UPPSALA
UNIVERSITET

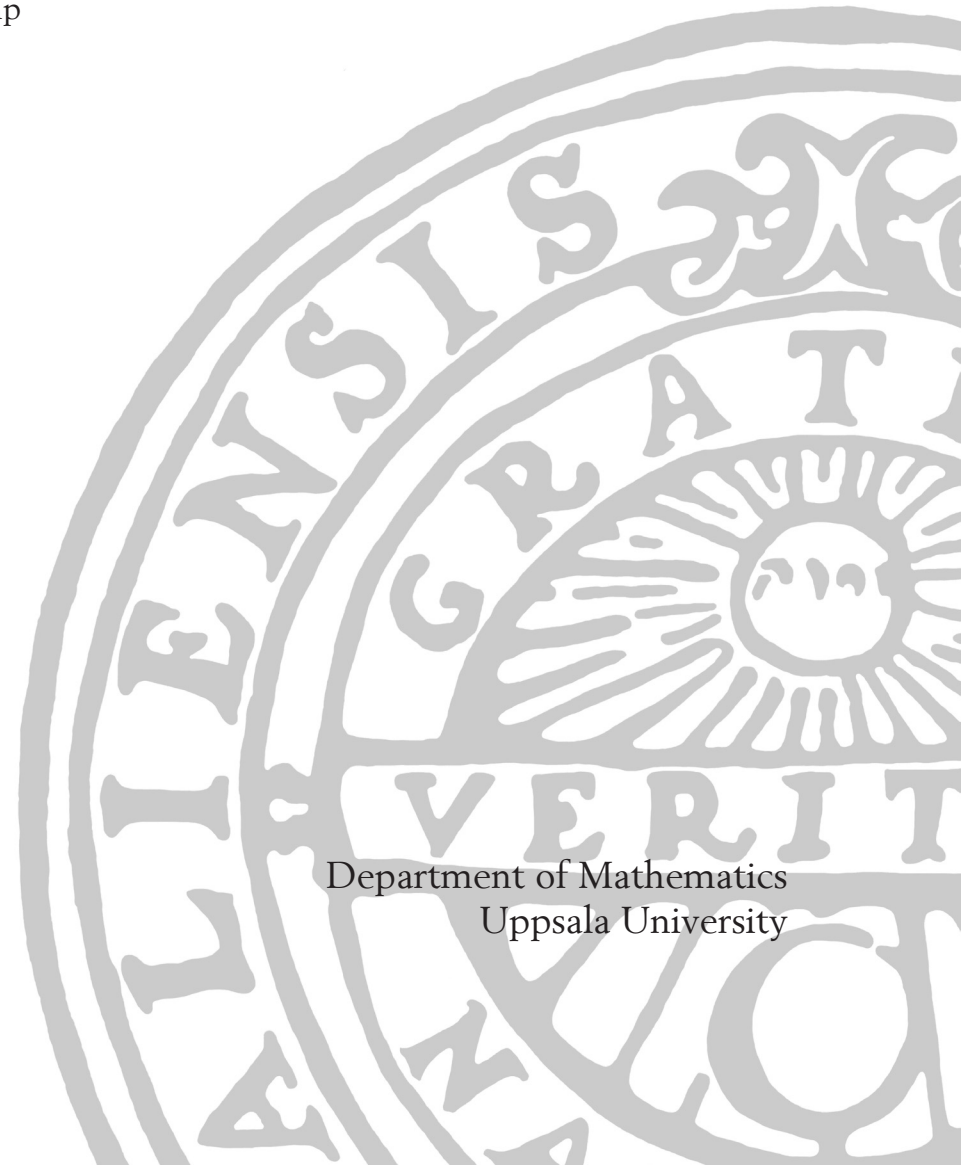
U.U.D.M. Project Report 2013:3

# Dimension Reduction and Adaptivity to Price Basket Options

Paria Ghafari

Examensarbete i matematik, 30 hp
Handledare: Lina von Sydow
Examinator: Anders Öberg

Januari 2013



Department of Mathematics
Uppsala University

# Dimension reduction and adaptivity to price basket options

Paria Ghafari

January 21, 2013

**Acknowledgement**

I would like to thank to my supervisor of this thesis work, Lina von Sydow for the valuable advice and constant supervision. It has been a great experience for me to work with her. Besides, I would like to thank Erik Lehto for his assistance in this study.

**Abstract**

A multi-dimensional Black-Scholes equation is numerically solved in order to price a basket option. These problems suffer from the "curse of dimensionality". The problem is tackled by using a dimension reduction technique, a principal component analysis together with an asymptotic expansion. Moreover, an adaptive technique is used in order to enhance the performance of the method.

In this study, it is shown that using adaptive grid-points instead of equidistant ones together with dimension reduction technique have considerably mitigated the curse of dimensionality.

# Contents

# 1 Introduction

In the fast growing financial markets today, increasing demand to get an accurate price under existing uncertainties is a great deal of matter. Options are essential tools in risk management and particularly in hedging. Following by such demand numerical methods have become imperative tools to price derivatives. This is particularly true for more complex derivatives like basket options which require computer simulations.

In this work an adaptive technique [8] is used to solve the Black-Scholes equation along with using a principal component analysis (PCA) and asymptotic expansion to reduce the number of dimensions [5].

## 1.1 Option Pricing

In 1900 option valuation was addressed for the first time by the French mathematician Louis Bachelier who developed his formula based on the assumption that the stock price follows a Brownian motion with zero drift. Since then, many researchers and scholars have developed various approaches to price options.

In finance, an option is a contract which gives the holder the right but not the obligation to buy *(call)* or sell *(put)* a certain amount of the underlying asset at a predetermined price called the *strike price* $K$ at some time in the future, the *maturity date* $T$. The underlying asset could be a physical asset (gold) or a financial asset (stock).

In this work the focus is on European basket options. *European options* can be exercised only at *maturity date* $T$ while the *American option* has the flexibility of being exercised prior to its *maturity date* $T$.

Even though market participants decide on the value of the option at the end, we need a mathematical model to calculate the value of the option. A widely accepted mathematical model is the Black-Scholes model introduced by Black and Scholes [3] and Merton [10] in 1973. The model will be described in more details later on.

## 1.2 High dimensions and the curse of dimensionality

A basket option is a contract on $d$ ($d \geq 2$) number of underlying assets. The most common example of such contracts are index options where the underlying asset is a stock index like American S&P 500 Index or British FTSE 100. The number of "space" dimensions in the Black-Scholes equation is determined by the number of underlying assets $d$.

Discretizing both time and space in Black-Scholes equation using a finite difference method in all dimensions results in a huge number of discrete points which grows exponentially in the number of dimensions. This problem is called the "curse of dimensionality".

The work presented here will address this problem by using a principal component analysis (PCA) and an asymptotic expansion [5] together with an adaptive method [8]. The outline of the paper is as follows:

In Section 2, the model used for pricing the basket option and the dimension reduction technique are presented. In Section 3, the numerical methods used to solve the problem are given and all the numerical results are presented in Section 4 and discussed in Section 5.

# 2 Theory

## 2.1 The Black-Scholes model

The Black-Scholes equation is a parabolic partial differential equation which was formulated by Fischer Black and Myron Scholes [3] and Merton [10] for the first time in 1973 for pricing the European option.
Assume that the market consists of two assets: a risk free asset with price process $B$ and a stock with process $S$ given by the following dynamics:

$$dB(t) = rB(t)dt, \tag{2.1}$$

$$dS(t) = S(t)\alpha(t, s(t))dt + S(t)\sigma(t, S(t))dW(t), \tag{2.2}$$

where $r \in R$ is the *short rate of interest*, $W$ is a *Wiener process*, $\alpha$ , $\sigma \in R$ are the *local mean of return* and *volatility of* $S$ respectively.
Moreover, a *simple contingent claim* $X$ has the form $X = \phi(S(T))$ where the function $\phi$ is called the *contract function*. In fact, a contingent claim is a contract which provides the holder $X$ *SEK* at maturity time $T$. From the definition we can say that the European call option is a simple contingent claim, for which the contract function is given by,

$$\phi(x) = max[x - K, 0] \tag{2.3}$$

Further, we assume that:

1. The derivative instrument can be traded on a market.
2. The market is *arbitrage free*.
3. The price process of the derivative has the following form:

$$\Pi(t; X) = F(t, S(t)) \tag{2.4}$$

In order to have a market $[\Pi(t), S(t), B(t)]$ free of arbitrage opportunities, $F$ must fulfill the following PDE:

$$F_t(t, s) + rsF_s(t, s) + \frac{1}{2}s^2\sigma^2(t, s)F_{ss}(t, s) - rF(t, s) = 0, \tag{2.5}$$

$$F(T, s) = \phi(s). \tag{2.6}$$

There exist another argument for deriving the pricing equation which is called *Risk Neutral Valuation* model. Assuming a market as described previously and arbitrage free price given by $\Pi(t; \phi) = F(t, S(t))$ where $F$ is the solution of the equations (2.5) and (2.6). Solving the equation by using the *Feynman-Kač* formula gives the following solution:

$$F(t,s) = e^{-r(T-t)}E^{t,s}[\phi(S(T))], \tag{2.7}$$

where, $S$ has the following dynamics:

$$dS(u) = rS(u)du + S(u)\sigma(u, S(u))dW(u), \tag{2.8}$$

$$S(t) = s. \tag{2.9}$$

The important thing to consider here is that in (2.2) $S$ has the local rate of return $\alpha$ while the $S$ process in (2.9) has the short rate of interest $r$ as its local rate of return, see [2] for the thorough derivation of the pricing formula.

## 2.2 The multi-dimensional model problem

In this section a $d$ dimensional Black-Scholes model is presented, where we have a number of underlying risky assets apart from the risk free asset. We assume that we have $d$ number of risky assets ("stocks") with the following price processes shown in the column vector:

$$\mathbf{S}(t) = \begin{pmatrix} S_1(t) \\ . \\ . \\ . \\ S_d(t) \end{pmatrix}, \tag{2.10}$$

where the $S_i$ dynamics are given by:

$$dS_i(t) = \alpha_i S_i(t)dt + S_i(t)\sum_{j=1}^{d} \sigma_i\sigma_j\rho_{ij}d\mathbf{W}_j(t). \tag{2.11}$$

For the same market model in the previous section and the given contingent claim $X = \Phi(\mathbf{S}(T))$, $F$ has to satisfy the following final value problem in order to have a market free of arbitrage opportunities:

$$\begin{cases} \frac{\partial F}{\partial t} + \sum_{i=1}^{d} rs_i\frac{\partial F}{\partial s_i} + \frac{1}{2}\sum_{i,j=1}^{d}\sigma_i\sigma_j\rho_{ij}s_is_j\frac{\partial^2 F}{\partial s_i\partial s_j} - rF = 0, \\ \\ F(T,\mathbf{S}) = \Phi(\mathbf{S}), \end{cases} \tag{2.12}$$

where $\rho_{ij}$ is the correlation between the underlying assets $i$ and $j$. The value of the option at expiry date is

$$\Phi(\mathbf{S}) = \max(\frac{1}{d}\sum_{i=1}^{d} s_i - K, 0), \tag{2.13}$$

The equation (2.13) is also known as the *final condition* of the PDE problem [2].

## 2.3 Principal Component Analysis(PCA)

The idea behind principal component analysis (PCA) is to diminish the dimensionality of a high dimensional data set with interrelated variables in a way that it retains the existing variation of the data set as much as possible. The procedure is obtained by a transformation to a new set of variables which are uncorrelated and yet the first few components keep the highest variation of the full primary data set, see [7] for exhaustive derivation. The rest of this section addresses a brief outline of the derivation of the principal components of the variables (stock prices $\mathbf{S}$) and transformation of the Black-Scholes equation.

Assume $\mathbf{S}$ represents the original set of variables and $\mathbf{x}$ is the transformed set obtained by PCA. The first step is to find a linear function $\bar{q}_1^T\mathbf{S}$ of the primary data set $\mathbf{S}$ with maximum variance:

$$x_1 = \bar{q}_1^T\mathbf{S} = q_{11}S_1 + q_{12}S_2 + \ldots + q_{1d}S_d = \sum_{j=1}^{d} q_{1j}S_j, \tag{2.14}$$

where $\bar{q}_1$ is a vector of $d$ elements $q_{11}, q_{12}, \ldots, q_{1d}$.

The next step is to find $\bar{q}_2^T\mathbf{S}$ having maximum variance and uncorrelated from $\bar{q}_1^T\mathbf{S}$. The same procedure shall be continued until the $d$th linear function $\bar{q}_d^T\mathbf{S}$ which has the maximum variance and is uncorrelated from $\bar{q}_1^T\mathbf{S}, \bar{q}_2^T\mathbf{S}, \ldots, \bar{q}_{d-1}^T\mathbf{S}$. The $i$th variable obtained by PCA is called $i$th PC (principal component). In order to derive the PCs, consider $\bar{q}_1^T\mathbf{S}$ where the vector $\bar{q}_1$ maximizes

$$\mathrm{var}[\bar{q}_1^T\mathbf{S}] = \bar{q}_1^T\Sigma\bar{q}_1 \tag{2.15}$$

where $\Sigma$ is the covariance matrix of the original variables. A normalization constraint is required to obtain a finite $\bar{q}_1$. The constraint used is $\bar{q}_1^T\bar{q}_1 = 1$. The standard procedure to solve the maximization problem $\bar{q}_1^T\Sigma\bar{q}_1$ subject to $\bar{q}_1^T\bar{q}_1 = 1$ is applying the method of Lagrange multipliers. The solution of the maximization problem is:

$$\Sigma\bar{q}_1 = \lambda\bar{q}_1 \tag{2.16}$$

where $\lambda$ is an eigenvalue of $\Sigma$ and $\bar{q}_1$ is the corresponding eigenvector. Since $\bar{q}_1^T\Sigma\bar{q}_1 = \lambda$ is the quantity to be maximized, $\bar{q}_1$ is the eigenvector corresponding to the largest eigenvalue $\lambda$ of $\Sigma$. The vector $\bar{q}_1$ is called the principal axis. The second PC of $\mathbf{S}$ is derived in the same manner with an additional constraint which makes sure that $\bar{q}_1^T\mathbf{S}$ and $\bar{q}_2^T\mathbf{S}$ are uncorrelated or equivalently $\mathrm{cov}[\bar{q}_1^T\mathbf{S}, \bar{q}_2^T\mathbf{S}] = 0$, where $\bar{q}_2$ is corresponding to the eigenvector with second largest eigenvalue $\lambda_2$. Consequently, for the third, fourth, ... and $d$th PCs, $\lambda_3$, $\lambda_4$, ..., $\lambda_d$ are the third largest, fourth largest, ..., and

smallest eigenvalues of $\Sigma$ and $\bar{q}_3$, $\bar{q}_4$, ..., $\bar{q}_d$ the corresponding eigenvectors. The new set of variables obtained by PCA can be written as follows:

$$\mathbf{x} = \mathbf{Q}^T \mathbf{S}. \tag{2.17}$$

Using transformation (2.17) for Black-Scholes equation would result in an unwieldy equation. Alternatively, a change to logarithmic coordinates $\ln \mathbf{S}$ along with a time axis reversion $\tau \to T - t$, a translation to exclude the drift and application of the eigenvectors $\mathbf{Q}$ yields:

$$\mathbf{x} = \mathbf{Q}^T ln\mathbf{S} + \bar{\mathbf{b}}\tau \tag{2.18}$$

where, $b_i = \sum_{j=1}^{d} q_{ji}(r - \frac{\sigma_j^2}{2})$. Implementing the changes to the Black-Scholes equation (2.12) and final condition (2.13) gives us:

$$\frac{\partial F}{\partial t} - \frac{1}{2}\sum_{i=1}^{d} \lambda_i \frac{\partial^2 F}{\partial x_i^2} + rF = 0, \qquad (\mathbf{x}, t) \in \mathbb{R}^d \times (0, T), \tag{2.19}$$

$$F(0, \mathbf{x}) = \max(\sum_{i=1}^{d} \mu_i e^{\sum_{j=1}^{d} q_{ji} x_j} - K, 0), \qquad \mathbf{x} \in \mathbb{R}^d. \tag{2.20}$$

where $\lambda_i$ are the eigenvalues of the covariance matrix, see [5]. A trivial way to use the PCA would be to truncate the summation over $d$ in (2.19). However, even for highly correlated underlying assets this gives fairly large errors.

## 2.4 Asymptotic expansion

Previous studies [5] and [11] have shown that disregarding the variables with small variation in PCA give large errors. The actual dimension reduction process is completed by an asymptotic expansion where we approximate each of the non-principal dimensions by a linear asymptotic expansion. Following [5] the asymptotic expansion is given by

$$F = F^{(1)} + \sum_{i=2}^{d} \lambda_i \left.\frac{\partial F}{\partial \lambda_i}\right|_{\bar{\lambda}=\bar{\lambda}^{(1)}} + \mathcal{O}(||\bar{\lambda} - \bar{\lambda}^{(1)}||^2) \tag{2.21}$$

where $F^{(1)}$ is the solution to the one-dimensional problem in the principal direction (corresponding to the largest eigenvalue), $\bar{\lambda}$ is a parameter vector of eigenvalues and $\bar{\lambda}^{(1)} = (\lambda_1, 0, \ldots, 0)$ is the parameter vector of the truncated equation to one dimension. The derivatives in (2.21) can be approximated by a finite difference method

$$\left.\frac{\partial F}{\partial \lambda_i}\right|_{\bar{\lambda}=\bar{\lambda}^{(1)}} = \frac{F^{(1,i)} - F^{(1)}}{\lambda_i} + \mathcal{O}(\lambda_i^2) \tag{2.22}$$

where $F^{(1,i)}$ is the solution to the two-dimensional problem on the plane spanned by the principal axis and the variable $i$ corresponding the $i$th largest eigenvalue. Thus, the $d$-dimensional problem is broken down to a one-dimensional and $(d-1)$ two-dimensional problems. First we solve the one-dimensional problem $F^{(1)}$ with a finite difference method using an adaptive technique and then the correction part is added in order to obtain a more accurate solution. The correction part is obtained by solving the two-dimensional problems using adaptive finite differences giving $F^{(1,2)}, F^{(1,3)}, \ldots, F^{(1,d)}$ which are used in (2.21) and (2.22).

# 3 Numerical Methods

## 3.1 Discretization and the finite difference method implementation

By defining the spatial difference operator $\mathcal{L}$ as

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{d} \lambda_i \frac{\partial^2}{\partial x_i^2} - r, \tag{3.1}$$

the partial differential equation (2.19) can be written as:

$$\frac{\partial F}{\partial t} = \mathcal{L}F \tag{3.2}$$

We apply a semi-discretization in space by using second-order finite differences (FD) on a structured but non-equidistant grid.
The number of grid points in dimension $i$ is $N_i$, $i = 1, \ldots, d$. Assuming that $F_h$ is a vector of lexicographically ordered unknowns gives

$$\frac{dF_h}{dt} = A_h F_h, \tag{3.3}$$

where $A_h$ is a very large, sparse matrix with the FD discretization of $\mathcal{L}$.
The second derivative in the direction $i$ is approximated as in [8] and [6]:

$$\frac{\partial^2 F(x_{ik})}{\partial x_k^2} \approx a_{ik}F(x_{i(k+1)}) + b_{ik}F(x_{ik}) + c_{ik}F(x_{i(k-1)}), \tag{3.4}$$

where

$$a_{ik} = \frac{2}{h_{ik}(h_{i,k-1}+h_{ik})}, \qquad b_{ik} = \frac{-2}{h_{i,k-1}h_{ik}}, \qquad c_{ik} = \frac{2}{h_{i,k-1}(h_{i,k-1}+h_{ik})}.$$

The approximation in (3.4) is second-order in space if there is a smooth variation of the grid such that $h_{i,k-1} = h_{i,k}(1 + \mathcal{O}(h_{i,k}))$. Since we are dealing with $(d-1)$ two-dimensional problems, we only have a principal direction ($i = 1$) and non principal direction ($i = d$). For instance, if we

have 5 underlying stocks, $d = 5$, then we have one one-dimensional problem and 4 distinct two-dimensional problems to solve.

In time the *Backward differential formula* BDF-2 scheme is applied which is unconditionally stable and have second order accuracy. BDF-2 uses two previous time-steps which force us to use another scheme to complete the initial time-step. A typical choice to perform the first time-step is the Backward *Euler method*.
The BDF-2 scheme is as follow:

$$a_0 F^n = \Delta t \mathcal{L}(F^n) - a_1 F^{n-1} - a_2 F^{n-2}, \tag{3.5}$$

where $a_0 = 1.5$, $a_1 = -2$ and $a_2 = -0.5$.

In the one-dimensional case $\mathcal{L}(F^n)$ is:

$$\mathcal{L}(F^n) = \frac{1}{2}\lambda_1 \frac{\partial^2 F}{\partial x_1^2} - rF^n, \tag{3.6}$$

and in the two-dimensional case we have:

$$\mathcal{L}(F^n) = \frac{1}{2}\lambda_1 \frac{\partial^2 F}{\partial x_1^2} + \frac{1}{2}\lambda_j \frac{\partial^2 F}{\partial x_j^2} - rF^n. \tag{3.7}$$

Implementing the second order derivative approximations (3.4) in the one-dimensional and two-dimensional problem will give us (some notations are changed for simplicity):

- One-dimensional problem:

$$\mathcal{L}(F_k^n) = \frac{1}{2}\lambda_1 (a_{1k} F_{k+1}^n + b_{1k} F_k^n + c_{1k} F_{k-1}^n) - rF_k^n \tag{3.8}$$

- Two-dimensional problem:

$$\begin{aligned}\mathcal{L}(F_k^n) = &\frac{1}{2}\lambda_1 (a_{1k} F_{k+1}^n + b_{1k} F_k^n + c_{1k} F_{k-1}^n) + \\ &\frac{1}{2}\lambda_d (a_{dk} F_{k+N_p}^n + b_{dk} F_k^n + c_{dk} F_{k-N_p}^n) - rF_k^n\end{aligned} \tag{3.9}$$

where $N_p$ is the number of grid points in the principal axis.

## 3.2 Space adaptive FD-method

To obtain high accuracy by controling the local discretization error of the numerical solution of a PDE, the adaptive method places the discretization nodes where they are most needed. As a result, it helps avoiding dense grids, since it is not needed for regions where the solution is smooth. Some previous works using adaptive methods to solve option valuation problem can be found in [1] and [9], where a mesh adaptation finite element method is used for discretizing the Black-Scholes equation for a European put option and American call option respectively. In another study, [4], an adaptive Radial Basis Function is used for pricing the European and American options. In [8], a space-time adaptive method is used to solve the multidimensional Black-Scholes equation for a European call option. The results show a considerable improvement in accuracy in comparison to equidistant grids. However, the "curse of dimensionality" is still an issue to be solved. In this work, the attempt is to address the "curse of dimensionality" by applying a PCA and an asymptotic expansion together with the adaptive method in [8]. Instead of implementing the adaptive technique to the original multidimensional Black-Scholes PDE (2.12), the transformed PDE (2.19) in combination with (2.21) in section 2.3 will be used.

The summary of the adaptive algorithm is the same as in [8]:

1. Solve the PDE once with a coarse grid giving low accuracy.
2. Create a new spatial grid aiming to capture the required accuracy.
3. Solve the PDE with the new grid-points.

In other words, the problem at hand should be solved twice; in the first round, the problem will be solved quickly with lower accuracy. Then, grid-points will be placed in desired regions based on estimates of the local truncation errors in the solution we have obtained in the first solve.

### 3.2.1 One-dimensional problem

To begin with, we address the one-dimensional problem in the principal axis and determine how the space adaptivity is accomplished. We begin with looking at the local discretization error $\tau_h$. Let's assume that the following relation holds for any smooth solution $F(x)$

$$A_h F_h = AF + \tau_h \tag{3.10}$$

where $F_h$ is the vector of unknowns, $A_h$ is the discrete approximation of the spatial operator $\mathcal{L}$, $AF$ is the exact operator $\mathcal{L}F$, evaluated in the grid

points. The local truncation error $\tau_h$ can be approximated with the leading term

$$\tau_h = h^p \eta(x) + \mathcal{O}(\text{h.o. terms}) \tag{3.11}$$

and for the second order discretization in space we have $p = 2$. We define $\delta_h$ and $\delta_{2h}$:

$$\begin{aligned} \delta_h &= A_h F_h = AF + \tau_h \\ \delta_{2h} &= A_{2h} F_{2h} = AF + \tau_{2h} \end{aligned} \tag{3.12}$$

and using the equations in (3.11) and (3.12) we can get the local discretization error $\tau_h$

$$\tau_h = \frac{\delta_{2h} - \delta_h}{3}. \tag{3.13}$$

Since the adaptive method is aiming at distributing the grid points in the most efficient way, we set the local discretization error $\tau_h$ smaller than some tolerance $\epsilon > 0$

$$|\tau(x)| = |h^2(x)\eta(x)| \leqslant \epsilon. \tag{3.14}$$

To approximate $\eta(x)$, we compute a solution by using the step-length $\bar{h}$. From (3.11) we have

$$\eta(x) = \frac{\tau_{\bar{h}}(x)}{\bar{h}^2(x)} \tag{3.15}$$

and

$$|\tau(x)| = |h^2(x)\eta(x)| \approx |h^2(x)\frac{\tau_{\bar{h}}(x)}{\bar{h}^2(x)}| \leqslant \epsilon, \tag{3.16}$$

where $\tau_{\bar{h}}(x)$ is estimated using (3.13). By choosing the step-length $h(x)$ for the adaptive grid

$$h(x) = \sqrt{\frac{\epsilon \bar{h}^2(x)}{\tau_{\bar{h}}(x)}} \tag{3.17}$$

the desired accuracy is obtained. Since our problem is time-dependent the local discretization errors will depend not only on $x$, but also on time. Since we for simplicity want to have the same spatial grid throughout our computations we have to incorporate this in our algorithm. We do as follow:

**coarse grid solve**

- for $n = 1$ to number of time steps
  - Solve for $F^n$ on coarse equidistant grid $\bar{h}$.
  - If time equals $[T/3 \quad 2T/3 \quad T]$, save this solution.
- end for
- Compute $\tau_{\bar{h}}^{(\frac{T}{3})}(x), \tau_{\bar{h}}^{(\frac{2T}{3})}(x), \tau_{\bar{h}}^{(T)}(x)$

- Compute $\tau_{\bar{h}}(x) = \max(|\tau_{\bar{h}}^{(\frac{T}{3})}(x)|, |\tau_{\bar{h}}^{(\frac{2T}{3})}(x)|, |\tau_{\bar{h}}^{(T)}(x)|)$
- Compute new $h(x)$ as in (3.17)

Note that in order to avoid the non-smooth behaviour of the approximated local discetization error $\tau_{\bar{h}}(x)$, a weighted mean-value filter is implemented. The value of $\tau_{\bar{h}}(x, j)$ in point $j$ is

$$\tau_{\bar{h}}(x_j) = (\tau_{\bar{h}}(x_{j-1}) + 2\tau_{\bar{h}}(x_j) + \tau_{\bar{h}}(x_{j+1})/4. \tag{3.18}$$

We have used this filter for a different number of repetition times $rep = 5, 10, 15$ and 10 repetitions gives the most desirable results.

Yet, if the estimate of $\tau_{\bar{h}}$ is very small, there is a risk of using too large new $h(x)$. Therefore, we need an extra parameter $\gamma$ as:

$$h(x) = \bar{h}(x)\sqrt{\frac{\epsilon}{\epsilon\gamma + |\tau_{\bar{h}}(x)|}} \tag{3.19}$$

For the calculations $\gamma = 0.01$ is used.

We also introduced a new filtration over $\tau_{\bar{h}}(x)$ and we are expecting to get a more dense allocation of adaptive grid points around $x^0$ which is the transformed value of todays's stock price. The new $\tau_{\bar{h}}^{\star}(x)$ is the Gaussian function multiplied by $\tau_{\bar{h}}(x)$,

$$\tau_{\bar{h}}^{\star}(x) = \tau_{\bar{h}}(x)e^{-\alpha(x-x^0)^2}. \tag{3.20}$$

Note that by using this new $\tau_{\bar{h}}^{\star}(x)$ we can no longer expect to satisfy the restriction on the local discretization error (3.16). The algorithm coarse grid solve will be modified to:

**coarse grid solve using $\tau_{\bar{h}}^{\star}(x)$**

- for $n = 1$ to number of time steps
  - Solve for $F^n$ on coarse equidistant grid $\bar{h}$
  - If time equals $[T/3 \quad 2T/3 \quad T]$, save this solution.
- Compute $\tau_{\bar{h}}^{(\frac{T}{3})}(x), \tau_{\bar{h}}^{(\frac{2T}{3})}(x), \tau_{\bar{h}}^{(T)}(x)$
- Compute $\tau_{\bar{h}}(x) = \max(\tau_{\bar{h}}^{(\frac{T}{3})}(x), \tau_{\bar{h}}^{(\frac{2T}{3})}(x), \tau_{\bar{h}}^{(T)}(x))$
- Compute $\tau_{\bar{h}}^{\star}(x)$ as in (3.20) Apply smoothing filtration (3.18) over $\tau_{\bar{h}}^{\star}(x)$ Compute new $h(x)$ as in (3.19)

The calculations for both algorithms are presented in section 4.

### 3.2.2 Two-dimensional problem

Now, we consider a two-dimensional problem. The theoretic derivation of the discretization error $\tau_{h_1,h_j}$ gives

$$\begin{aligned}\tau_{h_1,h_j} &= h_1^2(\eta_1(x_1,x_j)) + h_j^2(\eta_j(x_1,x_j)) + \mathcal{O}(\text{h.o. terms}) \\ &\approx \tau_{h_1} + \tau_{h_j}. \end{aligned} \tag{3.21}$$

To approximate $\eta_1(x_1,x_j)$ and $\eta_j(x_1,x_j)$ the same procedure as in the one-dimensional case can be followed by first using $\bar{h}_1$ and $2\bar{h}_1$ to estimate $\eta_1(x_1,x_j)$ and similarly for $\eta_j(x_1,x_j)$. Since we already have computed the new step-length $h(x)$ in the principal axis $x_1$ we are only interested in computing a new grid in the non-principal axes which we denote by $x_j$,$j = 2,\ldots,d$. We are interested in a one-dimensional variable $\tau_{h_j}$ in $x_j$ which can be accomplished by taking the maximum of $\tau_{h_j}$ over $x_1$.

# 4 Numerical Results

In this section, we investigate the performance of the dimension reduction technique together with the adaptive method in various tests. The tests are performed for one- and five-dimensional problems. The contract function is the European basket option (2.13). The implementation of the algorithm is done in MATLAB. Our results are compared to the results obtained by [5].

## 4.1 Numerical Results for One-dimensional Problem

The computational domain is set to $S \in [0.05K, 4K]$. Transforming it gives us $x_1 \in [\mathbf{Q}^T \ln 0.05K + \bar{\mathbf{b}}\tau, \mathbf{Q}^T \ln 4K + \bar{\mathbf{b}}\tau]$. The boundary conditions used are:

$$F = 0, \qquad x_1 = \mathbf{Q}^T \ln 0.05K + \bar{\mathbf{b}}\tau \tag{4.1}$$

$$F = \mu e^{\sum_{j=1}^{n} q_{1j}(x_j - b_j\tau)} - Ke^{-r\tau}, \qquad x_1 = \mathbf{Q}^T \ln 4K + \bar{\mathbf{b}}\tau \tag{4.2}$$

Parameters for the one-dimensional problems are chosen as presented in Table 1. Results for the one-dimensional problem are compared with the exact solution of the European call option given by

$$F(t,s) = s\mathcal{N}(d_1(t,s)) - Ke^{-rt}\mathcal{N}(d_2(t,s)), \tag{4.3}$$

where

$$d_1(t,s) = \frac{\ln(s/K) + (r + \frac{1}{2}\sigma^2)t}{\sigma\sqrt{t}}, \tag{4.4}$$

$$d_2(t,s) = d_1(t,s) - \sigma\sqrt{t}. \tag{4.5}$$

| $r$ | $T$ | $K$ | $\mu$ | $\bar{\sigma}$ | $\rho_{ij}$ |
|---|---|---|---|---|---|
| 0.05 | 1.0 | 25 | 1 | 0.25 | 1 |

Table 1: Parameters for one-dimensional problem

where $\mathcal{N}$ is the standard Gaussian cumulative distribution function [3].

The experiments are done for both equidistantly distributed and adaptive grid-points. In order to calculate the error the exact solution given by (4.3) is used. The space tolerance levels $5 \times 10^{-3}$, $10^{-3}$, $5 \times 10^{-4}$, $10^{-4}$, $5 \times 10^{-5}$ and $10^{-5}$ have been studied. Moreover, several stock prices $S_0 =$ 20, 24, 26 and 30 are tested with respect to their distance from the strike price $K = 25$. It is interesting to investigate how the error varies with respect to the position of the grid-points in the neighbourhood of the bending point, $s_1 - K = 0$, in the principal direction; thus, the ratios $1/2 - 1/2$, $1/3 - 2/3$, $2/3 - 1/3$, $1/4 - 3/4$ and $3/4 - 1/4$ are selected, see figure 1.

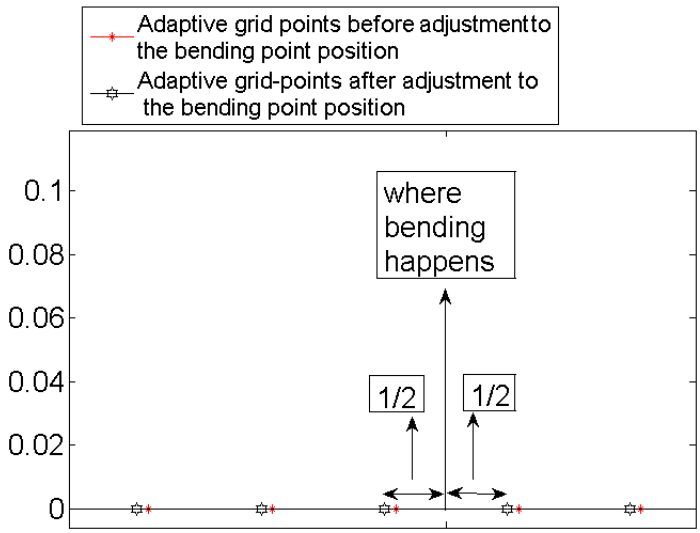



Figure 1

### 4.1.1 Adaptive

Comparisons of the algorithms coarse grid solve with and without using $\tau_{\bar{h}}^{\star}(x)$ where $\alpha = 0, 5, 10, 20, 30$ has been studied. Note that $\alpha = 0$ is the algorithm without using $\tau_{\bar{h}}^{\star}(x)$. Applying $\tau_{\bar{h}}^{\star}(x)$ is not leading us to a decisive conclusion. Depending on the choice of stock price and the ratios, they behave differently. As a result, we continue our calculation using the original algorithm, i.e. $\alpha = 0$, see section (3.2).

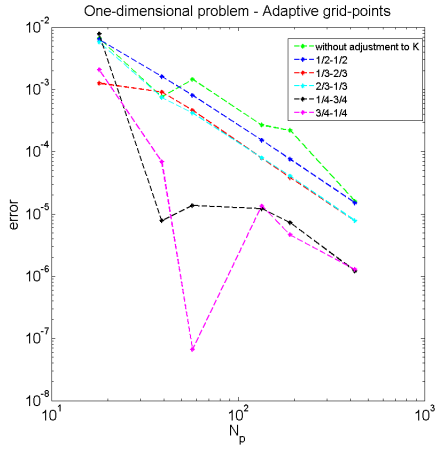



Figure 2: Error versus number of grid-points in the principal direction for $S_0 = 20$

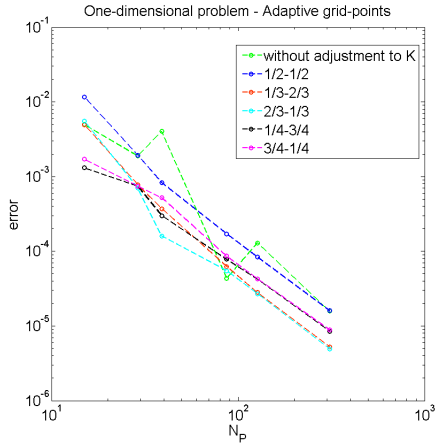



Figure 3: Error versus number of grid-points in the principal direction for $S_0 = 30$

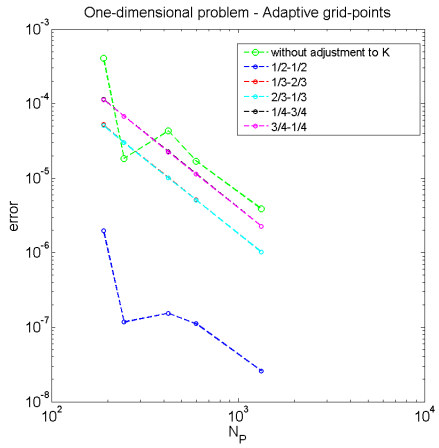



Figure 4: Error versus number of grid-points in the principal direction for $S_0 = 24$

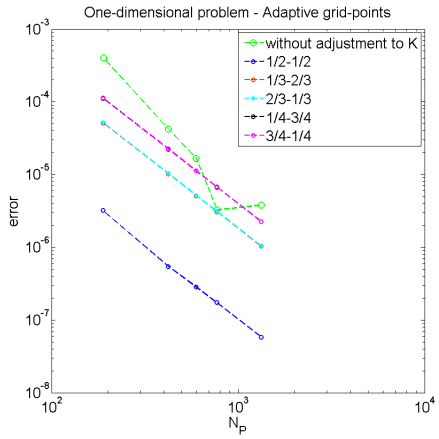



Figure 5: Error versus number of grid-points in the principal direction for $S_0 = 26$

In Figures 2 and 3 where the stock prices 20 and 30 have a rather large distance from the strike price 25, the ratio $1/4 - 3/4$ shows smaller error in comparison to the other ratios and the experiment without making adjustment to the strike price $K$. While in the Figures 4 and 5 the experiment with ratio $1/2 - 1/2$ shows better results. Note that, in Figures 4 and 5 the stock prices are chosen close to the strike price.

### 4.1.2 Equidistant

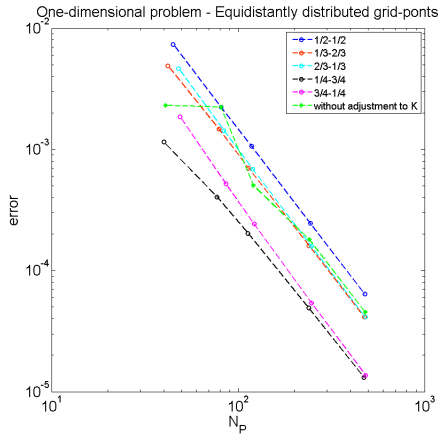



Figure 6: Error versus number of grid-points in the principal direction for $S_0 = 20$

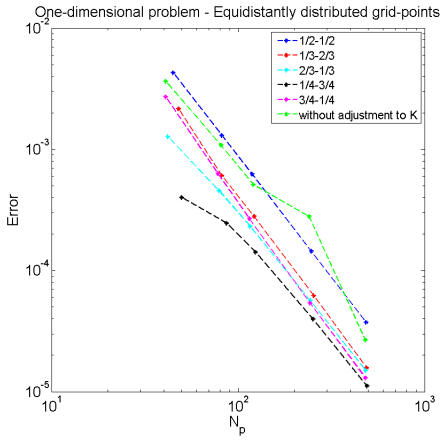



Figure 7: Error versus number of grid-points in the principal direction for $S_0 = 30$

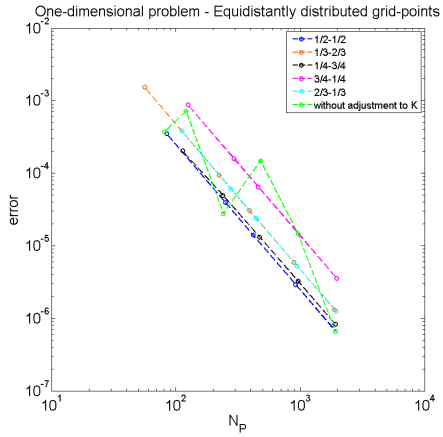



Figure 8: Error versus number of grid-points in the principal direction for $S_0 = 24$

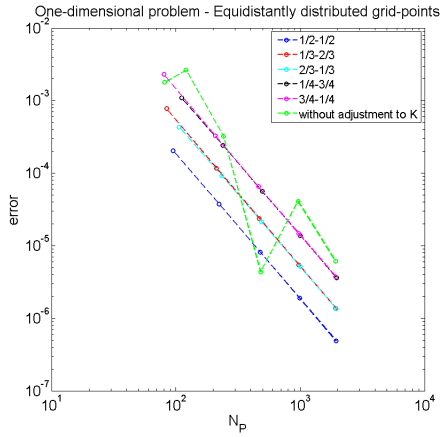



Figure 9: Error versus number of grid-points in the principal direction for $S_0 = 26$

Using the same assumption during the calculations using equidistant grid-points, a similar pattern is observed. In the Figures 6 and 7 better results are obtained using the ratio $1/4 - 3/4$ and for the Figures 8 and 9 the ratio $1/2 - 1/2$ has smaller error. The same pattern was observed in the adaptive results.

Comparison between the results of the dimension reduction technique using adaptive and equidistant grid-points is shown in figure 10.

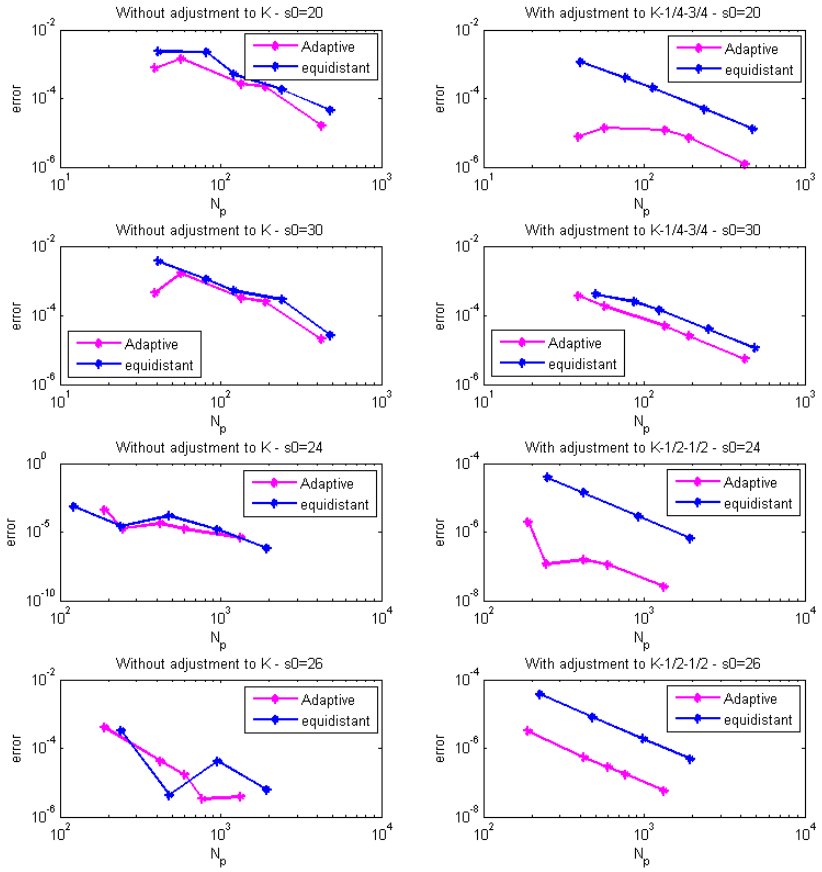



Figure 10: One-dimensional problem - comparisons between adaptive and equidistant grid-points

From Figure 10, we can see that the dimension reduction technique with adaptivity technique gives a considerable improvement in the results.

## 4.2 Numerical Results for Five-dimensional Problem

In this section a highly correlated basket option has been studied. For the five-dimensional problem we use the results from [5] as reference solution with a very fine grids where 1001 grid-points are used in all space dimensions. Results are shown in Table 2.
The Monte Carlo solution presented in table 2 is the result of $n = 10^8$

| Method | F |
|---|---|
| 1D PCA-Equidistant | 36.4158 |
| Asymptotic Expansion PCA-Equidistant | 40.9383 |
| Monte Carlo | 41.0243 |

Table 2: $N_p = 2001$ and $N_d = 1001$ using PCA-Equidistant

simulations. The asymptotic expansion PCA method using equidistant or adaptive grid-points have some similarities to Monte Carlo method. In contrast to the standard finite difference method which gives the solution for all space- and time steps, in this method we only calculate the solution for one price $S_0$. This is also true for Monte Carlo method where we only get the option price for $S_0$.

Parameters for five-dimensional problems are chosen as presented in Table 3. $\alpha_p = 0$ in the principal axis and $\alpha_d = 0$, 0.5, 1, 2, 5 in the other axes are studied for the algorithm coarse grid solve using $\tau_h^\star(x)$. Since using $\alpha_p = 0$ and $\alpha_d = 0$ resulted in too many grid-points in all space dimensions, we will not consider it in our comparisons. The computational domain was first set as in the one-dimensional problem with only the change at the far-field dirichlet boundary, where we truncate the domain by multiplying the $4 \times K$ by the number of dimensions in order to have the far-field boundary at four times from the location of the discontinuity of the derivative of the initial function $\phi$ in all directions [8]. The computational domain chosen in this way was very small in the non-principal axes, so we chose the same boundaries as in [5] which is proved to be working well

$$F = 0, \qquad x_1 = 0,$$

$$F = \sum_{i=1}^{d} \mu_i e^{\sum_{j=1}^{n} q_{ij}(x_j - b_j\tau)} - Ke^{-r\tau}, \qquad x_1 = 2x_1^0 \tag{4.6}$$

$$\frac{\partial^2 F}{\partial x_d^2} = 0, \qquad x_d = x_d^0 - 3, x_d^0 + 3 \tag{4.7}$$

| $r$ | $T$ | $K$ | $\bar{S}^0$ | $\mu$ | $\bar{\sigma}$ | $\rho_{ij}$ |
|---|---|---|---|---|---|---|
| 0.05 | 2.0 | 125 | $\begin{pmatrix}25\\25\\25\\25\\25\end{pmatrix}$ | $\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}$ | $\begin{pmatrix}0.518\\0.648\\0.623\\0.570\\0.530\end{pmatrix}$ | $\begin{pmatrix}1 & 0.79 & 0.82 & 0.91 & 0.84\\0.79 & 1 & 0.73 & 0.80 & 0.76\\0.82 & 0.73 & 1 & 0.77 & 0.72\\0.91 & 0.80 & 0.77 & 1 & 0.90\\0.84 & 0.76 & 0.72 & 0.90 & 1\end{pmatrix}$ |

Table 3: Parameters for five-dimensional problem

In the adaptive experiments space tolerances $9\times10^{-3}$, $8\times10^{-3}$, $7\times10^{-3}$, $6\times10^{-3}$, $5\times10^{-3}$, $4\times10^{-3}$, $3\times10^{-3}$, $2\times10^{-3}$, $10^{-3}$, $9\times10^{-4}$ $8\times10^{-4}$, $7\times10^{-4}$, $7\times10^{-4}$, $6\times10^{-4}$, $5\times10^{-4}$ and $4\times10^{-4}$ are studied.

An error comparison between adaptive and equidistant PCA without changing the position of the grid-points with respect to $x_1^0$ is presented in Table 4, 5, 6 and 7. It is also shown that the error changes with respect to the way the $x_j^0$ is inserted for $\alpha_p = 0$ and $\alpha_d = 0.5$, 1, 2, 5. Note that equidistant grid-points are chosen simply by dividing the computational domain to the number of grid points in all space dimensions.

| space tolerance | $N_p$ | $N_{d_2}$ | $N_{d_3}$ | $N_{d_4}$ | $N_{d_5}$ | Adaptive (pushed grid-points) | Adaptive ($x^0$ manually inserted) | Equidistant |
|---|---|---|---|---|---|---|---|---|
| $9 \times 10^{-3}$ | 61 | 58 | 51 | 49 | 50 | 0.0484 | 0.0252 | 0.0044 |
| $8 \times 10^{-3}$ | 65 | 61 | 54 | 52 | 53 | 0.0290 | 0.0086 | 0.0211 |
| $7 \times 10^{-3}$ | 69 | 65 | 57 | 55 | 57 | 0.0112 | 0.0138 | 0.0330 |
| $6 \times 10^{-3}$ | 75 | 71 | 62 | 59 | 61 | 0.0009 | 0.0086 | 0.0397 |
| $5 \times 10^{-3}$ | 82 | 77 | 68 | 65 | 67 | 0.0230 | 0.0008 | 0.0046 |
| $4 \times 10^{-3}$ | 91 | 86 | 76 | 73 | 75 | 0.0146 | 0.0098 | 0.0093 |
| $3 \times 10^{-3}$ | 105 | 100 | 87 | 84 | 86 | 0.0046 | 0.0041 | 0.0029 |
| $2 \times 10^{-3}$ | 128 | 122 | 106 | 102 | 105 | 0.0001 | 0.0081 | 0.0071 |
| $10^{-3}$ | 181 | 172 | 150 | 144 | 148 | 0.0017 | 0.0021 | 0.0018 |
| $9 \times 10^{-4}$ | 191 | 181 | 158 | 152 | 156 | 0.0079 | 0.0008 | 0.0001 |
| $8 \times 10^{-4}$ | 202 | 192 | 168 | 161 | 166 | 0.0018 | 0.0028 | 0.0075 |
| $7 \times 10^{-4}$ | 216 | 205 | 179 | 172 | 177 | 0.0057 | 0.0012 | 0.0002 |
| $6 \times 10^{-4}$ | 233 | 222 | 194 | 186 | 191 | 0.0005 | 0.0010 | 0.0004 |
| $5 \times 10^{-4}$ | 255 | 243 | 212 | 204 | 209 | 0.0003 | 0.0017 | 0.0029 |
| $4 \times 10^{-4}$ | 285 | 272 | 237 | 228 | 234 | 0.0023 | 0.0001 | 0.0003 |

Table 4: Error comparison, $\alpha_p = 0$, $\alpha_d = 0.5$, $N_p$ is the number of grid-points in the principal direction and $N_{d_j}$, $j = 2 \cdots 5$, is the number of grid-points in other space dimensions

| space tolerance | $N_p$ | $N_{d_2}$ | $N_{d_3}$ | $N_{d_4}$ | $N_{d_5}$ | Adaptive (pushed grid-points) | Adaptive ($x^0$ manually inserted) | Equidistant |
|---|---|---|---|---|---|---|---|---|
| $9 \times 10^{-3}$ | 61 | 42 | 39 | 37 | 38 | 0.0465 | 0.0236 | 0.0088 |
| $8 \times 10^{-3}$ | 65 | 44 | 41 | 39 | 40 | 0.0268 | 0.0073 | 0.0246 |
| $7 \times 10^{-3}$ | 69 | 47 | 44 | 41 | 42 | 0.0095 | 0.0148 | 0.0354 |
| $6 \times 10^{-3}$ | 75 | 51 | 48 | 45 | 46 | 0.0002 | 0.0075 | 0.0418 |
| $5 \times 10^{-3}$ | 82 | 55 | 52 | 49 | 50 | 0.0223 | 0.0001 | 0.0007 |
| $4 \times 10^{-3}$ | 91 | 62 | 58 | 55 | 56 | 0.0141 | 0.0091 | 0.0111 |
| $3 \times 10^{-3}$ | 105 | 71 | 67 | 63 | 64 | 0.0052 | 0.0046 | 0.0041 |
| $2 \times 10^{-3}$ | 128 | 87 | 82 | 77 | 79 | 0.0003 | 0.0084 | 0.0199 |
| $10^{-3}$ | 181 | 123 | 115 | 108 | 111 | 0.0019 | 0.0020 | 0.0023 |
| $9 \times 10^{-4}$ | 191 | 129 | 121 | 114 | 117 | 0.0078 | 0.0007 | 0.0004 |
| $8 \times 10^{-4}$ | 202 | 137 | 128 | 120 | 124 | 0.0017 | 0.0026 | 0.0079 |
| $7 \times 10^{-4}$ | 216 | 146 | 137 | 129 | 132 | 0.0056 | 0.0013 | 0.0003 |
| $6 \times 10^{-4}$ | 233 | 158 | 148 | 139 | 142 | 0.0004 | 0.0009 | 0.0005 |
| $5 \times 10^{-4}$ | 255 | 173 | 162 | 152 | 156 | 0.0003 | 0.0015 | 0.0030 |
| $4 \times 10^{-4}$ | 285 | 193 | 181 | 170 | 174 | 0.0022 | 0.0001 | 0.00008 |

Table 5: Error comparison$\alpha_p = 0$, $\alpha_d = 1$, $N_p$ is the number of grid-points in the principal direction and $N_{d_j}$, $j = 2 \cdots 5$, is the number of grid-points in other space dimensions

| space tolerance | $N_p$ | $N_{d_2}$ | $N_{d_3}$ | $N_{d_4}$ | $N_{d_5}$ | Adaptive (pushed grid-points) | Adaptive ($x^0$ manually inserted) | Equidistant |
|---|---|---|---|---|---|---|---|---|
| $9 \times 10^{-3}$ | 61 | 31 | 30 | 29 | 30 | 0.0411 | 0.0198 | 0.0140 |
| $8 \times 10^{-3}$ | 65 | 33 | 32 | 31 | 31 | 0.0235 | 0.0044 | 0.0286 |
| $7 \times 10^{-3}$ | 69 | 35 | 34 | 33 | 34 | 0.0068 | 0.0171 | 0.0387 |
| $6 \times 10^{-3}$ | 75 | 38 | 37 | 35 | 36 | 0.0028 | 0.0061 | 0.0452 |
| $5 \times 10^{-3}$ | 82 | 42 | 41 | 39 | 40 | 0.0202 | 0.0019 | 0.0029 |
| $4 \times 10^{-3}$ | 91 | 46 | 45 | 43 | 44 | 0.0126 | 0.0079 | 0.0136 |
| $3 \times 10^{-3}$ | 105 | 53 | 52 | 49 | 51 | 0.0062 | 0.0054 | 0.0062 |
| $2 \times 10^{-3}$ | 128 | 65 | 63 | 60 | 62 | 0.0008 | 0.0090 | 0.0212 |
| $10^{-3}$ | 181 | 91 | 89 | 85 | 87 | 0.0022 | 0.0017 | 0.0030 |
| $9 \times 10^{-4}$ | 191 | 96 | 93 | 89 | 91 | 0.0075 | 0.0004 | 0.0010 |
| $8 \times 10^{-4}$ | 202 | 102 | 99 | 95 | 97 | 0.0015 | 0.0024 | 0.0085 |
| $7 \times 10^{-4}$ | 216 | 109 | 106 | 101 | 103 | 0.0054 | 0.0014 | 0.0002 |
| $6 \times 10^{-4}$ | 233 | 117 | 114 | 109 | 112 | 0.0003 | 0.0008 | 0.0009 |
| $5 \times 10^{-4}$ | 255 | 128 | 125 | 119 | 122 | 0.0001 | 0.0020 | 0.0033 |
| $4 \times 10^{-4}$ | 285 | 143 | 139 | 133 | 136 | 0.0021 | 0.0003 | 0.0002 |

Table 6: Error comparison, $\alpha_p = 0$, $\alpha_d = 2$, $N_p$ is the number of grid-points in the principal direction and $N_{d_j}$, $j = 2 \cdots 5$, is the number of grid-points in other space dimensions

| space tolerance | $N_p$ | $N_{d_2}$ | $N_{d_3}$ | $N_{d_4}$ | $N_{d_5}$ | Adaptive (pushed grid-points) | Adaptive ($x^0$ manually inserted) | Equidistant |
|---|---|---|---|---|---|---|---|---|
| $9 \times 10^{-3}$ | 61 | 23 | 23 | 23 | 23 | 0.0239 | 0.0049 | 0.0271 |
| $8 \times 10^{-3}$ | 65 | 24 | 24 | 24 | 24 | 0.0067 | 0.0075 | 0.0371 |
| $7 \times 10^{-3}$ | 69 | 26 | 26 | 25 | 26 | 0.0059 | 0.0268 | 0.0474 |
| $6 \times 10^{-3}$ | 75 | 27 | 27 | 27 | 28 | 0.0122 | 0.0031 | 0.0522 |
| $5 \times 10^{-3}$ | 82 | 30 | 30 | 29 | 30 | 0.0075 | 0.0108 | 0.0083 |
| $4 \times 10^{-3}$ | 91 | 33 | 33 | 33 | 33 | 0.0045 | 0.00001 | 0.0199 |
| $3 \times 10^{-3}$ | 105 | 38 | 38 | 37 | 38 | 0.0105 | 0.0096 | 0.0094 |
| $2 \times 10^{-3}$ | 128 | 46 | 46 | 46 | 47 | 0.0031 | 0.0105 | 0.0232 |
| $10^{-3}$ | 181 | 65 | 65 | 64 | 65 | 0.0031 | 0.0009 | 0.0043 |
| $9 \times 10^{-4}$ | 191 | 68 | 68 | 68 | 69 | 0.0067 | 0.0003 | 0.0022 |
| $8 \times 10^{-4}$ | 202 | 72 | 73 | 72 | 73 | 0.0008 | 0.0018 | 0.0094 |
| $7 \times 10^{-4}$ | 216 | 77 | 78 | 77 | 78 | 0.0048 | 0.0020 | 0.0012 |
| $6 \times 10^{-4}$ | 233 | 83 | 84 | 83 | 84 | 0.0002 | 0.0003 | 0.0017 |
| $5 \times 10^{-4}$ | 255 | 91 | 92 | 90 | 92 | 0.0002 | 0.0023 | 0.0040 |
| $4 \times 10^{-4}$ | 285 | 102 | 102 | 101 | 102 | 0.0018 | 0.0005 | 0.0006 |

Table 7: Error comparisons, $\alpha_p = 0$, $\alpha_d = 5$, $N_p$ is the number of grid-points in the principal direction and $N_{d_j}$, $j = 2 \cdots 5$, is the number of grid-points in other space dimensions

| space tolerance | $N_p$ | $N_{d_2}$ | $N_{d_3}$ | $N_{d_4}$ | $N_{d_5}$ | Adaptive (pushed grid-points) | Adaptive ($x^0$ manually inserted) | Equidistant |
|---|---|---|---|---|---|---|---|---|
| $9 \times 10^{-3}$ | 61 | 20 | 20 | 20 | 20 | 0.0128 | 0.0311 | 0.0073 |
| $8 \times 10^{-3}$ | 65 | 21 | 21 | 21 | 21 | 0.0103 | 0.0482 | 0.0174 |
| $7 \times 10^{-3}$ | 69 | 22 | 23 | 23 | 23 | 0.0120 | 0.0512 | 0.0344 |
| $6 \times 10^{-3}$ | 75 | 24 | 24 | 24 | 24 | 0.0245 | 0.0563 | 0.0094 |
| $5 \times 10^{-3}$ | 82 | 26 | 26 | 26 | 27 | 0.0036 | 0.0132 | 0.0144 |
| $4 \times 10^{-3}$ | 91 | 29 | 29 | 29 | 29 | 0.0004 | 0.0209 | 0.0022 |
| $3 \times 10^{-3}$ | 105 | 33 | 33 | 33 | 34 | 0.0175 | 0.0132 | 0.0143 |
| $2 \times 10^{-3}$ | 128 | 39 | 40 | 40 | 40 | 0.0088 | 0.0257 | 0.0159 |
| $10^{-3}$ | 181 | 56 | 56 | 56 | 57 | 0.0038 | 0.0049 | 0.0002 |
| $9 \times 10^{-4}$ | 191 | 59 | 60 | 60 | 60 | 0.0061 | 0.0029 | 0.0009 |
| $8 \times 10^{-4}$ | 202 | 62 | 63 | 63 | 64 | 0.0003 | 0.0102 | 0.0012 |
| $7 \times 10^{-4}$ | 216 | 66 | 67 | 67 | 68 | 0.0043 | 0.0017 | 0.0025 |
| $6 \times 10^{-4}$ | 233 | 71 | 72 | 73 | 73 | 0.0006 | 0.0024 | 0.0001 |
| $5 \times 10^{-4}$ | 255 | 78 | 79 | 80 | 80 | 0.0006 | 0.0045 | 0.0027 |
| $4 \times 10^{-4}$ | 285 | 87 | 89 | 89 | 90 | 0.0015 | 0.0011 | 0.0008 |

Table 8: Error comparisons, $\alpha_p = 0$, $\alpha_d = 8$, $N_p$ is the number of grid-points in the principal direction and $N_{d_j}$, $j = 2 \cdots 5$, is the number of grid-points in other space dimensions

| space tolerance | $N_p$ | $N_{d_2}$ | $N_{d_3}$ | $N_{d_4}$ | $N_{d_5}$ | Adaptive (pushed grid-points) | Adaptive ($x^0$ manually inserted) | Equidistant |
|---|---|---|---|---|---|---|---|---|
| $9 \times 10^{-3}$ | 61 | 19 | 19 | 19 | 19 | 0.0260 | 0.0391 | 0.0217 |
| $8 \times 10^{-3}$ | 65 | 20 | 20 | 20 | 20 | 0.0123 | 0.0482 | 0.0288 |
| $7 \times 10^{-3}$ | 69 | 21 | 21 | 21 | 22 | 0.0291 | 0.0574 | 0.0415 |
| $6 \times 10^{-3}$ | 75 | 22 | 23 | 23 | 23 | 0.0244 | 0.0589 | 0.0139 |
| $5 \times 10^{-3}$ | 82 | 24 | 25 | 25 | 25 | 0.0010 | 0.0168 | 0.0164 |
| $4 \times 10^{-3}$ | 91 | 27 | 27 | 28 | 28 | 0.0027 | 0.0247 | 0.0040 |
| $3 \times 10^{-3}$ | 105 | 31 | 31 | 32 | 32 | 0.0165 | 0.0143 | 0.0144 |
| $2 \times 10^{-3}$ | 128 | 37 | 37 | 38 | 38 | 0.0100 | 0.0265 | 0.0168 |
| $10^{-3}$ | 181 | 52 | 53 | 53 | 54 | 0.0045 | 0.0057 | 0.0004 |
| $9 \times 10^{-4}$ | 191 | 55 | 56 | 56 | 57 | 0.0055 | 0.0036 | 0.0014 |
| $8 \times 10^{-4}$ | 202 | 58 | 59 | 60 | 60 | 0.0001 | 0.0107 | 0.0009 |
| $7 \times 10^{-4}$ | 216 | 62 | 63 | 64 | 64 | 0.0039 | 0.0021 | 0.0027 |
| $6 \times 10^{-4}$ | 233 | 67 | 68 | 69 | 69 | 0.0009 | 0.0026 | 0.0003 |
| $5 \times 10^{-4}$ | 255 | 73 | 74 | 75 | 76 | 0.0008 | 0.0048 | 0.0010 |
| $4 \times 10^{-4}$ | 285 | 81 | 83 | 84 | 84 | 0.0014 | 0.0013 | 0.0028 |

Table 9: Error comparisons, $\alpha_p = 0$, $\alpha_d = 10$, $N_p$ is the number of grid-points in the principal direction and $N_{d_j}$, $j = 2 \cdots 5$, is the number of grid-points in other space dimensions

Note that in the first experiment for adaptive one, the calculation is made by inserting the $x^0$ manually as an extra grid-point in the grid-points in all dimensions. In the second experiment the calculation is made shifting the grid-points to the right until $x^0$ becomes included in the grid-points. In order not to introduce a new error, see Table 2, the error should be $< 10^{-2}$. This is achieved by adaptive grid-points faster than equidistant ones for $\alpha_d = 5$, $\alpha_d = 8$ and $\alpha_d = 10$.

# 5 Discussion

In this study, a multidimensional Black-Scholes equation is solved numerically using a principal component analysis together with asymptotic expansion in order to derive the price for a basket option. An adaptive technique is used in order to place the grid-points efficiently.

The innovation in this work is implementing these two methods to solve the problem. The problem is previously solved by each of these methods individually while in this study both methods are applied to fill the gap that existed in each study separately. Using principal component analysis together with asymptotic expansion addresses the unsolved curse of dimensionality in the problem solved merely using the adaptive method. On the other hand, allocating the grid-points in desired areas avoids the unnecessary distribution of the grid-points where higher accuracy is not needed.

Comparing the performance of the dimension reduction technique using adaptive grid-points instead of equidistant ones, we can observe a considerable improvement in the results. Results confirm that using adaptive grid-points decreases the number of grid-points required for a given error.

Results from the one-dimensional problem using various ratios show improvement in the results depending on the distance between the strike price and the stock price however it is difficult to draw conclusion for the five-dimensional experiments.

## References


[1] Yves Achdou and Olivier Pironneau. *Computational methods for option.* Society for Industrial and Applied Mathematics, 2005.

[2] Tomas Björk. *Arbitrage Theory in Continuous Time.* Oxford University Press, New-York, 1998.

[3] Fischer Black and Myron Scholes. The Pricing of Options and Corporate liabilities, Journal of Political Economy, 81:637-659, 1973.

[4] Ron Chan. *Pricing Options under Jump-Diffusion Models by Adaptive Radial Basic Functions*, Working Paper. Department of Economics, University of Bath, 2010.

[5] Erik Ekedahl, Eric Hansander and Erik Lehto. *Dimension Reduction for the Black-Scholes Equation*, Department of Information Technology, Uppsala University, 2007.

[6] Lötstedt, P., Persson, J., von Sydow, L., Tysk, J.: *Space-time adaptive finite difference method for European multi-asset options.* Technical report / Department of Information Technology, Uppsala University, ISSN 1404-3203; 2004-055(2004).

[7] I. T. Jolliffe. *Principal Component Analysis*, Sprimger-Verlag New York Inc., 1986.

[8] Jonas Persson and Lina von Sydow. Pricing European multi-asset options using a space-time adaptive FD-method. In Computing and Visualization in Science, 10:173-183, 2007.

[9] Olivier Pironneau, Frédéric Hecht. Mesh adaption for the Black&Scholes equations. East - West Journal of Numerical Mathematics, 8:25-35, 2000.

[10] Robert C. Merton. Theory of Rational Option Pricing. The Bell Journal of Economics and Management Science, 4:141-183, 1973.

[11] Christopher Reisinger and Gabriel Wittum, Efficient Hierarchical Approximation of high-dimensional Option Pricing Problem. SIAM Journal on Scientific Computing, 29:440-458, 2007.

[12] Rüdiger U.Seydel. *Tools for Computational Finance*, Springer, 2009.

[13] A Vande Wouwer , Ph. Saucez and W.E. Schiesser. *Adaptive Method of Lines.* Chapman and Hall/CRC, 2001.

[14] Paul Wilmot, Sam Howison and Jeff Dewynne. *The Mathematics of Financial Derivatives.* Cambridge University Press, 1995.