UNIVERSALITY OF POLYHEDRAL LINKAGES

ROBERT MIRANDA

Abstract. Planar linkages are a classicalal area of study motivated by practical applications in engineering mechanisms. A central result is Kempe's Universality Theorem, which states that polynomial functions can be realized by planar linkages on a bounded set. However, these linkages are often not planar graphs, and so it is not immediate if a physical mechanism modeling this functionality could be built. In this paper, we introduce the notion of polyhedral linkages is $n \geqslant 2$ dimensions, (which correspond to existing notions when $n = 3$ and $n = 2$) and generalize Kempe's Universality Theorem to this linkages with an embedded construction when $n \geqslant 3$.

1. Introduction

1.1. **Previous Work.** Recall the definition of a planar linkage as an edge weighted graph that is realized in $\mathbb{R}^2$ such that the weight of each edge corresponds to the distance between the two adjacent vertices. Planar linkages can be used to 'compute' functions in the following sense. Suppose $U \subset \mathbb{R}^2$ is an open set and $F : U \to \mathbb{R}^2$ is a function. We say that a planar linkage *defines* F on the neighborhood U if the linkage has an input vertex x and an output vertex y such that whenever x is realized in U, then y is contrained to be realized at $F(x)$. (See [KM] for a complete definition.)

Remarkably, planar linkages are universal in the sense that they can realize any polynomial function in any bounded region. This result is known as Kempe's Universality Theorem after Kempe's 1875 paper, but the first correct proof was not published until 2002 by Kapovich and Milson. (See [KM], also [Kem].) Alternatively, we may fix the output vertex to a specific point, which constrains the input vertex to trace the zero locus of the defined function. Such linkages are called *closed*. Thus planar linkages can trace any bounded region of an algebraic set, or as popularized by Thurston, they can "sign your name". (See [King].)

However, it is not immediate from Kempe's Universality Theorem whether a physical machine which can model the behavior of an arbitrary planar linkage could actually be built. Not all realizations of planar linkages produce planar graphs, and so any physical implementation must necessarily utilize 3-dimensions to allow for difference edges to cross each other, or different vertices to overlap.

In this paper, we generalize the definition of a planar linkage to a *polyhedral linkage* in $n \geqslant 2$ dimensions. In higher dimensions, a polyhedral complex satisfying certain conditions plays the role of an edge weighted graph in planar linkages. (See § 2.) This recovers the definition of planar linkages when $n = 2$, and previous notions of polyhedral linkages which were only defined in the $n = 3$ case. (See e.g. [Gol].) Our main focus is in the $n = 3$ case.

1.2. **Main Results.** We say that a realization of a polyhedral linkage is *embedded* if the interiors of the maximal polytopes do not intersect. Embedded realizations correspond to configurations that could be achieved by a physical mechanism. For $n = 2$, most functional linkages cannot be embedded. One problem is the Peaucellier inversor, a fundamental construction used to define inversion and multiplication. While the Peaucellier inversor can be embedded, it uses an output vertex which is not adjacent to the exterior

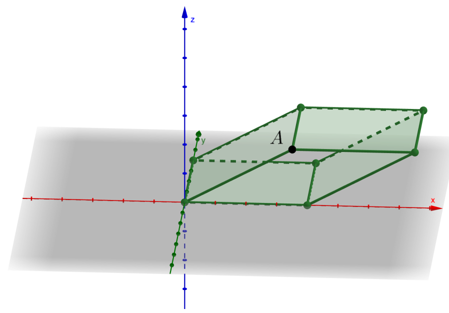

FIGURE 1. A polyhedral linkage.

face, so after any nontrivial composition the resulting linkage will not be a planar graph. (See § 5.) However, in 3-dimensions and higher, we are able to extend Kempe's Universality Theorem with embedded polyhedral linkages.

Theorem 1.1. *Let $n \geqslant 3$, $U \subset \mathbb{R}^n$ be a bounded, open set, and $F : U \to \mathbb{R}^n$ be a polynomial function. There exists an embedded, functional polyhedral linkage $\mathcal{P}$ which defines F on U.*

We can also generalize the corollary of Kempe's Universality Theorem for closed linkages to higher dimensions. That is, to use Thurston's phrasing, there exist embedded polyhedral linkages which can "sign" any algebraic set in $\mathbb{R}^n$.

Theorem 1.2. *Let $n \geqslant 3$, $U \subset \mathbb{R}^n$ be a bounded, open set, and $S \subset \mathbb{R}^n$ be an algebraic set. Then there is an embedded polyhedral linkage which realizes $S \cap U$.*

1.3. **Embedded linkages with multiple inputs and outputs.** Kempe's Universality Theorem extends to a stronger result that any polynomial function $F : (\mathbb{R}^2)^{m_1} \to (\mathbb{R}^2)^{m_2}$ can be realized by planar linkages on an open bounded set. (See [KM].) However, allowing multiple inputs and outputs poses a separate problem for embedded linkages. For example, consider a function with two inputs. If this were simulated by a physical machine, then the two input vertices may be realized at the same point. This would mean that the physical mechanisms should be able to 'pass through each other'. A similar problem can occur with output vertices.

To address this problem, we propose a modification to the definition of a functional linkage which considers each input and output vertex relative to its own reference coordinate frame. Specifically, in the forgetful maps defining the input and output vertices, we introduce a postcomposition which translates each coordinate. (See § 2, and [KM] for comparison.) This allows us to extend the vector version of Kempe's Universality Theorem to $n \geqslant 3$ dimensions in the embedded case.

1.4. **Computation in polyhedral linkages.** Our construction of functional polyhedral linkages centers around an efficient way to generate linear motion in $n \geqslant 3$ dimensions. Specifically, for any $0 < a < b$, we produce an embedded polyhedral linkage whose length in one dimension varies continuously in the interval (a, b), and which is arbitrarily small in all other dimensions. (See § 3.)

We set up an array of these *extender* linkages arranged parallel to each other in a fixed hyperplane. The length of each extender encodes a scalar value, and the register of extenders allows us to encode a vector value. Note that this differs fundamentally from registers used in digital computers because each extender stores a value with infinite precision, but the range of values an extender can hold is still limited by its construction, i.e., by the chosen values of a and b.

Computations are decomposed into a sequence of elementary operations. Each elementary operation uses an input register and an output register, which is a parallel translate of the input register. The output of the operation is stored in an extender in the output register by attaching both the output extender, and all input extenders, to a specific elementary linkage. (See § 4) These are generalizations of elementary planar linkages to n-dimensions, sometimes with slight modifications so that they are embedded. (See §2 and [KM, §6] for comparison.) All other values in the original register are faithfully copied to the output register by rigid linkages.

1.5. **Structure of the paper.** We prove our main results, Theorem 1.1 and Theorem 1.2, by showing that all polynomial functions can be decomposed into elementary operations. Then we construct a polyhedral linkage in $\mathbb{R}^n$ which has a point that can be moved freely in an n-dimensional region, and which records the coordinates of this point in a register of n extenders. (See § 5.) We use this register as the beginning of our computation, and attach a similar linkage to combine the final output register into the coordinates of a single output vertex. The generalization to multiple inputs and outputs is simple with our convention of separate reference frames. (See § 2.)

2. Polyhedral linkages

Our definition of a polyhedral linkage is a direct generalization of the definition of a planar linkage to include an $(n-1)$-dimensional polyhedral complex realized in $\mathbb{R}^n$. See [KM] for motivation and full details of the original definition for planar linkages.

2.1. **Polyhedral linkages.** Let $\mathcal{P}$ be a polyhedral complex in $\mathbb{R}^n$. We say that $\mathcal{P}$ is *pure* if every maximal polytope has the same dimension, and we say that $\mathcal{P}$ is *proper* if the intersection of any two k-dimensional faces in $\mathcal{P}$ is either $\varnothing$, or a $(k-1)$-dimensional face in $\mathcal{P}$. An edge weighted graph is exactly a 1-dimensional pure, proper polyhedral complex in $\mathbb{R}^2$.

A *polyhedral linkage* of dimension n is a pair $(\mathcal{P}, W)$, where $\mathcal{P}$ is a n-dimensional pure, proper polyhedral complex in $\mathbb{R}^{n+1}$, and $W \subset \mathcal{V}(\mathcal{P})$ is a subset of the vertices of $\mathcal{P}$ called the *fixed vertices*. When clear by context, we will refer to the polyhedral linkage as $\mathcal{P}$ without reference to W.

A *realization* of $\mathcal{P}$ is a map $\phi : \mathcal{V}(\mathcal{P}) \to \mathbb{R}^{n+1}$ such that for each maximal face $F \in \mathcal{P}$, the vertices $\{v \in \mathcal{V}(\mathcal{P}) \mid v \in F\}$ form the vertices of a polytope congruent to F. I.e., the maximal polytopes can be rearranged by individual rigid motions as long as they all fit together in the same structure. The $(n-1)$-dimensional faces act as "hinges", allowing the structure to flex. The set of all realization, $C(\mathcal{P})$, is called the *configuration space* of $\mathcal{P}$.

Given a set of points Z in $\mathbb{R}^{n+1}$ in bijection with W, a realization $\phi : \mathcal{V}(\mathcal{P}) \to \mathbb{R}^{n+1}$ is said to be *relative to* Z if $\phi(w) = z$ for each $w \in W$ and the corresponding $z \in Z$. The set of all relative realizations to Z, $C(\mathcal{P}, Z)$, is called the *relative configuration space*.

We say that a realization $\phi \in C(\mathcal{P}, Z)$ is *embedded* if the interiors of all maximal faces are pairwise disjoint. The set of all embedded realizations, $C^e(\mathcal{P}, Z)$, forms an open subset of $C(\mathcal{P}, Z)$. For example, consider the 2-dimensional polyhedral linkage $\mathcal{P}$ formed by removing two opposite faces of a cube. (See Figure 1.)

Let W be the vertices of one face of the cube, and let Z be the four points $\{(0,0,0), (1,0,0), (1,1,0), (0,1,0)\}$ in the xy-plane. The relative configuration space $C(\mathcal{P}, Z)$ consists of three intersecting smooth curves and is naturally identified with the moduli space of the square, [KM, §3]. However, the embedded realization space $C^e(\mathcal{P}, Z)$ is identified with the set $\{x^2 + z^2 = 1; z \neq 0\}$, where the location of the vertex

A determines the entire configuration when A is not on the xy-plane. The points where A is on the xy-plane correspond to self-intersecting realizations of $\mathcal{P}$, and are included in the other two curves in $C(\mathcal{P}, Z)$. Any physical model which is built to emulte the behavior of $\mathcal{P}$ can naturally move within a single connected component of $C^e(\mathcal{P}, Z)$.

2.2. **Functional linkages.** Next we define *functional linkages.* Let $m_1, m_2 \geqslant 1$ and $F : \mathbb{R}^{(n+1)m_1} \to \mathbb{R}^{(n+1)m_2}$ be a function. A functional linkage is a polyhedral linkage $\mathcal{P}$ of dimension n with two distinguished sets of vertices $\{P_1, \dots, P_{m_1}\}$ called *input vertices* and $\{Q_1, \dots, Q_{m_2}\}$ called *output vertices.* We also define two forgetful maps $p : C(\mathcal{P}, Z) \to \mathbb{R}^{(n+1)m_1}$ and $q : C(\mathcal{P}, Z) \to \mathbb{R}^{(n+1)m_2}$ as

$$p(\phi) = (\phi(P_1) + X_1, \dots, \phi(P_{m_1}) + X_{m_1}),$$

$$q(\phi) = (\phi(Q_1) + Y_1, \dots, \phi(Q_{m_2}) + Y_{m_2}),$$

for some choice of translations $X_1, \dots, X_{m_1}, Y_1, \dots, Y_{m_2} \in \mathbb{R}^{n+1}$. We say that $\mathcal{P}$ *defines* the function F at a point $\mathcal{O} \in \mathbb{R}^{(n+1)m_1}$ if there is a commutative diagram

$$\begin{tikzcd} & C(\mathcal{P}, Z) \arrow[dl, "p"'] \arrow[dr, "q"] & \\ \mathbb{R}^{(n+1)m_1} \arrow[rr, "F"] & & \mathbb{R}^{(n+1)m_2} \end{tikzcd}$$

and p is a regular topological branched cover of a bounded open set $U \subset \mathbb{R}^{(n+1)m_1}$ containing the point $\mathcal{O}$. Alternatively, we say that $\mathcal{P}$ is a *functional linkage* for the germ $(F, \mathcal{O})$. Moreover, we say that a functional linkage $\mathcal{P}$ is *embedded* if there exists a connected component $E \subset C^e(\mathcal{P}, Z)$ such that the restrictions $p|_E, q|_E$ still form a commutative diagram.

$$\begin{tikzcd} & E \arrow[dl, "p|_E"'] \arrow[dr, "q|_E"] & \\ \mathbb{R}^{(n+1)m_1} \arrow[rr, "F"] & & \mathbb{R}^{(n+1)m_2} \end{tikzcd}$$

Finally, we say that a functional linkage $\mathcal{P}$ is *closed* if the output vertices are also fixed vertices. In this case, the image of the input map corresponds to the zero set of the defined function, and we say the $\mathcal{P}$ *realizes* U, where $U := p(C(\mathcal{P}, Z))$, or $U := p|_E(E)$ in the embedded case.

3. Linear motion in three dimensions

In $\mathbb{R}^2$, finding a linkage which produces linear motion was a fundamental problem in the field. (For a history of the solution, see [KM, §14].) However, the construction in $\mathbb{R}^3$ is compartively simple. In fact, in our current terminology a polyhedral linkage which achieves linear motion was first discovered by Sarrus in 1853, several years before any solution in $\mathbb{R}^2$ has been published. (See [Sar], [Gol] and also [WKA].) Here we give a construction which is a modification of Sarrus' linkage. It has the added property that the linkage is periodic, and remains periodic during flexion. This is important for scalability in the construction of embedded polyhedral linkages.

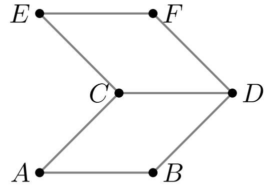

Figure 2. A planar linkage which achieves a 2-dimensional range of motion.

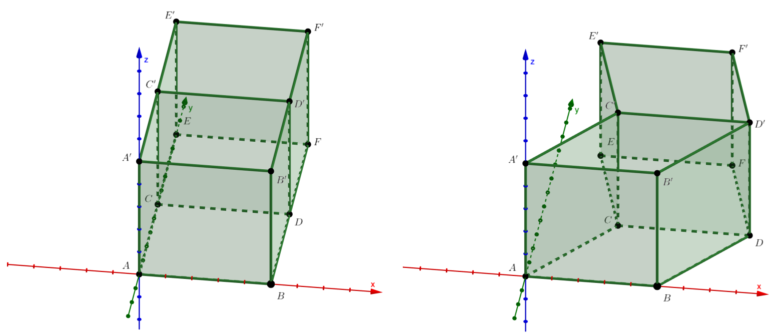

Figure 3. A polyhedral linkage which achieves a 2-dimensional range of motion.

3.1. **Basic construction.** Consider the following planar linkage. The vertices A and B are fixed at $(0,0)$ and $(1,0)$, respectively. All other vertices are allowed to move freely, and all edges have length 1. See Figure 2.

The vertex E may move within the 2-dimensional region $\{x^2+y^2 \leqslant 4\}$. However, when this planar linkage is extruded to a polyhedral linkage in 3-dimensions, the domain of the corresponding vertex remains 2-dimensional. It does not gain an extra dimension of flexibility.

This extruded linkage is shown in Figure 3. The linkage is marked by fixing the vertices A, B and A' to be at $(0,0,0), (1,0,0)$ and $(0,0,1)$, respectively. All other vertices are allowed to move freely, and all edges have length 1.

The vertex E may move within the region $\{x^2+y^2 \leqslant 4; z=0\}$, which is still 2-dimensional. Moreover, we can constrain this linkage to restrict the motion to be 1-dimensional. Suppose we add two vertices, X and Y, which form two squares $ABYX$ and $EFYX$. See Figure 4.

The lengths of all the new edges are set to 1. The squares $ABYX$ and $EFYX$ constrain the vertices E and A to have the same y-coordinate. Therefore, the vertex E may move in the 1-dimensional region $\{0 \leqslant x \leqslant 2; y=0; z=0\}$. This is a notably simpler solution to linear motion in 3-dimensions than in 2-dimensions. Also note that Sarrus' original linkage is obtained by removing the vertices D and D', and all edges and faces adjacent to them. (See [Sar].)

3.2. **Periodic construction.** We can extend this construction periodically as follows. Note that in any realization of the previous linkage, the squares $EFF'E'$ and $ABB'A'$ are parallel translates of each other by a scalar multiple of the normal vector to each face. We attach multiple copies of the linkage together by identifying the $EFF'E'$ square on the i^{th} copy with the $ABB'A'$ square of the $i+1^{\text{st}}$ copy.

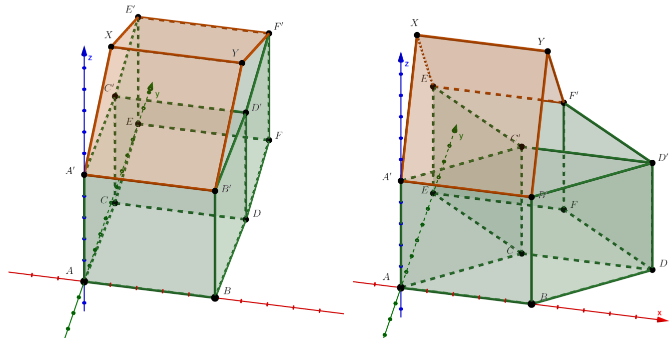

FIGURE 4. A polyhedral linkage which achieves a 1-dimensional range of motion.

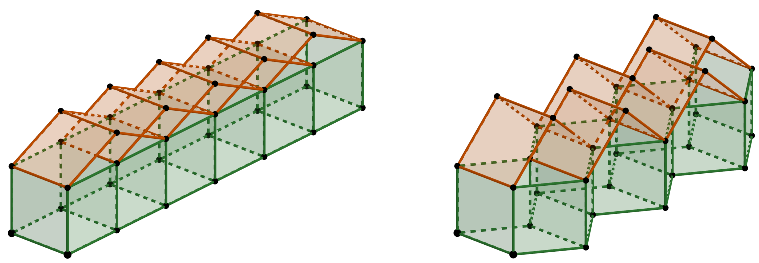

FIGURE 5. A portion of a periodic polyhedral linkage.

With n copies attached, the E-vertex of the n^{th} copy can be moved within the region $\{0 \leqslant x \leqslant 2n; y = 0; z = 0\}$. Moreover, note that if we also add 'roofs' connnecting the CD edge of the i^{th} copy to the CD edge of the $i+1^{\text{st}}$ copy, then the y-coordinates of the C and D vertices of all constituent linkages will be the same, so they all will flex at the same rate. See Figure 5.

Extending infinitely, we obtain a periodic polyhedral linkage whose configuration space is 1-dimensional, and whose periodicity is preserved during flexion. The direction of the periodicity vector does not change, only the magnitude. This construction can also be embbed by using parallelograms instead of rectangles for the 'roofs', see Figure 6.

Restricting this linkage to a connected component of the embedded realization space $C^e(\mathcal{P}, Z)$, we see that the each link has a length in the range $(0, 1)$ measured along the x-axis. To summarize our results in this section, our construction has proved the following theorem:

Theorem 3.1. *Let $0 < a < b$ and $\epsilon > 0$. There exists an embedded polyhedral linkage $\mathcal{P}$ with two faces F_1, F_2 and a 1-dimensional flex such that F_1 and F_2 remain parallel translates of each other during flexion while the distance between them varies continuously in the range (a, b), and $\mathcal{P}$ can be contianed in a cylinder of radius ϵ and length $b - a$ under any realization.*

Proof. We scale our given construction so that the side length of each of the squares is a number d such that (1) under any flexion, each unit of the linkage is contained in a ball of radius ϵ, and (2) $b - a = N \cdot 2d$

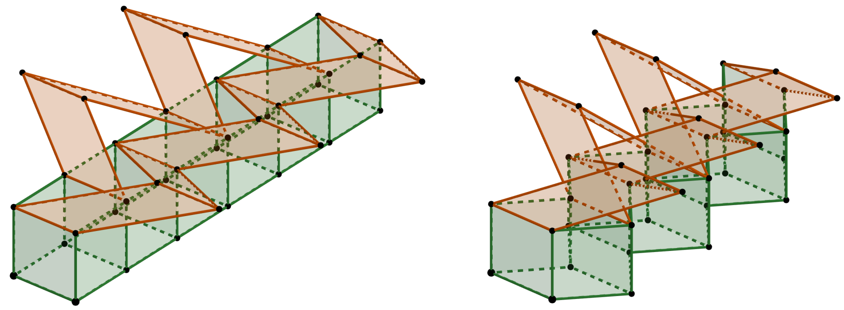

FIGURE 6. A portion of an embedded periodic polyhedral linkage.

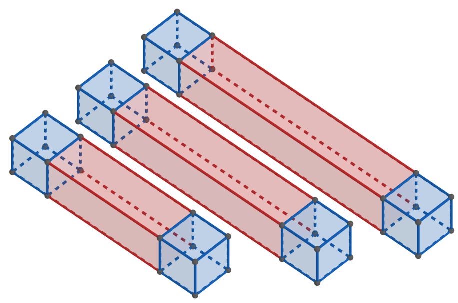

FIGURE 7. A set of extender linkages representing computational vertices.

for some integer N. Thus a linkage with N units, along with a rigid component of length a, will vary continuously in length between a and b. □

We call this construction an *extender linkage*. To simplify future figures, we represent an extender with a red rectangular prism which is understood to extend and contract in length. In constrast, blue faces are rigid faces and cannot be extended. (See Figure 7.)

4. Scalar computation

Suppose we have an array of extender linkages, pointed parallel to the x-axis in the xy-plane, and spaced at equal intervals so that the linkages never intersect under any realization. We fix one end of each extender on the y-axis and allow the other end to move freely parallel to the x-axis in a set range. Each linkage encodes a single real number represented by its length, and in this section we will describe how to use these extenders to perform computations on these inputs via embedded polyhedral linkages. (See Figure 7.)

Every extender linkage is taken to have the same range, (a, b), for two real numbers $0 << a < b$ to be chosen later. The midpoint $m = \frac{1}{2}(a + b)$ is taken to represent 0, and in general, an extender linkage that is extended a distance x from the y-axis represents the value $x - m$. In this case we call the extender a *computational linkage* and say that $x - m$ is its *value*. We set $N = b - m$, so a computational linkage may represent any value in the range $(-N, N)$. The array of n computational linkages will be

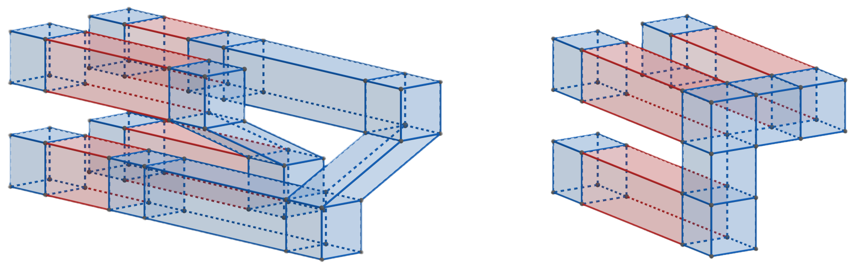

FIGURE 8. Left: a computational linkage which simulates a swap operation. Right: a computational linkage which simulates a copy operation.

called a *register*. We refer to the individual computational linkages as $C_1, \dots, C_n$ and their values as $|C_1|, \dots, |C_n|$, respectively.

Computations are decomposed into a sequence of elementary unary and binary operations which are performed on registers. Each operation takes one or two computational linkages as inputs, and stores its outputs in the values of another computational linkages. To embbed this operation, computations are performed vertically. For a single operation, an output register is created, aligned in the xy-plane but offset vertically in the z-axis with respect to the input register. The output of the operation is attached to the output computational linkages in the output register. The values of all other original computational linkages are preserved by attaching each to their offset counterpart with rigid linkages.

Let $U \subset (-N, N)$ be a connected set and $f : U \to (-N, N)$ be a function. We say that we can *simulate* the function f on computational linkages if there is an embedded polyhedral linkage which connects input computational linkage C_i to an output computational linkage C_i' such that if $|C_i| \in U$, then $|C_i'| = f(|C_i|)$. This definition also naturally generalizes if U is a connected subset of $(-N, N) \times (-N, N)$ using two input computational linkages C_i, C_j.

In the following subsections, we describe constructions which simulate elementary operations, including addition and multiplication by scalars, negation, addition, and multiplication. Together, these allow us to simulate any polynomial function on any bounded set.

4.1. **Swap and copy.** Before constructing polyhedral linkages to perform actual computations, we need to set up basic operations to manipulate memory stored in registers. We describe operations to swap two values stored in different computational linkages, as well as copy the value of one computational linkage to another.

If $C_1, \dots, C_n$ are the original computational linkages, and $C_1', \dots, C_n'$ are the updated computational linkages, then the swap operation s_i sends $|C_i| \mapsto |C_{i+1}'|$ and $|C_{i+1}| \mapsto |C_i'|$. For all $j \neq i, i+1$, $s_i : |C_j| \mapsto |C_j'|$. By composition, we can freely permute the computational linkages and so we will always assume that the input and output computational linkages are in a desirable configuration.

First C_i is connected to C_{i+1}' by a simple set of parallelograms. However, the connection from C_{i+1} to C_i' needs to be rerouted to avoid intersecting with the C_i to C_{i+1}' connection. Thus we add a rectangular prism of length at least $2N$ before adding the parallelograms to ensure that, under any realization, the

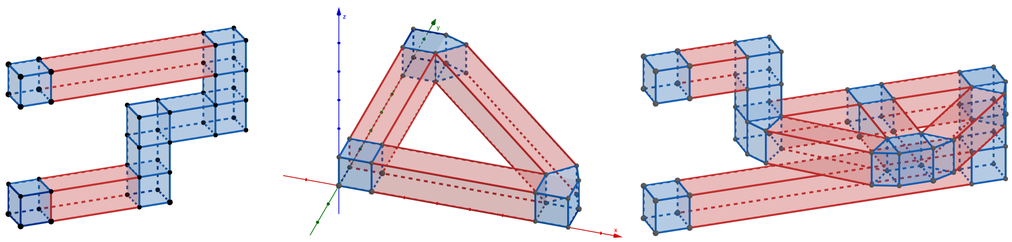

FIGURE 9. Left: a computational linkage for scalar addition. Middle: a $45° - 45° - 90°$ triangle linkage. Right: A polyhedral linkage for negation.

operation is embedded. All other computational linkages are attached vertically by rectangles. (See Figure 8, left.)

A copy operation is comparatively simpler, because there is no need for rerouting. First, C_i is connected to C'_i by a set of rectangles, and then C'_i is connected to C'_{i+1} by a rigid linkage. The entire construction is rigid, and all other computational linkages are attached vertically. (See Figure 8, right.)

Thus we can always assume without loss of generality that our linkages are in a desirable configuration. Usually, this means that input and output computational linkages are aligned vertically, or else are offset by one of two linkages. Additionally, we will assume enough space between linkages so that our constructions will always be embedded.

4.2. **Scalar addition.**

Lemma 4.1. *For $\lambda \in (0, N)$, the functions $(-N, N-\lambda) \to (-N, N), x \mapsto x+\lambda$ and $(\lambda-N, N) \to (-N, N)$, $x \mapsto x - \lambda$ can be simulated on computational linkages.*

Proof. For $\lambda \in (0, N)$, we can define the operation of scalar addition via the following linkage. (See Figure 9, left.) A rigid structure is attached to the end of the computational linkage C_i, which adds an offset of length λ before attaching to the end of the new computational linkage C'_i.

□

4.3. **Negation.**

Lemma 4.2. *The function $(-\frac{1}{2}N, \frac{1}{2}N) \to (-N, N), x \mapsto -x$ can be simulated on computational linkages.*

Proof. We can simulate the operation of negation $(0, N) \to (-N, N), x \mapsto -x$, by using two copies of the $45° - 45° - 90°$ triangle linkage. (Figure 9, middle.) Extender linkages are attached at $90°$ and connected by a third extended linkage, attached at $45°$ relative to both legs. This constrains all three linkages to flex at the same rate, and the perpendicular sides will always be the same length. The negation linkage is formed by joining two $45° - 45° - 90°$ triangle linkages along a common edge, enforcing the positive and negative legs to have the same length during all realizations. (See Figure 9, right.) The offset induced by the rigid central cube should be considered, but we can postcompose with a scalar addition operation. We leave the details to the reader.

Further, we can adjust the simulated domain to include a neighborhood of 0 by precomposing and postcomposing with scalar addition $x \mapsto x + \mu$. The largest domain centered at 0 is obtained by choosing $\mu = N/2$, which lets us define a negation map $(-N/2, N/2) \to (-N, N)$. □

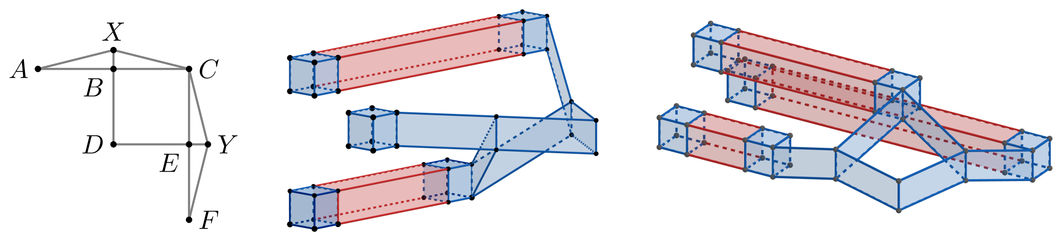

FIGURE 10. Left: a rigidifed pantograph. Middle: a computational linkage for scalar multiplication. Right: a computational linkage for addition.

4.4. **Scalar multiplication.**

Lemma 4.3. *For $\lambda \in (1, N)$, the functions $(-\frac{1}{2\lambda}N, \frac{1}{2\lambda}N) \to (-N, N), x \mapsto \lambda x$ can be simulated on computational linkages. For $\lambda \in (0, 1)$, the functions $(-\frac{1}{2}N, \frac{1}{2}N) \mapsto (-N, N), x \mapsto \lambda x$ can be simulated on computational linkages.*

Proof. We prove the case when $\lambda \in (1, N)$ first. Consider the rigidified pantograph linkage used for scalar multiplication (Figure 10, left) in planar linkages. We fix the lengths $|AC| = \lambda|AB|$ and $|CF| = \lambda|EF|$. Vertex A is fixed at the origin, D is used as an input vertex, and F is used as an output vertex. The structure is rigidified so that the pairs of edges AB, BC and CE, EF remain parallel. To preserve embeddedness, this is achieved by adding the auxilliary vertices X and Y along with the triangles ABX, BCX, CEY and EFY. From the geometry of the construction, the linkage constrains $|AF| = \lambda|AD|$, achieving scalar multiplication. (See [KM, §6.2])

We take this construction and extrude it to three dimensions (Figure 10, middle.) The vertices corresponding to D and F are offset vertically so that the corresponding computational linkages do not intersect. (Note: for figure clarity, we do not include the extrusion of the auxilliary vertices corresponding to X and Y, and the details of rigidifying are left to the reader.) This simulates the operation $(0, \frac{1}{\lambda}) \to (-N, N), x \mapsto \lambda x$, because nonpositive inputs would cause self intersections.

As with negation, we can adjust the simulated domain to include a neighborhood of 0 by precomposing with $x \mapsto x + \mu$ and then postcomposing with $x \mapsto x - \lambda\mu$. The largest domain centered at 0 is obtained by choosing $\mu = \frac{N}{2\lambda}$, which lets us define scalar multiplication on $(-\frac{N}{2\lambda}, \frac{N}{2\lambda}) \to (-N, N)$. Lastly, we can handle multiplication by $\lambda \in (0, 1)$ by switching the input and output computational linkages. This is the same process as in planar linkages. (See [KM, §6.2].) □

4.5. **Addition.**

Lemma 4.4. *The function $(-\frac{1}{4}N, \frac{1}{4}N) \times (-\frac{1}{4}N, \frac{1}{4}N) \to (-N, N), (x, y) \mapsto x + y$ can be simulated on computational linkages.*

Proof. For planar linkages, we can use the pantograph to simulate the function $x, y \mapsto \frac{1}{2}(x+y)$. (Figure 10, left.) Here we fix $\lambda = 2$ and take the vertices A and F to be inputs and the vertex D to be the output. (See [KM, §6.3]) As with scalar multiplication, we extrude the linkage to three dimensions and offset the input and outputs vertically. (See Figure 10, right.)

We can simulate the operation $x, y \mapsto x + y$ by postcomposition with $x \mapsto 2x$. Multiplication by 2 is defined on the domain $D := \{x, y \in (-N, N) \mid \frac{1}{2}(x + y) \in (-N/4, N/4)\}$. The largest square

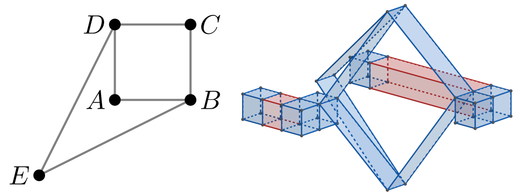

FIGURE 11. Left: a Peaucellier inversor. Right: a computational linkage for inversion.

domain containing $(0,0)$ is $(-N/4, N/4) \times (-N/4, N/4) \subset D$, which gives the simulation of the function $(-N/4, N/4) \times (-N/4, N/4) \to (-N, N), (x, y) \mapsto x + y$. $\square$

Remark: This proves the stronger statement that we can simulate $D \to (-N, N), (x, y) \mapsto x + y$. However, we want a definition of the operation whose domain is not dependent on its output, and the choice is inconsequential because we will later choose N large enough to neglect this potential inefficiency.

4.6. **Inversion.**

Lemma 4.5. *The function* $(\frac{1}{N}, N) \to (-N, N), x \mapsto \frac{1}{x}$ *can be simulated on computational linkages.*

Proof. Consider the Peaucellier inversor which is used for inversion in planar linkages. (See Figure 11, left.) The vertex E is fixed at the origin. The vertex C is used as an input vertex and the vertex A is used as an output. From the geometry of the construction, we see that $|EA| = \frac{t^2}{|EC|}$ where $t^2 = |DE|^2 - |DA|^2$. We choose edge lengths such that $t^2 = 1$. Note that the classical Peaucillier inversor is subject to degenerate configurations when the square $ABCD$ collapses. (See Kapovich and Milson's use of a 'hook' to prevent this in [KM, §6.4].) In our case, however, restricting to the embedded realizations prevents this collapse, so no special treatment is needed. (See also §6.2.)

The edges extruded from the vertices A and C are offset vertically to attach to computational linkages. (See Figure 11, right.) Choosing appropriate edge lengths, we can simulate the function $x \mapsto \frac{1}{x}$ on the domain $(\frac{1}{N}, N)$, where the bounds come from the possible values representable on computational linkages. $\square$

4.7. **Multiplication.** To simplify notation, let $\widetilde{N} = \frac{1}{2}\left(\sqrt{N} - \sqrt{1 + \frac{1}{N}}\right)$ for the rest of this section.

Lemma 4.6. *The function* $(-\widetilde{N}, \widetilde{N}) \to (-N, N), x \mapsto x^2$ *can be simulated on compuational linkages.*

Proof. We use the identity

$$\frac{1}{2}\left(\frac{1}{x-1} - \frac{1}{x+1}\right) = \frac{1}{x^2-1}$$

Thus we can simulate the function $x \mapsto x^2$ by composing with scalar addition, inversion, negation, and addition. The domain is only restricted by the scalar addition and inversion. Note that we only use half-addition, $x, y \mapsto \frac{1}{2}(x + y)$, which is defined on the domain $(-N, N) \times (-N, N)$. Also because $(\frac{1}{N}, N) \to (-N, N), x \mapsto \frac{1}{x}$ produces only positive outputs we can use inversion defined on the domain $(0, N)$. Thus no restrictions are imposed by addition and negation.

In total, we can simulate the function $\left(\sqrt{1 + \frac{1}{N}}, \sqrt{N}\right) \to (-N, N),\ x \mapsto x^2$. To adjust this domain to include 0, we can precompose by $x \mapsto x + \mu$ and postcompose with $y \mapsto y - 2x\mu - \mu^2$, where x is

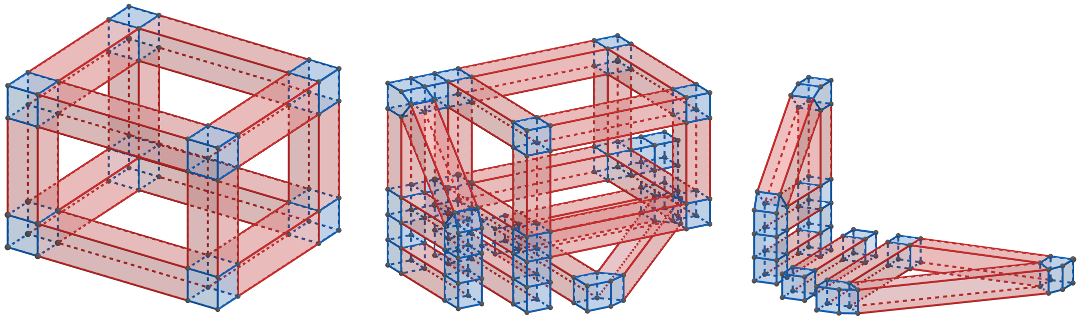

FIGURE 12. Left: an embedded polyhedral linkage with a 3-dimensional flexion. Middle/Right: a mechanism for translating 3 dimensional motion into computational linkages.

the original input to the function. The optimal μ is the average of the endpoints, giving the largest domain containing 0 as $(-\widetilde{N}, \widetilde{N})$. With this value of μ, no additional restrictions come from the pre or postcomposition. $\square$

Lemma 4.7. *The function* $(-\frac{1}{2}\widetilde{N}, \frac{1}{2}\widetilde{N}) \times (-\frac{1}{2}\widetilde{N}, \frac{1}{2}\widetilde{N}) \to (-N, N), (x, y) \mapsto xy$ *can be simulated on compuational linkages.*

Proof. We use the identity
$$\frac{1}{2}\left((x+y)^2 - (x-y)^2\right) = xy$$
Thus we can simulate the function $x, y \mapsto xy$ by composing addition, negation, and squaring. The composition of adding x and y and then squaring adds a new restriction on the domain, so we require that $x, y \in (-\frac{1}{2}\widetilde{N}, \frac{1}{2}\widetilde{N})$. The other operations do not impose any new restrictions, where again we note that in the last step we use half-addition, $x, y \mapsto \frac{1}{2}(x+y)$, which is defined everywhere. $\square$

4.8. All polynomials can be simulated on computational linkages.

Theorem 4.8. *Let* $f : \mathbb{R}^n \to \mathbb{R}^m$ *be a polynomial function and* $U \subset \mathbb{R}^n$ *be a bounded set. Then for some* N*, we can simulate* (f, U) *on computational linkages.*

Proof. A polynomial function comes from the composition of scalar addition, scalar multiplication, negation, addition and multiplication. Each operation can be simulated on computational linkages, and for each operation, the domain is defined on some neighborhood of 0 whose size only depends on N. Thus we can choose N large enough that each operation is always defined on every input from U. $\square$

5. Vector computation

For 3 dimensional motion, we consider the 1-skeleton of a cube made of twelve extender linkages joined by eight rigid cubes. (See Figure 12, left.) If one cube is fixed, the opposite cube has a 3-dimensional range of motion. Moreover, note that if this cube records the position (x, y, z), then the three cubes adjacent to the fixed cube record the coordinates $(x, 0, 0)$, $(0, y, 0)$ and $(0, 0, z)$.

By adding two $45° - 45° - 90°$ triangle linkage, we can transfer 3 dimensional motion to an array of three computational linkages. By offsetting the linkages, this construction can be embedded. (See Figure 12 middle, right.)

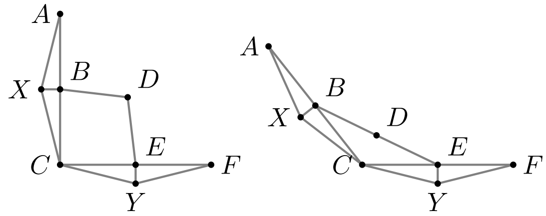

FIGURE 13. A skew pantograph that achieves a bounded flexible angle.

Conversly, we can also take three computational linkages C_1, C_2, C_3 and simulate three dimensional motion with a vertex aligned at $(|C_1|, |C_2|, |C_3|)$. This allows us to generalize the result given in 1 dimension on computational linkages.

Theorem 1.1. *Let $F : \mathbb{R}^{3m_1} \to \mathbb{R}^{3m_2}$ be a polynomial function, and $U \subset \mathbb{R}^{3m_1}$ be a bounded open set. There exists an embedded polyhedral linkage which realizes F on U.*

As a corollary, we can also restrict a polyhedral linkage to trace out a semialgebraic set.

Theorem 1.2. *Let $V \subset \mathbb{R}^3$ be an algebraic set, and $U \subset \mathbb{R}^3$ be a bounded subset. There there is a fixed embedded polyhedral linkage which realizes $V \cap U$.*

Proof. An algebraic set V is defined as the vanishing locus of a set of polynomials $f_1, \ldots, f_n : \mathbb{R}^3 \to \mathbb{R}$. Let F be the function $\mathbb{R}^3 \to \mathbb{R}^{3n}$ defined by

$$F : (x, y, z) \mapsto \Big((f_1(x, y, z), 0, 0), \ldots, (f_n(x, y, z), 0, 0) \Big)$$

In this linkage we fix all of the output vertices to be at their relative origins $(0, 0, 0)$, thus for each f_i, we have the constraint that $f_i(x, y, z) = 0$. Thus the input vertex is constrained to be in the intersection $V \cap U$, completing the proof. □

6. FINAL REMARKS

6.1. **Higher dimensions.** Note that our proofs for Theorem 1.1 and Theorem 1.2 naturally generalize to all dimensions $n > 3$. By extruding a polyhedral linkage into higher dimensions, no added flexibility is gained. Thus by extruding our construction of extender linkages, we can generalize the results of Theorem 3.1 into higher dimensions. Moreover, all of our techniques for performing scalar computation will carry over as well. And finally we can modify our construction of translating 3-dimensional motion into computational linkages by employing the 1-skeleton of an n-cube made of extender linkages to translate n-dimensional motion into a register of n extender linkages, as in Figure 12. Thus our results hold for embedded polyhedral linkages in dimension $n \geqslant 3$.

6.2. **Embedded linkages.** In § 2.1, we chose the convention to only consider the subset of embedded realizations so that the resulting linkages could be realized as physical mechanisms. An alternate approach is to constrain the linkages so that only embedded realizations are possible. Consider a rigified pantograph, with $|BD| = |DE| < |CB| = |CE|$. (See Figure 13.)

If $t = \frac{|BD|}{|BC|}$, then the angle $\angle ACF$ can open to a maximum measure of $2\sin^{-1}(t) < \pi$. Thus another way to prevent degenerate realizations of the square planar linkage is to replace each corner of the square with a copy of the skew pantograph. Then no degenerate realization is possible because every angle of the original square is strictly less than 2π. Extruding to a 3-dimensional polyhedral linkage gives a linkage whose entire configuration space is embedded. So one could provide an alternate proof of Theorem 3.1 based on this construction and produce an extender linkage whose entire realization space is embedded.

References

[Abb] T. Abbott, Generalizations of Kempe's universality theorem, Masters thesis, Massachusetts Institute of Technology, 2008, 86pp.; https://web.mit.edu/tabbott/www/papers/mthesis.pdf.

[King] H. King, Planar linkages and algebraic sets, in Proceedings of 6th Gökova Geometry-Topology Conference, *Turkish J. Math.* **23** (1999), pp. 33–56.

[Kem] A. Kempe, On a general method of describing plane curves of the nth degree by linkwork, *Proceedings of the London Mathematical Society* (1875).

[KM] M. Kapovich and J. Milson, Universality theorems for configuration spaces of planar linkages, *Topology* **41** (2002), 1051–1107; arXiv: math.AG/9803150.

[Gol] M. Goldberg, Polyhedral linkages, *National Mathematics Magazine* **16** (1942), 323-332.

[Sar] P. Sarrus, Note sur la transformation des mouvements rectilignes alternatifs, en mouvements circulaires, et reciproquement, *Académie des Sciences* **36** (1853), 1036-1038.

[WKA] K. Waldron, G. Kinzel and S. Agrawal, Kinematics, dynamics, and design of machinery, John Wiley & Sons, West Sussex, United Kingdom, (2016), 361-392.