Filomat 30:11 (2016), 2973–2984
DOI 10.2298/FIL1611973K

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

# Computation of Replicated Exercise Prices by Using Positive Bases

**Vasilios N. Katsikis**[a]

[a]*National and Kapodistrian University of Athens, Department of Economics, Division of Mathematics and Informatics, Greece.*

**Abstract.** A procedure is provided for computing the replicated exercise prices of a given portfolio. We highlight a matrix-based framework for analyzing option replication. The new matrix formulation allows the development of efficient computational methods in order to determine the replicated exercise prices of a given portfolio by using the theory of positive bases in vector lattices.

## 1. Introduction

An important implication of the work of Ross in [14], is the existence of options that cannot be replicated by the primitive securities when markets are incomplete. Therefore, perfect risk transfer is not possible since some payoffs cannot be replicated by trading in marketed securities. In [1], Aliprantis and Tourky concluded that Ross's findings are, in fact, negative since they assert that in an incomplete market one cannot expect to replicate the payoff of each option even if the underlying asset is traded. Moreover, they reach the following remarkable complementary conclusion: If the markets are strongly resolving and the number of securities is less than half the number of states of the world, then (generically) not a single option can be replicated by traded securities. In [4], Baptista extended the aforementioned result in [1], to accommodate cases where the condition on the number of primitive securities is not imposed. In particular, it is proved that if there exists no binary payoff vector in the asset span, then for each portfolio there exists a set of nontrivial exercise prices of full measure such that any option on the portfolio with an exercise price in this set is non-replicated. Also, as it is remarked in [4], the class of markets without binary vectors is dense in $\mathbb{R}^m$, in the sense of Lebesgue measure. It is also worth mentioning that the results of Ross for two-date security markets with finitely many states holds for security markets with more than two dates, see [2, 3]. In [13], the authors examined both the results of Baptista, [4], and Aliprantis and Tourky, [1]. They proposed a characterization of markets without binary vectors as follows: Consider a two-period security market $X$, if $\mathbf{1} = (1, 1, ..., 1) \in X$, then $X$ does not contain binary vectors if and only if for any nonconstant vector $x \in X$ at least one nontrivial option of $x$ is non-replicated. In [13] it is proved that in these markets for any $x \in X$ the set of nontrivial exercise prices of $x$ contains at most $k-3$ replicated exercise prices, where $k \leq m$. In previous work, [5–12], we have shown that it is possible to construct computational methods in order to efficiently compute vector sublattices and lattice-subspaces of $\mathbb{R}^m$ or $C[a, b]$. In addition, these methods have been successfully applied in portfolio insurance, completion of security markets and option replication. This work further explores the results provided in [13] as well as various techniques

2010 *Mathematics Subject Classification*. 90C90; 90-08; 91-08; 91G10
*Keywords*. Replicated exercise price; positive bases.
Received: 25 July 2014; Accepted: 13 October 2014
Communicated by Predrag Stanimirović
*Email address:* vaskatsikis@econ.uoa.gr, vaskats@gmail.com (Vasilios N. Katsikis)

for the computation of positive bases and vector sublattices presented in [5–12]. This paper proceeds as follows: Section 2 briefly introduces the basic notation and terminology. In Section 3, we present our main results as well as the related algorithmic procedure in order to determine replicated exercise prices for a given portfolio. Conclusions are provided in Section 4. The Appendix, i.e. Section 5, presents a Matlab implementation of the proposed algorithm.

## 2. Preliminaries

For any $x = (x(1), x(2), ..., x(m)) \in \mathbb{R}^m$, the set $supp(x) = \{i | x(i) \neq 0\}$ is the *support* of $x$. The vectors $x, y \in \mathbb{R}^m$ have *disjoint supports* if $supp(x) \cap supp(y) = \emptyset$. An ordered subspace $Z$ of $\mathbb{R}^m$ is a *vector sublattice* or a *Riesz subspace* of $\mathbb{R}^m$ if for any $x, y \in Z$ the supremum and the infimum of the set $\{x, y\}$ in $\mathbb{R}^m$ belong to $Z$. Assume that $X$ is an ordered subspace of $\mathbb{R}^m$ and $B = \{b_1, b_2, ..., b_\mu\}$ is a basis for $X$. Then $B$ is a *positive basis* of $X$ if for each $x \in X$ it holds that $x$ is positive if and only if its coefficients in the basis $B$ are positive.

A positive basis $B = \{b_1, b_2, ..., b_\mu\}$ is a *partition of the unit* if the vectors $b_i$ have disjoint supports and $\sum_{i=1}^{\mu} b_i = 1$. The existence of positive bases is not always ensured, but in the case where $X$ is a vector sublattice of $\mathbb{R}^m$ then $X$ always has a positive basis. Recall that a nonzero element $x_0$ of $X_+$ is an extremal point of $X_+$ if, for any $x \in X, 0 \leq x \leq x_0$ implies $x = \lambda x_0$, for a real number $\lambda$. Since each element $b_i$ of the positive basis of $X$ is an extremal point of $X_+$, a positive basis of $X$ is unique in the sense of positive multiples. For computational methods regarding positive bases with applications in economics we refer to [5–9, 11].

Suppose that, agents trade $x_1, x_2, ..., x_n$ non-redundant (linearly independent) securities in period $t = 0$, future payoffs of $x_1, x_2, ..., x_n$ are collected in a matrix $A = \left[x_i(j)\right]_{i=1,2,...n}^{j=1,2,...,m} \in \mathbb{R}^{m \times n}$ where $x_i(j)$ is the payoff of one unit of security $i$ in state $j$. It is clear that the matrix $A$ is of full rank and the *asset span* is denoted by $X = Span(A)$. A vector in $\mathbb{R}^m$ is said to be *marketed* or *replicated* if $x$ is the payoff of some portfolio $\theta$ (called the *replicating portfolio* of $x$), or equivalently if $x \in X$. In the following, we shall denote by $\mathbf{1}$ the *riskless bond* i.e., the vector $\mathbf{1} = (1, 1, ..., 1)$. A vector $x$ is a *binary vector* if $x \neq \mathbf{0} = (0, 0, ..., 0)$, $x \neq \mathbf{1}$ and $x(i) = 0$ or $x(i) = 1$, for any $i$. If both $c(x, \alpha) = (x - \alpha\mathbf{1})^+ > 0$ and $p(x, \alpha) = (\alpha\mathbf{1} - x)^+ > 0$, we say that the call option $c(x, \alpha)$ and the put option $p(x, \alpha)$ are *non trivial* and the exercise price $\alpha$ is a *non trivial exercise price* of $x$. If $c(x, \alpha)$ and $p(x, \alpha)$ belong to $X$ then we say that $c(x, \alpha)$ and $p(x, \alpha)$ are *replicated*. If we suppose that $\mathbf{1} \in X$ and at least one of $c(x, a)$, $p(x, a)$ is replicated, then both of them are replicated since by the put-call parity we have, $x - \alpha\mathbf{1} = c(x, \alpha) - p(x, \alpha)$. For notation not defined here the interested reader may refer to [9, 13] and the references therein.

We present a new matrix formulation of the study of markets without binary vectors, in order to develop efficient computational methods for option replication. Suppose that a security market $X$ is generated by a given collection of non-redundant securities $x_1, x_2, ..., x_n$ of $\mathbb{R}^m$. In the theory of security markets it is a usual practice to take call and put options with respect to the riskless bond $\mathbf{1} = (1, 1, ..., 1)$. The completion, $F_{\mathbf{1}}(X)$, of $X$ by options is exactly the subspace of $\mathbb{R}^m$ generated by all options written on the elements of $X \cup \{\mathbf{1}\}$. Since the payoff space is $\mathbb{R}^m$, which is a vector lattice, in the case where $\mathbf{1} \in X$ then $F_{\mathbf{1}}(X)$ is exactly the vector sublattice generated by X. In addition, if $X$ is a vector sublattice of $\mathbb{R}^m$ then $F_{\mathbf{1}}(X) = X$ therefore any option is replicated. Note that the vectors $x_1, x_2, ..., x_n$ are not presupposed to be positive. A *basic set of marketed securities* is a set of linearly independent and positive vectors of $X$ such that the vector sublattice of $\mathbb{R}^m$ generated by such a set is $F_{\mathbf{1}}(X)$. It is well-known that a basic set of marketed securities always exist. For any $x = \sum_{i=1}^{\mu} \lambda_i b_i \in F_{\mathbf{1}}(X)$, let $a_1 = \min\{\lambda_i | i = 1, 2, ..., \mu\}$, $a_2 = \min\{\lambda_i | \lambda_i > a_1\}$, ..., $a_k = \min\{\lambda_i | \lambda_i > a_{k-1}\}$. The numbers $a_1, a_2, ..., a_k$ will be referred as the *essential coefficients* of $x$, with respect to the basis $\{b_i\}$. A vector $x \in \mathbb{R}^m$ is a *nonconstant vector* if $x$ is not a multiple of $\mathbf{1}$, i.e., $x \neq \lambda\mathbf{1}$ for any $\lambda \in \mathbb{R}$.

## 3. Main Results

We highlight a matrix-based framework, for analyzing option replication. The proposed matrix formulation will enable us to develop efficient computational methods in order to determine the replicated exercise prices of a portfolio by using the theory of vector lattices and positive bases. Suppose that a security market $X$ is generated by a given collection of linearly independent vectors $y_1, y_2, ..., y_n$ of $\mathbb{R}^m$. As we have already mentioned, a basic set of marketed securities $x_1, x_2, ..., x_n$ for the market $X$ always exists. Also, the sublattice generated by a basic set of marketed securities is exactly $F_{\mathbf{1}}(X)$ and $F_{\mathbf{1}}(X)$ has a positive basis which is a partition of the unit, i.e., the vectors $b_1, b_2, ..., b_\mu$ have disjoint supports and $\sum_{i=1}^{\mu} b_i = 1$. Let us denote by $A$ the matrix

$$A = \begin{bmatrix} x_1(1) & x_2(1) & \dots & x_n(1) \\ x_1(2) & x_2(2) & \dots & x_n(2) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(m) & x_2(m) & \dots & x_n(m) \end{bmatrix}$$

where $x_i(j)$ is the payoff of one unit of security $i$ in state $j$. Also, if $b_1, b_2, ..., b_\mu$ is a positive basis for $F_{\mathbf{1}}(X)$ which is a partition of the unit, then

$$B = \begin{bmatrix} b_1(1) & b_2(1) & \dots & b_\mu(1) \\ b_1(2) & b_2(2) & \dots & b_\mu(2) \\ \vdots & \vdots & \vdots & \vdots \\ b_1(m) & b_2(m) & \dots & b_\mu(m) \end{bmatrix}$$

is the matrix where $b_i(j)$ is the $j$ coordinate of the vector $b_i$.

**Lemma 3.1.** *For each $x \in F_{\mathbf{1}}(X)$, the essential coefficients of $x$ are the different coordinates of $x$.*

*Proof.* Let $x = (x(1), x(2), ..., x(m))$ and $x = \sum_{i=1}^{\mu} \lambda_i b_i$. If $j \in supp(b_k)$, for some $k \in \{1, ..., \mu\}$, then $x(j) = \sum_{i=1}^{\mu} \lambda_i b_i(j) = \lambda_k b_k(j)$. Since the basis is a partition of the unit we have that $b_k(j) = 1$, so $x(j) = \lambda_k$. From the definition of the essential coefficients of $x$ the lemma is true. □

Suppose that

$$S_1 = supp(b_1), S_2 = supp(b_2), ..., S_\mu = supp(b_\mu)$$

and denote by $m_i$ the minimum element of the set $S_i$ for each $i = 1, ..., \mu$, i.e.,

$$m_1 = \min(S_1), m_2 = \min(S_2), ..., m_\mu = \min(S_\mu).$$

**Lemma 3.2.** *For each $x \in X$ it holds $x = \sum_{i=1}^{\mu} x(m_i) b_i$.*

*Proof.* Suppose that $x = (x(1), x(2), ..., x(m))$ and $x = \sum_{i=1}^{\mu} \lambda_i b_i$ then $x(m_1) = \sum_{i=1}^{\mu} \lambda_i b_i(m_1)$. Since $m_1 \in S_1$, we have that $b_1(m_1) = 1$ and $b_\rho(m_1) = 0$, for $\rho = 2, 3, ..., \mu$. Therefore, $x(m_1) = \lambda_1$ and by using similar arguments one can prove that $x(m_i) = \lambda_i$, for $i = 2, 3, ..., \mu$. □

Let $\beta_1 = \min\{x(m_i) | i = 1, 2, ..., \mu\}$, $\beta_2 = \min\{x(m_i) | x(m_i) > \beta_1, | i = 1, 2, ..., \mu\}$, $\dots$, $\beta_k = \max\{x(m_i) | i = 1, 2, ..., \mu\}$, then we have the following proposition.

**Proposition 3.3.** *For any $x \in F_{\mathbf{1}}(X)$ and $a \in \mathbb{R}$ we have:*

(*i*) *$c(x, a) = \sum_{i=1}^{\mu} (x(m_i) - a)^+ b_i$ and $p(x, a) = \sum_{i=1}^{\mu} (a - x(m_i))^+ b_i$,*
(*ii*) *the interval $K_x = (\beta_1, \beta_k)$ is the set of nontrivial exercise prices of $x$.*

*Proof.* (*i*) From lemma 3.2 we have $x = \sum_{i=1}^{\mu} x(m_i) b_i$, so $c(x, a) = (x - a\mathbf{1})^+ = \left(\sum_{i=1}^{\mu} x(m_i) b_i - a\mathbf{1}\right)^+$. Since the basis $\{b_i\}$ is a partition of the unit we have $\sum_{i=1}^{\mu} b_i = \mathbf{1}$, and the vectors $b_i$ have disjoint supports, therefore

$$c(x, a) = \Big(\sum_{i=1}^{\mu} x(m_i) b_i - a \sum_{i=1}^{\mu} b_i\Big)^+ = \Big(\sum_{i=1}^{\mu} \big(x(m_i) - a\big) b_i\Big)^+ = \sum_{i=1}^{\mu} \max\{x(m_i) - a, 0\} b_i = \sum_{i=1}^{\mu} \big(x(m_i) - a\big)^+ b_i.$$

The proof for the put option is analogous.

(*ii*) Trivial.

□

The following analysis based on row leader elements is important for the construction of our proposed computational method. Let us call the *row leader*, the leftmost nonzero element of each row of the matrix

$$C = B^T = \begin{bmatrix} b_1(1) & b_1(2) & \dots & b_1(m) \\ b_2(1) & b_2(2) & \dots & b_2(m) \\ \vdots & \vdots & \vdots & \vdots \\ b_\mu(1) & b_\mu(2) & \dots & b_\mu(m) \end{bmatrix},$$

and let us denote by $c_{1j_1}, c_{2j_2}, ..., c_{\mu j_\mu}$ the row leader elements of $C$.

**Proposition 3.4.** *For each portfolio $x \in X$ with $x = (x(1), x(2), ..., x(m))$ we have that*

$$x = \sum_{i=1}^{\mu} x(j_i) b_i,$$

*where $j_1, j_2, ..., j_\mu$ are the column indices of the row leader elements of $C$.*

*Proof.* It is clear that, $j_i = m_i$, for $i = 1, 2, ..., \mu$, therefore from lemma 3.2 it holds $x = \sum_{i=1}^{\mu} x(m_i) b_i = \sum_{i=1}^{\mu} x(j_i) b_i$. □

**Theorem 3.5.** *Suppose that the asset span $X$ does not contain binary vectors and $x$ is a nonconstant vector of $X$.*

(*i*) *If $k = 2$, each nontrivial call option of $x$ is non-replicated. If $k > 2$, each of the intervals $(\beta_1, \beta_2)$, $[\beta_2, \beta_3)$, ..., $[\beta_{k-2}, \beta_{k-1})$ contains at most one call-replicated exercise price, therefore there are at most $k-2$ call-replicated exercise prices of $x$.*

(*ii*) *If $k = 2$, each nontrivial put option of $x$ is non-replicated. If $k > 2$, each of the intervals $(\beta_2, \beta_3]$, ...,$(\beta_{k-2}, \beta_{k-1}]$, $(\beta_{k-1}, \beta_k)$ contains at most one put-replicated exercise price, therefore there are at most $k-2$ put-replicated exercise prices of $x$.*

(*iii*) *If we suppose moreover that $\mathbf{1} \in X$, we have: If $k = 3$, each nontrivial option of $x$ is non-replicated. If $k > 3$, each of the intervals $(\beta_2, \beta_3), [\beta_3, \beta_4), ..., [\beta_{k-2}, \beta_{k-1})$ contains at most one replicated exercise price, therefore there are at most $k-3$ replicated exercise prices of $x$.*

*Proof.* (*i*) From propositions 3.3 and 3.4 we have that $c(x, a) = \sum_{i=1}^{\mu} \left(x(j_i) - a\right)^+$. Note that $\beta_1 = \min\{x(m_i) | i = 1, 2, ..., \mu\} = \min\{x(j_i) | i = 1, 2, ..., \mu\}$ and $\beta_k = \max\{x(m_i) | i = 1, 2, ..., \mu\} = \max\{x(j_i) | i = 1, 2, ..., \mu\}$. Suppose that $a \in (\beta_1, \beta_k)$ then

$$c(x, a) = \sum_{i \in \Phi} (x(j_i) - a) b_i, \quad \textit{where} \quad \Phi = \{i | x(j_i) > a\}.$$

Let $\beta_k = x(j_k)$ then $a < x(j_k)$. If, in addition, $a > x(j_i)$ for each $i \in \Phi \setminus \{k\}$ then

$$c(x, a) = (x(j_k) - a) b_k = (\beta_k - a) b_k \notin X,$$

since it is a positive multiple of a binary vector and $X$ does not contain binary vectors. Therefore, for any $a \in [\beta_{k-1}, \beta_k)$ we have that $c(x, a) \notin X$ hence $a$ is not a call-replicated exercise price. Since $x$ is a nonconstant vector it has at least two different coordinates, so $k \geq 2$.

If $k = 2$, then for each $a \in (\beta_1, \beta_2)$ any call option written on $x$ with replicated exercise price $a$ is nonreplicated. Indeed,

$$c(x, a) = (\beta_1 - a)^+ b_1 + (\beta_2 - a)^+ b_2 = (\beta_2 - a) b_2$$

which is a positive multiple of a binary vector, contradiction.
Let $a$, $a'$ are different exercise prices and $a, a' \in (\beta_1, \beta_2)$ or $a, a' \in [\beta_r, \beta_{r+1})$ for some $r = 2, 3, ..., k-2$. Suppose that $c(x,a) = \sum_{i\in\Phi}(x(j_i) - a)b_i$ and $c(x,a') = \sum_{i\in\Phi}(x(j_i) - a')b_i$ belong to $X$ then

$$c(x,a) - c(x,a') = \sum_{i\in\Phi}(a' - a)b_i \in X,$$

contradiction since $X$ does not contain binary vectors. Hence there are at most $k-2$ call-replicated exercise prices.

(*ii*) The proof is analogous to (*i*).

(*iii*) Trivial.

□

Note that the proofs of Proposition 3.3 and Theorem 3.5 are presented here for consistency reasons regarding the notion of row leader elements. The original proofs are included in [13].

Our previous analysis based on row leader elements will provide an efficient computational method in order to calculate the replicated exercise prices of a given portfolio. According to theorem 3.5, we shall present the proposed formulation for the option replication problem based on a matrix notation.

Recall that $A$ is the $m \times n$ matrix with columns being the vectors of the basic set of marketed securities. Recall that, the matrix $A$ has full column rank i.e., $rank(A) = n$. $B$ denotes the $m \times \mu$ matrix whose columns are the vectors of the positive basis of $F_1(X)$ and this basis is a partition of the unit. First, we express the columns of $A$ in terms of the positive basis by solving $n$ linear systems as follows:

$$B \cdot Y = A.$$

The resulting matrix $Y$, is the matrix whose $i$ column entries are the coefficients in terms of the positive basis $B$ of the $i$ column of $A$. If $x \in X$ is a nonconstant portfolio then, from proposition 3.4, $x = \sum_{i=1}^{\mu} x(j_i)b_i$ and let us denote by $W$ the matrix

$$W = \begin{bmatrix} x(j_1) \\ x(j_2) \\ \vdots \\ x(j_\mu) \end{bmatrix}.$$

For $r = 1, ..., k-3$, we define the matrices $W_r$, from the matrix $W$, by putting $x(j_i) = 0$, $i = 1, 2, ..., \mu$ whenever $x(j_i) < \beta_{r+2}$. For each matrix $W_r = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_\mu \end{bmatrix}$ we define a corresponding matrix $J_r$, where $J_r = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_\mu \end{bmatrix}$ such that $z_i = \begin{cases} 1, & w_i \neq 0 \\ 0, & w_i = 0 \end{cases}$, $i = 1, 2, ..., \mu$. Also, for $r = 1, ...k-3$, we define the matrices $Z_r = \left[ Y \,\vdots\, J_r \right]$.

In order to calculate the replicated exercise prices of $x$, we must compute all $a \in (\beta_1, \beta_k)$ such that $c(x,a) = \sum_{i=1}^{\mu}(x(m_i) - a)^+ b_i \in X$.

According to the previous discussion, we have to solve for $r = 1, 2, ..., k-3$ the corresponding linear system

$$Z_r \cdot \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \\ a_r \end{bmatrix} = W_r.$$

If $a_r \in (\beta_{r+1}, \beta_{r+2})$ then $a_r$ is a replicated exercise price for $x$.

**Remark 3.6.** *It is clear that a slight modification of the previous procedure can provide the corresponding computational method for the put options.*

**Example 3.7.** Consider the following 5 vectors $x_1, x_2, ..., x_5$ in $\mathbb{R}^{10}$,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

and $X = [x_1, x_2, ..., x_5]$.

A positive basis which is a partition of the unit is the following

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Let $x = -x_1+x_2+3x_3+3x_4 = (7, 6, 7, 10, 6, 6, 6, 12, 8, 7)$. According to lemma 3.1 the essential coefficients of $x$ are the different elements of $x$ i.e., essential coefficients $= \{7, 6, 10, 12, 8\}$. Also, $S_1 = supp(b_1) = \{1, 3\}$, $S_2 = supp(b_2) = \{2, 5, 6, 7\}$, $S_3 = supp(b_3) = \{4\}$, $S_4 = supp(b_4) = \{8\}$, $S_5 = supp(b_5) = \{10\}$, $S_6 = supp(b_6) = \{9\}$. Then $m_1 = \min(S_1) = 1$, $m_2 = \min(S_2) = 2$, $m_3 = \min(S_3) = 4$, $m_4 = \min(S_4) = 8$, $m_5 = \min(S_5) = 10$, $m_6 = \min(S_6) = 9$. Finally, one gets $x = x(1)b_1 + x(2)b_2 + x(4)b_3 + x(8)b_4 + x(10)b_5 + x(9)b_6$.

Suppose that $a$ is a replicated exercise price of $x$. The row leader elements are the entries $c_{11}$, $c_{22}$, $c_{34}$, $c_{48}$, $c_{5\,10}$, $c_{69}$ of the matrix $C = B^T$. So, $j_1 = 1$, $j_2 = 2$, $j_3 = 4$, $j_4 = 8$, $j_5 = 10$, $j_6 = 9$ and

$$c(x, a) = \big(x(1) - a\big)^+ b_1 + \big(x(2) - a\big)^+ b_2 + \big(x(4) - a\big)^+ b_3 + \big(x(8) - a\big)^+ b_4 + \big(x(10) - a\big)^+ b_5 + \big(x(9) - a\big)^+ b_6$$

Also, $\beta_1 = 6$, $\beta_2 = 7$, $\beta_3 = 8$, $\beta_4 = 10$, $\beta_5 = 12$ and since $\mathbf{1} \in X$ ($\mathbf{1} = x_5 - x_4 + x_1$), we have that $a \notin (\beta_1, \beta_2]$ and $a \notin [\beta_4, \beta_5)$, therefore we search for replicated exercise prices inside the intervals $(7, 8)$ and $[8, 10)$. Let $a \in (7, 8)$, since $x(1) = 7$, $x(2) = 6$, $x(4) = 10$, $x(8) = 12$, $x(10) = 7$, $x(9) = 8$ we have

$$c(x, a) = \big(x(4) - a\big)b_3 + \big(x(8) - a\big)b_4 + \big(x(9) - a\big)b_6. \tag{1}$$

On the other hand, $a$ is a replicated exercise price of $x$ i.e., $c(x, a) \in X$ hence

$$c(x, a) = \sum_{i=1}^{5} \rho_i x_i = \rho_1(b_2 + b_3 + 2b_4 + b_5 + 2b_6) + \rho_2(b_1 + b_2 + 2b_3 + 2b_4 + 2b_5 + b_6) + \rho_3(b_1 + b_2 + 2b_3 + 2b_4 + b_5 + b_6) + \rho_4(b_1 + b_2 + b_3 + 2b_4 + b_5 + 2b_6) + \rho_5(2b_1 + b_2 + b_3 + b_4 + b_5 + b_6) = (\rho_2 + \rho_3 + \rho_4 + 2\rho_5)b_1 + (\rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5)b_2 + (\rho_1 + 2\rho_2 + 2\rho_3 + \rho_4 + \rho_5)b_3 + (2\rho_1 + 2\rho_2 + 2\rho_3 + 2\rho_4 + \rho_5)b_4 + (\rho_1 + 2\rho_2 + \rho_3 + \rho_4 + \rho_5)b_5 + (2\rho_1 + \rho_2 + \rho_3 + 2\rho_4 + \rho_5)b_6.$$

Therefore, by using equation (1) we have,

$$(\rho_2 + \rho_3 + \rho_4 + 2\rho_5)b_1 + (\rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5)b_2 + (\rho_1 + 2\rho_2 + 2\rho_3 + \rho_4 + \rho_5 + a)b_3 + (2\rho_1 + 2\rho_2 + 2\rho_3 + 2\rho_4 + \rho_5 + a)b_4 + (\rho_1 + 2\rho_2 + \rho_3 + \rho_4 + \rho_5)b_5 + (2\rho_1 + \rho_2 + \rho_3 + 2\rho_4 + \rho_5 + a)b_6 = x(4)b_3 + x(8)b_4 + x(9)b_6$$

from which we conclude to the following system of equations

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 2 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ x(4) \\ x(8) \\ 0 \\ x(9) \end{bmatrix}. \tag{2}$$

If the resulting $a$ belongs to the interval $(7, 8)$ then $a$ is a replicated exercise price for $x$. For the case $a \in [8, 10)$ one has to follow a similar procedure. In this example there is no replicated exercise price $a$ for $x$.

In matrix terms and according to our definitions for the matrices $W_r$, $J_r$ and $Z_r$, the previous procedure is equivalent to the following analysis,

$$A = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 \\ 2 & 1 & 1 & 2 & 1 \\ 1 & 2 & 1 & 1 & 1 \end{bmatrix},\ B = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and by solving the system $B \cdot Y = A$ we have

$$Y = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 \\ 1 & 2 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 & 1 \end{bmatrix}.$$

Recall that the row leader elements are the entries $c_{11}$, $c_{22}$, $c_{34}$, $c_{48}$, $c_{5\,10}$, $c_{69}$ of matrix $C = B^T$. Therefore, we have

$$W = \begin{bmatrix} x(j_1) \\ x(j_2) \\ x(j_3) \\ x(j_4) \\ x(j_5) \\ x(j_6) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(2) \\ x(4) \\ x(8) \\ x(10) \\ x(9) \end{bmatrix} = \begin{bmatrix} 7 \\ 6 \\ 10 \\ 12 \\ 7 \\ 8 \end{bmatrix}$$

and for $r = 1, 2$ we have matrices $W_r$, $J_r$ and $Z_r$ as follows,

$$W_1 = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 12 \\ 0 \\ 8 \end{bmatrix},\ W_2 = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 12 \\ 0 \\ 0 \end{bmatrix},\ J_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix},\ J_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$Z_1 = \left[ Y \,\vdots\, J_1 \right] = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 2 & 1 & 1 \end{bmatrix}, \; Z_2 = \left[ Y \,\vdots\, J_2 \right] = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 2 & 1 & 0 \end{bmatrix}.$$

Then we compute $a$ by solving the systems

$$Z_1 \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ a \end{bmatrix} = W_1, \; Z_2 \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ a \end{bmatrix} = W_2.$$

Note that, the first system is the system provided by equation (2).

The basic steps of an algorithmic procedure that allow the accurate implementation of the previous ideas and leads to the computation of the replicated exercise prices, are collected in the following algorithm:

**Algorithm 1** Algorithm for finding the replicated exercise prices of a given portfolio.

**Require:** The matrix $A$, i.e., the payoff matrix with the non-redundant security vectors $x_1, x_2, ..., x_n$ specified as columns.
1: Determine a basic set $y_1, y_2, ..., y_n$ of marketed securities.
2: Calculate the vector sublattice generated by $y_1, y_2, ..., y_n$, which is exactly the completion by options $F_1(X)$ of $X$.
3: Calculate a positive basis for $F_1(X)$ which is a partition of the unit.
4: Compute the row leader elements, according to proposition 3.4, and the interval of replicated exercise prices for a given portfolio $x$.
5: Expand the securities $\{x_1, x_2, ..., x_n\}$ in terms of the positive basis $\{b_1, b_2, ..., b_\mu\}$.
6: For $r = 1, 2, ..., k - 3$ construct the matrices $W_r$, $J_r$, and $Z_r$.
7: Solve the corresponding linear systems in order to compute possible replicated exercise prices.
8: Decide which of the resulting solutions are replicated exercise prices.

In the Appendix we present a Matlab-based imlementation of the previous algorithm. In order to solve the corresponding linear systems in Step 7 of the previous algorithm the Matlab backslash operator has been used.

*3.1. Execution Times*

For the purpose of monitoring the performance, we present a table with the execution times of the `reprices` function for a collection of payoff matrices with various dimensions. All numerical tasks have been performed by using the Matlab R2013a environment on an Intel(R) Core(TM) i7-3770 CPU 3.40 GHz, 3.40 GHz running on the Windows 7 Operating System. It is evident that the proposed numerical method, based on the introduction of the `reprices` function (see the appendix), enables us to perform fast computations for a variety of dimensions. A closer look at the algorithmic process suggest that a manual procedure in order to determine the replicated exercise prices of a given nonconstant portfolio can easily become prohibitive due to the elaborate computations involved. In fact, for a given portfolio, the more different the coordinates are the greater the amount of calculations required. Hence, we are convinced that the proposed approach can serve as a nice complement, with practical relevance, to the existing set of tools for option replication.

Table 1: Computation times

| Payoff matrix dimension | Time (in seconds) |
|---|---|
| $10 \times 7$ | 0.006137 |
| $11 \times 8$ | 0.0057819 |
| $12 \times 9$ | 0.0064074 |
| $13 \times 10$ | 0.0076597 |
| $14 \times 11$ | 0.0087804 |
| $15 \times 12$ | 0.010308 |
| $16 \times 13$ | 0.010978 |
| $17 \times 14$ | 0.012091 |
| $18 \times 15$ | 0.01731 |
| $19 \times 16$ | 0.01976 |
| $20 \times 17$ | 0.016359 |

## 4. Conclusions

In this paper, we propose a matrix-based framework, for analyzing option replication. The new matrix formulation allowed the development of efficient computational methods in order to determine the replicated exercise prices of a given portfolio by using the theory of vector lattices and positive bases. We are hopeful that the results of this work provide an important tool in order to study the interesting problem of option replication, in which the space of marketed securities is a subspace of $\mathbb{R}^m$.

## 5. Appendix

We present the Matlab code for the proposed computational method that enables us to determine replicated exercise prices for a given nonconstant portfolio. The code is self contained.

```
function [Reprices,Npb] = reprices(X,x)
%***************************%
%   General  Information.   %
%***************************%
% Synopsis:
% Reprices = reprices(X,x)
% [Reprices,Npb] = reprices(X,x)
% Input:
%    X = the payoff matrix with the non-redundant
%        security vectors x_1, x_2,...,x_n specified
%        as columns.
%    x = a given nonconstant portfolio of X.
% Output:
%    Reprices = is a cell array containing the
%               replicated exercise prices of x.
%    Npb = positive basis of F_1(X) which is a partition
%          of the unit. The i column of the Npb matrix is
%          the vector bi of the positive basis.
%*****************************************************%
% Determination of a basic set of marketed securities. %
%*****************************************************%
if any(any(X < 0)) ~= 0
a = max(max(abs(X)));
B = a*ones(size(X)) -X;
 if any(any(B < 0)) ~= 0
 B = 2*a*ones(size(X)) - X;
```

```
 end
else
B = X;
end
Matrix = zeros(size(B));
%**********************************%
%    Range of the basic curve.     %
%**********************************%
% Determination of the basic curve.
N = length(B(:,1));
for i = 1:N,
  if norm(B(i,:),1) ~= 0,
  Matrix(i,:) = 1/norm(B(i,:),1)*B(i,:);
  end
end
% Find the unique elements of the range of the basic curve.
[Unique,m] = unique(Matrix,'rows','first');
Sort_m = sort(m);
Matrixnew = Matrix(Sort_m,:);
r = length(m);
%**********************************************%
% Calculation of the vector sublattice F_1(X). %
%**********************************************%
% Choose which vectors are linearly independent.
S = rref(Matrixnew');
[I,J] = find(S);
Linearindep = accumarray(I,J,[rank(Matrixnew),1],@min)';
M = length(B(1,:));
% A) If X=F_1(X).
if r == M
    disp('X is a vector sublattice hence any option is replicated')
end
% B) If X~=F_1(X).
Index1 = 1:r;
Index2 = setdiff(Index1,Linearindep);
Index = 1:N;
YY = sum(B,2)';
TTT = setdiff(Index,Linearindep);
Id = eye(N);
KK = Id(TTT,:);
TT = YY(1,TTT)';
T = diag(TT)*KK;
K = zeros(N);
K(TTT,:) = T;
Vec = zeros(r-M,N);
DDD = cell(r-M,1);
for i = 1:length(Index2)
DD = strmatch(Matrixnew(Index2(i),:),Matrix,'exact');
R = length(DD);
 if R >= 2,
  Vector = sum(K(DD,:));
 else
  Vector = K(DD,:);
 end
DDD{i,:} = DD;
Vec(i,:) = Vector;
```

```
end
Sublattice = [B Vec'];
%****************************************************%
% Determination of a positive basis for F_1(X) which %
% is a partition of the unit.                        %
%****************************************************%
% Calculate the new basic curve for F_1(X).
Matrixnew2 = zeros(size(Sublattice));
for i = 1:N,
 if norm(Sublattice(i,:),1) ~= 0,
 Matrixnew2(i,:) = 1/norm(Sublattice(i,:),1)*Sublattice(i,:);
 end
end
u = Matrixnew2([Sort_m(Linearindep)' cell2mat(DDD)'],:);
Test_Pb = u'\Sublattice';
[f,ff] = find(Test_Pb);
Pb = Test_Pb(unique(f),:);
% Normalization of the positive basis (Npb).
Npb1 = diag(1./max(Pb,[ ],2))*Pb;
Npb = Npb1';
Npb(Npb < 10*eps) = 0;
Npb(Npb < 1+10*eps & Npb > 1-10*eps) = 1;
%************************************************************%
% Determination of the row leader elements and the interval  %
% of replicated exercise prices.                             %
%************************************************************%
esscoef = unique(x);
mu = length(esscoef);
if mu <= 3
  Reprices = [];
  return
end
[i,j] = find(Npb');
Z = accumarray(i,j,[r,1],@min)';
L = x(Z);
disp('The interval of nontrivial exercise prices of x:');
disp([esscoef(1,1),esscoef(end,1)]);
%************************************************************%
% Expansion of the primitive securities in terms of the      %
% positive basis (Npb) of F_1(X). Construct the matrices Wr, %
% Jr and Zr. Solve the corresponding systems.                %
%************************************************************%
X1 = Npb\X;
R_Sol = length(X(1,:))+1;
Solution = zeros(R_Sol,mu-3);
Reprices = cell(1,mu-3);
for r = 1:mu-3
 K = L;
 K(K < esscoef(r+2)) = 0;
 Wr = K;
 K(K~=0) = 1;
 Jr = K;
 Zr = [X1 Jr];
 Solution(:,r) = Zr\Wr;
%*********************************************************%
% Decide which of the resulting solutions are replicated  %
```

```
% exercise prices.                                         %
%**********************************************************%
 if Solution(R_Sol,r) >= esscoef(r+1,1) && ...
    Solution(R_Sol,r) < esscoef(r+2,1)
    Reprices{1,r} = Solution(R_Sol,r);
 else
    Reprices{1,r} = '-';
 end
end
```

## References


[1] C.D. Aliprantis, R. Tourky, Markets that don't replicate any option, *Economics Letter,* **76** (2002), 443-447.
[2] A.M. Baptista, Spanning with american options, *Journal of Economic Theory,* **110** (2003), 264-289.
[3] A.M. Baptista, Options and efficiency in multidate security markets, *Mathematical Finance,* **15** (2005), 569-587.
[4] A.M. Baptista, On the non-existence of redundant options, *Economic Theory,* **31** (2007), 205-212.
[5] V.N. Katsikis, Computational methods in portfolio insurance, *Applied Mathematics and Computation,* **189** (2007), 9-22.
[6] V.N. Katsikis, Computational methods in lattice-subspaces of $C[a, b]$ with applications in portfolio insurance, *Applied Mathematics and Computation,* **200** (2008), 204-219.
[7] V.N. Katsikis, A Matlab-based rapid method for computing lattice-subspaces and vector sublattices of $\mathbb{R}^n$: Applications in portfolio insurance, *Applied Mathematics and Computation,* **215** (2009), 961-972.
[8] V.N. Katsikis. Computational and Mathematical Methods in Portfolio Insurance. A MATLAB-Based Approach., Matlab - Modelling, Programming and Simulations, (Ed.), ISBN: 978-953-307-125-1, InTech, 2010 (Book chapter).
[9] V.N. Katsikis, Computational methods for option replication, *International Journal of Computer Mathematics,* **88** (2011), 2752-2769.
[10] V.N. Katsikis, MATLAB Aided Option Replication, MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 3, ISBN 978-953-51-0752-1, InTech, 2012 (Book chapter).
[11] V.N. Katsikis, I. Polyrakis, Computation of vector sublattices and minimal lattice-subspaces of $\mathbb{R}^k$. Applications in finance. *Applied Mathematics and Computation,* **218** (2012), 6860-6873.
[12] V.N. Katsikis, A new characterization of markets that don't replicate any option through minimal-lattice subspaces. A computational approach. *Filomat,* **27:7** (2013), 1357-1372.
[13] C. Kountzakis, I.A. Polyrakis, F. Xanthos, Non replication of options, *Mathematical Finance,* **22** (2012), 569-584.
[14] S.A. Ross, Options and efficiency, *Quarterly Journal of Economics,* **90** (1976), 75-89.