

# Computational and mathematical methods in portfolio insurance - A MATLAB-based approach

Vasilios N. Katsikis  
General Department of Mathematics,  
Technological Education Institute of Piraeus,  
12244 Athens  
Greece

## 1. Introduction

Portfolio insurance is based on the principal of risk transfer i.e., one person's protection is another person's liability. The cost of portfolio insurance is the mechanism to equilibrate its demand with supply. In the theory of finance minimum-cost portfolio insurance has been characterized as a very important investment strategy. In this chapter, we discuss the investment strategy called minimum-cost portfolio insurance as a solution of a cost minimization problem and we propose computational methods that translate the economics problem into the language of computing. This strategy not only enables an investor to avoid losses but also allows him/her to capture the gains at the minimum cost. In general, it is well known that the minimum-cost insured portfolio depends on security prices. The cases where it is price-independent (i.e., it does not depend on arbitrage-free security prices) are very important not only because the insured portfolio can be selected without knowledge of current security prices but also because we can present it in a simple form. Market structures in which minimum-cost portfolio insurance is price-independent relies on the theory of vector lattices (Riesz spaces). In particular, we focus our study in two very important classes of subspaces of a vector lattice, namely vector sublattices and lattice-subspaces. Vector lattices have been used by Brown & Ross (1991) and by Green & Jarrow (1987) in the framework of options markets. Also, Ross (1976) gave a characterization of complete markets by observing that derivative markets are complete if and only if the asset span is a vector sublattice of  $\mathbb{R}^k$ . Completeness of derivative markets is a sufficient but not necessary condition for the minimum-cost portfolio insurance to be price-independent. Let us denote by  $X$  the subspace of payoffs of all portfolios of securities; then in Aliprantis et al. (2000) it is proved that the minimum-cost insured portfolio exists and is price-independent for every portfolio and at every floor if and only if  $X$  is a lattice-subspace of  $\mathbb{R}^k$ . An equivalent necessary and sufficient condition so that  $X$  is a lattice-subspace is the existence of a positive basis for  $X$ , that is a basis of limited liability payoffs such that every marketed limited liability payoff has a unique representation as a nonnegative linear combination of basis payoffs. The notion of a positive basis for  $X$  is a generalization to incomplete markets of a basis of Arrow securities for complete markets. From the previous discussion, it is evident that the mathematical theory of lattice-subspaces has been used in order to provide a characterization of market structures in which the cost minimizing portfolio is price-independent. In general, the theory of lattice-subspaces has been extensively used in

the last years in Mathematical Economics, especially in the areas of incomplete markets and portfolio insurance (e.g., Aliprantis et al. (1997; 2000; 2002); Polyrakis (2003)) as well as in completion of security markets (Kountzakis & Polyrakis (2006)). The study of finite-dimensional lattice-subspaces is important since many economic models are finite, such as, for example, the well-known Arrow-Debreu model. Additional applications of lattice-subspaces in economics appear in Aliprantis et al. (1998); Henrotte (1992). In this chapter, the main advantage of the computational techniques that we present is that we are able to solve the minimization problem without making use of any linear programming method. This is possible by using the theory of positive bases in vector lattices; specifically, we are able to provide a practical numerical way to check whether a subspace  $X$  is a lattice-subspace or a vector sublattice.

In Polyrakis (1996; 1999), lattice-subspaces and vector sublattices are studied in the space of continuous real valued functions  $C(\Omega)$  defined on a compact Hausdorff topological space  $\Omega$ . In the case where  $\Omega$  is finite, for example  $\Omega = \{1, 2, \dots, k\}$ , then  $C(\Omega) = \mathbb{R}^k$  and the results of Polyrakis (1996; 1999) can be applied for the determination of the lattice-subspaces and vector sublattices of  $\mathbb{R}^k$ . In particular, in Polyrakis (1996) it is provided a solution to the problem of whether a finite collection of linearly independent positive functions in  $C(\Omega)$  generates a lattice-subspace. In addition, he proposed an algorithm under which one can check whether the vector subspace<sup>1</sup>  $X = [x_1, \dots, x_n]$  is a lattice-subspace, where  $x_1, \dots, x_n$  are linearly independent positive functions in  $C(\Omega)$ . Another approach to the same problem of whether  $X$  forms a lattice-subspace of  $\mathbb{R}^k$  is presented in Abramovich et al. (1994).

In Katsikis (2007), based on Abramovich et al. (1994), a computational solution is given to the problem of whether a finite collection of linearly independent, positive vectors of  $\mathbb{R}^k$  generates a lattice-subspace. In addition, in Katsikis (2007), applications to the cost minimization problem that ensures the minimum-cost insured portfolio are discussed. The same reference concludes with a Matlab function which is an elegant and accurate tool in order to provide whether or not a given collection of vectors forms a lattice-subspace.

Also, in Katsikis (2008), a different computational method is presented based upon the Polyrakis algorithm (1996), in order to solve the corresponding problem in  $C[a, b]$ . This computational method implements a general algorithmic process and when slightly modified, this process can also be used in the case of lattice-subspaces of  $\mathbb{R}^k$ . Following this remark, Katsikis (2009) presents the translation followed by the implementation of this algorithm in  $\mathbb{R}^k$  within a Matlab function. This function provides an important tool in order to investigate lattice-subspaces and vector sublattices of  $\mathbb{R}^k$  with direct applications to portfolio insurance. Finally, the results of Katsikis (2009) can be applied in completion of security markets and the theory of efficient funds.

The material in this chapter is spread out in 8 sections. Section 2 gives the fundamental properties of lattice-subspaces and vector sublattices of  $\mathbb{R}^k$  together with the solution to the problem of whether a finite collection of linearly independent, positive vectors of  $\mathbb{R}^k$  generates a lattice-subspace or a vector sublattice. Section 3 studies, in detail, from the computational point of view the mathematical problem stated in Section 2 and presents an efficient computational method in order to solve it. Comparison results with other existing computational methods are also provided. Section 4 studies finite dimensional lattice-subspaces of  $C[a, b]$  and presents the solution to the problem stated in Section 2, in the case where the initial space is  $C[a, b]$ . Section 5 presents computational methods in order to determine finite dimensional lattice-subspaces of  $C[a, b]$ . Section 6 provides the most important interrelationship between lattice-subspaces and the minimization problem of minimum-cost portfolio insurance. Also,

<sup>1</sup>  $[x_1, \dots, x_n]$  denotes the  $n$ -dimensional vector subspace generated by  $x_1, \dots, x_n$ .

the reader will find in this Section a study that involves the computational techniques presented in Section 3 and Section 5 in order to calculate the minimum-cost insured portfolio both in the case of  $\mathbb{R}^k$  and  $C[a, b]$ . Section 7 provides a computational technique, based on Section 3, in order to solve the problem of completion by options of a two-period security market in which the space of marketed securities is a subspace of  $\mathbb{R}^k$ . Methods on computing the efficient funds of the market are also presented. Conclusions and research directions are provided in Section 8.

In this chapter, all the numerical tasks have been performed using the Matlab 7.8 (R2009a) environment on an Intel(R) Pentium(R) Dual CPU T2310 @ 1.46 GHz 1.47 GHz 32-bit system with 2 GB of RAM memory running on the Windows Vista Home Premium Operating System.

**2. Lattice-subspaces and vector sublattices of  $\mathbb{R}^k$**

In this section, a brief introduction is provided to the theory of lattice-subspaces and vector sublattices of  $\mathbb{R}^k$ . In addition, we present the solution to the problem of whether a finite collection of linearly independent, positive vectors of  $\mathbb{R}^k$  generates a lattice-subspace or a vector sublattice.

**2.1 Preliminaries and notation**

We view  $\mathbb{R}^k$  as an ordered space, then the *pointwise order* relation in  $\mathbb{R}^k$  is defined by

$$x \leq y \text{ if and only if } x(i) \leq y(i), \text{ for each } i = 1, \dots, k.$$

The positive cone of  $\mathbb{R}^k$  is defined by  $\mathbb{R}^k_+ = \{x \in \mathbb{R}^k | x(i) \geq 0, \text{ for each } i\}$  and if we suppose that  $X$  is a vector subspace of  $\mathbb{R}^k$  then  $X$  ordered by the pointwise ordering is an *ordered subspace* of  $\mathbb{R}^k$  with positive cone  $X_+$  defined by  $X_+ = X \cap \mathbb{R}^k_+$ . For a two-point set  $S = \{x, y\}$ , we denote by  $x \vee y$  ( $x \wedge y$ ) the *supremum* of  $S$  i.e., its least upper bound (the *infimum* of  $S$  i.e., its greatest lower bound). Thus,  $x \vee y$  ( $x \wedge y$ ) is the componentwise maximum (minimum) of  $x$  and  $y$  defined by

$$(x \vee y)(i) = \max\{x(i), y(i)\} \text{ } ((x \wedge y)(i) = \min\{x(i), y(i)\}), \text{ for all } i = 1, \dots, k.$$

An ordered subspace  $X$  of  $\mathbb{R}^k$  is a *lattice-subspace* of  $\mathbb{R}^k$  if it is a vector lattice in the induced ordering, i.e., for any two vectors  $x, y \in X$  the supremum and the infimum of  $\{x, y\}$  both exist in  $X$ . Note that the supremum and the infimum of the set  $\{x, y\}$  are, in general, different in the subspace than the supremum and the infimum of this set in the initial space. An ordered subspace  $Z$  of  $\mathbb{R}^k$  is a *vector sublattice* or a *Riesz subspace* of  $\mathbb{R}^k$  if for any  $x, y \in Z$  the supremum and the infimum of the set  $\{x, y\}$  in  $\mathbb{R}^k$  belong to  $Z$ . Suppose that  $X$  is an ordered subspace of  $\mathbb{R}^k$  and  $B = \{b_1, b_2, \dots, b_m\}$  is a basis for  $X$ . Then  $B$  is a *positive basis* of  $X$  if the positive cone  $X_+$  of  $X$  has the form,

$$X_+ = \{x = \sum_{i=1}^m \lambda_i b_i | \lambda_i \geq 0, \text{ for each } i\}.$$

Therefore, if  $x = \sum_{i=1}^m \lambda_i b_i$  and  $y = \sum_{i=1}^m \mu_i b_i$  then  $x \leq y$  if and only if  $\lambda_i \leq \mu_i$  for each  $i = 1, 2, \dots, m$ . The existence of positive bases is not always ensured, but in the case where  $X$  is a vector sublattice of  $\mathbb{R}^k$  then  $X$  has always a positive basis. Moreover, it holds that an ordered subspace of  $\mathbb{R}^k$  has a positive basis if and only if it is a lattice-subspace of  $\mathbb{R}^k$ . If  $B = \{b_1, b_2, \dots, b_m\}$  is a positive basis for a lattice-subspace (or a vector sublattice)  $X$  then the

lattice operations in  $X$ , namely  $x \nabla y$  for the supremum and  $x \Delta y$  for the infimum of the set  $\{x, y\}$  in  $X$ , are given by

$$x \nabla y = \sum_{i=1}^m \max\{\lambda_i, \mu_i\} b_i \text{ and } x \Delta y = \sum_{i=1}^m \min\{\lambda_i, \mu_i\} b_i,$$

for each  $x = \sum_{i=1}^m \lambda_i b_i, y = \sum_{i=1}^m \mu_i b_i \in X$ . A vector sublattice is always a lattice-subspace, but the converse is not true as shown in the next example.

**Example 0.1.** Let  $X = [x_1, x_2, x_3]$  be the subspace of  $\mathbb{R}^4$  generated by the vectors  $x_1 = (6, 0, 0, 1), x_2 = (6, 4, 0, 0), x_3 = (8, 4, 2, 0)$ . An easy argument shows that the set  $B = \{b_1, b_2, b_3\}$  where

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 2 & 0 \\ 12 & 8 & 0 & 0 \\ 6 & 0 & 0 & 1 \end{bmatrix}$$

forms a positive basis of  $X$  therefore  $X$  is a lattice-subspace of  $\mathbb{R}^4$ . On the other hand, let us consider the vectors  $y_1 = 2x_1 + x_2 = (18, 4, 0, 2)$  and  $y_2 = x_3 - x_2 = (2, 0, 2, 0)$  of  $X$ . Then,  $y_1 \vee y_2 = (18, 4, 2, 2)$  and since  $y_1 = \frac{1}{2}b_2 + 2b_3, y_2 = b_1$ , it follows that  $y_1 \nabla y_2 = b_1 + \frac{1}{2}b_2 + 2b_3 = (20, 4, 2, 2)$ . Therefore,  $X$  is not a vector sublattice of  $\mathbb{R}^4$ , since the supremum on the subspace  $X$  is different than the supremum on the whole space.

For an extensive presentation of lattice-subspaces, vector sublattices and positive bases the reader may refer to Abramovich et al. (1994); Polyakis (1996; 1999).

**2.2 The mathematical problem**

Suppose that  $\{x_1, x_2, \dots, x_n\}$  is a collection of linearly independent, positive vectors of  $\mathbb{R}^k$ . The problem is, under what conditions the subspace  $X = [x_1, x_2, \dots, x_n]$  is a lattice-subspace or a vector sublattice of  $\mathbb{R}^k$ ?

Let us denote by  $\beta$  the basic function of  $x_1, x_2, \dots, x_n$ , that is,  $\beta : \{1, 2, \dots, k\} \rightarrow \mathbb{R}^k$  such that

$$\beta(i) = \left( \frac{x_1(i)}{z(i)}, \frac{x_2(i)}{z(i)}, \dots, \frac{x_n(i)}{z(i)} \right),$$

for each  $i \in \{1, 2, \dots, k\}$  with  $z(i) > 0$ , where  $z = \sum_{i=1}^n x_i$ . The set

$$R(\beta) = \{\beta(i) | i = 1, 2, \dots, k, \text{ with } z(i) > 0\},$$

is the range of the basic function and the cardinal number,  $cardR(\beta)$ , of  $R(\beta)$  is the number of different elements of  $R(\beta)$ . Let  $cardR(\beta) = m$  then it is clear that  $n \leq m \leq k$ . Denote by  $K$  the convex hull of  $R(\beta)$ . Since  $K$  is the convex hull of a finite subset of  $\mathbb{R}^k$  it is a polytope with  $d$  vertices and each vertex of  $K$  belongs to  $R(\beta)$  therefore  $n \leq d \leq m$ .

Suppose that  $R(\beta) = \{P_1, P_2, \dots, P_m\}$  such that, under a proper enumeration, the vertices  $P_1, P_2, \dots, P_n$  are linearly independent and  $P_1, P_2, \dots, P_d$  are the vertices of  $K$ , i.e.,

$$R(\beta) = \left\{ \overbrace{P_1, P_2, \dots, P_n}^{\text{vertices of } K}, P_{n+1}, \dots, P_d, \dots, P_m \right\}.$$

*linearly independent*

The following theorem, from (Polyakis, 1999), provides a full answer to the stated problem.

**Theorem 0.1.** *Suppose that the above assumptions are satisfied. Then,*

- (i) *X is a vector sublattice of  $\mathbb{R}^k$  if and only if  $R(\beta)$  has exactly  $n$  points (i.e.,  $m = n$ ) and a positive basis  $\{b_1, b_2, \dots, b_n\}$  for  $X$  is defined by the formula*

$$(b_1, b_2, \dots, b_n)^T = A^{-1}(x_1, x_2, \dots, x_n)^T,$$

*where  $A$  is the  $n \times n$  matrix whose  $i$ th column is the vector  $P_i$ , for each  $i = 1, 2, \dots, m$ .*

- (ii) *X is a lattice-subspace of  $\mathbb{R}^k$  if and only if the polytope  $K$  has  $n$  vertices (i.e.,  $d = n$ ) and a positive basis  $\{b_1, b_2, \dots, b_n\}$  for  $X$  is defined by the formula*

$$(b_1, b_2, \dots, b_n)^T = A^{-1}(x_1, x_2, \dots, x_n)^T,$$

*where  $A$  is the  $n \times n$  matrix whose  $i$ th column is the vector  $P_i$ , for each  $i = 1, 2, \dots, d$ .*

### 2.3 The algorithm

The basic steps of an algorithmic process that will accurately implement the ideas of Theorem 0.1 are the following:

- (1) Determine  $R(\beta)$ .
- (2) Compute the number  $m = \text{card}R(\beta)$ , and the number  $d$  of vertices of the polytope  $K$ .
- (3) If  $n = m$  (vector sublattice case) or  $n = d$  (lattice-subspace case) then, determine a positive basis for  $X$ .

Based on a theorem of Edmonds, Lovász and Pulleybank in Edmond et al. (1982), we close this section with some remarks on the existence of a polynomial-time decision procedure, in order to decide whether the collection of vectors  $\{x_1, x_2, \dots, x_n\}$  generates a lattice-subspace or a vector sublattice. We shall present this result, in a suitable form for our analysis, as it is presented in Aliprantis et al. (1997).

**Theorem 0.2.** *There exists a polynomial-time algorithm that for any polytope,  $P$ , defined as the convex hull of a given finite set of vectors, determines the affine hull of  $P$ . Specifically the algorithm finds affinely independent vertices  $u_0, u_1, \dots, u_\ell$  of  $P$  such that*

$$\text{aff}(P) = \text{aff}(\{u_0, u_1, \dots, u_\ell\}).$$

Recall that, algorithms which have a polynomial or sub-polynomial time complexity (that is, they take time  $O(g(n))$  where  $g(n)$  is either a polynomial or a function bounded by a polynomial), are practical. Such algorithms with running times of orders  $O(\log n), O(n), O(n \log n), O(n^2), O(n^3)$  etc. are called polynomial-time algorithms. There are several arguments to support the thesis that “polynomial” is a synonym to practical and the general conclusion is that a problem can be considered “efficiently solved” when a polynomial-time algorithm has been found for it.

In order to implement algorithm 2.3, we shall use the Quickhull algorithm from Barber et al. (1996) for computing the convex hull of a given set of points. According to Barber et al. (1996) (Theorem 3.2) if  $d$  is the dimension,  $n$  is the number of input points,  $r$  the number of processed points, and  $f_r$  the maximum number of facets of  $r$  vertices ( $f_r = O(r^{\lfloor \frac{d}{2} \rfloor} / \lfloor \frac{d}{2} \rfloor!)$ ) then the worst-case complexity of Quickhull is  $O(n \log r)$  for  $d \leq 3$  and  $O(n f_r / r)$  for  $d \geq 4$ .

### 3. The computational method

#### 3.1 Method presentation

In this section we present the translation followed by the implementation of algorithm 2.3 within a Matlab function named `SUBlatSUB` from Katsikis (2009). This function provides an important tool in order to investigate lattice-subspaces and vector sublattices of  $\mathbb{R}^k$  since we are able to perform fast testing for a variety of dimensions and subspaces. Recall that, the numbers  $n, m, d, k$  denote the dimension of  $X$ , the cardinality of  $R(\beta)$ , the number of vertices of the convex hull of  $R(\beta)$  and the dimension of the initial Euclidean space, respectively.

The function `SUBlatSUB` first checks if the given collection of vectors generates a vector sublattice by examining the validity of condition (i) of Theorem 0.1. In the case of a vector sublattice, i.e.,  $m = n$ , the program responds with the output:

```
vector sublattice
```

followed by a  $n \times k$  matrix whose rows are the vectors of the positive basis.

If, instead, the collection does not generate a vector sublattice, that is  $m \neq n$ , then the function `SUBlatSUB` checks if the given collection generates a lattice-subspace by examining the validity of condition (ii) of Theorem 0.1. In the case of a lattice-subspace, i.e.,  $d = n$ , the program responds with the output:

```
lattice-subspace
```

followed by a  $n \times k$  matrix whose rows are the vectors of the positive basis.

If  $m \neq n$  and  $d \neq n$  then the program responds with the output:

```
not a lattice-subspace
ans=
     []
```

So, in order to decide whether a given collection of linearly independent, positive vectors generates a lattice-subspace or a vector sublattice of  $\mathbb{R}^k$ , we construct a matrix whose columns are the vectors of the given collection and then we apply the function `SUBlatSUB` on that matrix. It is possible to produce the numbers  $n, m, d, k$ , with this order, as a  $4 \times 1$  matrix with the following code,

```
>> [positivebasis, dimensions]=SUBlatSUB(a)
```

where `a` is the matrix whose columns are the given vectors.

#### 3.2 Numerical examples

In order to describe the most important features of `SUBlatSUB`, we illustrate some examples featured in Katsikis (2007) for various collections and dimensions. Also, we close this section with comparison results of the `SUBlatSUB` function and the alternative function, namely `K` function, presented in Katsikis (2007).

**Example 0.2.** Consider the following 7 vectors  $x_1, x_2, \dots, x_7$  in  $\mathbb{R}^{10}$ ,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

According to the definition of the  $\beta$  function, the rows of the following matrix  $u$  are the different elements of  $R(\beta)$ ,

```
u =
0 0 0 0 0 0 1
0 0 0 0 0 0.5 0.5
0 0 0 0 1 0 0
0 0 0 1 0 0 0
0 0 1 0 0 0 0
0 0.5 0 0 0 0 0.5
0 1 0 0 0 0 0
0.5 0 0 0 0.5 0 0
```

Thus,  $m = 8$  and it is clear that rows  $u(1), \dots, u(5), u(7), u(8)$  of  $u$  are linearly independent. This means that these vectors belong to the convex hull of  $R(\beta)$ . Also, it is easy to see that row  $u(6)$  of  $u$  is a convex combination of the other rows. Therefore,  $d = 7$  and  $X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]$  is a lattice-subspace.

For a numerical solution, we invoke the `SUBlatSUB` function by typing in the command window of the Matlab environment:

```
>> [positive basis, dimensions]=SUBlatSUB(a)
```

The results, then, are as follows:

```
lattice-subspace
positivebasis =
0 1 0 0 0 0 0 0 1
0 0 0 0 0 0 2 0 0
0 0 0 0 1 1 0 0 1
0 0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0
2 0 0 0 0 0 0 0 0
dimensions =
7
8
7
10
```

We conclude with some comments based on the results of the `SUBlatSUB` function. A positive basis is unique in the sense of positive multiples since each element of the basis is an extremal<sup>2</sup> point of the positive cone of the subspace. If we denote by  $\{b_1, b_2, \dots, b_7\}$  the positive basis that we obtained by using the `K` function (see Katsikis (2007)) and by  $\{B_1, B_2, \dots, B_7\}$  the positive basis we found with the `SUBlatSUB` function then it holds

$$(B_1, B_2, B_3, B_4, B_5, B_6, B_7) = (b_7, 2b_5, b_4, b_3, b_2, b_6, 2b_1).$$

**Example 0.3.** Consider the following 7 vectors  $x_1, x_2, \dots, x_7$  in  $\mathbb{R}^{10}$ ,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 4 & 3 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 2 & 3 & 1 & 3 & 4 & 4 \\ 3 & 3 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 \end{bmatrix}$$

where following the same procedure, as before, one gets

```
vector sublattice
positivebasis =
    0     0     0     0     0     0    12     0     0     0
    0     0     0     0     3     0     0     0     0     0
    0     0     0     0     0     4     0     4     0     0
    0     0     0     0     0     0     0     0    12    12
    6     6     0     0     0     0     0     0     0     0
    0     0     0     5     0     0     0     0     0     0
    0     0     6     0     0     0     0     0     0     0
dimensions =
    7
    7
    10
```

If we denote by  $\{b_1, b_2, \dots, b_7\}$  the positive basis that we obtained by using the `K` function and by  $\{B_1, B_2, \dots, B_7\}$  the positive basis we found with the `SUBlatSUB` function then it holds

$$(B_1, B_2, B_3, B_4, B_5, B_6, B_7) = (12b_6, 3b_4, 4b_5, 12b_7, 6b_1, 5b_3, 6b_2).$$

**Example 0.4.** Consider the following 5 vectors  $x_1, x_2, \dots, x_5$  in  $\mathbb{R}^{10}$ ,

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

For the above set, the program yields

<sup>2</sup> 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  $\lambda$ .



```

lattice-subspace
positivebasis =
  0  0  0  0  0  0  0  8  0  0
  0  7/4  0  7  7/4  7/4  7/4  0  0  0
  6  3/2  6  0  3/2  3/2  3/2  0  0  0
  0  0  0  0  0  0  0  0  0  6
  0  7/4  0  0  7/4  7/4  7/4  0  7  0
dimensions =
  5
  6
  5
  10
    
```

In this case it holds

$$(B_1, B_2, B_3, B_4, B_5) = (8b_3, 7b_2, 6b_1, 6b_5, 7b_4).$$

For the purpose of monitoring the performance, we present in Table 1 the execution times of the SUBlatSUB function and the method presented in Katsikis (2007) (**K** function).

Matlab functions	Example 0.2	Example 0.3	Example 0.4
SUBlatSUB	0.052	0.030	0.035
<b>K</b>	0.516	0.828	0.969

Table 1. Time in seconds

### 3.3 The case of coplanar points

The correct performance of the SUBlatSUB function requires the use of the **convhulln** Matlab function which is based on Qhull<sup>3</sup> and Qhull implements the Quickhull algorithm (Barber et al. (1996)) for computing the convex hull of a given set of points. Suppose that  $a$  denotes the matrix whose rows are the coefficients of the given points, then the **convhulln** function returns the indices of the points in  $a$  that comprise the facets of the convex hull of  $a$ . The **convhulln** function is facing problems during the calculation of the convex hull of points that lie in a  $q$ -manifold, with  $q \leq n - 1$ , in the  $n$ -dimensional space of the given data. So, we cannot use **convhulln** to solve our problem directly.

We illustrate the details in this case through the following example and we also provide an improvement technique to solve the resultant problem in this particular case.

**Example 0.5.** Consider the following four vectors  $x_1, x_2, x_3, x_4$  in  $\mathbb{R}^7$ ,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 0 & 1 & 1 & 4 \\ 0 & 1 & 1 & 1 & 1 & 0 & 2 \\ 2 & 1 & 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

Following the second step of algorithm 2.3, the calculation of the convex hull of  $R(\beta)$  is required to check whether  $X = [x_1, x_2, x_3, x_4]$  is a lattice-subspace or a vector sublattice of  $\mathbb{R}^7$ .

<sup>3</sup> For information about Qhull see <http://www.qhull.org/>

In this case, and after the necessary calculations it is clear that  $R(\beta) = \{P_1, P_2, P_3, P_4, P_5, P_6\}$  where

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{4} & 0 & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}.$$

Therefore, in order to determine the convex hull of  $R(\beta)$  we have used the **convhulln** function. In this case the **convhulln** function yields with the following warning message:

```
??? qhull precision warning: The initial hull is narrow (cosine of
min. angle is 1.0000000000000002). A coplanar point may lead to a
wide facet.
```

If we use the following simple rank test:

```
>> y=[0 1/3 1/3 1/3;1/4 0 1/2 1/4;1/2 0 1/2 0;1/3 1/3 0 1/3;
1/2 1/4 1/4 0;1/4 1/4 1/4 1/4];
>> rank(bsxfun(@minus,y,y(6,:)))
```

then one gets

```
ans =
     3
```

Thus, by our previous analysis it is clear that the points  $P_1, P_2, P_3, P_4, P_5, P_6$  of  $\mathbb{R}^4$  lie in a 3-manifold and we cannot use **convhulln** to solve our problem directly. A solution to this problem can be given under the following methodology:

- Translate the points to a hyper-plane that passes through the origin.
- Determine a set of basis vectors for the subspace.
- Transform the points into an equivalent lower dimensional space.
- Form the convex hull triangulation in the lower dimensional space.

Let us describe, in detail, the procedure for this particular example. First, one has to translate the given points to a hyper-plane that passes through the origin by subtracting one of the vectors from the others,

```
>> ytrans = bsxfun(@minus,y,y(6,:))
ytrans =
-0.2500    0.0833    0.0833    0.0833
     0   -0.2500    0.2500     0
 0.2500   -0.2500    0.2500   -0.2500
 0.0833    0.0833   -0.2500    0.0833
 0.2500     0         0   -0.2500
     0         0         0         0
```

Then, we form an orthonormal basis for the range of **ytrans**.

```
>> rot = orth(ytrans')
rot =
    0.5   -0.64505    0.28967
    0.5   -0.28967   -0.64505
   -0.5    0.64505   -0.28967
    0.5    0.28967    0.64505
```

Recall that, if  $\{v_1, v_2, \dots, v_r\}$  is an orthonormal basis for a finite dimensional subspace  $W$  of an inner product space  $V$  and  $u$  is any vector of  $V$ , then the projection of the vector  $u$  in  $W$  is given by the formula  $proj_W u = \langle u, v_1 \rangle v_1 + \dots + \langle u, v_r \rangle v_r$ .

Now, we project the points into an equivalent lower dimensional space where `rot` is a basis for this space. Hence,

```
>> yproj = ytrans*rot
yproj =
    0.16667    0.21502   -0.096557
   -0.25     0.23368    0.088845
   -0.5     3.0531e-16  5.5511e-17
    0.16667   -0.21502    0.096557
   -0.25    -0.23368   -0.088845
    0         0         0
```

Note that, the rows of `yproj` matrix are the coordinates of the initial points in terms of the basis in the lower dimensional space.

Finally, we form the convex hull triangulation in the projected subspace `yproj`. That is,

```
>> tri = convhulln(yproj)
tri =
    1 2 3
    2 4 3
    4 2 1
    5 1 3
    4 5 3
    5 4 1
```

Since we are only interested for the number of vertices of the convex hull of  $R(\beta)$ , we can only determine the number of vertices of the convex hull in the projected subspace. Therefore,

```
>> length(unique(tri(:)))
ans =
    5
```

So, there are 5 vertices in the hull. This procedure is included in the `SUBlatSUB` function therefore, for a direct answer in the previous example, one can apply the `SUBlatSUB` function directly to the given collection by using the code,

```
>> [positive basis, dimensions]=SUBlatSUB(a)
```

the results, then, are as follows:

```

not a lattice-subspace
positivebasis =
      []
dimensions =
      4
      6
      5
      7
    
```

where  $a$  denotes a matrix that has the vectors  $x_i, i = 1, 2, 3, 4$  as columns.

### 3.4 Comparison results

In this section, we compare the performance of the `SUBlatSUB` function to that of the `K` function. The numerical method, based on the introduction of the `SUBlatSUB` function, enables us to perform fast and accurate estimations of the lattice-subspace or the vector sublattice for a finite collection of positive, linearly independent vectors of  $\mathbb{R}^k$  for a variety of dimensions. For this purpose we have used the Matlab function `rand` in order to produce 50 full rank matrices for each rank  $n, n = 3, \dots, 30$ . The cumulative results are presented in Figure 1 (Figure 1 shows the time efficiency curves, i.e., the rank of the 50 tested matrices versus the total computation time (in seconds)) and in Table 2. From the previous results (see Figure 1, Table 2) it is evident

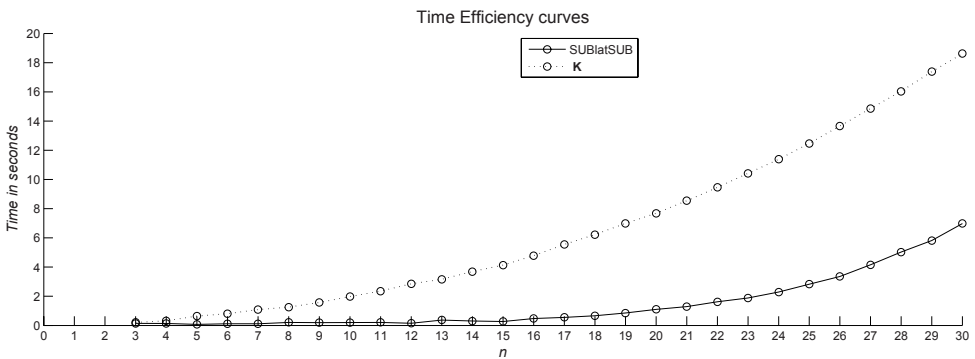


Fig. 1. Time efficiency curves for the `K` function and the `SUBlatSUB` function

that using the `SUBlatSUB` function the interested user can reach a fast computational solution using a reduced amount of computational resources.

### 4. Finite dimensional lattice-subspaces of $C[a, b]$

In what follows we shall denote by  $C[a, b]$  the space of all continuous real functions defined on the interval  $[a, b]$ . As in the case of  $\mathbb{R}^k$ , lattice-subspaces of  $C[a, b]$  are subspaces which are vector lattices in the induced ordering, i.e., for any two vectors  $x, y$  of the subspace the supremum and the infimum of the set  $\{x, y\}$  both exist in the subspace. Recall that the supremum and the infimum of the set  $\{x, y\}$  are, in general, different in the subspace than the supremum and the infimum of this set in the initial space. In this section we present a brief introduction to the theory of lattice-subspaces in  $C[a, b]$ . In addition, we describe in detail the construction of a powerful and efficient package for the translation, into the language of computing, of the

Rank	SUBlatSUB (Time in seconds)	K (Time in seconds)	Rank	SUBlatSUB (Time in seconds)	K (Time in seconds)
3	0.141	0.205	17	0.553	5.548
4	0.140	0.322	18	0.665	6.230
5	0.073	0.641	19	0.853	6.992
6	0.115	0.805	20	1.100	7.681
7	0.113	1.090	21	1.290	8.545
8	0.207	1.257	22	1.620	9.457
9	0.191	1.572	23	1.882	10.420
10	0.198	1.980	24	2.292	11.382
11	0.209	2.358	25	2.823	12.470
12	0.153	2.858	26	3.353	13.662
13	0.371	3.161	27	4.154	14.854
14	0.308	3.694	28	5.030	16.040
15	0.272	4.135	29	5.815	17.384
16	0.477	4.781	30	6.978	18.617

Table 2. Results for 50 tested full rank matrices for each rank  $n, n = 3, \dots, 30$ .

mathematical problem of whether the vector subspace  $X = [x_1, \dots, x_n]$  is a lattice-subspace of  $C[a, b]$ , where  $x_1, \dots, x_n$  are linearly independent positive functions in  $C[a, b]$ .

**4.1 Preliminaries and notation**

Let  $C[a, b]_+$  be the positive cone of  $C[a, b]$  and assume that  $X$  is a subspace of  $C[a, b]$ . The *induced ordering* on  $X$  is the ordering defined by  $X_+ = X \cap C[a, b]_+$  (induced cone of  $X$ ). An *ordered subspace* of  $C[a, b]$  is a subspace of  $C[a, b]$  under the induced ordering. A *lattice-subspace* of  $C[a, b]$  is an ordered subspace  $X$  of  $C[a, b]$  which is a vector lattice in its own, that is, for each  $x, y \in X$  the supremum and the infimum of the set  $\{x, y\}$  exists in  $X$ . If  $X$  is a lattice-subspace of  $C[a, b]$  then we will denote by  $x \nabla y$  the supremum of the set  $\{x, y\}$  in  $X$ . Similarly,  $x \Delta y$  stands for the infimum of the set  $\{x, y\}$  in  $X$ . If  $x \vee y$  denotes the supremum and  $x \wedge y$  the infimum in  $E$  of the set  $\{x, y\}$  and we suppose that  $x \Delta y, x \wedge y, x \vee y, x \nabla y$  exists, then it follows that

$$x \Delta y \leq x \wedge y \leq x \vee y \leq x \nabla y \tag{1}$$

For example, consider  $C[0, 1]$ , the space of all continuous real functions in the interval  $[0, 1]$  and  $X = \{ax + b | a, b \in \mathbb{R}\}$ . Then  $X$  is a lattice-subspace of  $C[0, 1]$  and (1) holds for each  $x, y \in X$  (Figure 2).

One of the most serious difficulties in the study of lattice-subspaces comes from the fact that the supremum and the infimum depend both on the subspace.

For a general definition of a positive basis, let  $E$  be a (partially) ordered Banach space. Then a sequence  $\{e_n\}$  of positive vectors of  $E$  is a *positive basis* if it is a Schauder basis of  $E$  and

$$E_+ = \{x = \sum_{i=1}^{\infty} \lambda_i e_i \in E | \lambda_n \geq 0, \text{ for all } n \in \mathbb{N}\}.$$

Equivalently, one can say that  $\{e_n\}$  is a positive basis of  $E$  if

$$x = \sum_{i=1}^{\infty} \lambda_i e_i \geq 0 \Leftrightarrow \lambda_n \geq 0, \text{ for all } n \in \mathbb{N}.$$

Let  $Y$  be a closed subspace of  $E = C[a, b]$  with basis  $\{b_n\}$  (not necessarily positive). Fix  $t \in [a, b]$  and  $m \in \mathbb{N}$ . Following the terminology introduced in Polyrakis (1996), if  $b_m(t) \neq 0$  and

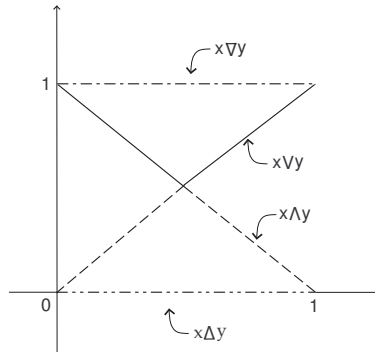


Fig. 2. Relation (1) for  $x=t, y=1-t, t \in [0,1]$ .

$b_n(t) = 0$  for each  $n \neq m$ , then we shall say that the point  $t$  is an  $m$ -node (or simply a node) of the basis  $\{b_n\}$ . If for each  $n$  there exists an  $n$ -node  $t_n$  of the basis  $\{b_n\}$ , then we shall say that  $\{b_n\}$  is a basis of  $Y$  with nodes and that  $t_n$  is a sequence with nodes of  $\{b_n\}$ . If  $\dim Y = n$  and for each  $m \in \{1, 2, \dots, n\}$  there exists an  $m$ -node  $t_m$  of the basis of  $Y$ , then we shall say that  $\{b_1, b_2, \dots, b_n\}$  is a basis of  $Y$  with nodes and that the points  $t_1, t_2, \dots, t_n$  are nodes of the basis  $\{b_1, b_2, \dots, b_n\}$ . The support of a function  $x \in C[a, b]$  is the closure of the set  $\{t \in [a, b] : x(t) > 0\}$  and shall be denoted by  $\text{supp}x$ .

**4.2 The mathematical problem**

In this section, we present the method developed in Polyrakis (1996), for the determination of the finite-dimensional lattice-subspaces of  $C[a, b]$  and we shall discuss the necessary and sufficient conditions for a collection of linearly independent, positive functions,  $x_1, x_2, \dots, x_n$  of  $C[a, b]$  to generate a lattice-subspace. Recall that the Wronski determinant of the functions  $x_i, i = 1, \dots, n$  is the  $n \times n$  determinant which  $i$ th row is constituted of the  $(i - 1)$ th derivatives of the functions  $x_i$ . Our first approach to the problem is given through the following Wronskian criterion:

**Theorem 0.3.** Consider the closed interval  $[a, b]$  of  $\mathbb{R}$  and  $\dim X > 2$ , where  $X = [x_1, x_2, \dots, x_n]$ . Suppose that  $(c, d)$  is an open interval of  $\mathbb{R}$  which contains  $[a, b]$ . If the functions  $x_i$  have continuous derivatives up to the  $n$ th order in  $(c, d)$  and the Wronskian of the functions  $x_i$  is nonzero for any point of  $(c, d)$ , then  $X$  is not a lattice-subspace of  $C[a, b]$ .

As in the case of  $\mathbb{R}^k$ , let  $x_1, \dots, x_n$  in  $C[a, b]$ , we shall denote by  $z$  the sum  $z = \sum_{i=1}^n x_i$  and by  $\beta$  the function  $\beta : [a, b] \rightarrow \mathbb{R}^n$  such that

$$\beta(t) = \left( \frac{x_1(t)}{z(t)}, \frac{x_2(t)}{z(t)}, \dots, \frac{x_n(t)}{z(t)} \right)$$

for each  $t \in [a, b]$  with  $z(t) > 0$ . We shall refer to  $\beta$  as the basic curve of the functions  $x_1, x_2, \dots, x_n$ . Also, we shall denote by  $D(\beta)$  the domain and by  $R(\beta)$  the range of the basic curve  $\beta$  of  $x_1, x_2, \dots, x_n$ . If  $K$  is a subset of  $\mathbb{R}^n$  then we shall denote by  $\bar{K}$  the closure of  $K$ , by  $\text{int}(K)$  the interior of  $K$  and by  $\partial K$  the boundary of  $K$ . We shall denote by  $\text{co}(K)$  the convex hull of  $K$  and by  $\overline{\text{co}}(K)$  the closure of  $\text{co}(K)$ .

The following theorem, is a criterion for lattice-subspaces and provides a full answer to the problem of whether a collection of positive functions  $x_1, x_2, \dots, x_n$  of  $C[a, b]$  generates a lattice-subspace. In addition, if we are in the case of a lattice-subspace then the theorem determines a positive basis for  $X = [x_1, x_2, \dots, x_n]$ .

**Theorem 0.4.** *The following statements are equivalent,*

- (i)  $X$  is a lattice-subspace of  $C[a, b]$ .
- (ii) There exist  $n$  linearly independent  $P_1, P_2, \dots, P_n$  vectors in  $\mathbb{R}^n$ , belonging to the closure of the range of  $\beta$  such that for each  $t \in D(\beta)$  the vector  $\beta(t)$  is a convex combination of the vectors  $P_1, P_2, \dots, P_n$ , i.e.,  $R(\beta) \subseteq \text{co}(\{P_1, P_2, \dots, P_n\})$ .

If (ii) is true,  $P_i = \lim_{v \rightarrow \infty} \beta(\omega_{iv})$  for each  $i$ ,  $A$  is the  $n \times n$  matrix whose  $i$ th column is the vector  $P_i$  and  $b_1, b_2, \dots, b_n$  are the functions defined by the formula

$$(b_1(t), b_2(t), \dots, b_n(t)) = A^{-1}(x_1(t), x_2(t), \dots, x_n(t))^T,$$

then  $X$  has the following properties:

- (a) The set  $\{b_1, b_2, \dots, b_n\}$  is a positive basis of  $X$ . In addition, if  $t_i$  is a limit point of the sequence  $\{\omega_{iv} : v = 1, 2, \dots\}$ , then  $t_i \in \text{suppb}_i$  and  $b_k(t_i) = 0$ , for each  $k \neq i$ .
- (b) The closed convex hull of  $R(\beta)$  and the convex polygon with vertices the points  $P_1, P_2, \dots, P_n$  coincide.
- (c) If  $P_k = \beta(t_k)$ , then  $t_k$  is a  $k$ -node of the basis  $\{b_1, b_2, \dots, b_n\}$ .
- (d) If  $P_k = \beta(t_k)$  for some interior point  $t_k$  of  $[a, b]$  and  $x_i$  are  $C^2$ - functions in a neighborhood of  $t_k$ , then

$$\beta'(t_k) = \mathbf{0}.$$

The set  $E(\beta)$  is the extreme subset of the basic curve  $\beta$  if there exists a subset  $G$  of  $\overline{R(\beta)}$  consisting of  $n$  linearly independent vectors such that  $R(\beta) \subseteq \text{co}(G)$ , then we put  $E(\beta) = G$ , otherwise we put  $E(\beta) = \emptyset$ .

From Theorem 0.4 and the preceding definition the following proposition should be immediate.

**Proposition 0.1.** *The subspace  $X$  satisfies the properties*

- (i)  $X$  is a lattice-subspace if and only if  $E(\beta) \neq \emptyset$ .
- (ii) If  $\beta(t) \in E(\beta)$ , then  $t$  is a node of the positive basis of  $X$ .
- (iii)  $X$  has a positive basis with nodes if and only if  $E(\beta)$  is a nonempty subset of  $R(\beta)$ .

From Theorem 0.4 it is evident that if  $P \in E(\beta)$  and  $P \notin R(\beta)$ , then we have that  $P = \lim_{v \rightarrow \infty} \beta(t_v)$ , where  $t_v$  is a sequence of  $D(\beta)$  having all limit points in the boundary  $\partial D(\beta)$  of  $D(\beta)$ .

So, the limit set  $L(\beta)$  of the curve  $\beta$  is defined as follows:

$$L(\beta) = \{P \in \mathbb{R}^n : \exists \{t_v\} \subseteq D(\beta) \text{ with its limit points in } \partial D(\beta), P = \lim_{v \rightarrow \infty} \beta(t_v)\}.$$

Also, if  $a, b \in D(\beta)$  then we shall denote by  $\beta(\partial[a, b])$  the set

$$\beta(\partial[a, b]) = \{\beta(a), \beta(b)\}.$$

If  $t$  is an interior point of  $[a, b]$  and  $\beta(t) \in E(\beta)$ , then  $t$  is a root of the equation

$$\beta'(t) = 0, \quad (2)$$

and we shall denote by  $I(\beta)$  the images of the roots of equation (2), i.e.,

$$I(\beta) = \{\beta(t) : t \in \text{Int}([a, b]) \cap D(\beta) \text{ and } t \text{ is root of the equation (2)}\}.$$

Any subset of  $L(\beta) \cup I(\beta) \cup \beta(\partial[a, b])$  consisting of  $n$  linearly independent vectors will be called a *possible extreme subset* of  $\beta$ .

**Proposition 0.2.** *If the functions  $x_1, x_2, \dots, x_n$  are  $C^2$ -functions in the set  $\text{Int}([a, b]) \cap D(\beta)$ , then  $E(\beta) \subseteq L(\beta) \cup I(\beta) \cup \beta(\partial[a, b])$ .*

The set  $\beta(\partial[a, b])$  is known because  $\partial[a, b] = \{a, b\}$  and if  $D(\beta) = [a, b]$  then  $L(\beta) = \emptyset$ . In addition, if we assume that the domain of  $\beta$  has the form

$$D(\beta) = [a, t_1) \cup (t_1, t_2) \cup \dots \cup (t_{n-1}, t_n) \cup (t_n, b]$$

and the limits

$$P_i = \lim_{t \rightarrow t_i} \beta(t)$$

exist for each  $i$ , then

$$L(\beta) = \{P_1, P_2, \dots, P_n\}.$$

In view of Proposition 0.2 one has to determine the set  $L(\beta) \cup I(\beta) \cup \beta(\partial[a, b])$  and then must investigate when one of the possible extreme subsets of  $\beta$  is indeed an extreme subset of  $\beta$ . The details are included in the next algorithm.

### 4.3 The algorithm

Based upon Theorem 0.3, Theorem 0.4 and the discussion in Subsection 4.2, next, we illustrate the steps of an algorithm in order to decide whether the collection  $\{x_1, x_2, \dots, x_n\}$  generates a lattice-subspace.

- (1) Does the Wronskian of the functions  $x_1, x_2, \dots, x_n$  have at least one root in the interval  $[a, b]$ ?
- (2) Determine the sets  $L(\beta), I(\beta), \beta(\partial[a, b])$  and the possible extreme subsets of  $\beta$ .
- (3) Is one of the possible extreme subsets an extreme subset of  $\beta$ ?
- (4) If step (3) holds, determine a positive basis of  $X$ .

## 5. The computational method

### 5.1 Method presentation

In this section, we present a procedure that will accurately implement the ideas of algorithm 4.3 while the main concern is to further calculate the positive basis (if one exists) in order to provide an exact description of the lattice-subspace. So, the first step of our approach consists of describing the functionality of the functions **wr**, **V**, **L**, **I**, **sisets** and **xitest** from Katsikis (2008).

According to Theorem 0.3, function **wr** checks if the Wronskian of the given collection  $\{x_1, x_2, \dots, x_n\}$  of  $C[a, b]$  has at least one root in the interval  $[a, b]$ . In addition, the **wr** function provides the roots (if there exist any) of the Wronskian. So, in this case the program responds with the output:



The Wronskian has at least one root in  $[a, b]$

and it yields the roots of the Wronskian. If, instead, the Wronskian does not have any roots in the interval  $[a, b]$ , then the program provides only the roots outside the interval  $[a, b]$  (if there exist any).

Suppose that the given collection passes the Wronskian test, then we try to determine the possible extreme subsets of the basic curve  $\beta$  starting with the computation of the set  $\beta(\partial[a, b])$ . So, in our next step we call the function **V**. Function **V** first checks whether there are any real roots of the function  $z(t) = \sum_{i=1}^n x_i$ . If that is the case it displays the message

Possible non empty limit set

so that we can continue in order to determine the limit set of the curve  $\beta$ . Function **V** responds with a matrix whose columns are the elements of the set  $\beta(\partial[a, b])$ .

In the case of a non empty limit set, we use the function **L** in order to determine the limit set of the curve  $\beta$ . The output of the function **L** is a matrix whose columns are the elements of the set  $L(\beta)$ .

In order to determine the set

$$I(\beta) = \{\beta(t) : t \in \text{Int}([a, b]) \cap D(\beta) \text{ and } t \text{ is root of the equation (2)}\},$$

we use the function **I**. The function **I** provides a matrix whose columns are the elements of the set  $I(\beta)$ .

Suppose that  $\{P_1, \dots, P_n\}$  is a possible extreme subset of  $\beta$  and

$$\beta(t) = \zeta_1(t)P_1 + \dots + \zeta_n(t)P_n.$$

In order to prove that  $\{P_1, \dots, P_n\}$  is an extreme subset of  $\beta$  we must show that  $\zeta_i(t) \geq 0$ , for each  $i$  and each  $t \in [a, b]$ . So, for the next step in our approach, we need to construct all the possible extreme subsets of  $\beta$  and check whether there exists an extreme subset of  $\beta$ . To this end, we make use of the function **sisets** in order to generate all the possible extreme subsets of the curve  $\beta$ . Note that, **sisets** calls automatically the function **xitest** in order to determine the domain where each one of the  $\zeta_i(t)$  are negative. Let us denote by  $\Delta_i$  the domain of negativity that corresponds to the function  $\zeta_i(t)$ , for  $i = 1, \dots, n$  then, if for at least one of the  $\zeta_i(t)$ ,  $\Delta_i$  has non empty intersection with the interval  $[a, b]$ , the set  $\{P_1, \dots, P_n\}$  is not an extreme subset of  $\beta$ . In the case where an extreme subset exists we determine the positive basis by using the formula

$$(b_1(t), b_2(t), \dots, b_n(t)) = A^{-1}(x_1(t), x_2(t), \dots, x_n(t))^T,$$

from Theorem 0.4.

### 5.2 Numerical examples

For the purpose of monitoring the performance, in the following we present some examples in  $C[a, b]$  for various collections of functions together with the time responses we obtained when running these examples (Table 3).

**Example 0.6.** Let  $x_1(t) = t^2 - 2t + 2$ ,  $x_2(t) = -t^3 + 2t^2 - t + 2$  and  $x_3(t) = t^3 - 3t^2 + 3t$  and  $X$  be the subspace of  $C[0, 2]$  generated by the functions  $x_1, x_2, x_3$ .

Our first step consists of loading the data of the problem by using the following commands:

## Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

