December 2023, No 12

Scripta-Ingenia

The future of theoretical physics

In the realm of mathematical physics, a vast tapestry of knowledge unfolds before our eyes. From the elegant symmetries of classical mechanics to the mind-bending implications of quantum theory, the field has undergone incredible transformations, driven by the relentless pursuit of understanding the fundamental nature of the universe. In the early 20th century, classical mechanics gave a foundation to mathematical physics by describing the motion of particles and systems. However, the revolutionary theories of special and general relativity emerged to explain observations that couldn't be explained by classical mechanics. These theories showed that spacetime is a dynamic entity that can be curved by mass and energy. In the twenties, quantum mechanics challenged classical intuition by introducing uncertainty, wave-particle duality, and non-local entanglement. It provided a powerful mathematical tool, using complex numbers and linear algebra, to understand the behavior of microscopic particles.

The current focus of mathematical physics is on reconciling general relativity and quantum mechanics. Looking into the future, mathematical physics holds immense potential for unraveling the deepest mysteries of the universe. As we venture into the uncharted territories of dark matter, dark energy, and the nature of the early universe, mathematical tools will continue to be a guiding force in our exploration. From the perspective of mathematics, the beauty lies in its abstract nature, allowing us to construct models and theories that go beyond what we can directly observe.

One fascinating area of mathematical physics is Roger Penrose's twistor theory. Twistor theory is a mathematical framework that aims to provide a deeper understanding of the fundamental nature of spacetime and particles. Its elegance lies in the notion of twistors, which are mathematical objects that encode both space and time information.

Twistors offer an alternative approach to describing physical phenomena, departing from traditional approaches that rely on spacetime coordinates. By using twistor space, one can bypass the complexities associated with the treatment of singularities and infinities in physics, and instead focus on the geometric relationships between twistors. By studying the geometric properties and symmetries of twistors, researchers aim to shed light on the underlying structures that may give rise to phenomena such as dark matter and dark energy. Moreover, twistor theory has implications for understanding the early universe.

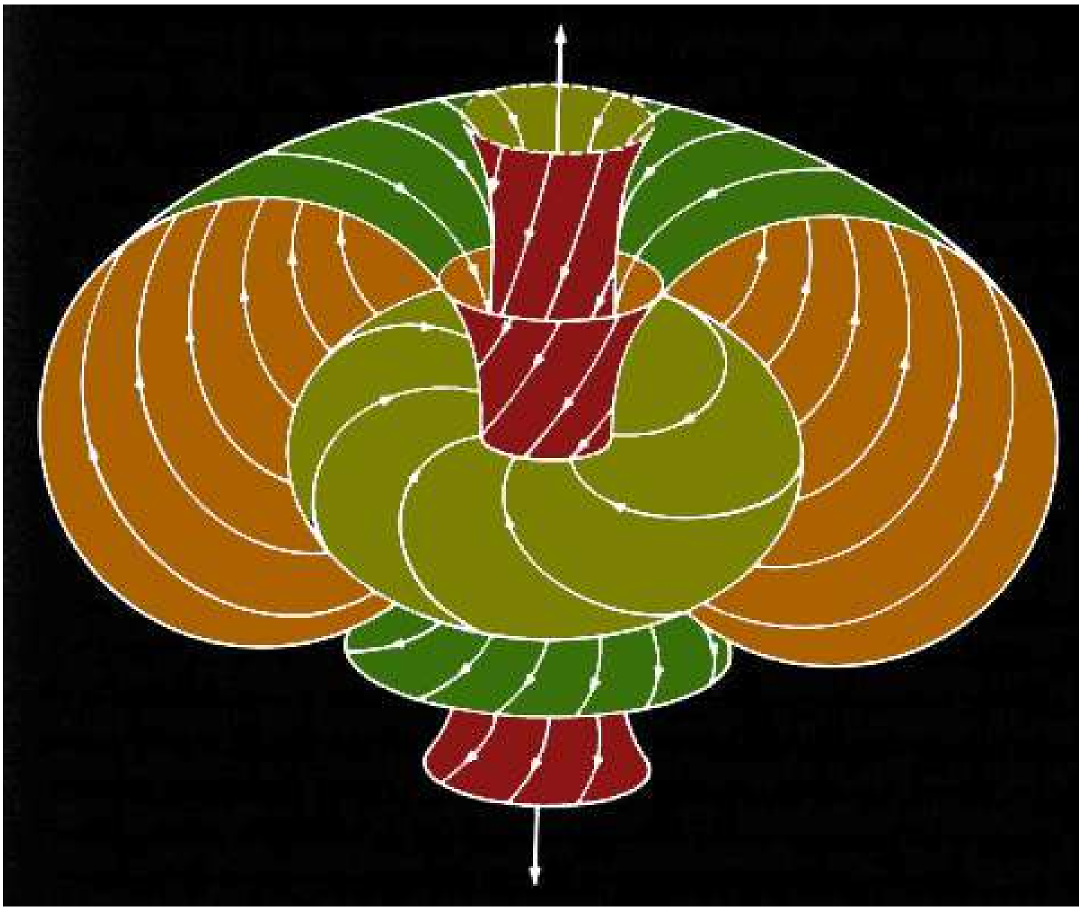
In summary, as we push the boundaries of our knowledge in physics, mathematicians and physicists continue to collaborate to develop new tools and theories. Roger Penrose's twistor theory is an example of a mathematical framework that holds immense potential for addressing some of the deepest mysteries of the universe.

H. Poincaré, *The present and the future of Mathematical Physics*, Bull. Amer. Math. Soc. 12 (1906), 240–260.

R. Penrose, The Road to Reality : A Complete Guide to the Laws of the Universe, Random House, London, (2005).

A Scripta-Ingenia assume-se como uma revista de divulgação científica tratando temas da ciência e da tecnologia, cobrindo todas as áreas do saber no domínio das ciências exactas ou aplicadas. Interessa-se ainda por artigos de opinião, sobre tópicos científicos ou não, desde que escritos por autores na área das ciências e da engenharia, e que reflitam as suas opiniões enquanto membros dessa comunidade. Este é o seu número 12 e corresponde ao Solstício de Inverno de 2023.

Director and Chief Editor — Nelson Martins-Ferreira
CDRSP-ESTG, IPLeiria



Un rayo de luz (fotón) como un twistor según la teoría de Roger Penrose
De Arenillas - Trabajo propio, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=21365811

Ficha Técnica
Director e Editor: Nelson Martins Ferreira
Proprietário: Instituto Politécnico de Leiria, anotado na ERC
Morada e Sede de Redação:
Rua General Norton de Matos; Apartado 4133; 2411-901 Leiria, Portugal
Contacto: scripta.ingenia@ipleiria.pt
Colaboradores e estatuto editorial: `http://cdrsp.ipleiria.pt/scriptaingenia/`

Ternary operations in classical algebraic structures

by J. P. Fatelo and N. Martins-Ferreira

Centre for Rapid and Sustainable Product Development
School of Technology and Management
Polytechnic Institute of Leiria
Author email: jorge.fatelo@ipleiria.pt; martins.ferreira@ipleiria.pt

Abstract Most commonly studied algebraic structures involve unary and binary operations solely. However, there have been proposals for defining various algebraic systems using ternary operations, such as Boolean algebras or distributive lattices. This article reviews successful efforts in this area and proposes suggestions for ternary MV-algebras.

1 Introduction

Algebraic structures are widely studied and exhibit great diversity. Examples of such systems include lattices, rings, and their special cases or generalizations. However, these structures typically involve constants and unary or binary operations only. The use of ternary operations, which encompass these cases as well, is not commonly employed. Ternary operations can offer a deeper understanding of the axiomatic system, simplify it, and potentially aid in finding unconventional generalizations.

In Section 2, the ternary definition proposed in 1948 by Birkhoff and Kiss [2] is revisited, and its equivalence to the conventional definition of bounded distributive lattice is proved in detail while employing a more explicit proof than the one presented in [2].

Another example can be found in [9] where Boolean algebras are defined successfully through a ternary operation. Ternary operations have also been used recently ([5, 6, 7]) to introduce mobi algebras and mobi spaces. Following [8], this article elaborates on a possible ternary version of MV-algebras. MV-algebras are briefly recalled in Section 3, and Section 4 presents an axiomatic framework with a ternary operation for these algebras.

2 Bounded distributive lattices

A bounded distributive lattice may be defined as an algebraic structure $(A, \vee, \wedge, 0, 1)$, where A is a set, $\vee$ and $\wedge$ are binary operations, 0 and 1 are elements of A, satisfying the usual axioms of Definition 1 below.

Definition 1. *A bounded distributive lattice is a system* $(A, \vee, \wedge, 0, 1)$ *such that*

(L1) $a \vee b = b \vee a, \quad a \wedge b = b \wedge a$

(L2) $a \vee (b \vee c) = (a \vee b) \vee c, \quad a \wedge (b \wedge c) = (a \wedge b) \wedge c$

(L3) $a \vee (b \wedge a) = a = (a \vee b) \wedge a$

(L4) $0 \vee a = a = a \wedge 1$

(L5) $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$

In [2], a structure with a ternary operation p and two constants were proposed to define bounded distributive lattices by postulating the four axioms of Definition 2 below.

Definition 2. *A bounded distributive lattice is a system* $(A, p, 0, 1)$ *such that*

(A1) $p(0, a, 1) = a$

(A2) $p(a, b, a) = a$

(A3) $p(a, p(b_1, b_2, b_3), c) = p(p(a, b_1, c), b_2, p(a, b_3, c))$

(A4) $p(a, b, c) = p(b, a, c) = p(b, c, a)$

The two structures are indeed isomorphic as will be proved in the next propositions.

Proposition 1. *Let* $(A, p, 0, 1)$ *be a structure verifying Definition 2. Then, with*

$$a \vee b = p(a, b, 1) \quad \text{and} \quad a \wedge b = p(0, a, b), \tag{2.1}$$

$(A, \vee, \wedge, 0, 1)$ *verifies Definition 1*

Demonstração. Each axiom of Definition 1 is proved:

(L1) $a \vee b = p(a, b, 1) = p(b, a, 1) = b \vee a$
$a \wedge b = p(0, a, b) = p(0, b, a) = b \wedge a$

(L2)
$$\begin{aligned} a \vee (b \vee c) &= p(a, p(b, c, 1), 1) \\ &= p(p(a, b, 1), c, p(a, 1, 1)) \\ &= p(p(a, b, 1), c, p(1, a, 1)) \\ &= p(p(a, b, 1), c, 1) = (a \vee b) \vee c \end{aligned}$$
$$\begin{aligned} (a \wedge b) \wedge c &= p(0, p(0, a, b), c) \\ &= p(p(0, 0, c), a, p(0, b, c)) \\ &= p(p(0, c, 0), a, p(0, b, c)) \\ &= p(0, a, p(0, b, c)) = a \wedge (b \wedge c) \end{aligned}$$

(L3) $$\begin{aligned} a \vee (b \wedge a) &= p(a, p(0, b, a), 1) \\ &= p(p(a, 0, 1), b, p(a, a, 1)) \\ &= p(p(0, a, 1), b, p(a, 1, a)) \\ &= p(a, b, a) = a \end{aligned}$$

$$\begin{aligned} (a \vee b) \wedge a &= p(0, p(a, b, 1), a) \\ &= p(p(0, a, a), b, p(0, 1, a)) \\ &= p(p(a, 0, a), b, p(0, a, 1)) \\ &= p(a, b, a) = a \end{aligned}$$

(L4) $0 \vee a = a \wedge 1 = p(0, a, 1) = a$
Remark:
$1 \vee a = p(1, a, 1) = 1$
$a \wedge 0 = p(0, a, 0) = 0$

(L5) $$\begin{aligned} a \vee (b \wedge c) &= p(a, p(0, b, c), 1) \\ &= p(a, p(b, 0, c), 1) \\ &= p(p(a, b, 1), 0, p(a, c, 1) \\ &= p(0, p(a, b, 1), p(a, c, 1) \\ &= (a \vee b) \wedge (a \vee c) \end{aligned}$$

$$\begin{aligned} a \wedge (b \vee c) &= p(0, a, p(b, c, 1)) \\ &= p(0, p(b, 1, c), a) \\ &= p(p(0, b, a), 1, p(0, c, a)) \\ &= p(p(0, a, b), p(0, a, c), 1) \\ &= (a \wedge b) \vee (a \wedge c). \end{aligned}$$

□

In the next proposition, it is shown that the ternary operation p has an explicit expression in terms of the two binary operations $\vee$ and $\wedge$.

Proposition 2. *Let $(A, p, 0, 1)$ be a structure verifying Definition 2 and consider the operations $a \vee b = p(a, b, 1)$ and $a \wedge b = p(0, a, b)$. Then the following relations hold.*

$$c \vee p(a, b, c) = c \vee (a \wedge b) \tag{2.2}$$
$$p(a, b, c) \wedge c = (a \vee b) \wedge c \tag{2.3}$$
$$p(a, b, c) = (a \wedge b) \vee (b \wedge c) \vee (c \wedge a). \tag{2.4}$$

Demonstração. First, we prove (2.2) and (2.3):
$$\begin{aligned} c \vee p(a, b, c) &= p(c, p(a, b, c), 1) \\ &= p(p(c, a, 1), b, p(c, 1, c)) \\ &= p(p(c, a, 1), b, c) \\ &= p(p(c, a, 1), b, p(c, 0, 1)) \\ &= p(c, p(a, b, 0), 1) = c \vee (a \wedge b) \end{aligned}$$
$$\begin{aligned} p(a, b, c) \wedge c &= p(0, p(a, b, c), c) \\ &= p(p(0, a, c), b, p(c, 0, c)) \\ &= p(p(0, a, c), b, c) \\ &= p(p(0, a, c), b, p(0, 1, c)) \\ &= p(0, p(a, b, 1), c) = (a \vee b) \wedge c. \end{aligned}$$
Now we prove (2.4). It is easy to check that, in a distributive lattice $(A, \vee, \wedge)$, the expression f defined by

$$f(a, b, c) = (a \wedge b) \vee (b \wedge c) \vee (c \wedge a)$$

is equal to

$$f(a, b, c) = (a \vee b) \wedge (b \vee c) \wedge (c \vee a)$$

and verifies $\begin{cases} c \vee f(a, b, c) &= & c \vee (a \wedge b) \\ f(a, b, c) \wedge c &= & (a \vee b) \wedge c \end{cases}$.
Consequently, we can deduce that $p = f$, indeed:
$$\begin{aligned} p &= p \wedge (c \vee p) = p \wedge (c \vee f) = (p \wedge c) \vee (p \wedge f) \\ &= (f \wedge c) \vee (f \wedge p) = f \wedge (c \vee p) = f \wedge (c \vee f) = f. \end{aligned}$$
□

Proposition 3. *Let $(A, \vee, \wedge, 0, 1)$ be a structure verifying Definition 1. Then, with*

$$p(a, b, c) = (a \wedge b) \vee (b \wedge c) \vee (c \wedge a), \tag{2.5}$$

$(A, p, 0, 1)$ verifies Definition 2.

Demonstração. Each axiom of Definition 2 is proved next.

(A1) $p(0, a, 1) = (0 \wedge a) \vee (a \wedge 1) \vee (1 \wedge 0) = 0 \vee a \vee 0 = a$

(A2) $p(a, b, a) = (a \wedge b) \vee (b \wedge a) \vee (a \wedge a) = (a \wedge b) \vee a = a$

(A3) $$\begin{aligned} &p(a, p(b_1, b_2, b_3), c) \\ &= (a \wedge p(b_1, b_2, b_3)) \vee (p(b_1, b_2, b_3) \wedge c) \vee (c \wedge a) \\ &= (a \wedge b_1 \wedge b_2) \vee (a \wedge b_2 \wedge b_3) \vee (a \wedge b_3 \wedge b_1) \\ &\vee (c \wedge b_1 \wedge b_2) \vee (c \wedge b_2 \wedge b_3) \vee (c \wedge b_3 \wedge b_1) \\ &\vee (c \wedge a) \vee (c \wedge a \wedge b_2) \vee (c \wedge a \wedge b_2) \\ &= (b_2 \wedge p(a, b_1, c)) \vee (b_2 \wedge p(a, b_3, c)) \\ &\vee (a \wedge b_3 \wedge b_1) \vee (c \wedge b_3 \wedge b_1) \vee (c \wedge a) \\ &= (b_2 \wedge p(a, b_1, c)) \vee (b_2 \wedge p(a, b_3, c)) \\ &\vee (p(a, b_1, c) \wedge p(a, b_2, c)) \\ &= p(p(a, b_1, c), b_2, p(a, b_3, c)) \end{aligned}$$

(A4) follows from the commutativity of $\vee$ and $\wedge$.

□

Theorem 1. *The structure $(A, \vee, \wedge, 0, 1)$ verifying Definition 1 and the structure $(A, p, 0, 1)$ verifying Definition 2 are isomorphic.*

Demonstração. Starting from $(A, \vee, \wedge, 0, 1)$, a structure $(A, p, 0, 1)$ is obtained by Proposition 3 and, from it, Proposition 1 gives back a bounded distributive lattice $(a, \vee', \wedge', 0, 1)$ equal to the original one:
$$\begin{aligned} a \wedge' b &= p(0, a, b) \\ &= (0 \wedge a) \vee (a \wedge b) \vee (b \wedge 0) = 0 \vee (a \wedge b) \vee 0 \\ &= a \wedge b \end{aligned}$$
$$\begin{aligned} a \vee' b &= p(a, b, 1) \\ &= (a \wedge b) \vee (b \wedge 1) \vee (1 \wedge a) = (a \wedge b) \vee b \vee a \\ &= b \vee a = a \vee b. \end{aligned}$$
Starting from $(A, p, 0, 1)$, a structure $(A, \vee, \wedge, 0, 1)$ is obtained by Proposition 1 and, from it, Proposition 3 gives back a ternary structure $(A, p', 0, 1)$ equal to the original one:

$$p'(a, b, c) = (a \wedge b) \vee (b \wedge c) \vee (c \wedge a) = p(a, b, c),$$

due to 2.4. □

3 MV-algebras

The well-known MV-algebras (see e.g. [1, 3, 4] and [8]) are structures of type $(2, 1, 0)$ which may be defined as follows.

Definition 3. *A MV-algebra is a system* $(X, \circ, \overline{()}, 0)$ *such that:*

(M1) $x \circ (y \circ z) = (x \circ y) \circ z$

(M2) $0 \circ x = x$

(M3) $\overline{\overline{x}} = x$

(M4) $\overline{0} \circ x = \overline{0}$

(M5) $x \circ \overline{x \circ \overline{y}} = y \circ \overline{y \circ \overline{x}}$

Usually, the commutativity of the binary operation $\circ$ is included as an axiom of MV-algebras. The proof of Property **(P4)** below, showing that commutativity of $\circ$ is a consequence of the other axioms, is due to Kolařík [10].

Proposition 4. *Let* $(X, \circ, \overline{()}, 0)$ *be a MV-algebra. It follows that:*

(P1) $x \circ 0 = x$

(P2) $x \circ \overline{x} = \overline{0}$

(P3) $(x \circ y) \circ \overline{y \circ x} = \overline{0}$

(P4) $x \circ y = y \circ x$

Demonstração. We begin with the proof of **(P1)**:

$$x\circ 0 = 0\circ\overline{0 \circ \overline{x \circ 0}} = x\circ 0\circ\overline{x \circ 0 \circ \overline{0}} = x\circ\overline{x \circ \overline{0}} = 0\circ\overline{0 \circ \overline{x}} = x.$$

Using **(P1)**, the complement property **(P2)** is easily proved:

$$x \circ \overline{x} = x \circ \overline{x \circ 0} = x \circ \overline{x \circ \overline{\overline{0}}} = \overline{0} \circ \overline{\overline{0} \circ \overline{x}} = \overline{0}.$$

Now, we can use **(P2)** to prove **(P3)**:

$$(x \circ y) \circ \overline{y \circ x} = x \circ (y \circ \overline{y \circ \overline{\overline{x}}}) = x \circ \overline{x} \circ \overline{\overline{x} \circ \overline{y}} = \overline{0} \circ \overline{\overline{x} \circ \overline{y}} = \overline{0}.$$

Properties **(P1)** and **(P3)** are used to prove **(P4)**:

$$\begin{aligned} x \circ y = (x \circ y) \circ 0 &= (x \circ y) \circ \overline{(x \circ y) \circ \overline{y \circ x}} \\ &= (y \circ x) \circ \overline{(y \circ x) \circ \overline{x \circ y}} = y \circ x. \end{aligned}$$

□

In a MV-algebra $(X, \circ, \overline{()}, 0)$, the operation $\circ$ is cancellative within the conditions presented in the next proposition.

Proposition 5. *Let* $(X, \circ, \overline{()}, 0)$ *be a MV-algebra, then:*

$$\left\{ \begin{array}{lcl} z \circ x & = & z \circ y \\ \overline{x} \circ \overline{z} & = & \overline{0} \\ \overline{y} \circ \overline{z} & = & \overline{0} \end{array} \right. \quad \Rightarrow \quad x = y. \tag{3.1}$$

Demonstração. The proof follows the one presented in [3].

$$\begin{aligned} x &= \overline{\overline{x} \circ 0} = \overline{\overline{x} \circ \overline{\overline{x} \circ \overline{z}}} = \overline{z \circ \overline{z \circ x}} = \overline{z \circ \overline{z \circ y}} = \overline{\overline{y} \circ \overline{\overline{y} \circ \overline{z}}} \\ &= \overline{\overline{y} \circ 0} = y. \end{aligned}$$

□

Remark: In a MV-algebra, the natural partial order is:

$$x \leq y \Leftrightarrow \overline{x} \circ y = \overline{0}. \tag{3.2}$$

Therefore the previous proposition can be written as:

$$z \circ x = z \circ y, \quad x \leq \overline{z} \quad \text{and} \quad y \leq \overline{z} \quad \Rightarrow \quad x = y.$$

In a MV-algebra, several derived operations are of interest.

Definition 4. *Let* $(X, \circ, \overline{()}, 0)$ *be a MV-algebra. Then the following binary operations are defined:*

$$\begin{aligned} x \cdot y &= \overline{\overline{y} \circ \overline{x}} && (3.3) \\ x \vee y &= x \circ (y \cdot \overline{x}) && (3.4) \\ x \wedge y &= (\overline{y} \circ x) \cdot y. && (3.5) \end{aligned}$$

In particular, **(M3)** implies that $\circ$ and $\cdot$ are dual operations as well as $\vee$ and $\wedge$:

$$\begin{aligned} x \circ y &= \overline{\overline{y} \cdot \overline{x}} && (3.6) \\ x \wedge y = \overline{\overline{y} \vee \overline{x}} \quad &, \quad x \vee y = \overline{\overline{y} \wedge \overline{x}}. && (3.7) \end{aligned}$$

With these operations, Axiom **(M5)** may be rewritten as

$$x \circ (y \cdot \overline{x}) = y \circ (x \cdot \overline{y}) \quad \text{or} \quad x \vee y = y \vee x.$$

Or, by duality, it can be rewritten as

$$(\overline{x} \circ y) \cdot x = (\overline{y} \circ x) \cdot y \quad \text{or} \quad y \wedge x = x \wedge y.$$

Note also that (3.2) and (3.5) imply $y \leq x \Leftrightarrow x \wedge y = y$, and that, in any MV-algebra, the following ordering is true:

$$x \cdot y \leq x \wedge y \leq x \leq x \vee y \leq x \circ y.$$

For convenience, we will call

$$\overline{0} = 1. \tag{3.8}$$

We list here some other obvious rewriting of previous properties using the derived operations:

$$\begin{array}{rcl} \overline{1} & = & 0 \\ x \cdot y & = & y \cdot x \\ x \cdot (y \cdot z) & = & (x \cdot y) \cdot z \\ x \cdot 1 = & x & = 1 \cdot x \\ x \cdot 0 = & 0 & = 0 \cdot x \\ x \cdot \overline{x} & = & 0 \end{array}$$

$$\left\{ \begin{array}{lcl} z \circ x & = & z \circ y \\ z \cdot x & = & 0 \\ z \cdot y & = & 0 \end{array} \right. \quad \Rightarrow \quad x = y.$$

It is well known that a distributive lattice is present in each MV-algebra through derived operations.

Proposition 6. *Let* $(X, \circ, \cdot, \overline{()}, 0)$ *be a MV-algebra. Then, the structure* $(X, \vee, \wedge, 0, 1)$ *is a bounded distributive lattice.*

Demonstração. This is routine of MV-algebras ([1, 4]). □

The way the derived operations interact with the original ones is also widely known. The next proposition only presents a few relations, in particular **(P7)** that will be needed in Section 4.

Proposition 7. *Let* $(X, \circ, \cdot, \overline{()}, 0)$ *be a MV-algebra and* $(X, \vee, \wedge, 0, 1)$ *the underlying lattice structure. Then:*

(P5) $x \circ (y \wedge z) = (x \circ y) \wedge (x \circ z)$

(P6) $x \cdot (y \vee z) = (x \cdot y) \vee (x \cdot z)$

(P7) $(\overline{y} \wedge x) \circ y = x \circ y$.

Demonstração. We refer to [4] for the proofs of **(P5)** and **(P6)**. Property **(P7)** is just a consequence of **(P5)**, commutativity of $\circ$, **(P2)** and the lattice property $1 \wedge x = x$. □

4 A ternary MV structure

Consider, in any structure $(A, p, 0, 1)$ of type $(3, 1, 1)$, the following derived operations:

$$\overline{a} = p(1, a, 0) \quad (4.1)$$

$$a \cdot b = p(0, a, b) \quad (4.2)$$

$$a \circ b = p(a, b, 1) \quad (4.3)$$

$$\tilde{p}(a, b, c) = p(a, p(\overline{c}, b, \overline{a}), c) \quad (4.4)$$

The notations $\overline{()}$, $\circ$ and $\cdot$, already used in the previous section, are applied again here because it will be proved that, within the framework used here, the operations are the same. With respect to $\vee$ and $\wedge$, these operations can be obtained, from $\tilde{p}$, the following way:

$$a \vee b = \tilde{p}(a, b, 1) \quad (4.5)$$

$$a \wedge b = \tilde{p}(0, a, b). \quad (4.6)$$

It will also be proved in Proposition 11 that in a Ternary MV-algebra, as defined next, the derived ternary operation $\tilde{p}$ is completely commutative.

Definition 5. *A system* $(A, p, 0, 1)$ *of type* $(3, 1, 1)$ *is said to be a Ternary MV-algebra when:*

(T1) $p(0, a, 1) = a$

(T2) $p(a, 0, b) = a = p(b, 1, a)$

(T3) $p(a, b, a) = a$

(T4) $p(a, p(b, c, 1), 1) = p(p(a, b, 1), c, 1)$

(T5) $\tilde{p}(a, b, 1) = \tilde{p}(b, a, 1)$

(T6) $p(0, b, a) = \overline{p(\overline{a}, \overline{b}, 1)}$

Proposition 8. *If* $(A, p, 0, 1)$ *is a Ternary MV-algebra then:*

i. $\overline{0} = 1, \quad \overline{1} = 0$

ii. $\overline{\overline{a}} = a$

iii. $b \cdot a = \overline{\overline{a} \circ \overline{b}}$

Demonstração. Only **(T2)** and **(T6)** are used.

i. $\overline{0} = p(1, 0, 0) = 1, \quad \overline{1} = p(1, 1, 0) = 0$

ii. $a = p(0, 1, a) = \overline{p(\overline{a}, 0, 1)} = \overline{\overline{a}}$

iii. It is just a rewriting of **(T6)**.

□

Proposition 9. *If* $(A, p, 0, 1)$ *is a Ternary MV-algebra then* $(A, \circ, \overline{()}, 0)$ *is a MV-algebra.*

Demonstração. The proof is quite straightforward.

(M1) $x \circ (y \circ z) = p(x, p(y, z, 1), 1) = p(p(x, y, 1), z, 1) = (x \circ y) \circ z$

(M2) $0 \circ x = p(0, x, 1) = x$

(M3) $\overline{\overline{x}} = x$ by Proposition 8

(M4) $1 \circ x = p(1, x, 1) = 1$

(M5) $\tilde{p}(x, y, 1) = \tilde{p}(y, x, 1)$

$$\Rightarrow p(x, p(0, y, \overline{x}), 1) = p(y, p(0, x, \overline{y}), 1)$$

$$\Rightarrow x \circ (y \cdot \overline{x}) = y \circ (x \cdot \overline{y}) \Rightarrow x \circ \overline{x \circ \overline{y}} = y \circ \overline{y \circ \overline{x}}$$

□

The next proposition show how to construct a Ternary MV-algebra from a MV-algebra.

Proposition 10. *If* $(A, \circ, \overline{()}, 0)$ *is a MV-algebra then* $(A, p, 0, 1)$ *is a Ternary MV-algebra with*

$$p(a, b, c) = (\overline{b \cdot a} \cdot a) \circ (b \cdot c). \quad (4.7)$$

Demonstração. First, the consistency of definitions (4.1),(4.2) and (4.3) is checked:

$p(1, a, 0) = (\overline{a \cdot 1} \cdot 1) \circ (a \cdot 0) = \overline{a} \circ 0 = \overline{a}$

$p(0, a, b) = (\overline{a \cdot 0} \cdot 0) \circ (a \cdot b) = 0 \circ (a \cdot b) = a \cdot b$

$p(a, b, 1) = (\overline{b \cdot a} \cdot a) \circ (b \cdot 1) = ((\overline{a} \circ \overline{b}) \cdot a) \circ b = (\overline{b} \wedge a) \circ b = a \circ b$, using **(P7)**.

Now, axioms of Definition 5 are proved:

(T1) $p(0, a, 1) = (\overline{a \cdot 0} \cdot 0) \circ (a \cdot 1) = 0 \circ a = a$

(T2) $p(a, 0, b) = (\overline{0 \cdot a} \cdot a) \circ (0 \cdot b) = (1 \cdot a) \circ 0 = a$
$p(a, 1, b) = (\overline{1 \cdot a} \cdot a) \circ (1 \cdot b) = (\overline{a} \cdot a) \circ b = 0 \circ b = b$

(T3) $p(a, b, a) = (\overline{b \cdot a} \cdot a) \circ (b \cdot a) = (b \cdot a) \vee a = (b \cdot a) \vee (1 \cdot a) = (b \vee 1) \cdot a = 1 \cdot a = a$

(T4) $p(a,p(b,c,1),1) = a\circ(b\circ c) = (a\circ b)\circ c = p(p(a,b,1),c,1)$

(T5) $\tilde{p}(a,b,1) = a\vee b = b\vee a = \tilde{p}(b,a,1)$

(T6) $\overline{p(\overline{a},\overline{b},1)} = \overline{\overline{a}\circ\overline{b}} = b\cdot a = p(0,b,a)$

□

Proposition 11. *If $(A,p,0,1)$ is a Ternary MV-algebra and $\tilde{p}$ defined as in (4.4) then:*

1. $\tilde{p}(0,a,1) = a$
2. $\tilde{p}(a,b,a) = a$
3. $\tilde{p}(b,a,c) = \tilde{p}(a,b,c) = \tilde{p}(b,c,a)$
4. $\tilde{p}(a,\tilde{p}(b_1,b_2,b_3),c) = \tilde{p}(\tilde{p}(a,b_1,c),b_2,\tilde{p}(a,b_3,c))$.

Demonstração. Those 4 properties are just a consequence of Propositions 9 and 6 and Theorem 1. □

5 Conclusion

In Section 2, we revisited the system proposed by Birkhoff and Kiss in [2], with a ternary operation and two constants, which is isomorphic to bounded distributive lattices. The complete proof was presented as an illustration of the procedure to construct an isomorphism of structures. Section 4 was an essay to find an isomorphism between MV-algebras and a structure involving a ternary operation p. A derived operation $\tilde{p}$ is found and, as expected, verifies all the axioms of the ternary system of Birkhoff and Kiss. If the expression of p, given in (4.7) can be deduced from a Ternary MV-algebra then this structure is isomorphic to MV-algebras. In any case, if (4.7) is added to Definition 5 as an extra axiom, a ternary structure isomorphic to MV-algebras is obtained.

Acknowledgements This work is supported by the Fundação para a Ciência e a Tecnologia (FCT) and Centro2020 through the Project references: UID/-Multi/04044/2019; PAMI - ROTEIRO/0328/2013 (Nº 022158); Next.parts (17963), and also by CDRSP and ESTG from the Polytechnic Institute of Leiria.

Referências

[1] M. Bergmann, *An introduction to many-valued and fuzzy logic*, Cambridge University Press (2008).

[2] G. Birkhoff, S. A. Kiss, *A ternary operation in distributive lattices*, Bull. Amer. Math. Soc. **53** (8) (1947) 749–752.

[3] C. C. Chang, *Algebraic analysis of many valued logics*, Transactions of the American Mathematical society **88** (2) (1958) 467-490.

[4] R.L.O. Cignoli, I.M.L. D'Ottaviano, D. Mundici, *Algebraic foundations of many-valued reasoning*, Trends in Logic (2000).

[5] J. P. Fatelo, N. Martins-Ferreira, *Mobi algebra as an abstraction to the unit interval and its comparison to rings*, Communications in Algebra **47** (3) (2019) 1197–1214.

[6] J. P. Fatelo, N. Martins-Ferreira, *Affine mobi spaces*, Boll Unione Mat Ital **15** (2022) 589–604.

[7] J. P. Fatelo, N. Martins-Ferreira, *Mobi spaces and geodesics for the N-sphere*, Cah. Topol. Géom. Différ. Catég. **63** (1) (2022) 59–88.

[8] J. P. Fatelo and N. Martins-Ferreira, *A brief survey on MV-algebras*, Scripta-Ingenia **11** (December) (2022) 16–19.

[9] A. A. Grau, *Ternary Boolean algebras*, Bull. Amer. Math. Soc. **53** (6) (1947) 567–572.

[10] M. Kolařík, *Independence of the axiomatic systems for a MV-algebras*, Math. Slovaca **63** (2013) 1–4.

A generalized construction for algebraic number systems

by N. Martins-Ferreira[1] and R. A. P. Perdigão[2,3]

(1) Centre for Rapid and Sustainable Product Development
School of Technology and Management
Polytechnic Institute of Leiria, Portugal
(2) IUC Physics of Complex Coevolutionary Systems and Fluid Dynamical Systems
Meteoceanics Institute for Complex Systems Science, Washington, DC, USA
(3) Synergistic Manifolds, Portugal
Author email: martins.ferreira@ipleiria.pt; perdigao@meteoceanics.org

Abstract A generalized construction procedure for algebraic number systems is hereby presented. This procedure offers an efficient representation and computation method for complex numbers, quaternions and other algebraic structures. The construction method benefits, applications, and potential impacts across various mathematical and scientific fields are discussed. In particular, the novel developments reported herein provide a generalized form of the Cayley-Dickson construction, thereby allowing for the treatment of more general spaces other than vector spaces, which underlie the associated Algebra structure.

1 Introduction

The Cayley-Dickson construction is a powerful algebraic tool that generates a sequence of number systems, known as the Cayley-Dickson algebras. These algebras extend the real numbers to more complex structures, incorporating properties such as addition, multiplication, and conjugation. Originally introduced in 1845 by the mathematician Arthur Cayley, and later analyzed by Leonard Eugene Dickson in 1919, the Cayley-Dickson construction has found significant applications in various branches of mathematics, including algebra, analysis, and geometry.

In this paper, we propose a generalized Cayley-Dickson construction that extends the original construction to a broader class of number systems. By relaxing certain constraints imposed by the traditional construction, we explore new possibilities and expand the scope of algebraic structures that can be generated.

The motivation behind this investigation is to provide a systematic framework for studying and understanding a more diverse range of algebraic systems. By leveraging the generalized Cayley-Dickson construction, we aim to uncover novel properties and structures that arise when extending the real numbers to higher dimensions. Additionally, we anticipate that the construction will shed light on the connections between previously disparate algebraic systems and reveal underlying patterns and symmetries.

We will present a generalized Cayley-Dickson construction methodology and illustrate its applicability by constructing higher-dimensional algebras. As a future work it will be interesting to investigate the algebraic properties of these extended number systems, such as associativity, commutativity, and the existence of inverses. It will also be worthwhile to explore how these new structures relate to existing algebraic systems and investigate their geometric interpretations when applicable.

The significance of our generalized construction lies in its potential to enrich our understanding of mathematical structures and open up new avenues for research in various domains. By extending the reach of the Cayley-Dickson construction beyond the traditional boundaries, we hope to contribute to the advancement of algebraic theory and its applications in diverse scientific fields.

In the subsequent sections of this paper, we will provide the necessary background on the original Cayley-Dickson construction, formalize the generalized approach, illustrate its application through concrete examples, and analyze the algebraic, geometric, and analytical implications of the generated structures. Finally, we will discuss potential future directions for research and the broader impact of this generalized construction in mathematics.

2 The classical construction

The Cayley-Dickson construction is a mathematical process used to construct a sequence of algebras with more complex properties from a given algebra. It was introduced independently by the mathematicians Arthur Cayley and Robert Dickson in the 19th century.

The construction starts with a given algebra, often the real numbers or the complex numbers, and defines a new algebra based on the original one. This process involves doubling the dimension of the original algebra by introducing new elements and defining new multiplication rules.

Specifically, for a given algebra with dimension n, the Cayley-Dickson construction defines a new algebra with dimension $2n$. This is achieved by extending the original algebra with new elements called "imaginary units"that satisfy certain multiplication rules.

In each step of the Cayley-Dickson construction, the new algebra inherits the addition and scalar multiplication properties of the original algebra. However, the multiplication operation is modified to incorporate the properties of the new imaginary units.

More specifically, Let $K = (K, 0, 1, +, \cdot)$ be a field and let $A = (A, 0, 1, +, \cdot)$ be a K-algebra with an involution $\overline{()}\colon A \to A$ with $\overline{x \cdot y} = \overline{y} \cdot \overline{x}$ and $\overline{x + y} = \overline{x} + \overline{y}$. The cayley-Dickson construction considers an algebra structure on the set $A \times A$ with involution $(a_1, a_2)^* = (\overline{a_1}, -a_2)$ and product

$$(a_1, a_2) \cdot (b_1, b_2) = (a_1 b_1 - \overline{b_2} a_2, a_2 \overline{b1} + b_2 a_1) \tag{2.1}$$

The Cayley-Dickson construction can be iterated multiple times to create a sequence of algebras, known as Cayley-Dickson algebras. These algebras possess interesting properties, such as division algebras for the complex numbers and quaternions. However, at each step, some of the properties of the original algebra are lost, such as commutativity and associativity of multiplication.

We observe that in essence, the process starts with an algebraic system of the form $A = (A, 0, 1, +, -, \cdot, ()^*)$ and construct a new algebraic system on the product $A \times A$ with component-wise addition and subtraction, with $(0, 0)$ as neutral element for addition and $(1, 0)$ as neutral element for multiplication, whereas conjugation is defined as $\overline{(a_1, a_2)} = (a_1^*, -a_2)$ and multiplication as

$$(a_1, a_2) \cdot (b_1, b_2) = (a_1 b_1 - {b_2}^* a_2, a_2 {b_1}^* + b_2 a_1) \tag{2.2}$$

This suggests that scalar multiplication is not involved and that we may even relax the associativity conditions on addition and multiplication from the beginning. This will be done in the following section.

The celebrated example of quaternions $q = t + xi + yj + zk$ viewed as a pair of complex numbers (z_1, z_2) with the quaternion conjugate $q = t - xi - yj - zk$ written as $(\bar{z_1}, -z_2)$ with $z_1 = t + xi$ and $z_2 = y + zi$ will be recovered as a special case.

Overall, the Cayley-Dickson construction provides a powerful and systematic way to generate new and increasingly complex algebras from a given starting point, leading to the exploration of non-commutative and non-associative algebraic structures.

3 A more general construction

As observed in the previous section, the vector space part of the algebra structure does not play a significant role in it. Therefore, instead of classical algebras we will consider algebraic systems, hereby denoted *unitary dimagmas*. These consist of two binary operations over the same set, each of which with an identity element and satisfying suitable conditions. More specifically, a *unitary dimagma* is an algebraic system $(A, +, \cdot, 0, 1)$ such that every equation of the form $x + a = b$ has a unique solution, namely $b - a \in A$. Thus, in particular we write $-a$ to signify $0 - a$. Moreover, the involution map will be given as part of the structure.

Before particularizing our general construction to the unitary case, we describe a more general construction which does not assume the neutral elements 0 and 1.

Let $X = (X, +)$ be a *magma* and $B = (B, +, \cdot)$ a *dimagma*, i.e. a set equipped with two independent binary operations. Suppose that X and B are connected via two maps given by:

$$\begin{tikzcd} X \arrow[r, shift right, "h"'] & B \arrow[l, shift right, "t"'] \end{tikzcd} \tag{3.1}$$

For the moment, we are not assuming the two maps to preserve any of the magma structures involved.

We hereby define a 4-*tensor-action* structure on X as the following representation of four *tensor action* operations defined on the set X given by:

$$x \otimes_1 x' \tag{3.2}$$
$$x \otimes_2 b' \tag{3.3}$$
$$b \otimes_3 x' \tag{3.4}$$
$$b \otimes_4 b' \tag{3.5}$$

for every $x, x' \in X$ and $b, b' \in B$. In order to simplify the notation, we will omit the subscripts in the symbol $\otimes$, since ambiguity and misinterpretation are unlikely to emerge. In doing so, we write

$$x \otimes x' \tag{3.6}$$
$$x \otimes b' \tag{3.7}$$
$$b \otimes x' \tag{3.8}$$
$$b \otimes b' \tag{3.9}$$

With this data we define a dimagma structure on the set $X \times B$ as follows (note that the use of parenthesis cannot be avoided!):

$$\begin{aligned} (x, b) + (x', b') =& ((x + t(b)) + (x' + t(b')), \\ & (h(x) + b) + (h(x') + b')) \end{aligned} \tag{3.10}$$

$$\begin{aligned} (x, b) \cdot (x', b') =& ((x \otimes x' + t(h(x) \cdot h(x'))) \\ & + (x \otimes b' + t(h(x) \cdot b')) \\ & + (b \otimes x' + t(b \cdot h(x))) \\ & + (b \otimes b' + t(b \cdot b')), \\ & + (h(x \otimes x') + h(x) \cdot h(x')) \\ & + (h(x \otimes b') + h(x) \cdot b') \\ & + (h(b \otimes x') + b \cdot h(x)) \\ & + (h(b \otimes b') + b \cdot b')) \end{aligned} \tag{3.11}$$

In order to make the product operation more visible we can present it as

$$(x, b) \cdot (x', b') = (u(x, b, x', b'), v(x, b, x', b')) \tag{3.12}$$

where u and v are defined as

$$\begin{aligned} u = {} & (x \otimes x' + t(h(x) \cdot h(x'))) && (3.13)\\ & + (x \otimes b' + t(h(x) \cdot b')) \\ & + (b \otimes x' + t(b \cdot h(x))) \\ & + (b \otimes b' + t(b \cdot b')) \\ v = {} & (h(x \otimes x') + h(x) \cdot h(x')) && (3.14)\\ & + (h(x \otimes b') + h(x) \cdot b') \\ & + (h(b \otimes x') + b \cdot h(x)) \\ & + (h(b \otimes b') + b \cdot b') \end{aligned}$$

These developments can be seen as a generalization of the Cayley-Dickson construction, in the sense that we are no longer limited to having the Algebra generation restricted to doubling the dimension of the Algebra (e.g. Real, Complex, Quaternions, Octonions, Sedenions). This further allows to explore Algebras with dimensions other than powers of 2 (i.e. other than 2^k).

For each fixed $(X, +)$ and $(B, +, \cdot)$ with fixed h and t, equipped with a 4-tensor action structure as above, a measure of entropy can be obtained by finding all possible subsets $R \subseteq X \times B$ which are closed under the two magma operations $+$ and $\cdot$ defined on $X \times B$ and contain the pairs $(x, h(x))$ and $(t(b), b)$ for all $x \in X$ and $b \in B$. This will be conducted in a follow-up study.

4 An illustrative case

We now consider the following illustrative case of the generalization laid out in the previous section. Given an algebra $(A, 0, 1, +, -, \cdot, \overline{()})$ let $X = B = A$ and take $t(b) = -b$, $h(x) = \bar{x}$ for every $x, b \in A$. Moreover, let us set:

$$x + x' \text{ and } b + b' \text{ as the addition in } A \qquad (4.1)$$

$$b \cdot b \text{ and } x \otimes x' \text{ as the product in } A \qquad (4.2)$$

$$x \otimes b' = 0 \qquad (4.3)$$

$$b \otimes x' = \bar{b}\bar{x}' - x'\bar{b} \qquad (4.4)$$

$$b \otimes b' = bb' - b'\bar{b} \qquad (4.5)$$

noting that equations 4.2 to 4.5 comprise an implementation of the 4-tensor-action defined in the previous section.

It is readily checked that the formula 3.12 for the product in $A \times A$ is the same as the one in the Cayley-Dickson construction [2]. We further observe that the addition is no longer the usual component-wise addition as it would be expected in the Cayley-Dickson case. In our case we obtain:

$$(x, b) + (x', b') = ((x - b) + (x' - b'), (\overline{x} + b) + (\overline{x'} + b')) \qquad (4.6)$$

This new operation for the sum shall require a physical and geometrical interpretation, which will be developed in a follow-up study. Moreover, it may be relevant to investigate the case where X is the set of real numbers and B is the set of complex numbers (or vice versa), in order to explore the possibility of defining a meaningful product operation in $\mathbb{R}^3$ other than the usual cross product. For instance, we may consider a higher-order product operation admitting more than two input arguments. In this sense, since the additive structure is not required to be associative, there is still potential for development towards fulfilling the original dream of Hamilton in having a way to multiply triplets (see e.g. [3]).

Let us consider the following concrete example. First, take $X = \mathbb{R}^3$ and $B = \mathbb{R}$ where $x \otimes x'$ is the usual vector product and $x \otimes b'$ and $b \otimes x'$ are the usual scalar product. Next, take the map t to be such that $t(b) = 0$ for every $b \in \mathbb{R}$ and $h(x)$ to be the norm of the vector $x \in \mathbb{R}^3$. In this case the formula 3.12 becomes:

$$\begin{aligned} (x, b) \cdot (x', b') = {} & (x \otimes x' + x \otimes b' + b \otimes x', && (4.7)\\ & h(x)h(x')(1 + \sin(\theta)) + \\ & 2(b'h(x) + bh(x')) + bb') \end{aligned}$$

which can be compared to the usual quaternion multiplication:

$$\begin{aligned} (x, b) \cdot (x', b') = {} & (x \otimes x' + x \otimes b' + b \otimes x', && (4.8)\\ & - h(x)h(x')\cos(\theta) + bb') \end{aligned}$$

where θ is the angle between the two vectors x and x', in the sense that the vector operations coincide whereas the scalar operations differ. The difference between 4.7 and 4.8 is therefore only manifested in the second argument of the *rhs*, yielding:

$$h(x)h(x')(1 + \cos(\theta) + \sin(\theta)) + 2(b'h(x) + bh(x')) \qquad (4.9)$$

In particular, when $\theta = 0$ or $\theta = \frac{\pi}{2}$ the first term becomes $2h(x)h(x')$, whereas when $\theta = -\frac{\pi}{2}$ or $\theta = \pi$ the first term becomes 0.

5 A categorical example

Now we examine how to generate the underlying algebra A considered in the previous illustrative case. For that purpose, consider the unit interval $I = [0, 1]$ in the category of sets. Note, however, that our approach is also valid in any category with products and coproducts, with a selected object I playing the role of the unit interval. The essential structure of the unit interval is as follows. There is an involution, namely $t \mapsto 1 - t$, there is an addition, namely $\frac{t+s}{2}$ and the usual multiplication ts for any two elements $t, s \in I$.

Our Algebra shall consist of triplets (X, f, φ) where f is a function from X to I and φ is an endomap of X. Such an object is hereby called an I-link. Given two such

I-links (X, f, φ) and (Y, g, ψ) we define addition and multiplication operations as:

$$(X, f, \varphi) + (Y, g, \psi) = (X + Y, [f, g], \varphi + \psi) \tag{5.1}$$

$$(X, f, \varphi) \cdot (Y, g, \psi) = (X \times Y, f * g, \varphi \times \psi) \tag{5.2}$$

with $[f, g] \colon X + Y \to I$ the unique map such that $[f, g]\iota_X = f$ and $[f, g]\iota_Y = g$ which exists by the universal property of coproduct as illustrated.

$$\begin{tikzcd}
X \arrow[r, "\iota_X"] \arrow[dr, "f"'] & X + Y \arrow[d, "{[f,g]}"'] & Y \arrow[l, "\iota_Y"'] \arrow[dl, "g"] \\
& I &
\end{tikzcd}$$

Similarly, $\varphi + \psi$ and $\varphi \times \psi$ are induced as well. The map $f * g$ is defined as follows

$$(f * g)(x, y) = \sum_n \sum_m \frac{f(\varphi^n(x)) \cdot g(\psi^m(y))}{f(\varphi^n(x)) + g(\psi^m(y))}$$

with n, m assuming all values $0, 1, 2, \ldots$ and considering $\varphi^0 = 1_X$ and $\psi^0 = 1_Y$.

If we restrict to the case in which φ is a bijection, then n, m can vary over the range of the integers, rather than the natural numbers. In that case we can further define a conjugate $(f, \varphi)^* = (\overline{f}, \varphi^{-1})$ with $\overline{f}(t) = 1 - f(t)$, for all $t \in I$.

6 Conclusion

The categorical construction introduced in the present study generates an algebra $(A, +, \cdot, ()^*)$ endowed with properties deemed more general than those underlying the Cayley-Dickson construction. Two core generalization aspects pertain, on one hand, not being restricted to operating with vector spaces, and on the other hand allowing for dimensionalities other than natural powers of 2 in the construction of said algebras. The missing operation of subtraction can be seen as a property of whether the equation $a + x = b$ has solution $x = b - a$. All in all, this study equips us with the tools to build further constructions such as that considered in the illustrative case making use of the features brought up in our generalization.

7 Acknowledgements

N.M.F. acknowledges the Portuguese Foundation for Science and Technology *(Fundação para a Ciência e Tecnologia)* FCT/MCTES (PIDDAC) through the following Projects: Associate Laboratory ARISE LA/P/0112/2020; UIDP/04044/2020; UIDB/04044/2020; PAMI - ROTEIRO/0328/2013 (N° 022158); MATIS (CENTRO-01-0145-FEDER-000014 - 3362); Generative.Thermodynamic; by CDRSP and ESTG from the Polytechnic Institute of Leiria.

R.A.P.P. acknowledges the research flagships *Mathematical Physics and Predictability of Complex Coevolutionary Systems* (MR-220617) and *Quantum Information Technologies in the Earth Sciences* (QITES) from the Meteoceanics Institute for Complex System Science, along with the C2IMPRESS project supported by the European Union under the Horizon Europe grant 101074004.

8 References

1 C. Flaut, *About some properties of algebras obtained by the Cayley-Dickson process*, Palestine Journal of Mathematics, Vol. 3(Spec 1) (2014) , 388–394

2 L. E. Dickson, *On Quaternions and Their Generalization and the History of the Eight Square Theorem, Annals of Mathematics*, Second Series, Vol. 20, No. 3 (Mar., 1919), pp. 155-171 (17 pages)

3 A. Rice and E. Brown, *Why Hamilton Couldn't Multiply Triples*, The College Mathematics Journal 52(3) (May 2021), 185–192
DOI:10.1080/07468342.2021.1897418

A procedure for digraph chaining and its application to an iso-slice algorithm

by Nelson Martins-Ferreira[a] and Marta Pascoal[b,c]

[a] CDRSP and ESTG, Polytechnic of Leiria,
Campus 2, Morro do Lena - Alto do Vieiro
2411-901 Leiria - Portugal

[b] University of Coimbra, INESC-Coimbra
3030-290 Coimbra - Portugal

[c] Politecnico di Milano
P.za L. da Vinci 32, 20133 Milano, Italy

Abstract This paper investigates the structure of a link, which is defined as a morphism paired with an endomorphism of its domain, without any additional requirements. Furthermore, the paper examines digraphs, which consist of two parallel morphisms without any further conditions. We establish a straightforward forgetful functor that maps the category of links to the category of digraphs. Additionally, we introduce a construction that, although not functorial, associates a link to every digraph and finds several applications in computer science. The construction assigns the domain morphism of the digraph with the morphism of the link, and the codomain morphism of the digraph is recovered as the composition of the morphism of the link with its endomorphism. As a biproduct we also obtain another endomap on the structure of the link associated with a digraph of which the orbits recover the vertices in the original digraph.

Keywords: Digraph, link, iso-slice algorithm, contour levels.

1 Introduction

The notion of a multi-link was introduced in [4] as a tool for the encoding and systematization of new and more efficient algorithms in the aim of 3D-printing. Our concern in this article is with the simpler case of a link. Its origin was motivated from computer science and 3D-printing and may be recalled as follows [4, 5]. From the point of view of memory storage, the notion of a 2-dimensional matrix, indexed rows and columns, is not very different than an arbitrary n-dimensional matrix. However, in practice, what is really stored into the computer's memory disk is an array, and the most efficient arrays are the linear ones.

The structure of a multi-link has been proposed as a structure which on the one hand can be stored as a linear array of information, while, on the other hand, it can be used to encode highly non-trivial structures such as surfaces and their properties.

The special case of a link can be seen as an abstraction for the notion of a curve (appropriate for computational purposes). An important example of application of the structure of a link is observed in one step of the iso-slice algorithm [4] which we recall here for convenience. This algorithm has itself several other applications such as the generation of cooling and refrigerating channels in a mould.

The construction described here is required in the iso-slice algorithm at steps 5(d) and 5(e), see Section 5.

2 The categorical setting

Internal to every category we may consider the category of digraphs whose objects are pairs of parallel morphisms (usually denoted as source and target) in the given base category and morphisms are pairs of morphisms preserving source and target, as illustrated in the diagram below

$$\begin{tikzcd}
A \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] \arrow[d, "f"'] & B \arrow[d, "g"] \\
A' \arrow[r, shift left, "s'"] \arrow[r, shift right, "t'"'] & B'
\end{tikzcd} \tag{2.1}$$

When we say that a morphism of digraphs preserves source and target we mean that $s'f = gs$ and $t'f = gt$.

We may also form the category of links whose objects consist of a morphism together with an endomorphism of its domain, which may be displayed as

$$\begin{tikzcd}
A \arrow[loop left, "\varphi"] \arrow[r, "d"] & B
\end{tikzcd}$$

and a morphism, say between a link (A, B, d, φ) and a link (A', B, d', φ') is a pair of morphisms in the base category, say $f\colon A \to A'$ and $g\colon B \to B'$ such that $d'f = gd$ and $\varphi' f = f\varphi$, as illustrated

$$\begin{tikzcd} A \arrow[loop left, "\varphi"] \arrow[r, "d"] \arrow[d, "f"'] & B \arrow[d, "g"] \\ A' \arrow[loop left, "\varphi'"] \arrow[r, "d'"] & B' \end{tikzcd} \tag{2.2}$$

It is now clear that there is a natural forgetful functor which assigns the digraph $(A, B, d, d\varphi)$ to every link structure (A, B, d, φ). In the following section we present the outline of a procedure which reverses the situation. That is, given a digraph (A, B, s, t) it creates an endomap $\varphi\colon A \to A$ such that $t = s\varphi$.

As a byproduct of the procedure, we also get another endomap, $\theta\colon A \to A$ such that s is recovered as the orbits of θ, or in other words, s is the coequalizer of θ and 1_A. In particular, $s = s\theta$.

3 A Matlab implementation

Here is a possible implementation in Matlab which receives a digraph with source and target maps s and t and returns two endomaps φ and θ. The maps φ is such that $t = \varphi s$ and the orbits of θ give back the vertices of the graph.

For simplicity we are considering a digraph with source and target maps s and t as being encoded in Matlab in the form of two column vectors with the same size and taking values in the same range of values. Is instead, source and target ate not indexed column vectors a simple standard procedure can be applied as illustrated below.

```
function [phi,theta,in,out] = mylinkdigraph(s,t)
% Given two column vectors of the same size, s and t, representing an indexed
%directed graph, returns endomaps phi and theta such that t=s(phi) and s=s(theta);
%'in' is the number of incoming edges at each vertex while 'out' is the number of
%outgoing edges at each vertex

id=(1:numel(s))';
if ~isequal(numel(s),numel(t)), error('s and t must be of the same size'), end
[s,is]=sort(s(:)); is_inv(is)=id;
[t,it]=sort(t(:)); it_inv(it)=id;
ds=logical([1; diff(s)]);
dt=logical([1; diff(t)]);
rs=id(ds); rt=id(dt);  % references for s and t
% in and out
in=diff([rt; numel(t)+1]);
out=diff([rs; numel(s)+1]);
% theta
m1=mod(id-rs(s)+1,out(s))+rs(s);  % disp([is is(m1)])
theta=is(m1(is_inv));
% phi
m2=mod(id-rt(s),out(t))+rs(t); % disp([it is(m2)])
phi=is(m2(it_inv))
end
```

The idea behind the algorithm is simple in the sense that we can always make a cyclic ordering, returning to the beginning every time there are more incoming than outcoming edges for a given vertex.

In the following sections we briefly recall the motivation underlying the need to introduce the abstract notion of a link structure as well as some of its applications. Before that we illustrate the more general situation of a digraph encoded in Matlab where the vertices are not indexed but are rather elements of an n-dimensional complex vector space. In that situation, instead of indexed maps s and t we may encode the information as two matrices, say `source` and `target`, with the same number of rows. Each of those rows is interpreted as a point on the n-dimensional complex space. A simple procedure is performed to obtain indexed maps s and t, such that $u(s) =$`source` and $u(t) =$`target`, as outlined below.

```
function [s,t,u]=digraph(source,target)
%Given two matrices of the same size,
%source and target, returns vectors s and
%t and a matrix u with the unique rows in
%both source and target such that
%u(s)=source and u(t)=target

f=source; g=target;
k=size(f,1);
[u,~,q]=unique([f;g],'rows');
s=q(1:k); t=q(k+1:k+size(g,1));
```

4 Motivation for the notions of link and multi-link

The purpose of this section is to briefly recall the motivation for introducing the notions of link and multi-link while providing a concrete example of application on the iso-slice algorithm. The interest reader can find further details in [4, 5].

4.1 A link as an abstraction of a planar curve

A classical planar curve is typically defined as a continuous mapping from the unit interval $[0, 1]$ into the set of complex numbers. While this definition works well from an analytical perspective and extends naturally to curves in 3D-space by replacing the complex numbers with euclidean three space, it is not ideal for computation. Various alternatives have been proposed to find a better definition, each with both advantages and disadvantages. In this context, the concept of link as introduced in [4] (see also [5]) with the purpose of being an abstraction for a planar curve in the complex plane. Indeed, by approximating a curve with a piecewise-linear sequence of directed edges, we can view it as a directed graph. A directed graph consists of a set of vertices, a set of edges, and two maps that assign a vertex to an edge, each representing its source or its target. It turns out that certain directed graphs, obtained from approximating a curve, possess a linking map. This linking map associates each edge with a successor edge along the curve's direction. Thus, we arrive at the abstract notion of a link as a computational model for a classical curve.

We define a link as a mathematical object which consists of a set equipped with an endomap and a map into a geometrical algebra of our choice. For example, in many cases the euclidean space $\mathbb{R}^n$ is enough. In those cases a link can be illustrated as a diagram of the form

$$\begin{tikzcd} A \arrow[loop left, "\varphi"] \arrow[r, "g"] & \mathbb{R}^n \end{tikzcd}$$

and it is thus interpreted as a generalized curve in $\mathbb{R}^n$ as follows. We consider that a curve is approximated by piecewise linear function, defined by a sequence of line segments, each of which determined by an indexing element in the set of indexes A. This is interpreted as the vector in $\mathbb{R}^n$ with the endpoints:

$$g(a) \longrightarrow g\varphi(a).$$

This means that each segment in the curve is indexed by an element a in A which can be used as a label for the corresponding edge in the directed graph

$$g(a) \xrightarrow{a} g\varphi(a).$$

The set A is called the set of indexes, the map φ is called the successor, or transition map, and it can be illustrated as follows

$$g(a) \xrightarrow{a} g\varphi(a) \xrightarrow{\varphi(a)} g\varphi^2(a) \xrightarrow{\varphi^2(a)} \cdots$$

For further examples the reader is referred to [1].

4.2 Linear representation of a matrix

By generalizing the concept of link as an abstraction for a curve, in order to find a suitable model for a surface, we could be led to consider a new abstract entity that would take the form of the following structure

$$X \times Y \xrightarrow{\varphi} X \times Y \xrightarrow{g} \mathbb{R}^n \ .$$

However, as soon as we try to interpret it as a surface we realize that $\varphi(x, y) = (\varphi_1(x, y), \varphi_2(x, y))$ should be of the form

$$\begin{tikzcd} (\varphi_1(x), y) \arrow[r] & (\varphi_1(x), \varphi_2(y)) \\ (x, y) \arrow[u] \arrow[r] & (x, \varphi_2(y)). \arrow[u] \end{tikzcd}$$

In other words, it should consist on two independent maps $\varphi_1 \colon X \to X$ and $\varphi_2 \colon Y \to Y$, together with the realization (or geometrical) map

$$X \times Y \xrightarrow{g} \mathbb{R}^n \ .$$

The role of φ_1 and φ_2 is to determine the behaviour of the transitions along the x-direction and the y-direction. Note that these directions are only abstract and they should not be confused with the directions of $\mathbb{R}^n$.

Let us see a concrete example. Suppose we are interested in modelling the cylinder

$$C = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 = 1, \quad 0 \leq z \leq 1\},$$

so that we could make an approximation, say, $X = \{1, 2, 3, \ldots, 360\}$, $Y = \{0, 1\}$ and define $\varphi_1 \colon X \to X$, $\varphi_2 \colon Y \to Y$ and $g \colon X \times Y \to \mathbb{R}^3$ such that: $\varphi_1(x) = x+1$ if $x < 360$ and $\varphi_1(360) = 1$; $\varphi_2(0) = \varphi_2(1) = 1$; and

$$g(u, v) = \left(\cos\left(\frac{2\pi u}{360}\right), \sin\left(\frac{2\pi u}{360}\right), v\right).$$

As remarked before, the crucial point here is to observe that we may exchange the set $X \times Y$ with another set, which is bijective to it, say A, and the endomap $\varphi \colon X \times Y \to X \times Y$ with two endomaps $\alpha, \beta \colon A \to A$ that are permutable, i.e., $\alpha\beta = \beta\alpha$. In this way we form squares indexed by the elements of A as illustrated

$$\begin{tikzcd} \beta(a) \arrow[r] & \alpha\beta(a) \\ a \arrow[u] \arrow[r] & \alpha(a). \arrow[u] \end{tikzcd}$$

If $|X| = n_X$ and $|Y| = n_Y$, then we can consider the set A as the set $\{1, 2, 3, \ldots, n_X n_Y\}$ and the well/known bijection which transforms pairs of indexes (i, j) into linear indexes $a(i, j) = i \cdot n_X + j$.

4.3 The multi-link structure

The concept of multi-link is a natural generalization of the concept of link and it is motivated by concrete examples as: square-links (like the one above), double-links (a structure that models arbitrary surfaces), and others designed for more specific purposes, such as contour filling algorithms or generating voxelized porous in 3-dimensional physical objects [2]. The concept of multi-link thus arises as the need to encode and organize a large amount of information in the form of data and algorithms that are relevant to the 3D-printing process.

A multi-link is a mathematical object displayed as

$$\begin{tikzcd} A \arrow[loop left, "\alpha_i"] \arrow[r, "g"] \arrow[d, "p_j"] & E \\ B_j & \end{tikzcd}$$

where A is an arbitrary object, E is some structured object such as a geometrical algebra and $i \in I$, $j \in J$ are arbitrary indexing sets. The maps p_j are required to be surjections. Furthermore, some conditions may be required on the structure such as commutativity between the arrows involved. For example, if $I = J$ then it is reasonable to assume that $p_j = p_j \alpha_j$ and think of the projections p_j as being the connected components for the orbits of the respective α_j.

The family of maps (α_i) is considered to be the topological part of the multi-link, the map g is the geometrical part, while the family of projections (p_j) is considered to be the logical or functional part of the structure. This is because in most of the examples the projection maps simply assign a functional behaviour to the edges, like color properties or materials or other kind of physical interpretation.

4.4 Particular cases as examples of multi-links

Let us recall from [4] the particular cases of multi-links which will be used in the applications of our chaining procedure, namely the iso-slice algorithm.

4.4.1 Coloured link

A coloured link is a link with a surjective map into a set of colors, C. In other words, it has the form

$$\begin{tikzcd} A \arrow[loop left, "\alpha"] \arrow[r, "g"] \arrow[d, "c"] & \mathbb{R}^n \\ C & \end{tikzcd}$$

where $c\alpha = c$. This is interpreted as a link in which every edge is associated withe certain color, and moreover, the edges in the same component (in the sense of orbits of α) have the same color. However, different components may have different colors.

4.4.2 Double-link

The concept of double-link is used to encode a surface, in its most general form. Here we will only give the definition and the simple example of the tetrahedron. Further examples and the study of its main properties are the matter of future work. All platonic solids can be found in [2] as double-links.

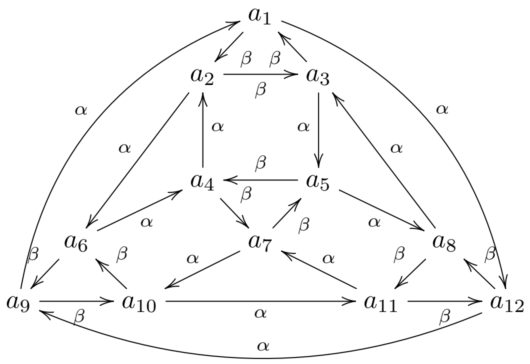
A double-link is an instance of a multi-link which can be displayed as

$$\begin{tikzcd} & A \arrow[loop left, "{\alpha,\beta}"] \arrow[r, "g"] \arrow[dl, "f"'] \arrow[dr, "v"] & \mathbb{R}^4 \\ F & & V \end{tikzcd}$$

such that

$$\begin{aligned} \alpha\beta\alpha\beta &= 1_A \\ \beta\alpha\beta\alpha &= 1_A \\ f\alpha &= f \\ v\beta &= v. \end{aligned}$$

Due to its simplicity, the example of a tetrahedron can be used to illustrate the concept. The following picture, which shows the planar graph representation of a tetrahedron (with the usual directed edges replaced by oriented squares that are build up from the labels α and β), is used to deduce a concrete example of a double-link, as follows.



Take A to be the set $\{a_1, \dots, a_{12}\}$ and let the endomaps α and β be defined by the labels indicated in the planar graph representation, that is $\alpha(a_1) = a_{12}$, $\beta(a_1) = a_2$, etc. The projection map f is the quotient over the orbits of α, thus defining the faces of the tetrahedron. The projection map v is the quotient over the orbits of β, thus defining the set of vertices. The map g is any realization map from A to any space.

Note that a triangulation is a particular instance of a double-link. Indeed, we observe $\theta\varphi\theta\varphi = \theta\theta^2 = \theta^3 = 1_A$ and $\varphi\theta\varphi\theta = \theta^2\theta = \theta^3 = 1_A$.

4.4.3 Square patch

A square patch is an intermediate level between a square-link and a double-link. It is the analogue to a triangulation, but made out of squares rather than triangles. The generalization of a triangulation is not difficult to obtain and we omit the details. Like an instance of a multi-link, a square patch is described as follows.

$$\begin{tikzcd} A \arrow[loop left, "{\theta,\varphi}"] \arrow[r, "g"] & \mathbb{H} \end{tikzcd} \tag{4.1}$$

such that

$$\begin{aligned} \theta^4 &= 1_A && (4.2)\\ \theta^3 &= \varphi\theta\varphi && (4.3)\\ g\varphi &= g. && (4.4) \end{aligned}$$

The orbits of θ are interpreted as (square) faces while the orbits of φ are interpreted as vertices.

5 The iso-slice algorithm

In this section we recall the necessary details for an implementation of an algorithm that efficiently computes level iso-contours. The contours are the ones obtained by slicing a triangulated surface in the euclidean 3D-space with respect to an iso-surface of a given level.

The algorithm may be decomposed in the following steps:

1. Consider a triangulation as input

$$\begin{tikzcd} T \arrow[r, shift left=2, "a"] \arrow[r, "b" description] \arrow[r, shift right=2, "c"'] & V \arrow[r, shift left=2, "x"] \arrow[r, "y" description] \arrow[r, shift right=2, "z"'] & \mathbb{R} \end{tikzcd}$$

Recall that an element $t \in T$ is interpreted as a triangle with vertices $[a(t), b(t), c(t)]$ and each element $v \in V$ is interpreted as a vertex in the 3D-space with coordinates $(x(v), y(v), z(v))$.

2. Transform the triangulation into a double-link as explained above (see also [3]). This gives a structure of the form

$$\begin{tikzcd} & A \arrow[loop left, "{\alpha,\beta}"] \arrow[r, "g"] \arrow[dl, "f"'] \arrow[dr, "v"] & \mathbb{R}^3 \\ F & & V \end{tikzcd}$$

with the meaning that the orbits of α are the faces (elements in F) and the orbits of β are the vertices (elements in V).

3. Suppose there is given a family of iso-surfaces in 3D-space, let us say defined by a map

$$F\colon \mathbb{R}^3 \to \mathbb{R}$$

which intuitively may be thought of as assigning a certain height to every point in space. The main example is $F(x, y, z) = z$, giving planar slices along the z-direction, but $F(x, y, z) = x^2 + y^2$ or $F(x, y, z) = x^2 + y^2 + z^2$ are also possible and correspond to cylinders and spheres, respectively. In general, we have arbitrary maps perhaps with a specific meaning. Nevertheless, the algorithm works similarly for every map.

In practice, what is needed is a scalar field defined on the vertices, which amounts to having a map $h\colon A \to \mathbb{R}$ with no further requirements.

4. Transform the structure of double link of step 2 in the structure of a coloured link by considering $\varphi = \beta\alpha$, $h = Fg$ and forgetting the projection map v. This produces

$$\begin{tikzcd} A \arrow[loop left, "\varphi=\beta\alpha"] \arrow[r, "h=Fg"] \arrow[d, "f"'] & \mathbb{R} \\ F \end{tikzcd}$$

and the map h is intuitively the height of each point in A relative to the iso-level-surface F of step 3.

5. For each contour level $r \in \mathbb{R}$ do:

   (a) Obtain the subset of A in the coloured link of step 4 defined as:

$$A_r = \{a \in A \mid h(a) \leq r < h\varphi(a)\}$$

   (b) Consider the directed graph whose edges are the elements in A_r as well as the vertices; the domain map is the identity map and the codomain map is $\varphi = \beta\alpha$, this will produce a picture which may be interpreted as

$$\begin{tikzcd} \cdots & {} \arrow[l, "\alpha"'] \arrow[r, "\beta"] & {} \arrow[dd, "\alpha"] & \cdots \arrow[l, "\alpha"'] & {} \arrow[l, "\alpha"'] \\ f(a) & & & f(a') & \\ \cdots \arrow[r, "\alpha"] & a \arrow[uu, "\alpha"] & {} \arrow[l, "\beta"'] \arrow[r, "\alpha"] & \cdots \arrow[r, "\alpha"] & a' \arrow[uu, "\alpha"] \end{tikzcd}$$

   in which we suppose a and a' to be in A_r. This means that $r \in \mathbb{R}$ is a height laying between $h(a)$ and $h\beta\alpha(a)$ as well as between $h(a')$ and $h\beta\alpha(a')$. The idea is to connect a and a' and in order to do so it suffices to identify the orbits of α via f. This procedure creates a directed graph.

   (c) Construct the directed graph

$$\begin{tikzcd} A_r \arrow[r, shift left, "d"] \arrow[r, shift right, "c"'] & F \end{tikzcd}$$

   from the subset $L\colon A_r \hookrightarrow A$ (step 5(a)) to the set of faces F (step 4), with $d = fL$ and $c = f\varphi L = f\beta\alpha L$. This graph is obtained by applying the quotient map f to the graph considered in step 5(b).

(d) Link the digraph of step 5(c), that is, find $\varphi_r \colon A_r \to A_r$ such that $d\varphi_r = c$. This is a general process and it can be performed in a unique way, provided that the faces are geometrically convex. Indeed, let $\begin{tikzcd} E \arrow[r, shift left, "d"] \arrow[r, shift right, "c"'] & V \end{tikzcd}$ be an arbitrary directed graph, it has a symmetry, that is, there exists a bijective map $\varphi \colon E \to E$ such that $d\varphi = c$ if and only if the incoming edges are in bijection with the out-coming ones for every vertex in V. In our case, if the faces are convex then they will either not be intersected by the iso-surface level r or they are intersected exactly in two different edges (in the picture displayed at step 5(b) this was assumed to happen at the edges starting at the indexes a and a').

(e) Construct the link structure

$$\begin{tikzcd} A_r \arrow[loop left, "\varphi_r"] \arrow[r, "g_r"] & \mathbb{R}^3 \end{tikzcd} ,$$

with $g_r(a) = g(a) + t_r(g\varphi(a) - g(a))$ where

$$t_r = \frac{r - h(a)}{h\varphi(a) - h(a)},$$

recall that $\varphi = \beta\alpha$ and $h = Fg$, come from step 4.

6. Collect all the link structures (A_r, φ_r, g_r) for all the contours $r \in \mathbb{R}$ in which we may be interested in and return this information as output.

6 An example of application

An example of application for the iso-slicing algorithm related to the generation of refrigeration channels was provided in [4]. Here we describe another general application which was in fact used in the main application but can be extracted independently. It has to do with converting a triangulation into a square patch. This is convenient because in some cases a square patch has better properties than a triangulation. The price to pay in this case is that the resulting surface may not be equivalent to the original one. This is expected as a square path can be seen as modeling a paralelizable surface whereas a triangulation can model an arbitrary one.

Suppose it is given a solid body object modeled as a subset of 3D-space, say $S \subseteq \mathbb{R}^3$.

Suppose (T, V, a, b, c, x, y, z) is a triangulation such as the one given on step 1 of Section 5, which is considered to be an approximation to the surface defined by the boundary of $S = closure(interior(S)) \subseteq \mathbb{R}^3$. We are interested in converting the triangulation into a square patch.

1. To obtain a square-link from the given triangulation:

   (a) find an appropriate set of contours equally spaced that can serve as a good approximation to the given triangulation;

   (b) for each one of the contour levels identified on the previous item, execute the slicing algorithm with $F(x, y, z) = z$;

   (c) re-sample the number of points obtained in each set of indexes from the final link (as in step 5(b) from the iso-slice algorithm) so that they all have the same number of points;

   (d) construct a square-link by letting A be the union of all A_r, assuming that we have chosen say, $r \in \{r_0, r_1, \ldots, r_n\} \subseteq \mathbb{R}$ and that each A_r has, say, $m = 100$ elements. This is done by letting the map α be given by the collection of φ_r and β to be the identify on each point in the level r_i with the closest one on the level r_{i+1}. Note that this does not necessarily give a structure for a surface which is homeomorphic to the initial one.

2. having a square link (A, α, β, g) as defined in Section 4.1.2 we now define the iso-surface family $F \colon A \to \mathbb{R}$ iteratively as follows (consider the bijection $\phi \colon A \to X \times Y$ with the sets $X = \{0, 1, \ldots, n\}$ and $Y = \{1, \ldots, m\}$). The base points, that is, the ones in the level r_0, are all zero $F(x, 0) = 0$; all the points at the same level will have the same value under F; suppose we have $F(x, y)$ given, then we define $F(x, y + 1)$ as the formula

$$F(x, y) + \|g\phi^{-1}(x, y) - g\phi^{-1}(x, y + 1)\|$$

3. use the iso-slicing algorithm with the new height value F;

4. the final result of this procedure is a family of contour levels parallel to the xy-direction which are isotropic along the geodesic paths measured on the surface.

We thus get contour trajectories enveloping the original surface in a way which is isotropic. A concrete example would be the sphere. If starting with a triangulation for the sphere then at the end we obtain a new representation of the sphere as a collection of parallel strips with two degenerate ones at the poles.

Acknowledgments

This work of NMF was partially supported by the Fundação para a Ciência e a Tecnologia (FCT) and Centro2020 through the following Projects: UIDB/04044/2020, UIDP/04044/2020 and PAMI - ROTEIRO/0328/2013 (Nº 022158). The work of MP was partially financially supported by the Portuguese Foundation for Science and Technology (FCT) under project grants UID/-MAT/00324/2020 and UID/MULTI/00308/2020.

Referências

[1] M. Gaspar and N. Martins-Ferreira, *Finite Orbit decomposition of endomaps*, Scripta-Ingenia **3** (December) (2014) 23–26.

[2] M. Gaspar, M. Leite, N. Martins-Ferreira and B. N. Panda, *Computer construction of Platonic Solids*, Scripta-Ingenia **4** (June) (2015) 6–11.

[3] N. Martins-Ferreira, *On the structure of a triangulation*, Scripta-Ingenia **5** (December) (2015) 22–23.

[4] N. Martins-Ferreira, *The notion of multi-link, its applications and examples*, Scripta-Ingenia **7** (December) (2016) 14–21.

[5] N. Martins-Ferreira, *Internal Categorical Structures and Their Applications* , Mathematics **11** (3) (2023) 660.
https://www.mdpi.com/2227-7390/11/3/660

Exploring Endomaps of Natural Numbers and its Finite Counterpart as a first step into a category of structured stes

by Nelson Martins-Ferreira

Centre for Rapid and Sustainable Product Development
School of Technology and Management
Polytechnic Institute of Leiria
Campus 2, Morro do Lena - Alto do Vieiro
2411-901 Leiria - Portugal

Abstract In this short paper, we provide an overview of the properties of the category with objects the endomaps of natural numbers and its finite counterpart. We also present a procedure for computing the orbits of these endomaps and discuss their connection with link structures. This computational approach to structured sets is a preliminary step towards defining a category that can serve as a mathematical foundation for computer science. It combines the fundamental notion of a set with a specific structure for storing set elements as a list in a computer.

Keywords: Structured set, endomap, category theory, category of endomaps, product, coproduct, type, monoid, link, monoid-link, multi-link, orbits.

1 Introduction

One way to introduce the semiring of natural numbers is to start with a set equipped with two operations, denoted by addition $(+)$ and multiplication $(\cdot)$, and two constants, 0 and 1. Using these operations and constants, we can construct the set of natural numbers denoted by $\mathbf{N}_0 = \{0, 1, 2, 3, \ldots, n, n+1, \ldots\}$. Each natural number can be obtained by starting with 0 and repeatedly adding the constant 1. So, the set of natural numbers with zero can be obtained as the free monoid on a one-element set. This already gives the additive structure. The multiplicative structure can be obtained by observing that there is a bijection between the set of prime numbers and the set of natural numbers, allowing us to identify the free commutative monoid generated by the prime numbers with the natural numbers and thus giving rise to the multiplicative structure.

Another method to obtain the set of natural numbers is through the concept of a natural number object in a category with a terminal object. In any category with a terminal object, denoted by 1, a natural number object (when it exists) is a triple $(N, 0, s)$, with N and object in the category, $0\colon 1 \to N$ a morphism from the terminal object into the object N and $s\colon N \to N$ is an endomap of N satisfying the following property. For any triple (X, a, r) with X and object in the category and $a\colon 1 \to X$ and $r\colon X \to X$ morphisms, there exists a unique morphism $x\colon N \to X$ such that $x0 = a$ and $xs = rs$. This construction can be applied in various contexts beyond the category of sets and maps. Further details can be found e.g. in:

Tom Leinster, *Rethinking Set Theory* The American Mathematical Monthly Vol. 121, No. 5 (May 2014), pp. 403-415 (13 pages)

The discrepancy between the concept of a set and its representation in a computer system poses difficulties in formalizing mathematical foundations for computer science. In mathematics, a set like $A = \{1, 2, 3, 4, 5\}$ is considered to have no other structure than the number of its elements, in this case 5, as well as the ability to recognize each one of its elements. However, in computer science, an additional step is required to represent the set A as an object. We have to specify an order in which the elements of the set A are recorded in an array, such as for example `a=[1 2 3 4 5]`. One of the purposes of these notes is to provide a first step towards combining the power of mathematical formalism with the power of scientific computing on a common ground.

To accomplish this, we will define a structured set as a collection of elements with a structure map that determines how the elements are stored in computer memory. The structure will be encoded using an endomap, while the different types of data will be encoded using an ordered monoid. Lastly, a map from a set of natural numbers to the monoid will specify the way in which the elements are to be listed in the set.

2 Structured sets

The category of structured sets is a first attempt to give a categorical foundation to computer science. Its objects are triples (d, φ, n) where n is a natural number, $\varphi\colon \{1, \ldots, n\} \to \{1, \ldots, n\}$ is an endomap and

$d\colon \{1,\ldots,n\} \to B$ is a map from the set with n indexes into an ordered monoid $(B, 0, +, \leq)$. The less or equal relation is a preorder (i.e., a reflexive and transitive binary relation) on B with the additional condition that if $x \leq y$ then $x + z \leq y + z$ for every $x, y, z \in B$. Intuitively we think of the relation $\leq$ as the relation is-a-prefix-of on the free monoid of words for a given alphabet. In an object (d, φ, n) we are of course omitting the monoid part B which is implicitly encoded as the codomain of the map d.

Morphisms of structured sets A morphism between two structured sets, say from (d, φ, n) to (d', φ', n'), is a triple (f, τ, α) where $f\colon B \to B'$ is a monotone monoid homomorphism, $\alpha\colon \{1,\ldots,n\} \to \{1,\ldots,n'\}$ is a map and $\tau\colon \{1,\ldots,n\} \to B'$ is a map satisfying the following conditions: $\varphi'\alpha = \alpha\varphi$ and $fd + \tau = d'\alpha$. If denoting by A the set $\{1,\ldots,n\}$ and similarly A' and A'' with respect to n' and n'' we observe that the law for composing morphisms, such as the ones displayed in the following diagram

$$\begin{tikzcd}
B \arrow[d, "f"'] & A \arrow[l, "d"'] \arrow[loop right, "\varphi"] \arrow[d, "\alpha"] \arrow[dl, "\tau"] \\
B \arrow[d, "f'"'] & A' \arrow[l, "d'"] \arrow[loop right, "\varphi'"] \arrow[d, "\alpha'"] \arrow[dl, "\tau'"] \\
B & A'' \arrow[l, "d''"] \arrow[loop right, "\varphi''"]
\end{tikzcd}$$

is obtained as

$$(f', \tau', \alpha') \circ (f, \tau, \alpha') = (f'f, f'\tau + \tau'\alpha, \alpha'\alpha)$$

The morphism τ is witnessing the requirement that $f(d(x)) \leq d'(\alpha(x))$ for every $x \in A$ and the same is true for τ' with respect to $f'(d'(x')) \leq d''(\alpha'(x'))$ for every $x' \in A'$. This category as well as its properties will be developed in follow-up studies. For the moment let us observe that we can define an equivalence of structure maps as $\varphi \simeq \varphi'$ if and only if there exists a bijective map α such that $\varphi' = \alpha\varphi\alpha^{-1}$. A picture illustrating the equivalence classes for a set with five elements is presented at the end of the paper. Furthermore, in order to recover the notion of a set from a structured set we use the standard procedure of computer science called unique factorization. It gives a (split-epi, mono) factorization for each triple (d, φ, n) as illustrated.

$$\begin{tikzcd}
B & A \arrow[l, "d"'] \arrow[loop right, "\varphi"] \arrow[d, shift left, "q"] \\
 & I \arrow[ul, "u"] \arrow[u, shift left, "r"] \arrow[loop right, "q\varphi r"]
\end{tikzcd}$$

Once again, $A = \{1,\ldots,n\}$, while u is a (uniquely determined) monomorphism such that $uq = d$, $I = \{1,\ldots,k\}$ is a set of indexes with k the number of unique elements in the image of d, and q and r are such that $d = uq$, $qr = 1_I$. As a consequence we also have $u = dr$. In order to work with two triples (d, φ, n) and (d', φ', n') as just sets which are the same as soon as their images by d and d' are the same, we define the following equivalence on the objects of the category of structured sets:

$$(d, \varphi, n) \simeq (d', \varphi', n')$$

if and only if $u = u'$ with u obtained as above and similarly u' being such that $u'q' = d'$. Under this equivalence we are forgetting the structure endomaps φ and φ' as well as the multiplicity of the elements in the inverse images of the maps d and d'. In other words, what remains after identifying structured sets via their unique factorization is the set of elements determined by the image of the map d.

3 Endomaps of natural numbers

The category of endomaps of natural numbers has objects which are endomaps $\varphi\colon \mathbb{N} \to \mathbb{N}$ and has morphisms $f\colon \varphi \to \varphi'$ which are also endomaps $f\colon \mathbb{N} \to \mathbb{N}$ that satisfy the condition $\varphi' f = f\varphi$. This category is worth studying from the perspective of structured sets, as mentioned earlier. We note that each object φ can also be seen as a morphism $\varphi\colon 1_{\mathbb{N}} \to 1_{\mathbb{N}}$.

The three endomaps, defined for every $x \in \mathbb{N}$ and taking values in $\mathbb{N}$

$$\begin{aligned}
\alpha(x) =& 2x && (3.1)\\
\beta(x) =& 2x - 1 && (3.2)\\
\gamma(x) =& \frac{x}{2}, \quad \text{if } x \text{ is even} && (3.3)\\
\gamma(x) =& \frac{x+1}{2}, \quad \text{if } x \text{ is odd} && (3.4)
\end{aligned}$$

as well as the bijective function $\delta\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined as

$$\delta(x, y) = 2^{x-1}(2y - 1)$$

are useful to describe products and coproducts. Indeed, in the diagram

$$\begin{tikzcd}
\mathbb{N} \arrow[r, "\alpha"] \arrow[dr, "u"'] & \mathbb{N} \arrow[d, "{[u,v]}"] & \mathbb{N} \arrow[l, "\beta"'] \arrow[dl, "v"] \\
 & \mathbb{N} &
\end{tikzcd}$$

we observe that for every u and v there exists a unique $[u, v]$ defined as

$$[u, v](x) = \begin{cases} u(x/2), & \text{if } x \text{ is even} \\ v((x+1)/2), & \text{if } x \text{ is odd} \end{cases} \qquad (3.5)$$

thus exhibiting the universal property of a coproduct.

Similarly, the diagram

$$\begin{tikzcd}
& \mathbb{N} \arrow[dl, "u"'] \arrow[d, "\langle u,v\rangle"] \arrow[dr, "v"] & \\
\mathbb{N} & \mathbb{N}\times\mathbb{N} \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\delta"] & \mathbb{N} \\
& \mathbb{N} \arrow[ul, "\pi_1\delta^{-1}"] \arrow[ur, "\pi_2\delta^{-1}"'] &
\end{tikzcd}$$

exhibits the universal property of the product since for every u and v there exists a unique $\langle u,v\rangle$ defined as $\langle u,v\rangle(x) = \delta(u(x), v(x))$.

In order to work in the finite case, we will consider pairs (φ, n) where n is a natural number and φ and endomap of the natural numbers with the property that if $x \leq n$ then $\varphi(x) \leq n$ and if $x > n$ then $\varphi(x) > n$. In this way we are able to work with the finite part of the endomap, namely its restriction to the set $A = \{1, \ldots, n\}$. In order to have a structured set as introduced before we need only to take a map $d\colon A \to B$ into any type of data that is modeled by a monoid with a preorder relation. For simplicity we will consider the complex numbers with the lexicographic order, as it is customary in computer science.

4 Some constructions on structured sets

Given two triples (d, φ, n) and (d', φ', n') as before we may construct its product and coproduct as follows, using Matlab language. The product is defined as

$$(d_1, \varphi_1, n_1) \times (d_2, \varphi_2, n_2) = (d_{12}, \varphi_{12}, n_1 n_2)$$

where

$$\varphi_{12}(x) = \texttt{sub2ind([n1,n2],phi1(x1),phi2(x2))}$$

with

$$\texttt{[x1,x2]=ind2sub([n1,n2],x)}$$

while

$$d_{12}(x) = (d_{12}^1(x), d_{12}^2(x))$$

where $d_{12}^1(x) \in B_1$ wheras $d_{12}^2(x) \in B_2$ respect their own types and moreover `d121=d1(x(:))` and `d122=d2(y(:))` with `[x,y]=ndgrid(1:n1,1:n2)`.

The coproduct construction is simpler and can be obtained as

$$(d_1, \varphi_1, n_1) + (d_2, \varphi_2, n_2) = ([d_1; d_2], [\varphi_1; \varphi_2 + n_1], n_1 + n_2)$$

Finally, we end this note with a procedure to compute the orbits of a finite endomap.

```
function [orb, ord, psi, deg, init, term, prin, conn] = orbits(phi)
% [ orb, ord, psi, deg, init, term, prin, conn] = orbits(phi)
%%
%%
% % given a vector phi, interpreted as an endomap from the set of indexes 1:numel(phi)
% into itself, returns vectors of the same size as phi with the orbits (orb),
% the order inside each orbit (ord), a pseudo-inverse to phi (psi), and the degree
% of each point (deg); it also returns a vector with the initial points
% (init), the terminal points associated with each initial point, that is,
% the last point in oder of its principal component, and the level of the
% component (prin), 1 means principal, 2 is for secondary, 3 terciary etc;
% (conn) encodes the connected components
%
% (a) the orbit is a label which identifies the points a, phi(a), phi^2(a),...,phi^k(a),
% until a point phi^(k+1)(a) is found which is already a member of the current
% sequence or has already been considered in a previous sequence; it is initiated
% with the numbers 1, 2, 3,... up to the total number n0 of initial
% points (the ones with degree 0); the sequence continues with the numbers
% n0+1, n0+2, n0+3,...
% for the cyclic components which are characterized by the fact that each point
% in it is a regular point (degree -1)
%
% (b) the  order is the number k described in the previous item
%
% (c) pseudo-inverse to phi in the sense that it is consistent with the choice
% of principal components
%
% (d) degree in the sense that deg(a)=p means that there are p elements x such
% that phi(x)=a; the points in a cyclic component are labeled −1
%
% Example
% id=@(x) [1:numel(x)]';
% phi=[4 5 5 7 10 8 1 10 10 11 9 8 13 13];
% [ orb, ord, psi, deg, init, term, prin, conn] = orbits(phi);
% disp([id(phi) phi(:) orb ord psi deg ])
%      1     4     6     1     7    −1
%      2     5     1     1     2     0
%      3     5     2     1     3     0
%      4     7     6     2     1    −1
%      5    10     1     2     2     2
%      6     8     3     1     6     0
%      7     1     6     3     4    −1
%      8    10     3     2     6     2
%      9    10     1     5    11     1
%     10    11     1     3     5     3
%     11     9     1     4    10     1
%     12     8     4     1    12     0
%     13    13     5     2    14     2
%     14    13     5     1    14     0
%
% disp([init; term; prin ; conn ])
%      2     3     6    12    14
%      9     3     8    12    13
%      1     2     2     3     1
%      1     1     1     1     5
%
% sortrows([orb ord id(orb)])
%      1     1     2
%      1     2     5
%      1     3    10
%      1     4    11
%      1     5     9
%      2     1     3
%      3     1     6
%      3     2     8
%      4     1    12
%      5     1    14
%      5     2    13
%      6     1     1
%      6     2     4
%      6     3     7

% given a vector phi
try
    phi=phi(:); % force phi to be a column vector
    phi(phi); % make sure that phi can be interpreted as an endomap
catch ME
    disp('The input vector phi must be interpreted as an endomap from the
    set of indexes 1:numel(phi) to itself:')
    error(ME.message)
end
n=numel(phi);
idn=(1:n)'; % the identity vector of the same size as phi
% initializing outputs
orb=zeros(n,1);
```

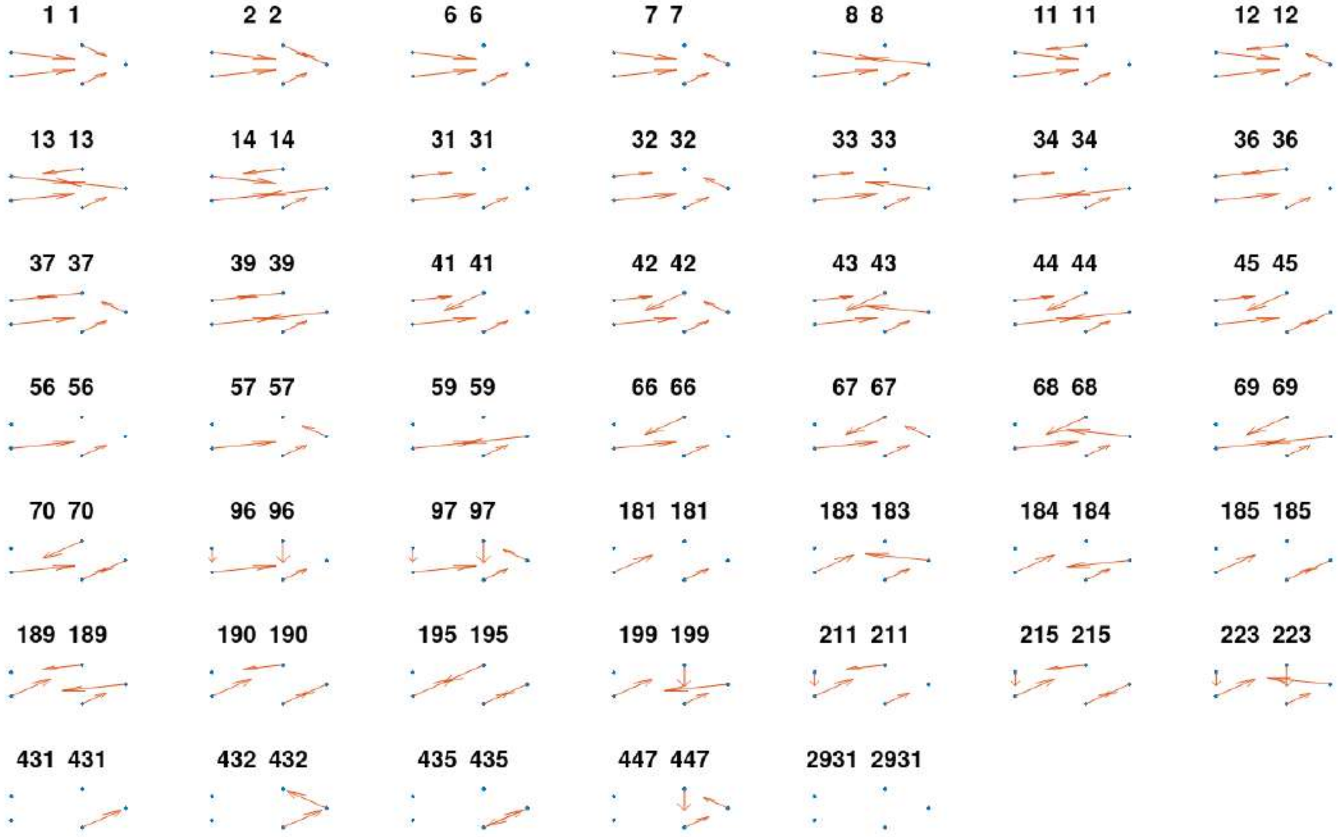
```
ord=zeros(n,1);
psi(phi)=idn; psi=psi(:); % ensure column vector
deg=repmat(-1,n,1);  % the degree is initiated by default as cyclic
% determining initial points (those with degree 0)
init=idn(~ismember(idn,phi))'; % vector with indexes of initial points
m=numel(init); % number of initial elements
term=zeros(1,m); % the vector term will contain the terminal point of each
                 % initial point with respect to the choice of principal components
prin=zeros(1,m); % the vector prin will contain the principal components,
                 % secondary, terciary, etc
conn=zeros(1,m); % the connected component, relative to the principal component

% the iterative process
for j=1:numel(init)
    a=init(j); k=1;
    orb(a)=j;
    ord(a)=k;
    psi(a)=a; % the initial points become fixed points when inverted
    deg(a)=0;
    while ~orb(phi(a))   % phi(a) is not yet a junction point
        psi(phi(a))=a; % is defined before a is uptaded as a=phi(a)
        a=phi(a); k=k+1;  % a is updated to phi(a) and k to k+1
        orb(a)=j;
        ord(a)=k;
        deg(a)=1;
    end
    % phi(a) is a junction point (orb(phi(a)) is not zero)
    if orb(a)==orb(phi(a))  %  principal component
        deg(phi(a))=deg(phi(a))+1;
        term(j)=a;
        prin(j)=1;
        conn(j)=j;  % the same as orb(a)
    else % secondary component
        deg(phi(a))=deg(phi(a))+1;
        term(j)=a; % each secondary component will use the terminal
                     % point of its principal component
        prin(j)=prin(orb(phi(a)))+1;
        conn(j)=conn(orb(phi(a)));
    end
end

% It remains to compute the cyclic orbits
%% e.g.:
% phi=[2 3 4 5 1 1 6 6 1 3 12 13 14 15 12 13 16 17]';
% phi=[phi' 20 21 22 19 23 25 24]; disp([idphi phi orb ord psi])
q=m;  % m=numel(init), initialize component label, in oder to continue
      % the non-cyclic components
while any(~orb)
    a=find(~orb,1);
    q=q+1;  % update component lablel
    orb(a)=q;  % set component label
    ord(a)=1;  % initialize counting the order of current component
    %deg(a)=-1;  % we have choosen to single out the cyclic points
    %              by pre-setting deg=-1
    while ~orb(phi(a))
        a=phi(a);
        orb(a)=q;
```

```
            ord(a)=ord(psi(a))+1;
deg(a)=-1;      end
end
```

Each one of the 47 different orbits (up to conjugation) of endomaps with $n = 5$ are displayed in the following picture. The numbers indicate the choice of a representative in each class. One of the application of the function `orbits` is to permit redrawing each one of the individual diagrams with a cyclic ordering.



Cartesian natural fibrous preorders and left-topological groups

by Nelson Martins-Ferreira

Centre for Rapid and Sustainable Product Development
School of Technology and Management
Polytechnic Institute of Leiria
Campus 2, Morro do Lena - Alto do Vieiro
2411-901 Leiria - Portugal

Abstract This essay explores Cartesian natural fibrous preorders, with a specific emphasis on quasitopological groups. Fibrous preorders were introduced as an extension of preorders, where the relationship between a topological space and a spatial fibrous preorder is akin to how an Alexandroff space is related to a preorder. The essay investigates the connection between these concepts within the realm of topological and quasitopological groups, and proposes potential directions for further research in this area.

Keywords: Structured set, endomap, category theory, category of endomaps, product, coproduct, type, monoid, link, monoid-link, multi-link, orbits.

1 Introduction

In mathematics, various structures and concepts are studied to better understand and analyze different mathematical phenomena. Three important such concepts are groups, preorders and spaces. These fields of study offer unique insights into the nature of symmetry, order relations and topology, and their applications can be found in diverse areas such as computer science, physics, and engineering.

Topology studies the properties of space that remain unchanged under continuous transformations. Topological groups, which combine the group structure with topological properties, provide an effective framework for studying these intertwining algebraic and topological aspects. However, when both structures are integrated, redundancy arises. This essay aims to outline ideas that minimize this redundancy.

Cartesian natural fibrous preorders were first introduced in [1] as a way to analyze topological spaces using preordered sets. In [2], it was proven that there exists a categorical equivalence between topological spaces and spacial fibrous preorders, which extends the well-known equivalence between Alexandroff spaces and preorders.

This essay aims to provide an overview of these two mathematical concepts, with a focus on elucidating their definitions, properties, and relationships. By exploring various areas where these concepts are used, we strive to emphasize their practical importance and potential for further investigation.

2 Monoids, order and topology

In the opening article of [3] it is observed a connection between the three concepts: monoids, order and topology and the following result is outlined.

Let $(X,+,0)$ and $(A,+,0)$ be two monoids (not necessarily commutative) and let $B \subseteq X$ be a subset of X together with two functions $s\colon A\times B \to B$ and $t\colon B\to A$ such that:

1. $b_1 + s(t(b_1), b_2) \in B$, for all $b_1, b_2 \in B$;
2. for all $a \in A$, $s(a, b_1 + s(t(b_1), b_2)) = s(a, b_1) + s(a + t(b_1), b_2)$;
3. $s(a_1 + a_2, b) = s(a_1, s(a_2, b)) = s(a_2, s(a_1, b))$, for all $a_1, a_2 \in A$, $b \in B$.

Under these conditions a topology τ on X is defined by

$$\mathcal{O} \in \tau \Leftrightarrow \forall x \in \mathcal{O}, \exists a \in A, N(x,a) \subseteq \mathcal{O},$$

with $N(x,a) = \{x + s(a,b) \in X \mid b \in B\} \cup \{x\}$, and we observe that $N(x,a) \in \tau$ for all $x \in X$ and $a \in A$. Furthermore, in the case where $t(b) = 0$ for all $b \in B$, we get, for every $a \in A$, a preorder relation on X determined as $xR_a y \Leftrightarrow y \in N(x,a)$.

Considering for example $X = (\mathbb{Q},+,0)$, $A = (\mathbb{N}_0,+,0)$, $B = \mathbb{Z}$, $t(b) = 0$ and $s(a,b) = bp^a$, with p any prime number, we get the so-called p-adic topology. On the other hand , if considering $X = (\mathbb{R},+,0)$, $A = (\mathbb{N},\cdot,1)$, $B =]-1,1[$, $s(a,b) = \frac{b}{a}$ and $t(b) \in \mathbb{N}$ such that $\frac{1}{t(b)} \leq 1 - |b|$ then we get the usual topology on the real line.

3 Cartesian natural fibrous preorders

As introduced in [2] a Cartesian natural fibrous preorder is a specialization of a fibrous preorder in two levels. First let us recall that a fibrous preorder [1] is a sequence $R \to A \to X$ with $R \subseteq A \times X$ satisfying some conditions. When the set A is the Cartesian product of a set I and the set X then we have a Cartesian fibrous preorder. Furthermore, if the set of indices I is the set of natural numbers $\mathbb{N}$ then we have a Cartesian natural fibrous preorder. In this case, R becomes a subset of $X \times \mathbb{N} \times X$ and we use the notation $x \leq^n y$ to mean $(x, n, y) \in R$.

Definition 6. *A Cartesian natural fibrous preorder on a set X is a sequence of binary relations, denoted as $\leq^n$, with $n \in \mathbb{N}$, satisfying the following conditions:*

1. *$x \leq^n x$, for all $n \in \mathbb{N}$ and $x \in X$;*
2. *for every $x, y \in X$ and every $n \in \mathbb{N}$ such that $x \leq^n y$ there exists a natural number $k = k(x, n, y) \in \mathbb{N}$ for which the following condition is satisfyed for every $z \in X$*
$$y \leq^k z \Rightarrow x \leq^n z$$
3. *for every $x, y \in X$ and every $n, m \in \mathbb{N}$ if $x \leq^{nm} y$ then $x \leq^n y$ and $x \leq^m y$.*

Some remarks are in order. Condition 1 asserts that each relation $\leq^n$ is reflexive while condition 2 is a generalization of transitivity. Indeed, if we take $k(x, ny) = n$ then each $\leq^n$ becomes a transitive relation. Condition 2 is here presented as a property, however, in [2] is was introduced as an extra structure with the purpose of internalization. Here we are simplifying our approach and restrict ourselves to the case of the category of sets and maps.

Condition 3 was not required in [2] as part of the definition of fibrous preorder, however, it has to be included as soon as we are interested in having an equivalence with topological spaces. This condition was called a spacial condition in [2] but we will consider it here as part of a Cartesian natural fiborus preorder.

The topology which is induced from a Cartesian natural fibrous preorder $(\leq^n)_{n\in\mathbb{N}}$ is obtained as expected, namely

$$\mathcal{O} \in \tau \Leftrightarrow \forall x \in \mathcal{O}, \exists n \in \mathbb{N}, N(x, n) \subseteq \mathcal{O},$$

where $N(x, n) = \{y \in X \mid x \leq^n y\}$.

Clearly, every metric space (X, d) is an example if we define $x \leq^n y$ if and only if $d(x, y) < \frac{1}{n}$. In this case we can choose $k(x, n, y)$ to be the smallest natural number such that

$$d(x, y) + \frac{1}{k} \leq \frac{1}{n}$$

which exists under the assumption that $d(x, y) < \frac{1}{n}$. Condition 2 in Definition 6 then follows by the triangular inequality of the metric distance.

The notion of continuity appears as the familiar definition of continuity when it is phrased in terms of epsilons and deltas, here replaced by n's and k's.

Definition 7. *Let $(A, \leq^n)$ and $(Y, \leq^n)$ be two Cartesian natural fiborus preorders. A function $f \colon X \to Y$ is said to be continuous at a point $x \in X$ if*

$$\forall n \in \mathbb{N}, \exists k \in \mathbb{N}, \forall y \in X, x \leq^k y \Rightarrow f(x) \leq^n f(y)$$

This notion is convenient and intuitive since it is a generalization of f being a monotone map and it resembles the definition of continuity at a point in terms of epsilons and deltas. Although we will not pursue that point of view here, it is nevertheless worth mentioning that, when $k = k(x, n)$ is considered as part of the structure different gradings of continuity can be observed [1,2]. Uniform continuity occurs when $k(x, n)$ does not depend on x. Additionally, other restrictions can be imposed on k when it is part of the structure.

In the following section we illustrate the case when X is a group.

4 Left-topological fibrous preordered groups

When X is a group we can take advantage of the group operation and define each binary relation $\leq^x$ in terms of its positive cone, that is, the set $B_n = \{u \in X \mid 0 \leq^n u\}$. We will use the

Definition 8. *A system (X, B_n) in which $X = (X, +, 0)$ is a group (not necessarity Abelian) and $B_n \subseteq X$ is a subset of X for every $n \in \mathbb{N}$ is called a fibrous preordered group if the following conditions are satisfied:*

1. *$0 \in B_n$, for every $n \in \mathbb{N}$*
2. *for every $n \in \mathbb{N}$, if $u \in B_n$ then there exists $k = k(n, u) \in \mathbb{N}$ such that*
$$v \in B_k \Rightarrow u + v \in B_n$$
3. *$B_{nm} \subseteq B_n \cap B_m$*

We observe that condition 2, in the case when $k(n, u) = n$, says that each B_n is closed under the group operation. This condition may also be written as:

$$\forall n \in \mathbb{N}, \forall u \in B_n, \exists k \in \mathbb{N}, u + B_k \subseteq B_n. \qquad (4.1)$$

The fibrous preordered structure is defined as

$$x \leq^n y \Leftrightarrow \exists u \in B_n, x + u = y$$

When$B_n = B$ for all n then we recover the usual notion of a preordered group in which the group operation if monotone on the first variable, that is, for every $x, y, w \in X$ if $x \leq y$ then $w + x \leq w + y$.

Once again, the definition of continuous mapping resembles the standard definition based on epsilons and deltas.

Definition 9. *Let (X, B_n) and (Y, B_n) be two quasitopological fibrous preordered groups. A map $f\colon X \to Y$ is said to be continuous at a point $x \in X$ if*

$$\forall n \in \mathbb{N}, \exists k \in \mathbb{N}, u \in B_k \Rightarrow -f(x) + f(x+u) \in B_n.$$

As usual, a function is said to be continuous if it is continuous at every point of its domain. We observe that the previous definition is compatible with the one saying that $f\colon X \to Y$ is continuous as soon as

$$\forall x, \forall n, \exists k, \forall y, x \leq^k y \Rightarrow f(x) \leq^n f(y)$$

with $x \leq^n y$ meaning that $x + u = y$ for some $u \in B_n$.

Clearly, when f is a group homomorphism then the continuity condition reduces to

$$\forall n, \exists k, f(B_k) \subseteq B_n.$$

The set $X \times X$ can be equipped with a cartesian natural fibrous preorder, which is induced by the family of subsets $B_n \times B_n \subseteq X \times X$. It becomes a fibrous preordered group as soon as we take $k((x_1, n_1, u_1), (x_2, n_2, u_2)) = k(x_1, n_1, u_1)k(x_2, n_2, u_2)$ in condition 2 of Definition 9.

Let us study the continuity of some elementary functions.

Definition 10. *Let:*

1. *$l_a\colon X \to X$, with $a \in X$ a fixed parameter, be the left translation by a, $l_a(x) = a + x$;*
2. *$r_a\colon X \to X$, with $a \in X$ a fixed parameter, be the right translation by a, $r_a(x) = x + a$;*
3. *$c_a\colon X \to X$, with $a \in X$ a fixed parameter, be conjugation by a, $c_a(x) = a + x - a$;*
4. *$i\colon X \to X$, be the symmetric operation $i(x) = -x$;*
5. *$s\colon X \times X \to X$ be the group operation $s(x, x') = x + x'$.*

The following results are immediate from the definition of continuity.

Proposition 12. *Let (X, B_n) be a quasitopological fibrous preordered group.*

1. *The function l_a is continuous for every $a \in X$. Hence (X, B_n) is a left-topological group.*
2. *The function r_a is continuous if and only if*
$$\forall n, \exists k, -a + B_k + a \subseteq B_n$$
3. *The function c_a is continuous if and only if*
$$\forall n, \exists k, a + B_k - a \subseteq B_n$$
4. *The function r_a is continuous if and only if the function c_{-a} is continuous.*
5. *The function i is continuous if and only if*
$$\forall x, \forall n, \exists k, x - B_k - x \subseteq B_n$$
6. *If r_a is continuous for every $a \in X$ then the following conditions are equivalent:*
   - *(a) i is continuous*
   - *(b) i is continuous at the origin*
   - *(c) $\forall n, \exists k, -B_k \subseteq B_n$*
7. *If $B_n = -B_n$, for every $n \in \mathbb{N}$, then i is continuous if and only if each r_a is continuous.*
8. *If $B_n = -B_n$, for every $n \in \mathbb{N}$, then i is continuous if and only if each c_a is continuous.*
9. *The function s is continuous if and only if*
$$\forall x, \forall n, \exists k, -x + B_k + x + B_k \subseteq B_n$$
10. *If every B_n is closed under conjugation then:*
    - *(a) each r_a is continuous*
    - *(b) each c_a is continuous*
    - *(c) i is continuous if and only if for every n there exists k with $-B_k \subseteq B_n$*
    - *(d) s is continuous if and only if for every n there exists k with $2B_k \subseteq B_n$ (it is continuous at the origin).*

5 Conclusion

Cartesian natural fibrous preorders have been introduced as a framework for analyzing topological spaces from the perspective of preordered sets. This approach recovers the usual intuition of analysis in terms of epsilons and deltas.

By examining the definitions, properties, and relationships of these mathematical concepts, this paper has shed light on their practical significance and potential for further research and exploration.

Overall, the study of groups, orders, and topological spaces provides a rich foundation for exploring mathematical structures and their applications. Continued research in these areas promises further insights into the nature of symmetry, order, and topology, with potential implications for a wide range of disciplines.

6 References

1 N. Martins-Ferreira, *On the structure of topological spaces*, Axioms **11** (2) (2022) .

2 N. Martins-Ferreira, *From A-spaces to arbitrary spaces via spatial fibrous preorders*, Math. Texts **46** (Categorical methods in algebra and topology) (2014) 221–235.

3 N. Martins-Ferreira, *Ordem, Monoides e Topologia*, Scripta-Ingenia 10 (December) (2021) 1-1.

Polyadic Inner Product and associated n-angle generalization

by Rui A. P. Perdigão [1,2]

(1) Meteoceanics Institute for Complex System Science, IUC Physics of Complex Coevolutionary Systems and Fluid Dynamical Systems, Washington, DC, USA; (2) Synergistic Manifolds, Lisbon, Portugal (perdigao@meteoceanics.org)

A polyadic inner product $\mathcal{P}^n$ is hereby defined as an n-ary generalization of the usual inner product to a higher-order operation admitting more than two input arguments or operands. For that purpose, consider a multivariate object $X = (x_1, \cdots, x_n)$ and an n-ary operator $\mathcal{P}^n$ admitting n input arguments and yielding a scalar output s:

$$\mathcal{P}^n(X) = s \tag{1.1}$$

Relative to the regular binary inner product, $\mathcal{P}$ allows for an n-ary algebraic-geometric way of finding redundancies through a single one-off operation, instead of a sequence of binary operations. This is akin to higher-order correlative or information theoretic metric of redundancy.

The practical implementation of the polyadic inner product shall require a higher-order generalization of geometric angle, which we term the $n-angle$. In this regard, the 1-angle is the regular polar angle between two vectors, and the 2-angle is related to the solid angle among three vectors as shall be seen below.

Consider the illustrative example of a triadic inner product $\mathcal{P}^3$, as a ternary operation among generic objects x, y, z. The triadic inner product $\mathcal{P}^3$ will be such that the n-angle α among x, y, x is hereby given by:

$$\alpha = 2\arctan\frac{\mathcal{P}^3(x,y,z)}{\mathcal{P}^1(x)\mathcal{P}^1(y)\mathcal{P}^1(z) + \mathcal{P}^2(x,y)\mathcal{P}^1(z) + \mathcal{P}^2(x,z)\mathcal{P}^1(y) + \mathcal{P}^2(y,z)\mathcal{P}^1(x)} \tag{1.2}$$

The difference between the n-angle definition and the classical definitions of multidimensional solid angle pertains the nature of the product involving three objects: here we have a full-fledged triad rather than the classical pseudo-triple products defined as sequences of binary product operations, e.g. the hybrid $x \cdot (y \otimes z)$.

The numerator features the triadic inner product as a pure ternary operation, whereas the denominator features a sequence of unitary and binary operations. This is reminiscent of a higher-order correlative relationship relating the triad with the combination of the lower-order m-ary dependencies, with $m < n$, thereby allowing to isolate, to some extent, synergistic information not found in any individual or combined parts up to order $m = n - 1$ i.e. the dimensionality of the full fledged system under consideration minus one.

Conversely, given the n-angle α among x, y, z, the $\mathcal{P}^3$ straightforwardly becomes:

$$\mathcal{P}^3(x,y,z) = \tan\left(\frac{\alpha}{2}\right)\left[\mathcal{P}^1(x)\mathcal{P}^1(y)\mathcal{P}^1(z) + \mathcal{P}^2(x,y)\mathcal{P}^1(z) + \mathcal{P}^2(x,z)\mathcal{P}^1(y) + \mathcal{P}^2(y,z)\mathcal{P}^1(x)\right] \tag{1.3}$$

where $\mathcal{P}^2$ is the binary inner product and $\mathcal{P}^1$ the norm or self-inner product.

Relative to a series implementation of binary inner products, $\mathcal{P}^3$ performs in one single operation what would otherwise require six binary operations. Moreover, it does so without requiring product associativity and commutativity relative to the binary products, which would be necessary in the binary setting to ensure equivalence of all possible product combinations involving x, y, z.

Therefore, this endows broader generality to the algebra underlying $\mathcal{P}^3$, which need no longer be associative as it was the case with the regular binary case. Moreover, it no longer requires conformity with the classic Cayley-Dickson algebras in terms of vector space positioning and operational dimensionality, thereby living in a broader algebra construct such as the recent generalization from Martins-Ferreira and Perdigão (2024, this issue).

Acknowledgments

R.A.P.P. acknowledges the research flagships *Mathematical Physics and Predictability of Complex Coevolutionary Systems* (MR-220617) and *Quantum Information Technologies in the Earth Sciences* (QITES) from the Meteoceanics Institute for Complex System Science, along with the C2IMPRESS project supported by the European Union under the Horizon Europe grant 101074004.

Diffusion-based methods for geometry processing in Matlab

by Miguel Belbut Gaspar [(1)]

(1) Centre for Rapid and Sustainable Product Development,
Polytechnic Institute of Leiria, Marinha Grande, Portugal
Email: miguel.gaspar@ipleiria.pt

Abstract In this note we give a brief overview of the application of diffusion-based methods to some geometry processing problems, with a focus on the practical implementation in Matlab.

Keywords: geometry processing, heat diffusion, triangulated surfaces, logarithmic map, Matlab.

1 Introduction

During most of the previous century, the field of geometry processing relied mostly on a casuistic approach, with many problems being handled in an approximate way, by mostly geometric reasoning. Many algorithms relied on iterative procedures (e.g. the Marching Cubes algorithm), resulting in slow and inneficient implementations. Methods based on Differential Geometry and PDE have become widespread in the last decade, having several benefits in terms of performance, simplicity of implementation, robustness and generality. We illustrate this approach with two very useful algorithms based on the application of the diffusion equation. In sections 2 and 3 we introduce the problem, method and core of the Matlab implementation for each of the algorithms, and in section 4 we discuss some applications and issues we are currently investigating in relation to these algorithms.

2 The heat method for geodesic distance

The problem: compute the geodesic distance from a given source vertex to all vertices on a triangulated surface.

A first rough approximation of the distances on a triangulated mesh can be obtained by finding shortest paths along edges using *e.g.* Dijkstra's method. This obviously doesn't give a correct distance, even in the limit of mesh refinement. More sophisticated methods extend the idea of Dijkstra's method to iteratively propagate the distance across the surface (like a wavefront), providing a more correct value at the cost of complexity and computational cost. Formally, the wave equation is used, which is hyperbolic.

The method. The heat method, introduced in [1], is based on the observation that the diffusion of heat approaches the geodesic distance as diffusion time tends to zero (Varadhan's formula):

$$\phi(x,y) = \lim_{t\to 0} \sqrt{-4t \log k_t(x,y)}$$

where k_t is the heat kernel after time t. Obtaining the geodesic distance directly from the above equation isn't practical, due to the non-linearity of the heat kernel; a discretization of the heat kernel will lead to large errors in magnitude of ϕ. However, the gradient of the diffused heat will still have the correct direction – opposite to the distance gradient. Intuitively, thinking of the heat transport as random walks, the direction where the heat diffuses faster is the direction of the shortest (geodesic) path.

The diffusion equation

$$\frac{du}{dt} = -\Delta u$$

where Δ is the laplacian, is used to compute the heat diffused from the source after a short time. Note that using a forward Euler scheme would require performing a number of iterations at least equal to the diameter of the graph; instead, we can use a backwards Euler scheme to obtain the diffused heat over the whole surface in a single step:

$$(\mathrm{id} + t\Delta)u = \delta_x$$

By definition, the magnitude of the gradient of the geodesic distance is

$$|\nabla\phi| := 1$$

so we have

$$\nabla\phi \approx \frac{-\nabla U_t}{|\nabla U_t|}$$

We want to find the scalar field ϕ; by applying the divergence operator to both sides, and noting that the divergence of the gradient is the laplacian, we get the Poisson equation which we can solve for ϕ:

$$\Delta\phi = \nabla\cdot\nabla\phi \approx -\nabla\cdot\frac{\nabla U_t}{|\nabla U_t|}$$

The implementation. For the implementation of this method we need the discrete analogues of the gradient, divergence and Laplacian operators.

Discrete gradient. The most straightforward and common way to compute a gradient of a per-vertex scalar field

f on a triangular mesh is the per-face constant estimation, which gives a constant gradient on each face as the sum of the area gradients relative to each vertex weighted by the value of the scalar field. Since the area gradient relative to a vertex i in a triangle ijk is determined only by the triangle's height relative to the opposed edge jk, the gradient on each face is given by

$$(\nabla f)_{ijk} = \frac{1}{2A_{ijk}} \sum_{lm \in ijk} f(v_{\bar{lm}}) e^{\perp}_{lm}$$

where $e^{\perp}_{lm}$ is an edge rotated by $pi/2$, $v_{\bar{lm}}$ is the vertex opposed to e_{lm} and A_{ijk} is the area of the face. This is implemented by the following routine:

```
function G=grad_op(tri)
% Each edge contributes to the gradient of the
% opposite vertex.
% Evec is a matrix of edge vectors (nE-by-3) in
% the same order they are defined in T.
oppv=[3,1,2]; % Map vertices to opposite edges
G=sparsev(...
   repmat((1:tri.nF)',1,3),tri.T(:,oppv),...
   (1/2)*cross(reshape(repmat(...
   (tri.FNorm./tri.FArea(:,[1,1,1])),3,1),[],3)
        ,...
   tri.Evec),...
   tri.nF,tri.nV);
end
```

The sparsev function wraps sparse and concatenates along the first dimension three sparse matrices constructed from the columns of the provided vector.

Discrete divergence. By Stokes theorem, in the smooth setting we can relate the gradient and the divergence as follows:

$$\int f \ \nabla \vec{u} \ dA + \int \vec{u} \cdot \vec{\nabla} f \ dA = 0$$

i.e. gradient and minus divergence are conjugate. This leads to the discrete analog

$$f^{\top} \ A_V \nabla \vec{u} + \vec{u} \cdot A_F \vec{\nabla} f = 0$$

and the relation defining the divergence operator

$$\nabla = -A_V^{-1} \ \vec{\nabla}^{\top} A_F$$

where A_V and A_F are the (dual) vertex areas and (primal) face areas.

We use the gradient operator for constructing the divergence using the relation above, except that we return the integrated divergence (don't multiply by A_V^{-1}):

```
function D=divg_op(tri)
% Return a linear divergence operator that acts
% on a per-face vector field.
% It gives the integrated divergence over the
% vertex (dual) area.
D=tri.grad'*diag(repmat(sparse(tri.FArea),3,1))
     ;
% Compute the area of vertex duals:
% VArea=sparse(tri.T,1,tri.FArea(:,[1 1 1]))/3;
% D=D./repmat(VArea,3,tri.nF);
end
```

Discrete laplacian. Having the gradient and the divergence, we can compute a laplacian operator by just multiplying their matrices; however, it may be useful to compute the laplacian directly. Many discrete formulations of the laplacian exist, depending in particular on how to define the dual of a vertex, each capturing some of the properties of the smooth counterpart, but none can exactly reproduce all the structure of the smooth laplacian. One commonly used discretization is the "cotan" laplacian, using the barycentric dual:

$$(\Delta u)_i = -\frac{1}{2A_i^*} \sum_{ij} (\cot \alpha_{ij} + \cot \beta_{ij}) (u_j - u_i)$$

where A_i^* is the area of the dual of vertex i and α_{ij}, β_{ij} are the angles opposite to edge ij to its left and right.

This leads to a matrix operator in the form of a nV-by-nV matrix with elements

$$[L_{ij}] = \begin{cases} -\frac{1}{2A_i^*}(\cot \alpha_{ij} + \cot \beta_{ij}) & i \neq j \\ -\sum L_{ij} & i = j \end{cases}$$

Note that for numerical stability and efficiency it is convenient to decompose this matrix in the product $L = M^{-1}C$, where $M = A_i^*$ is called the "mass matrix" and C is the "stiffness matrix"; which we implement in the following function.

```
function [C,A,t]=lapl_op(tri)
% Implements the "cotan laplacian".
% L=tri.divg*tri.grad;

% The vertex indices of each edge:
nF=tri.nF;
E2V=[tri.T(:),reshape(tri.T(:,[2,3,1]),[],1)];
E2F=[[1:nF,1:nF,1:nF]',...
   [ones(nF,1);2*ones(nF,1);3*ones(nF,1)]];

% The angle opposite to each edge:
oppA=tri.abc(E2F(:,1)+nF*mod(E2F(:,2)-2,3));
cotA=sparse(E2V(:,1),E2V(:,2),cot(oppA));
cotB=cotA';

C=-(cotA+cotB)/2;
C=C-diag(diag(C));
C=C-diag(sum(C));

% The mass matrix is the dual area of each
     vertex;
% the 'lumped' or barycentric mass matrix is
     just
% the face areas equally divided to its
     vertices:
A=diag(sparse(tri.T,1,tri.FArea(:,[1 1 1]))/3);

% We also return t (square of mean edge length)
t=mean(normv(...
   tri.V(tri.EE(:,:,2),:)-...
   tri.V(tri.EE(:,:,1),:))).^2;
end
```

The functions normv (and normalizev) return the norm (or the normalized vectors) of a $n \times 3$ matrix (interpreted as n vectors). The tri_angles function returns the interior angles a, b, c opposite to each oriented edge, in the same order as they appear in the faces matrix T.

The diffusion from a source can be computed by the function:

```
function U=lapl_vertex_diffuse(U0,C,A,t)
% Diffuses a (scalar? vector? complex?)
    quantity
% stored per-vertex: U = U0 - lapl(U0) * t
% Does one step backward Euler propagation.
U=((U0.'*A)/((A+C*t))).';
end
```

We can now compute the geodesic distance by:

```
function D=geodesic_distance(source,tri)
% Compute geodesic distance from weighted
    source
% field, via heat method. Sources are the
    vertices
% with non-zero initial weight.
% Note: we don't multiply RHS by mass matrix
% because our divergence is already integrated.
integrate_grad=@(G,d,C)((d*G)'/C)';
geodist=@(src,C,A,g,d,t)...
  integrate_grad(-normalizev(...
  g*lapl_vertex_diffuse(src,C,A,t)),...
  d,C);

D=geodist(source,...
  tri.lapl,tri.mass,tri.grad,tri.divg,tri.t);
D=D-min(D(any(source~=0,2)));
end
```

3 The diffusion method for parallel transport

The problem. If we have a tangent vector field on a surface, how do we compare directions at different points? The heat method gives us a consistent notion of distance accross the surface, but how do we define a consistent notion of direction across the whole surface? The answer is the parallel transport and the concept of connection.

The method. The vector heat method [2] extends the concepts of the heat method for geodesic distance to vector quantities. The basic idea is to diffuse a vector from a source to the whole surface. As in the scalar case, the Laplacian is used for the diffusion, but now this Laplacian must encode how a vector rotates as it moves along the surface. A most useful and natural way to transport a vector on the surface is the Levi-Civita connection, *i.e.* move a vector such that it keeps its direction relative to the geodesics (with zero geodesic curvature).

To represent a tangent vector space we can use a complex number at each vertex. Each vertex has a distinct tangent space, so we need to define a reference direction in each tangent space, which can be any arbitrary but fixed unit tangent vector at each vertex. Each edge ij then has an angle φ_{ij} in the tangent space of its start vertex i. We can incorporate the rotation in the Laplacian by augmenting the off-diagonal elements L_{ij} with the appropriate imaginary part, corresponding to the rotation as a vector is transported along the corresponding edge ij. As the edges are by definition geodesics between their endpoints, the angle of this rotation under the Levi-Civita connection is just the transformation from the reference direction at the start vertex to the end vertex, given by the difference between the angle of its endpoints

$$\theta_{ij} = (\varphi_{ji} + \pi) - \varphi_{ij}$$

Thus we define the (discrete) Connection Laplacian, denoted by Δ^∇ as

$$(\Delta^\nabla \mathbf{v})_i = -\frac{1}{2A_i^*}\sum_{ij}(\cot\alpha_{ij} + \cot\beta_{ij})\, e^{i\ \theta_{ij}}(\mathbf{v}_j - \mathbf{v}_i)$$

We can now perform diffusion of a vector from a source vertex by using the connection laplacian in the diffusion equation, but in this process the magnitude is not preserved. However, the decay in magnitude is the same as for the scalar diffusion, so we can find and correct for it. If we diffuse a single vector, we may just ignore the magnitude and set it to the magnitude of the original vector.

The implementation. Taking advantage of the routines presented before for the laplacian diffusion, we just need to compute the connection laplacian.

```
function [Lc,H,phi]=conn_lapl_op(tri)
% Construct a reference vector field H:
H=rand_tangent_field(tri);

EE=reshape(tri.EE,[],2);
% Edge unit vectors
Euv=normalizev(tri.V(EE(:,2),:)-tri.V(EE(:,1)
    ,:));
% Project to vertex tangent plane
tmp1=Euv-repmat(...
  dot(Euv,tri.VNorm(EE(:,1),:),2),1,3).*Euv;
% Compute the angle of each edge relative to
    the
% reference field:
tmp2=vec2ang(tmp1,H(EE(:,1),:),...
  tri.VNorm(EE(:,1),:));
% The rotation when moving along each edge:
phi=sparse(EE(:,1),EE(:,2),tmp2+2*pi);
ij=sub2ind([tri.nV,tri.nV],EE(:,1),EE(:,2));
ji=sub2ind([tri.nV,tri.nV],EE(:,2),EE(:,1));
rho=mod(phi(ji)+pi-phi(ij),2*pi);

% Multiply the off-diagonal elements of L:
r_ij=sparse(EE(:,1),EE(:,2),exp(1i*rho))+...
  speye(tri.nV);
Lc=tri.lapl.*r_ij;
end
```

The utility functions vec2ang, vec2ts and ts2vec perform the transformation between Cartesian vectors and the complex representation in the tangent space.

The vector diffusion or transport can then be computed by:

```
function V=vector_vertex_diffuse(V0,tri)
% The input vector field V0 could be in
    cartesian
```

```
% coordinates (vx,vy,vz) or in complex
% representation (tangent space). Note however
% that the cartesian representation does not
% ensure the result is a tangent vector field.
V=lapl_vertex_diffuse(V0,tri.conn_lapl,tri.mass
     );
end
```

4 Applications

Logarithmic map. One particular application of the parallel transport described in [2] is to obtain the logarithmic map. This is defined in relation to a specified source and associates to each point not only a distance to the source but also a direction – the initial direction of the shortest geodesic from the source to the target point. To obtain the logarithmic map, we just use the parallel transport of an initial gradient of the distance around the source. The critical part in this procedure is to obtain an accurate source gradient vector field around the source point.

Naively, we could use the gradient operator defined in section 2 (indeed, for the magnitude of the logarithmic map we may start by applying the heat method, where we compute this gradient). This however computes a gradient per-face, but we need a gradient on vertices because our parallel transport operates on vertex-based vector fields. Simply transfering the face gradient to vertices will lead to inconsistencies. It is possible to define a discrete gradient on vertices (see [7]), but we don't have to; we may just compute one explicitely around the source vertex. This approach is typical: use global PDE-based formulation derived from smooth analysis, and use explicit formulation only locally for the boundary conditions. In this case, restricting our domain to vertices around the source allows to define exactly a discrete gradient, although the derivation is too long to include here (see the appendix in [2]).

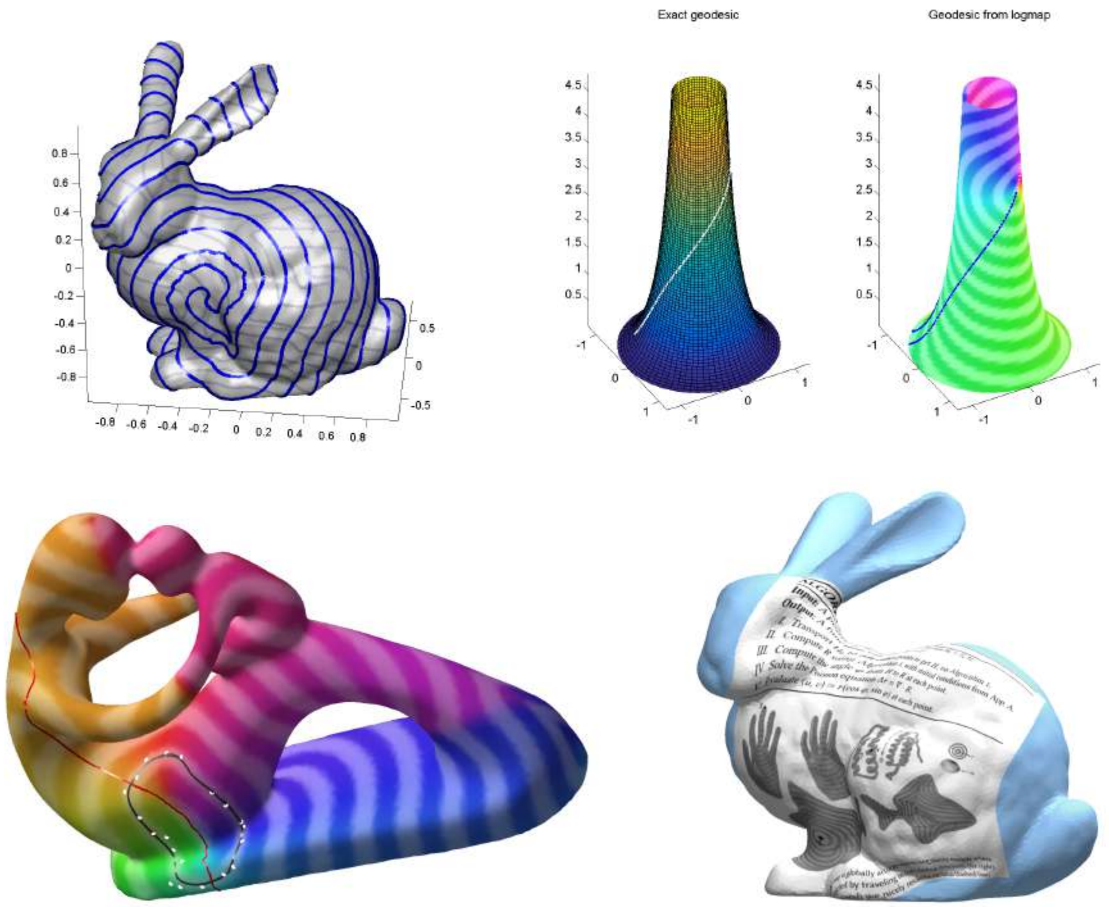


Figure 1: Some example applications of diffusion-based methods. top left) construction of uniformly spaced continuous spirals; top right) construction of geodesic segments; bottom left) the "curve logmap" for creating smooth curves from control points; bottom right) defining orthogonal coordinate system for mapping a region to the plane.

Note that, by using parallel transport, we obtained a per-vertex gradient of the geodesic distance over the whole domain; we could, then, obtain the geodesic distance from this gradient, using a per-vertex divergence operator and the Poisson equation as in section 2. By starting with more information, this could arguably provide a more accurate geodesic distance as well as ensure consistency between lines of constant angle and constant distance.

Multiple sources. In the previous discussion we considered just a single source vertex, which is encoded as a single nonzero in a single column source field. The way these algorithms are formulated naturally leads to the question of multiple sources. Indeed, for the logarithmic map we already mentioned applying the parallel transport to multiple source vectors, without providing a justification.

Let's focus first on the use of the heat method with multiple sources. The natural thing to ask when we have multiple sources is: *what is the distance at each point to the nearest source?* Additionally, we may be interested in knowing which of the sources is the nearest to each point. Three approaches can be used:

i – Clearly, we can reuse the operators constructed and apply them to compute distances from each source separately. For a small number of sources, this is already reasonably efficient, as it involves just some matrix multiplications and two linear solves, where only the right-hand side changes. We can then obtain the infimum over all sources for each point (while also finding the corresponding source).

ii – We could be tempted to combine all the sources in a single vector, and perform a single pass of the heat method; but this only approximately works, because the diffused heat from each source "overlaps" and adds up; the resulting heat distribution is influenced at every point by all sources, not just the nearest one. However, since the diffused heat decays very quickly with distance, at each point near a particular source the heat diffused from that source will dominate, thus giving a smoothed approximation of the gradient from the nearest source. This approach may be used in certain cases, and is extremely efficient. Note however that this does not allow us to identify which source is the closest.

iii – An alternative is to note that: 1) after we have a diffused heat field, we don't care about each individual source; 2) for equal initial heat, the heat diffused to a point is maximum for the source that is the nearest to that point. Thus, we can diffuse the heat from each source (in a single application of the backwards Euler, but with each source as a separate column), and obtain the final diffused heat as the supremum over all sources. From here, we can apply the remaining steps to this heat field. This is more efficient than method *i* and also allows identification of the source. Empirically, we find only subtle differences between this and the first approach, whose origin we are still investigating.

For the case of vector quantities (parallel transport), the problem of superposition of multiple sources is lessened to some extent by the fact that contibutions from different directions tend to cancel each other. This is in many cases enough to allow the use of the most efficient approach *ii* of "collapsed" sources in a single column; in other cases, both approaches *i* and *iii* above could be used also for vector quantities.

"Curve" sources. A particular case of multiple sources is that of a closed path or circuit on edges of the mesh. This deserves a distinct treatement from the general case because in this case all sources are topological adjacent – there are no "gaps". Empirically, we find that the algorithms give reasonable results in this case, even when using the approach of "collapsed" sources. A further generalization would be the case of a piecewise linear closed curve on the surface (*i.e.* linear on each face). We are still working on a more formal characterization of these cases.

"Curve log map". The above led us to consider the extension of the logarithmic map to a closed curve as source. The geodesic distance from multiple source vertices can be used to construct smooth closed curves (for example, for interactive drawing on the surface). Our extension of the logmap to closed curves then lets us construct parallel offsets of these curves.

The cut locus. In all of the above we have skipped over an important question: how accurate are these methods? In fact, they are not the most accurate – at least in some particular regions. Even in the smooth setting, although the geodesic distance is well defined for any pair of source and target points, the logarithmic map is not; in general, there are target points where more than one geodesic path have the same minimal length. The set of these target points forms the *cut locus*. Moreover, in the discrete setting, every vertex with positive angle defect is part of the cut locus. The practical consequence is that for an accurate description around the cut locus we may need to store our information per corner.

On the other hand, the diffusion methods rely on an arbitrarily chosen time step, that can not be made exceedingly small; in practice, for t smaller than the square of the average edge length the results start to degrade. This finite time diffusion results in a Laplacian smoothing where the geodesic isn't harmonic – in the vicinity of the cut locus. While this may be desirable in some cases (the degree of regularization being adjustable by varying t), it may preclude some applications. Recursive application of the algorithm is one of the proposed methods to overcome this error; we are investigating other ways to improve the results near the cut locus. For a more detailed analysis we refer the reader to [4, 3].

Computation on sub-domains. Besides the possibility of pre-computing many of the required structures (matrices) which can be reused for multiple sources, the structure of the algorithms also allows to easily perform computations on a simply connected compact subset of our domain. This means that in some cases, where exact/-global solution isn't needed, we can restrict our operators to a small subset of vertices, edges and faces, greatly improving performance of some computations. For example, if we already have a distance field computed from a point, we may want to perform some computation only on a ball of some radius around that point; we can easily remove the non-relevant rows and columns from the matrices defining our operators. This involves some possible inaccuracy due to the introduction of boundaries.

Handling of boundaries. The implementations described above apply to a closed surface, but the methods can be easily extended to bounded surfaces or to handle prescribed boundaries. The main modification has to do with the fact that half-edges on the boundary will not have a twin; we can solve this by a small modification to the construction of our operators. Note that, regarding the behaviour on the boundaries, contrary to what happens at the cut locus, we have the benefit of knowing beforehand its location, so we can adjust the boundary conditions to our need; in particular, a mix of Dirichlet and Neumann results in the most accurate representation of geodesic distance on the boundary.

5 Conclusion

Despite some limitations, diffusion based methods (and discrete differential geometry methods more generally) are extremely efficient and flexible tools which serve as a base for many geometry processing algorithms. By their nature, being based on linear operators, they lend themselves to very simple and efficient implementations in Matlab. This is an area of very active development, with several extensions and improvements of the heat method and vector heat method having been proposed recently, as well as alternative approaches (*e.g.* [5, 6, 8]). For further information on this subject, we refer the reader to [3][4].

Acknowledgments This work was supported by portuguese Fundação para a Ciência e Tecnologia under grant 2022.14868.BD.

Referências

[1] Crane, Weischedel, Wardetzky. The Heat method for distance computation (2013)

[2] Sharp, Soliman, Crane. The Vector heat method (2019)

[3] Crane, Livesi, Puppo, Qin. A Survey of algorithms for geodesic paths and distances (2020)

[4] Belyaev, Fayolle. On variational and PDE-based distance function approximations (2015)

[5] Trettner, Bommes, Kobbelt. Geodesic distance computation via virtual source propagation (2021)

[6] Edelstein, Guillen, Solomon, Ben-Chen. A convex optimization framework for regularized geodesic distances (2023)

[7] C. Mancinelli & M. Livesu & E. Puppo / Gradient Field Estimation on Triangle Meshes (2018)

[8] C. Mancinelli & E. Puppo. Computing the Riemannian Center of Mass on Meshes (2023)

[9] C. Mancinelli & E. Puppo. Nearly smooth differential operators on surface meshes (2022)