  
0.377,0.327,0.319,0.253,0.281,0.288,0.349,0.351,0.224,0.294,  
0.302,0.406,0.339,0.275,0.390,0.320,0.319,0.200,0.343,0.323,  
0.384,0.341,0.315,0.349,0.352,1,0.265,0.232,0.320,0.366;  
0.472,0.468,0.489,0.317,0.412,0.422,0.455,0.336,0.365,0.428,  
0.254,0.538,0.434,0.422,0.494,0.446,0.524,0.242,0.569,0.477,  
0.458,0.484,0.390,0.485,0.467,0.265,1,0.461,0.466,0.466;  
0.346,0.331,0.393,0.287,0.362,0.387,0.366,0.314,0.259,0.323,  
0.224,0.426,0.420,0.287,0.419,0.372,0.420,0.145,0.449,0.419,  
0.366,0.437,0.309,0.483,0.387,0.232,0.461,1,0.305,0.380;  
0.567,0.552,0.522,0.218,0.593,0.614,0.561,0.425,0.406,0.419,  
0.394,0.585,0.508,0.504,0.611,0.634,0.496,0.398,0.558,0.449,  
0.535,0.653,0.557,0.464,0.536,0.320,0.466,0.305,1,0.505;  
0.708,0.625,0.622,0.389,0.654,0.673,0.749,0.585,0.452,0.533,  
0.398,0.725,0.497,0.512,0.651,0.601,0.576,0.348,0.638,0.594,  
0.637,0.689,0.512,0.608,0.562,0.366,0.466,0.380,0.505,1]