On the Use of Matrices for Belief Revision

Giorgos Flouris and Dimitris Plexousakis

Institute of Computer Science, F.O.R.T.H.
F.O. Box 1385, GR 71110, Heraklion, Greece
{fgeo, dp}@ics.forth.gr

Abstract. A most crucial problem in knowledge representation is the revision of knowledge when new, possibly contradictory, information is obtained (belief revision). In this paper, this problem is addressed for propositional knowledge bases. A new, more expressive representation of propositional expressions is introduced, which uses 2-dimensional complex matrices to store knowledge. The increased expressiveness of this representation can be exploited for devising a solution to the problem of belief revision. A simple method for belief revision is proposed, and the new problems and opportunities arising for query answering under this new representation are addressed. Finally, some results regarding matrices are presented as well as their connection with logic.

1 Introduction

The problem of revising beliefs is the problem of maintaining a knowledge base (KB) so that the information contained in it is as accurate as possible in the face of changes in the modelled world and incomplete (or even faulty) information. This problem is greatly interwoven with the representation of knowledge; before deciding on a proper belief revision method, we should choose a proper knowledge representation scheme [2], [15].

The updating methods of knowledge are by no means obvious, even when we are concerned with the intuitive processes only. Let us consider the simple example of knowing facts \( \alpha \) and \( \alpha \rightarrow \beta \). One obvious implication of our knowledge is \( \beta \) (by *modus ponens*), which could be inserted into our KB as a new fact. Let us now consider the negation of \( \beta \) (\( \neg \beta \)) entering into the base as a new piece of knowledge (revision). This contradicts our previous assumption that \( \beta \) is true, forcing us to give up some (or all) of our previous beliefs or end up with an inconsistent KB. Alternatively, we could reject the revision as non-valid. Even in this trivial example, the approach to be taken is not at all clear. Extra-logical factors should be taken into account, like the source and reliability of each piece of information or some kind of bias towards or against revisions.

Such problems have been addressed by philosophers, computer scientists, logicians and others, in an effort to provide an intuitively correct method of revising beliefs. In this paper, we introduce a new representation of propositional expressions using complex matrices. Based on it, we devise an elegant solution to the problem of belief revision and explore some of its properties, as well as its relation with proposed algorithms from the literature.
2 Matrices and Knowledge Representation

Our method for addressing the problem of belief revision uses a special function named Table Transformation (TT) function, which transforms an expression of any finite propositional language into a complex matrix. The power of this representation stems from the concept of Reliability Factor (RF), which is a real number indicating the level of confidence in a belief.

We will first describe how the RF assignment is made on atoms. In any formula, say \( p \), an atom of the language, say \( \alpha \), can occur either as a positive (\( \alpha \)) or as a negative literal (\( -\alpha \)); alternatively, it may not occur at all. If \( \alpha \) does appear in \( p \), its occurrence indicates belief in the atom itself (\( \alpha \)) or its negation (\( -\alpha \)), or, equivalently, our confidence that, in the real world, \( \alpha \) is true or false respectively; RF indicates the level of this belief. RF is a real number, attached to all atoms’ truth and all atoms’ falsehood and may be positive, negative or zero. Using RF, all knowledge related to one occurrence of a given atom in a given proposition can be expressed using a pair of real numbers \((x, y)\), representing the RFs of the atom’s truth and falsehood respectively.

By using the well known isomorphism of the set \( \mathbb{R}^2 \) to the set of complex numbers \( \mathbb{C} \), we can equivalently express an atom’s knowledge using the complex number \( z = x + yi \). This number will be referred to as the atom’s knowledge, or the number attached to the atom, and we will use atoms and complex numbers interchangeably. The pair of RFs could be equivalently used (or any other system that pairs numbers), but the more compact form of complex numbers gives us more flexibility and allows us to directly use the operations studied for complex numbers and matrices.

The complex number so defined can express a variety of information for each atom, depending on the real and imaginary part of \( z = x + yi \in \mathbb{C} \). If \( x = 0 \), we know nothing about the truth of the respective atom. In a different case, the sign of \( x \) indicates whether we believe \( (x > 0) \) or not \( (x < 0) \) the atom’s truth, while its absolute value indicates the level of trust/distrust. Similar comments apply for \( y \) and the atom’s falsehood.

It must be noted that distrust in one proposition is not equivalent to believing that proposition’s negation, as far as belief revision is concerned. Disbelieving (negative RF) refers to retraction of knowledge. For example, retracting \( \alpha \) (ie disbelieving \( \alpha \)) from \( \alpha \land b \) should result in \( b \) while revising the same KB with \( -\alpha \) would lead to inconsistency, which should be remedied somehow to get an “acceptable” KB. Thus, a negative RF attached to the truth of an atom is not equivalent to a positive RF attached to its falsehood (and vice-versa). In older works of the authors, like [6], RF was a non-negative real number, thus unable to express disbelief; the current definition is more powerful.

By combining different values (sign and absolute value) of \( x \) and \( y \) we can express different types of knowledge regarding the atom. If \( x = y = 0 \), we have no knowledge on the atom’s state, so we must accept both the atom’s affirmation or negation as there is no reason to disallow any of the two. This case may occur when the atom does not appear in the proposition at hand. If one of \( x, y \) is 0, we know nothing about the truth (if \( x = 0 \)) or falsehood (if \( y = 0 \)) of the
atom, so our knowledge must rely on the non-zero element. Such numbers are assigned to literals; a positive literal indicates no knowledge regarding the atom’s falsehood, so \( y = 0 \); similarly, for a negative literal, \( x = 0 \). On the other hand, if both \( x > 0 \) and \( y > 0 \), then we believe in both the atom’s affirmation and negation. This implies a contradiction whose intensity is a function of \( x \) and \( y \); by comparing \( x \) and \( y \) the prevailing belief can be determined. If both \( x < 0 \) and \( y < 0 \) the same comments apply, for disbelieving (instead of believing). Finally, if \( x > 0 \) and \( y < 0 \), then the negative value of \( y \) enhances our initial impression that the atom must be true in the real world. The same comments apply if \( x < 0 \) and \( y > 0 \), but for the atom’s falsehood. All the above apply in different degrees depending on the absolute values of \( x \) and \( y \), and imply a significant ability regarding our power of expressiveness and a great improvement over the classical method of representation using propositional expressions.

In order to expand these thoughts to include arbitrary propositions and define \( TT \), we need to integrate knowledge from different atoms. Initially we restrict ourselves in formulas in Disjunctive Normal Form (DNF). For the transformation, each atom of the language is assigned to one column of the matrix, and there are as many lines in the matrix as the number of disjuncts in the DNF of the formula. Each element of the matrix is the complex number that expresses the knowledge regarding the respective atom (column) in the respective disjunct (line).

We show the transformation of the expression \( p = (\alpha \land b \land \neg c) \lor (\alpha \land \neg b \land d) \lor (c \land e) \) into its respective matrix. We suppose that language \( L \) consists of 5 atoms, namely \( \alpha, b, c, d \) and \( e \), so the matrix has 5 columns. The expression is in DNF, having 3 disjuncts, so the number of lines in the matrix is 3. As far as the elements of the matrix are concerned, the procedure is the following: an element has the value of 1 if the respective atom in the respective disjunct appears as a positive atom; if negative the value is \( i \); if the atom does not appear at all, then the value of the element is 0. The application of these rules on the expression \( p \) results in the first matrix shown below. If information regarding the elements’ RFs is available, it can be accommodated as in the second matrix:

\[
\begin{pmatrix}
\alpha \land b \land \neg c \\
\alpha \land \neg b \land d \\
c \land e
\end{pmatrix}
\begin{pmatrix}
1 & 1 & 0 & 0 \\
1 & i & 0 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
RF & 1 & 2 & i & 0 & 0 \\
3 & i & 0 & 4 & 0 \\
0 & 0 & 5 & 0 & 2
\end{pmatrix}
\]

To transform a matrix back to a logical expression, we must “decode” the information contained in each element of the matrix. This procedure is called Inverse Table Transformation (\( TTI \)). Due to the inherent inability of propositional logic to express disbelief, elements implying disbelief in a fact (having negative RF) are mapped into the negation of the fact, despite the difference in semantics. Similarly, since any RF (“intensity”) data cannot be expressed in logic, it is ignored during the transformation.

The above simple definition of \( TT \) captures the intuition that will be used throughout this paper to describe our belief revision scheme. It is used to transform a formula \( p \) into a matrix \( P \), including RF information (if available) in a very compact form. \( TT \) causes no loss of information, making matrices an attractive alternative for knowledge representation. Each line of \( P \) represents a
different clause in p, or a different possible state of the world represented by p. For this reason, we will sometimes refer to lines as possible world states. Each element of $P$ represents information regarding one atom in one possible world state. Each line is actually an ordered tuple of such information for one such state. The ordering of lines in the matrix is simply a syntactic convenience, as a different ordering would make no difference as far as the knowledge contained in the matrix is concerned.

In our attempt to expand this representation to propositions not in DNF we will encounter some problems making the above informal definition a special easy case and forcing us to change it somewhat. However, these issues are irrelevant as far as the knowledge representation and belief revision problems are concerned, so we postpone their discussion until a later point (see definition 7, section 5).

3 Matrices and Belief Revision

In order to use this transformation in belief revision we will initially assume that both the knowledge and the revision are represented by matrices. In other words, we assume that we already have a general method of transforming the abstract notion of "real world knowledge" to a specific weighted matrix that represents it. Intuitively, this is easy to do, using the method described in the previous section.

For the moment, we will additionally assume that both the base $K$ and the revision matrix $M$ have only one line, so they both represent only one possible state of the world. In this special case, the revision will be defined as the addition of the two matrices, because the inclusion of matrix $M$ in our knowledge increases our trust in all the information that $M$ carries, effectively increasing our reliance in each atom and/or its negation. This adjustment may or may not be enough to force us to change our beliefs. In general, when we believe something with reliability $x$ and a new piece of knowledge confirms this belief, coming from a source with reliability $y$, then the intuitively expected result of the revision should be the same belief with reliability $x+y$. This is the notion behind defining the revision as the addition of the two matrices. One can verify that the operation of addition is intuitively correct even in cases where $x$ and/or $y$ are negative numbers, or when they refer to contradicting data (one implying $\alpha$ and the other $\neg\alpha$ for some atom $\alpha$). For example, let $K = [i 3 0], M = [3 2 1]$. Using the TTI function we can verify that the matrices represent the expressions $K = \neg\alpha \land b$ (base) and $M = \alpha \land b \land c$ (revision). The matrices also show the reliability per atom in each proposition. In this case, $a$’s negation ($\neg\alpha$) is believed with an RF of 1 in $K$, but $M$ should force us to abandon this belief as $\alpha$ is believed with an RF of 3 in $M$. In $b$, there is no contradiction between $K$ and $M$; however this revision should increase our confidence in the truth of $b$. Finally, in $c$, we know now that $c$ is true, with a reliance of 1; we had no knowledge regarding $c$ before. The resulting matrix is $K'$ where the symbol "•" stands for the operation of revision: $K' = K \cdot M = [i 3 0] + [3 2 1] = [3 + i 5 1]$. 
The proposition related (via the TTI function) to the resulting matrix is: $F \land b \land c \equiv F$, because the first element $(3 + i)$ implies a contradiction. This is not generally acceptable, as such a KB contains no useful information. The result should be $\alpha \land b \land c$, as showed by the previous analysis. We will deal with this problem in the next section (which is in fact not a problem at all!).

In the general case where one (or both) of the matrices contains more than one line, each line represents one possible state of the world. In order to be sure that the real world will be represented by a line in the resulting matrix, we must add (revise) each line of $K$ with each line of $M$, creating one line per pair in the resulting matrix $K'$. Let us see one example:

$$K = \begin{bmatrix} 1 & 2 \\ 0 & 5i \end{bmatrix}, M = \begin{bmatrix} i & 3 \\ 2i & 3i \end{bmatrix}, K \bullet M = \begin{bmatrix} 1 + i & 5 \\ 1 + 2i & 2 + 3i \\ i & 3 + 5i \\ 2i & 8i \end{bmatrix}$$

4 Queries and Contradictions

Any given matrix has no intuitive meaning for a user, because finding the knowledge contained in a large KB matrix is not a trivial operation. More importantly, the user is normally not allowed nor interested to see the whole KB, but wishes to execute queries for finding specific information. To reply to a user query, we must transform the KB matrix back into a proposition expressing the knowledge contained in it. TTI is inadequate because there are cases (as the one above) when the result of a revision is a matrix corresponding via TTI to the logical constant $F$; equivalently, the KB related to the matrix via TTI is inconsistent. To overcome this problem, we will perform some pre-processing on the KB matrix before applying TTI, in order to remove the parts of the matrix that cause the contradictions and that contain data not properly describing the real world.

Before defining such a function, we notice that not all lines of a matrix properly model the real world, so a selection has to be made. One may argue that contradictory lines represent contradictory world states, so they contain false information and they could as well be deleted. This is the policy followed by the TTI function, as shown in the previous example. However, this is not entirely true. A single faulty revision may create a contradiction in an otherwise correct line, so we must be careful before discarding a contradictory line as non-true. On the other hand, even a non-contradictory line could be too far from the real world. The absence of contradictions indicates that nothing that overrules this line has been known up to now; still, we cannot be sure that the line properly describes the modelled world. Therefore, our policy is to keep the matrix as-is and to let inconsistency stay in the KB, even if some (or all) lines are contradictory.

One possible way to materialize these requirements is described in [7]. In order to estimate the proximity of a line to the real world, a function named Line Reliability $(RL)$ is used, depending on the Element Reliability $(RE)$ function. $RE$ is a function that calculates our reliance in the truth of the information that
one element carries. This quantity is a real number having nothing to do with
the truth or falsity of the respective atom; it expresses our estimation on whether
the information carried by the element corresponds to this atom’s truth value
in the real world. Similarly, RL is a function that uses RE in order to estimate
the proximity of the line’s information to the real world. For a KB matrix A,
RE returns a matrix B in which all elements of A have been replaced by their
reliability, a real number. RL produces a column matrix C composed of real
numbers, in which each element represents the reliability of the respective line.

Given the estimation provided by the RL function, we should select the
lines that are sufficiently close to the real world (according to our estimations,
as usual). This selection is made by a function named Line Selection function
(LS), which returns a set S of indices, corresponding to the selected lines of the
original matrix. This set is used by the Submatrix Selection function (MS) to
produce a new matrix \( D = MS(A, S) = MS(A, LS(C)) \), which is a submatrix
of the original matrix, according to our selection.

In order to remove the contradictions possibly appearing in the selected lines,
the Matrix Normalization function (MN) is used to transform each element to
its normalized (non-contradictory) counterpart. To decide whether to believe in
the truth or falsity of the atom (instead of both or neither, which is the source
of the contradiction), we compare the RF of the real and imaginary part of the
element by subtracting them. MN returns a new matrix \( E = MN(D) \).

This procedure ensures that there will be no contradictions in the matrix
that resulted by the selection implied by the LS function. This final matrix
(\( E \)) expresses the (consistent) KB that will be used in queries. It contains
the most reliable lines of our KB, which have been normalized in order to extract
the information contained in the contradictory elements. At this point, we can
eventually apply the TTI function to get the (satisfiable) logical expression \( p \)
related to matrix A, corresponding to the knowledge store in the KB.

The whole process (function) of transforming a matrix to a proposition for the
needs of queries will be denoted by \( QT \) (Query Transformation), so \( p = QT(A) \).
It is clear by the analysis above that QT is in fact a composite function, being
composed of six functions. The first three (RE, RL, LS) are not clearly defined,
because we claim that a reasonable definition of such functions is application-
dependent. Arguments for this claim, as well as its importance, will be provided
in section 6. The final three functions (MS, MN, TTI), will be formally defined
in the next section and express the process of extracting information out of (pos-
sibly contradictory) elements. Finally, we stress that the QT operation does not
actually change the matrix; it temporarily transforms it to a logical expression
for the needs of queries. The next revision will be executed upon the original
matrix and the whole process of query transformation should be repeated after
the revision to calculate the new related proposition.
5 Formal Framework

With the introduction of the QT function, we completed the description of the framework we propose for representing and revising knowledge. Summarizing our method, we can say that it consists of the following three parts:

1. **Knowledge Representation**: express the knowledge represented by the logical propositions of the input into matrices and encode the reliability information, if available, into the matrix elements, transforming our knowledge into the more expressive and convenient form of the matrix representation.

2. **Knowledge Revision**: whenever any new data is introduced (revision), apply the belief revision algorithm to accommodate the new knowledge.

3. **Knowledge Query**: apply the QT function upon the matrix representing the KB to extract the KB’s information, in order to reply to user queries.

In this section we provide a formal framework for the procedures informally described in previous sections. Proofs are omitted due to lack of space, but they can be found in [7]. At first, some notational conventions should be introduced. We denote by $\mathbb{C}^{(+)}$ the set of complex numbers whose real and imaginary part are both non-negative. For matrices, we define $\mathbb{C}^{m \times n}$ to be the set of $m \times n$ matrices whose elements are complex numbers, and $\mathbb{C}^{* \times n}$ the union of $\mathbb{C}^{m \times n}$ for all $m \in \mathbb{N}^*$. Analogously, we define the sets $\mathbb{C}^{(+)}_{m \times n}$ and $\mathbb{C}^{(+)* \times n}$. We will use the usual notation for addition / multiplication of matrices, and the multiplication of a number with a matrix. We define the operation of juxtaposition as follows:

**Definition 1.** Let $A, B \in \mathbb{C}^{* \times n}$. The juxtaposition of $A$ and $B$, denoted by $A|B$, is the matrix that results by placing the lines of $B$ below the lines of $A$.

We also define a partitioning on $\mathbb{C}$ as follows:

**Definition 2.** Let $z = x + yi \in \mathbb{C}$. Then:

- $z$ is called **positive** iff $x \geq 0$, $y \leq 0$. Such numbers (forming set $\mathbb{C}_+$) denote trust in the truth and distrust in the falsehood of the atom (positive literal).
- $z$ is called **negative** iff $x \leq 0$, $y \geq 0$. Such numbers (forming set $\mathbb{C}_-$) denote distrust in the truth and trust in the falsehood of the atom (negative literal).
- $z$ is called **contradictory** iff $x \cdot y > 0$ (so $x > 0$ and $y > 0$ or $x < 0$ and $y < 0$). Such numbers (forming set $\mathbb{C}_*$) denote trust or distrust in both the atom's truth and falsehood. Both these cases are invalid (contradictory), because an atom can be either true or false, but not both or neither.
- $z$ is called **zero** iff $z = 0$ (forming set $\mathbb{C}_0$). In this case, $x = y = 0$, so $z$ is both positive and negative and denotes lack of knowledge regarding both the atom and its negation; we have no reason to accept or discard the truth or falsehood of the atom, so we have to accept both.

We get that: $\mathbb{C}_0 = \{0\}$, $\mathbb{C}_+ \cap \mathbb{C}_- = \mathbb{C}_0$, $\mathbb{C}_+ \cap \mathbb{C}_* = \emptyset$, $\mathbb{C}_- \cap \mathbb{C}_* = \emptyset$, $\mathbb{C}_- \cup \mathbb{C}_+ \cup \mathbb{C}_* = \mathbb{C}$.

**Definition 3.** We define the following matrices:
- The k-atom: \( A_k = [0 \ldots 0 1 0 \ldots 0] \in C^{1 \times n} \), where the element 1 is in the k-th column.
- The generalized k-atom: \( A_k(z) = z \cdot A_k \in C^{1 \times n} \), for some \( z \in C \).
- The n-true matrix: \( T_n = [0 \ldots 0] \in C^{1 \times n} \).
- The n-false matrix: \( F_n = [1 + i \ldots 1 + i] \in C^{1 \times n} \).

**Definition 4.** We define the truth constants \( F = 0 \in C \) and \( T = 1 \in C \). Any sequence \( I \in \{0, 1\}^n \) is called an interpretation of space \( C^{* \times n} \), forming set \( I(n) \). Let \( I = (\alpha_1, \ldots, \alpha_n) \in I(n) \), \( A \in C^{1 \times n} \) such that: \( A = \Sigma_{j=1}^n A_j(a_j + (1 - a_j) \cdot i) \). A is called an interpretation matrix of space \( C^{* \times n} \).

Notice that there is a direct mapping between logical interpretations, interpretations and interpretation matrices. Moreover, the number \( z = a_j + (1 - a_j) \cdot i \) can be either \( z = 1 \) (for \( a_j = 1 \)) or \( z = i \) (for \( a_j = 0 \)). Therefore, an interpretation matrix is a matrix of the set \( C^{1 \times n} \), whose elements are from the set \( \{1, i\} \).

**Definition 5.** Let \( I = (\alpha_1, \ldots, \alpha_n) \in I(n) \), \( A \in C^{1 \times n} \). A is satisfied by \( I \) iff there exist \( z_1, \ldots, z_n \in C^+ \) : \( A = \Sigma_{j=1}^n A_j((2 \cdot a_j - 1) \cdot \overline{z_j}) \), where \( \overline{z_j} \) is the conjugate complex of \( z_j \in C^+ \). If \( A \in C^{m \times n} \) such that \( A = A^{(1)} \mid \ldots \mid A^{(m)} \), \( A^{(j)} \in C^{1 \times n} \), \( j = 1, \ldots, m \), then we say that A is satisfied by \( I \) iff there exists \( j \in \{1, \ldots, m\} \) such that \( A^{(j)} \) is satisfied by \( I \). The set of interpretations satisfying \( A \), called the set of models of \( A \), will be denoted by \( \text{mod}(A) \).

Notice that \( (2 \cdot a_j - 1) \in \{-1, 1\} \) and \( x_j = (2 \cdot a_j - 1) \cdot \overline{z_j} \in C_+ \) iff \( a_j = 1 \) and \( x_j \in C_- \) iff \( a_j = 0 \). Numbers \( z_j \) represent \( \text{RF} \); interpretation matrices, having elements in \( \{1, i\} \) lack “intensity” information, whereas a matrix may be weighted with \( \text{RF} \) information. To calculate \( \text{mod}(A) \), two easier, constructive methods can be devised:

**Proposition 1.** Let \( A = [w_{jk}] \in C^{m \times n} \). Then \( \text{mod}(A) = \bigcup_{j=1}^m (I_{j1} \times \ldots \times I_{jn}) \), where for any \( j \in \{1, \ldots, m\} \), \( k \in \{1, \ldots, n\} \):
- \( I_{jk} = \{0\} \) iff \( w_{jk} \in C_- \cap C_+ = C_- \cap C_0 \)
- \( I_{jk} = \{1\} \) iff \( w_{jk} \in C_+ \cap C_- = C_+ \cap C_0 \)
- \( I_{jk} = \{0, 1\} \) iff \( w_{jk} \in C_0 \)
- \( I_{jk} = \emptyset \) iff \( w_{jk} \in C_* \)

Furthermore, \( \text{mod}(A) = \bigcup_{j=1}^m \bigcap_{k=1}^n \text{mod}(A_k(w_{jk})) \).

The latter result indicates a close connection between juxtaposition and addition of matrices to union and intersection between models, respectively. Juxtaposition is closely related to the union of models:

**Proposition 2.** Let \( A, B \in C^{* \times n} \). Then \( \text{mod}(A|B) = \text{mod}(A) \cup \text{mod}(B) \).

Unfortunately, the connection between addition and intersection is not so straightforward. We explore their relation in [7]. In order to define \( \text{TT} \), we need operations on matrices emulating the usual operations of propositional logic:
Definition 6. We define 3 classes of functions in $C^{\times n}$, denoted by $F_\lor$, $F_\land$ and $F_\lnot$, called the classes of disjunction, conjunction and negation functions:

- A function $f_\lor : (C^{\times n})^2 \to C^{\times n}$ is said to belong in $F_\lor$ iff for any $A, B \in C^{\times n}$, $\text{mod}(f_\lor(A, B)) = \text{mod}(\text{mod}(A) \lor \text{mod}(B))$. We use $A \lor B$ to denote $f_\lor(A, B)$.
- A function $f_\land : (C^{\times n})^2 \to C^{\times n}$ is said to belong in $F_\land$ iff for any $A, B \in C^{\times n}$, $\text{mod}(f_\land(A, B)) = \text{mod}(\text{mod}(A) \land \text{mod}(B))$. We use $A \land B$ to denote $f_\land(A, B)$.
- A function $f_\lnot : C^{\times n} \to C^{\times n}$ is said to belong in $F_\lnot$ iff for any $A \in C^{\times n}$, $\text{mod}(f_\lnot(A)) = I(n) \setminus \text{mod}(A)$. We use $\lnot A$ to denote $f_\lnot(A)$.

The space $C^{\times n}$, equipped with functions $\lor \in F_\lor$, $\land \in F_\land$, $\lnot \in F_\lnot$, is called a logically complete matrix space of dimension $n$ and is denoted by $(C^{\times n}, \lor, \land, \lnot)$.

No further restrictions are set on the selection of operators $\lor, \land, \lnot$ and different selections imply different logically complete spaces. The same discipline can be used to define additional operators (like $\rightarrow$), if necessary, but the above are enough for the main definition of this section:

Definition 7. Assume $(C^{\times n}, \lor, \land, \lnot)$ and $L$ a finite propositional language. We denote by $\alpha_j$ the atoms of the language and by $A_j$ the atoms of $C^{\times n}$. We define the Table Transformation function, $TT : L^* \to C^{\times n}$, recursively, as follows:

- $TT(T) = T_n$, $TT(F) = F_n$
- For any $j \in \{1, \ldots, n\}$: $TT(\alpha_j) = A_j$
- $TT(p \theta q) = TT(p) \theta TT(q)$, for any $p, q \in L^*, \theta \in \{\land, \lor\}$
- $TT(\lnot p) = \lnot TT(p)$, for any $p \in L^*$

Similarly, for $j \in \{1, \ldots, n\}, z \in C$, we define the Inverse Table Transformation function, $TTI : C^{\times n} \to L^*$, recursively, as follows:

- $TTI(A_j(z)) = \alpha_j$, iff $z \in C_+ \setminus C_0 = C_+ \setminus C_0$
- $TTI(A_j(z)) = \lnot \alpha_j$, iff $z \in C_+ \setminus C_0 = C_+ \setminus C_0$
- $TTI(A_j(z)) = T$, iff $z \in C_0$
- $TTI(A_j(z)) = F$, iff $z \in C_0$

For $m \in N^*, A = [z_{ij}] \in C^{m \times n}$, $TTI(A) = \bigvee_{k=1}^m (A_{ij} \bigvee_{i=1}^n TTI(A_j(z_{ij})))$.

The transformations above have a very important property:

Proposition 3. For any proposition $p \in L^*$ and any matrix $P \in C^{\times n}$ it holds that $\text{mod}(p) = \text{mod}(TT(p))$ and $\text{mod}(P) = \text{mod}(TTI(P))$.

The above proposition shows that the transformation of a matrix to a logical expression and vice-versa does not cause any loss of (logical) information, as any interpretation that satisfies a given matrix satisfies its respective logical expression (via the TTI function) and vice-versa (via the TT function). This is true as far as logic is concerned; if RF information is available, TTI may cause information loss, in knowledge representation terms. Moreover, notice that the above proposition holds regardless of the selection of the operations $\lor, \land, \lnot$. 
We have already stressed the fact that matrices’ elements can have negative real and/or imaginary parts. Such numbers indicate lack of confidence to a given literal and/or its negation, so they do not give direct information on the truth or falsity of a literal; instead, they indirectly imply its truth or falsity by specifying distrust in its falsity or truth respectively. Such kind of knowledge will be denoted by the term negative knowledge contrary to elements with non-negative parts (real and imaginary), which will be referred to as positive knowledge. The distinction is justified by the fact that logic can only express positive knowledge. Negative knowledge is only useful as far as knowledge representation is concerned, and its importance will be set forth in the next section.

By restricting ourselves to positive knowledge, it becomes easier to define operations like \( \land, \neg \). For a specific selection of operators, for positive knowledge and for propositions in DNF, the simple informal definition of TT given in section 2 coincides with the operation given by definition 7 (see [7]). TT, as defined in definition 7 is only important as a theoretical construction, showing the relation between matrices and logic; it will also allow us to prove some results (e.g. proposition 4). TT cannot express the additional reliability information that we may have, nor can it express negative knowledge. This seriously reduces our expressive abilities. Furthermore, this approach requires the explicit definition of the propositional operations. However, it has no real intuitive meaning to define such operations in the general case, though possible; remember that negative knowledge cannot be expressed in propositional logic. All we need is their existence, to support definition 7.

Having completed the formalization of our knowledge representation scheme and its relation with propositional logic, we can now proceed to the definition of our revision scheme:

**Definition 8.** Let \( A, B \in \mathbb{C}^{n \times n} \), where \( A = A^{(1)} | \ldots | A^{(k)} \), \( B = B^{(1)} | \ldots | B^{(m)} \), for some \( k, m \in \mathbb{N}^* \), \( A^{(j)} \in \mathbb{C}^{1 \times n} \), \( j \in \{1, \ldots, k\} \) and \( B^{(j)} \in \mathbb{C}^{1 \times n} \), \( j \in \{1, \ldots, m\} \). We define the operation of revision, denoted by \( \bullet \), between those two matrices as follows: \( A \bullet B = \sum_{k=1, j=1}^{k=m} A^{(k)} + B^{(j)} \).

It is easy to verify that this definition concurs with the informal description of the revision operator of section 3. It has been proved in [7] that, for positive knowledge: \( \text{mod}(A \bullet B) = \text{mod}(A) \cap \text{mod}(B) \).

**Definition 9.** We define the submatrix selection function \( MS : \mathbb{C}^{n \times n} \times \mathbb{P}(\mathbb{N}^*) \to \mathbb{C}^{k \times n} \). For \( k \in \mathbb{N}^* \), \( A \in \mathbb{C}^{k \times n} \), \( S \subseteq \mathbb{N}^* \), such that \( A = A^{(1)} | \ldots | A^{(k)} \), \( A^{(j)} \in \mathbb{C}^{1 \times n} \), \( j \in \{1, \ldots, k\} \), we define:

- \( MS(A, S) = A \), iff \( S = \emptyset \) or there exists \( m \in S \) such that \( m > k \),
- \( MS(A, S) = \{ j \in S \mid A^{(j)} \}, \) otherwise.

We define the matrix normalization function \( MN : \mathbb{C}^{n \times n} \to \mathbb{C}^{k \times n} \) such that for \( A = [a_{ij}] \in \mathbb{C}^{k \times n} \): \( MN(A) = [\text{Re}(a_{ij}) - \text{Im}(a_{ij})] \in \mathbb{C}^{k \times n} \).

The above functions are used in the third part of our method, namely the querying of the knowledge in the KB. The MS function returns a submatrix
of A which consists of some of the lines of A, those whose indexes belong to the set S. Some abnormal cases have been included in the above definition for completeness, but they will not appear in this application. The MN function is used to remove any contradictions from the matrix, by comparing (subtracting) the real and imaginary part of \( a_{ij} \). T T T is the final step to transforming a matrix to its respective logical expression. The first three functions of the QT (RE, RL, LS) will not be explicitly defined for the reasons set forth in the next section.

6 Comments, Properties and Results

In this section, we will discuss some of the problems and considerations researchers have to face when dealing with the problem of belief revision, as well as how our method deals with these problems.

One such problem is the representation of knowledge. In our framework, the abstract term “knowledge” (or “belief”) will refer to any propositional formula over a finite language L along with its reliability information, or, equivalently, by the matrix that represents this information. So, a KB is a single matrix, constituting a compact knowledge representation scheme. This matrix contains all the information of the past revisions in an encoded fashion. A world state is an interpretation and the real world that we are trying to model is actually one such state which may change through time. Each revision describes the world partially by denoting a set of acceptable states (interpretations), namely those that it satisfies.

Another primary consideration regarding belief revision is the concurrence of the results with human intuition. Unfortunately, it is not absolutely clear how humans revise their beliefs, despite several efforts in the area. One example of disagreement is the representation of knowledge in the human brain. In [9] two general types of theories concerning this representation are described: foundation and coherence theories. According to foundational theorists, only some beliefs (called foundational, or reasons) can stand by themselves; the rest are derived from the most basic (foundational) beliefs. On the other hand, coherence theorists believe that each piece of knowledge has an independent standing and needs no justification, as long as it does not contradict with other beliefs. Experiments [9] have shown that the human brain actually uses the coherence paradigm, by showing that people tend to ignore causality relationships once a belief has been accepted as a fact, even if it has been accepted solely by deduction from other beliefs. There has been considerable controversy [9], [10] on the experiments' results based on the argument that humans do not actually ignore the causality relationships, but forget them.

The approach (foundational or coherence) chosen greatly influences the algorithms considered. Foundational KBs need to store the reasons for beliefs, whereas KBs based on the coherence paradigm need to store the set of beliefs, without any reason information. Reasons should be taken into account when revising a KB only under the foundational approach. In any case, the set of beliefs of any KB includes the derived beliefs, which, in general, may be too many or
even infinitely many. It has been proposed in [11], [18] that instead of the whole set of beliefs (belief set), a small number of propositions could be stored (belief base), enough to reproduce the whole set via deduction. The use of belief bases does not necessarily force us to use the foundational paradigm; the causality relationships possibly implied by the selection of a belief base may or may not be used, depending on the approach. The use of belief bases gives rise to the problem of the selection of a proper belief base, as there may be several for any given belief set. Different selections become critical under the foundational approach, but are irrelevant under the coherence paradigm. An example of belief base selection appears in [3], [4] where a single proposition is used as the KB.

In this work, we adopt Nebel’s proposition [18] for the selection of the belief base where the contents of the KB are the individual revisions, each of them expressing an observation, experiment, rule etc regarding a domain of interest. In principle, knowledge is derived from our observations about the world, so this is the best way to describe our knowledge. As in any belief base, the knowledge of the KB consists of these revisions as well as their logical consequences, but the selection of the belief base just described implies that this approach follows the foundational paradigm. The foundational approach is generally viewed as being more compatible with common sense [9].

There are cases when a revision contradicts previous ones or their consequences. In such cases, at least one of the previous revisions (old knowledge) or the new revision itself is usually modified or discarded altogether in order for the KB to remain consistent. This is generally done while revising the KB, but, in our method, contradictions are resolved at query time using QT (see also [14] for a similar approach). In most cases, the removal of contradictions is done following Dalal’s Principle of Primacy of New Information, which states that the revision is more reliable than the KB, so it is the KB that should be adapted to accommodate the revision [3]. Despite the fact that a revision represents the latest information about the world and can be usually assumed correct, the new data may come from a noisy or otherwise unreliable source, so it should not be always assumed correct.

This problem is overcome in our method by the use of the RF and the MN function. In queries, given that the respective line is selected (by the LS function), the information with the largest RF will be kept, regardless of whether it resulted from the last revision or the old data. This is concurrent with human intuition, as our beliefs, eventually, are unaffected by the order information is received. This fact additionally allows us to perform KB merging in a straightforward way. The important problem of data rejection in contradicting revisions is also solved in the same way (using the QT function). Rejection is only temporary, for the needs of queries, and depends on the RF and the functions RE and RL, being minimal with respect to these quantities. The notion of minimality depends on the LS function selection.

Some of the most important results of our method originate from the inexplicit definition of the RE, RL and LS functions. These functions determine the lines to be used for queries, because they select the “important” lines of a matrix.
The best selection is application-dependent. To see this, one could try to determine which of the (contradictory) elements $1+3\imath$ and $100+300\imath$ is more reliable (less contradictory). Many would argue that both elements are equally reliable, as the belief ratio of truth to falsehood of the atom is the same. Others would disagree, on the argument that the removal of the contradiction in $100+300\imath$ requires greater changes in the KB than in $1+3\imath$. In any case, the RE function will determine that. In an application where data is generally obtained through noisy channels, the contradiction in $1+3\imath$ is likely to have occurred due to some random noise (error) of the input; on the other hand, $100+300\imath$ is less likely so, statistically, therefore it could be safely assumed that this element implies a real contradiction. In an environment where the world often changes dynamically, the contradiction in both elements may be due to information received in a previous state of the world; thus, they can be assumed equally reliable as they have the same belief ratio. In some other application where decisions are based on subjective opinions, instead of facts, the fact that the number $100+300\imath$ implies a bigger sample may force us to consider it more reliable than $1+3\imath$.

The effect of the element reliability on the overall reliability of a line (RL parameter), as well as our tolerance in contradictory lines and the number of lines selected for use in a query (LS parameter) depends mainly on the reliability of the input devices. In a medical application, where the input devices are most reliable, even small contradictions should be “fatal” for the line that they appear in; contradictions in such a sensitive application should be unacceptable, unless there is no other choice. This is not the case in applications with often dynamic changes of the world’s state or with noisy input channels.

Moreover, the ability to freely define RE, RL and LS functions provides a considerable flexibility in the extraction of the knowledge in a matrix. This flexibility allows relating the result of any given revision (and any given matrix) to several different propositions; as far as the user is concerned, this is equivalent to supplying different belief revision algorithms. Consequently, our framework provides a whole class of belief revision algorithms and the problem of finding a good such algorithm is reduced to the problem of finding a good way to extract the information from a matrix. The search for some interesting members of this class of algorithms is an ongoing work, but it has already been proven that Dalal’s algorithm [3, 4], gives the same results as our method for a specific parameter selection, as shown by the following proposition:

**Proposition 4.** Let $p, q \in L^*$ be two satisfiable propositional expressions in DNF and let $r$ be the revision of $p$ with $q$ under Dalal’s algorithm ($r = p_{D}^{*}q$). Moreover, let $P \in C^{(+)}_{x} \times n$ the matrix related to $p$ via the TT function, using an RF of 1 for all atoms ($P = TT(p)$), $Q \in C^{(+)}_{x} \times n$ the matrix related to $q$ via the TT function, using an RF of 2 for all atoms ($Q = 2 \cdot TT(q)$) and $R \in C^{(+)}_{x} \times n$ the matrix resulting by the revision of $P$ with $Q$ under our framework ($R = P_{\bullet}^{*}Q$). Under these RF selections, there exist selections for the functions RE, RL and LS such that the resulting propositional expression (to be used in queries) is logically equivalent to the expression $r$ as defined above, that is: $QT(R) \cong r$. 
The phrasing of the above proposition implies the important fact that the selection of the RFs of a matrix's elements has great effects on the result of a revision. Its proof, as well as the definition of one possible set of RE, RL and LS functions that satisfy it, is given in [7].

One effort to formalize the concurrence of the results of belief revision with human intuition was made by Alchourron, Gärdenfors and Makinson in a series of papers [1], [8], [17]. Unlike other researchers, they receded from the search of any specific algorithm and attempted to formalize the notion of revision. As a result, a set of widely accepted properties of belief revision algorithms was introduced, in the form of postulates expressed as a set of logical propositions (named AGM postulates after the authors' initials). Using these postulates, a series of important theoretical results were proved and a series of other works were inspired. Nebel in [18], investigated generalizations of these postulates into the knowledge level. In [13], a different theoretical foundation of revision functions was proposed by reformulating the AGM postulates in terms of formulas and an elegant representation based on orderings of belief sets was provided. In the same paper, Dalal's algorithm was proven to satisfy all 8 AGM postulates; this implies that there exist parameters under which our method satisfies the AGM postulates for revision.

One of the most peculiar problems in belief revision is the fact that the result of a revision may not only depend on the data itself but on its source as well. Let us suppose that there are two lamps, A and B, in a room and we know that exactly one of them is on. Our knowledge can be represented by the proposition: \((a \land \neg b) \lor (\neg a \land b)\). If we make the observation that lamp A is on, we should revise our KB with the proposition \(a\) and the intuitively correct result for the revision is the proposition \(a \land \neg b\), as we know now that B is off. On the other hand, if a robot is sent into the room in order to turn lamp A on, then we would again have to revise with \(a\). The proper intuitive result of this revision is the proposition \(a\) in this case, as we know nothing about the state of lamp B; it could have been on or off before sending the robot in the room (and stayed so). This example shows that even identical (not just equivalent) KBs can give different intuitively correct results with identical revisions!

In order to overcome the problem, two different types of "knowledge change" have been defined in [12], namely revision and update. Revision is used when new information about a static world is obtained. This is the first case of our example where the observation did not change the state of A and B. A revision is performed when the source of the data is an observation regarding the world. Update is used when the world dynamically changes and we have to record that change. In the second case of our example, the robot changed the state of lamp A, thus the state of the world being modeled. Therefore, the result of an update must be different. An update is performed when the reason of change is an action, instead of an observation. An excellent study on the problem may be found in [12], where a new set of postulates, adequate for update, is presented.

An important novelty of our scheme is the introduction of negative knowledge. The disbelief expressed by such type of knowledge cannot be expressed in
propositional logic, so this is a direct improvement of our expressive abilities over the conventional knowledge representation schemes. Using negative knowledge we are able to deal with the above operations (revision, update, contraction, erasure) in terms of revision only, by changing the revision matrix's RF in such a way as to signify the operations' different nature.

We will see how this is possible with an example, which will also show the dominating effect of the RF selection and negative knowledge on the result of a revision as well as the power of parameterization. The reliability of an element $x + yi$ will be defined as $|x - yi|$, i.e., the difference between its true and false part. The reliability of a line will be defined as the sum of the reliabilities of all its elements and LS will select the lines with maximum reliability.

Revisiting the above example with the lamps, we could express our knowledge using the proposition: $p = (\neg \alpha \land b) \lor (\alpha \land \neg b)$. When we conclude (through observation) that A is on, the proposition representing the revision is $q = \alpha$. By assigning an RF of 1 on all atoms of the KB and an RF of $x > 0$ on the revision we get: $P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $Q = \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix}$, $P' = P \cdot Q = \begin{bmatrix} x + i & 1 \\ 1 & x + i \end{bmatrix}$.

Note that the first line contains a contradictory element $(x + i)$, whereas the second contains no contradictions. Using the parameterization above, we get reliabilities of $|x-1|+1$ for the first line and $|x+1|+1$ for the second. For all $x > 0$ it holds that $|x-1|+1 < |x+1|+1$, thus LS will only select the second line. Upon applying MS function, $P'$ matrix will be mapped to $P'' = [1 + x i]$ (containing only the second line of $P'$). Applying the MN function will have no effect on $P''$, as there are no contradictions to resolve, thus the respective proposition of $P''$ is: $QT(P'') = \alpha \land \neg b$, as expected. Alternatively, the RL function could be defined as the minimum over the reliabilities of all elements in the line (only the least reliable element of the line is considered in the calculation). In this case, for $x \geq 2$ both lines would have equal reliability (equal to 1), so the LS function would select them both. Thus, for this selection of the RL function, the RF of $Q$ is crucial; for $x \geq 2$ we get $QT(P') = (\alpha \land b) \lor (\alpha \land \neg b) \equiv \alpha$, whereas for $0 < x < 2$ we get $QT(P') = \alpha \land \neg b$.

A more intuitively correct approach for the LS function would be to select the non-contradictory lines only; if there are no such lines, then we go on by selecting the most reliable contradictory ones, as before. It can be verified that in this case, the result would have been $QT(P') = \alpha \land \neg b$, regardless of the selection of the parameters RE, RL or $x$.

Continuing this example, notice that if we had sent a robot into the room with the order “turn lamp A on”, then we should update (not revise) $p$ with $q = \alpha$. Matrix $Q' = \begin{bmatrix} 1 & -i \\ -i & 0 \end{bmatrix}$, corresponds to the same information as $Q$, because $TTI(Q') = \alpha = TTI(Q)$. However, $Q'$ includes some disbelief in the negation of $\alpha$ because the imaginary part of $1 - i$ is negative ($-1$). Thus, $Q'$ has been enhanced in such a way as to contain the additional information that the action of the robot voids the previous state of $A$, if $A$ was off, because the dynamic change indicated by the update renders any previous knowledge on the world irrelevant. The revision of $P$ with $Q'$ under our scheme gives: $P' = \begin{bmatrix} 1 & 1 \\ 2 - i & i \end{bmatrix}$. 
The important point here is that there are no contradictory lines in $P'$, therefore, by using the rational LS function previously described (regardless of the RE and RL selection), we would select both lines of $P'$, so $QT(P') = (\alpha \land b) \lor (\alpha \land \lnot b) \equiv \alpha$. This is the expected result of an update!

Update and revision are used to add knowledge to a KB. Essentially, new data enhances our knowledge about the domain of interest. In some cases though, we wish to remove knowledge from the KB. This operation is called contraction and it is dual to revision. It has been argued [8], [17] that it is intuitively simpler to deal with contraction instead of revision. We could also define the operation of erasure which is dual to update. For the class of revision schemes that satisfy the AGM postulates, it has been proven that revision and contraction can be defined in terms of each other. Similar results apply for the update/erasure schemes that satisfy the postulates for update/erasure defined in [12].

We notice that using negative information we can also get a contraction operation, by revising with matrix $-M$ whenever we want to contract $M$. Similar considerations can lead to the integration of the operation of erasure as well. This fact eliminates the need for additional operators, as they can all be defined in terms of revision. Moreover, we can perform partial operations, by, for example, contracting knowledge regarding some atoms and revising others (or any other combination), a property not available in conventional revision schemes.

Another important consideration in belief revision is the problem of iterated revisions. All the algorithms described so far work well with one revision, but there are sequences of revisions which give counter-intuitive results if we process each one individually. The main problem regarding these algorithms is the fact that the belief base is not properly selected after each revision, because the algorithms are only concerned with the result of the revision and do not keep information regarding the previous KB states. This can cause the loss of valuable information as far as future revisions are concerned. One proposed solution to this problem is to process the sequence of revisions as a whole [5], [16].

In our method, iterated revisions are inherently supported. Each line in the KB matrix contains the additive information over all revisions (for a certain combination of lines) regarding the truth and falsehood of each atom in the world. By not removing any lines from the KB, we lose no data regarding past revisions, because all world states are kept. Such states may be useful in future revisions.

An example will show this fact. Suppose the propositions: $p_1 = \alpha \leftrightarrow b$, $p_2 = \alpha \leftrightarrow c$, and $p_3 = b \leftrightarrow c$, with a reliability of 1, as well as the proposition $p_4 = \alpha \leftrightarrow \lnot b$ with a reliability of 2. It is easily verified that the intuitively correct result for the revisions $p_1 \bullet p_2$ and $p_1 \bullet p_3$ is $\alpha \leftrightarrow b \leftrightarrow c$. If we subsequently revise with $p_4$, the intuitively correct result is different in each case. Specifically, given the increased reliability of $p_4$, we should have $(p_1 \bullet p_2) \bullet p_4 = \alpha \leftrightarrow (\lnot b) \leftrightarrow c$ and $(p_1 \bullet p_3) \bullet p_4 = (\alpha) \leftrightarrow (b) \leftrightarrow c$. Most revision schemes (e.g., Dalal’s [3], [4]) would give $(p_1 \bullet p_2) \bullet p_4 = (p_1 \bullet p_3) \bullet p_4 = \alpha \leftrightarrow (\lnot b)$, thus losing all information regarding $c$. This happens because Dalal’s algorithm does not support iterated revisions.
Let us assume that the reliability of \( x+y \) is \(|x-y|\), that of a line is the sum of the element reliabilities and LS is the function described in the previous example. It is easy to verify that, under this parameterization, our revision operator gives the intuitively correct results for \( p_1 \cdot p_2, p_1 \cdot p_3, (p_1 \cdot p_2) \cdot p_4 \) and \((p_1 \cdot p_3) \cdot p_4\). The additional information that is needed in order to do the second revision correctly is “hidden” in the contradictory lines of the first revision in both cases. These are ignored when answering queries after the first revision, but they play an important role in the formulation of the result of the second revision. If such lines were permanently discarded after the first revision, we would get the same (incorrect) result as Dalal’s operator. A detailed proof of these facts is omitted due to lack of space.

Unfortunately, there is an annoying fact about the above method; the number of lines in a KB is exponential with respect to the number of revisions performed, making the KB matrix too large to be manageable. We propose the technique of abruption, a procedure that permanently prunes lines from the KB. The selection of the lines to be removed could be made using functions similar to the RE, RL and LS functions. However, we should bear in mind that such removal always implies loss of knowledge; thus, the use of abruption is a trade-off between knowledge integrity and processing speed and should be carefully designed to prune lines that are too far from the real world to ever affect the result of QT. The effect of abruption upon the complexity and the correctness of the revision algorithm varies from case to case and is another application-specific parameter for which no universally defined function can work satisfactorily.

7 Conclusions and Future Work

In this paper, we used an innovative representation of propositional expressions to address the problem of belief revision. The approach was proven fruitful, resulting in a very flexible and general-purpose method of revising beliefs. The introduction of RF and the quantitative nature of the representation introduce an increased expressiveness, allowing the use of features not normally available, like negative knowledge or the integration of “knowledge change” operators.

We believe that much more work needs to be done in order to fully exploit this representation’s capabilities. The behavior of our algorithm under different parameter selections (RE, RL and LS functions, RF selection, abruption effects etc) is only partially explored. Such a study may reveal interesting connections between our method and existing approaches, such as the general conditions under which the AGM postulates are satisfied. It would also allow us to formulate some general, formal methods regarding the integration of operators that was informally described above. Finally, knowledge representation issues could be addressed under the light of this new representation.

References