



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS MATEMÁTICAS Y
DE LA ESPECIALIZACIÓN EN ESTADÍSTICA APLICADA

IMPLEMENTACIÓN DEL ÁLGEBRA GEOMÉTRICA
CONFORME EN *MATHEMATICA* Y SU APLICACIÓN
AL PROBLEMA DE LA GEOMETRÍA DE DISTANCIAS.

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
MAESTRA EN CIENCIAS

PRESENTA:

ELIZABETH ALEJANDRA ORTIZ DURÁN

DIRECTOR DE TESIS:

DR. JOSÉ LUIS ARAGÓN VERA
CENTRO DE FÍSICA APLICADA Y TECNOLOGÍA AVANZADA (CFATA)

JURIQUILLA, QUERÉTARO, MAYO DEL 2023



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Agradezco cada guía y oportunidad que a lo largo de mi vida me ha permitido llegar hasta este día, y quiero aprovechar estas líneas para agradecer a las personas que han marcado esta etapa personal y académica.

Primero, quisiera agradecer a los académicos que fungieron como profesores, quienes con la enseñanza de sus valiosos conocimientos hicieron que creciera día a día como profesional. En especial, le agradezco infinitamente al Dr. José Luis Aragón Vera, mi tutor y director de tesis, por permitirme trabajar a su lado, por compartir su tiempo y conocimiento, por todo el apoyo brindado, por sus consejos y orientación determinantes para mi formación académica y profesional. Asimismo, le agradezco mucho por guiar el desarrollo de este trabajo y por ser un gran ejemplo de desarrollo profesional a seguir.

Agradezco también al Dr. Carlile Lavor por sus correcciones y valiosas aportaciones que influyeron en gran medida en la conclusión de esta tesis. Asimismo, agradezco al comité revisor por el tiempo y la dedicación que empeñaron en la lectura, revisión y corrección de este trabajo.

A mi familia, les agradezco por sus oraciones, apoyo y acompañamiento en mis sueños y metas. En forma especial, agradezco infinitamente a mis padres Elizabeth y Mario y a mi hermano Erick, quienes con su amor, cuidado, apoyo y esfuerzo me han permitido llegar aquí.

Por último, y desde luego no menos importante, me gustaría agradecer a mi confidente y compañero incondicional Daniel, por compartir y coincidir conmigo todos estos años en el gran entusiasmo, motivación y deseo de seguir aprendiendo, y por su incondicional respaldo, orientación y cariño a lo largo de estos años.

Querétaro, Mayo 2023

Esta tesis fue realizada con el apoyo CONACYT Becas Nacionales.

Tabla de contenidos

1	Resumen	2
1.1	Resumen	2
1.2	Objetivo de la tesis	3
1.3	Esquema general	3
2	Introducción	4
2.1	Introducción	4
2.2	Antecedentes	5
2.3	Estado del arte	6
2.4	Motivación	6
2.5	Metodología	7
3	Álgebra Geométrica Conforme	8
3.1	El espacio $\mathbb{R}^{p,q}$	8
3.2	Álgebra de Hamilton	9
3.3	Álgebra de Grassman	10
3.3.1	Subespacios	11
3.3.2	El pseudoescalar	12
3.3.3	Operando con el producto exterior en \mathbb{R}^3	12
3.3.4	Multivectores en \mathbb{R}^3	13
3.3.5	Contracción	13
3.3.6	Norma de un k -blade	17

3.3.7	Dualidad	18
3.3.8	Representaciones directa y dual	18
3.4	El Álgebra de Clifford	20
3.4.1	Multivectores	21
3.4.2	Paridad de Multivectores	21
3.4.3	El Álgebra de Grassman en el Álgebra de Clifford y el producto externo	22
3.4.4	El producto interno	23
3.4.5	El producto geométrico	24
3.4.6	Norma de un multivector	25
3.4.7	El inverso de un multivector	25
3.4.8	Espacios con signatura (p, q)	25
3.5	Álgebra Geométrica Conforme.	26
3.5.1	Producto geométrico	28
3.5.2	Producto interno	28
3.5.3	Producto Exterior	29
3.5.4	Objetos geométricos y sus transformaciones	31
4	Implementación en <i>Mathematica</i>	32
4.1	Programación en <i>Mathematica</i>.	32
4.2	Programación basada en reglas.	32
4.3	Paquete CGAlgebra	33
4.3.1	Esquema general y metodología.	33
4.3.2	Funciones internas auxiliares.	35
4.3.3	Funciones de apoyo	36
4.3.4	Funciones (internas) para el álgebra $C^{4,1}$	37
4.3.5	Funciones del Álgebra Geométrica Conforme	38
4.3.6	Funciones gráficas.	41
5	Aplicaciones de CGAlgebra	43
5.1	Empezando con CGAlgebra	43
5.1.1	Producto Geométrico	44
5.1.2	Producto interno y externo	44
5.2	Representación de objetos geométricos	46
5.2.1	Representación de puntos	46
5.2.2	Representación directas de líneas	48
5.2.3	Representación directa de planos	50
5.2.4	Representación dual de planos	52
5.2.5	Representación dual de líneas	55
5.2.6	Representación dual de puntos	56
5.2.7	Representación directa de esferas	56
5.2.8	Representación dual de esferas	58
5.2.9	Representación directa de círculos	59
5.2.10	Representación directa de un par de puntos	61
5.2.11	Representación dual de círculos	61

5.2.12	Representación dual de pares de puntos	62
5.3	Transformaciones conformes	62
5.3.1	Involución	62
5.3.2	Traslaciones	63
5.3.3	Rotaciones	64
5.3.4	Inversión por una esfera	65
5.3.5	Resumen	65
5.4	El problema de la geometría de distancias.	66
5.4.1	Planteamiento del problema DMDGP	67
5.4.2	Intersección de esferas	69
5.4.3	Cáscaras esféricas	75
6	Conclusiones	82
6.1	Resultados	83
6.2	Discusión	83
6.3	En perspectiva	84

1. Resumen

1.1 Resumen

En este trabajo se implementa el Álgebra Geométrica Conforme en el sistema *Wolfram Mathematica*, usando el paradigma de programación basado en reglas. Como resultado, se ofrece una herramienta para realizar cálculos en esta álgebra y, para ejemplificar su uso, se presenta la manipulación de diversos objetos geométricos junto con su visualización interactiva. Asimismo, se presenta una aplicación al problema de la geometría de distancias.

Abstract

In this work conformal geometric algebra is implemented in the system *Wolfram Mathematica* using a rule-based programming paradigm. As a result, a tool is offered to perform calculations in this algebra, and to exemplify its use, the manipulation of various geometric objects is presented along with their interactive visualization. Additionally, an application to the problem of distance geometry is presented.

Resumo

Neste artigo a Álgebra Geométrica Conforme é implementada no sistema *Wolfram Mathematica*, usando o paradigma de programação baseado em regras. Como resultado, é oferecida uma ferramenta para realizar cálculos nesta álgebra e, para exemplificar seu uso, é apresentada a manipulação de diversos objetos geométricos juntamente com sua visualização interativa. Da mesma forma, é apresentada uma aplicação ao problema de geometria de distâncias.

Palabras clave

Álgebra Conforme, Álgebra Geométrica Conforme, Álgebra de Clifford, Álgebra de Grassman, Álgebra Geométrica, Producto Externo, Producto Interno, Producto Geométrico, Espacio Conforme, Geometría

Conforme, Intersección de esferas, Intersección de esferas y planos, Geometría de Distancias, Representación Dual, Representación Directa, Inversor, Reflector, Dilator, Versor, Transformación Conforme, Wolfram Mathematica, Wolfram Language, CGAlgebra.

1.2 Objetivo de la tesis

Desarrollo de un paquete para el sistema *Wolfram Mathematica*, que permitirá realizar cálculos en el Álgebra Geométrica Conforme, incluyendo funciones para la visualización de objetos geométricos y su aplicación al problema de la geometría de distancias.

1.3 Esquema general

En el Capítulo 2 se presenta una introducción general al álgebra de Clifford, también conocida como álgebra geométrica, destacando su importancia en varias áreas de la ciencia y la ingeniería, así como sus aplicaciones. En el capítulo 3 se abordan conceptos matemáticos y definiciones que serán relevantes para el álgebra geométrica y el Álgebra Geométrica Conforme; esta última se refiere al modelo conforme en 5 dimensiones, desarrollado con el propósito de representar y manipular objetos geométricos tridimensionales por medio de identidades algebraicas, equipado con el producto geométrico también conocido como producto de Clifford.

Posteriormente, en el Capítulo 4, se presenta la implementación del Álgebra Geométrica Conforme en *Wolfram Mathematica* usando el paradigma de programación basada en reglas. Mas adelante, en el capítulo 5, se ilustran y ejemplifican los aspectos matemáticos fundamentales para la representación de objetos geométricos, su manipulación y sus transformación en el Álgebra Geométrica Conforme. Además, se describe el problema de la geometría de distancias, y se explica cómo se reduce a un problema de intersección de esferas y cómo se aplica el Álgebra Geométrica Conforme a este problema. Adicionalmente, se muestran algunos ejemplos de la aplicación del paquete a transformaciones conformes en \mathbb{R}^3 entre las que se incluyen traslaciones, rotaciones, reflexiones y dilataciones.

Finalmente, en el capítulo 6, se presentan resultados, discusión y conclusiones.

2. Introducción

En este capítulo se presenta una introducción general e histórica al álgebra geométrica, también conocida como álgebra de Clifford. Enfocándose en sus antecedentes, importancia, y posibles aplicaciones.

2.1 Introducción

William K. Clifford (1845-1879) fue un matemático inglés que realizó una notable contribución al combinar el álgebra de cuaternios de Hamilton y el producto exterior de Grassmann para desarrollar una álgebra de vectores que incluía un nuevo producto: el producto geométrico, posteriormente denominado como producto de Clifford.

De forma antagónica, Josiah W. Gibbs, un físico estadounidense, simplificó las álgebras de Hamilton y Grassmann hasta su forma más elemental para describir aspectos físicos. En su enfoque, redujo las operaciones vectoriales básicas al producto interno, producto vectorial y a los productos triples escalares. Mientras tanto de forma simultánea e independiente, Heaviside decidió “divorciar” los escalares de los vectores y redefinió el producto escalar, cambiando el signo de acuerdo al producto escalar cuaterniónico original, dejándolo tal como se conoce hoy en día. Asimismo, dejó de utilizar el producto cuaterniónico para usar solamente productos de un escalar por otro, productos de escalares por vectores, y productos escalares y vectoriales de pares de vectores.

Debido a que estas formulaciones, propuestas independientemente por Oliver Heaviside y por Josiah Willard Gibbs, son muy fáciles de entender y cubren la mayoría de las relaciones geométricas en \mathbb{R}^3 predominaron sobre la de Clifford y es lo que se enseña hoy en física e ingeniería bajo el nombre de "cálculo vectorial".

Debido al éxito del cálculo vectorial, en el siglo XX, el álgebra de Clifford fue en gran medida olvidada [1]. Sin embargo, a finales del siglo XX principios del siglo XXI, comenzó una reevaluación y resurgimiento de las viejas álgebras. Una figura central entre los científicos en este proceso fue David Hestenes, un físico

estadounidense que retomó el “álgebra geométrica”, nombre que Clifford originalmente le dio a su álgebra, y la difundió activamente [1]. Si bien, algunos matemáticos anteriores a Hestenes la llamaron "álgebra de Clifford" en honor de William Clifford, Hestenes prefirió llamarla "álgebra geométrica" para enfatizar sus aspectos de aplicación [1].

El álgebra geométrica propuesta por Hestenes es un redescubrimiento del álgebra de Clifford. Este redescubrimiento tuvo un gran impacto en diferentes ramas de la física y generó diferentes aplicaciones en diferentes ramas de la ciencia, como la ingeniería, la robótica, los gráficos por computadora y la visión por computadora. Sin embargo, este progreso no habría sido posible sin el desarrollo de herramientas de software que permitieran la ejecución automática de las operaciones involucradas en esa álgebra.

2.2 Antecedentes

El álgebra es un campo de estudio que se ocupa de las operaciones y manipulaciones de símbolos. Sus orígenes se remontan a la antigüedad, cuando se utilizaba para resolver ecuaciones algebraicas. Posteriormente en el siglo XIX, se desarrollaron técnicas para describir la geometría en términos algebraicos y nacieron álgebras más especializadas [1]

Una de estas álgebras especializadas fue el "Álgebra de Clifford", introducida por William K. Clifford (1845-1879) en un artículo titulado "Sobre la clasificación de álgebras geométricas" [2]. Clifford se dio cuenta de que el álgebra exterior de Grassmann y los cuaterniones de Hamilton pueden incorporarse a la misma álgebra mediante una ligera modificación del producto exterior. Con este nuevo producto, conocido como el producto geométrico, se pueden derivar directamente las reglas de multiplicación de los cuaterniones a partir de combinaciones de vectores base, manteniendo la estructura del álgebra exterior de Grassmann. Además, con estas definiciones los números complejos y las matrices de Pauli, que son fundamentales en la mecánica cuántica, tienen también una representación natural en el álgebra de Clifford. A pesar de estas ventajas, durante la mayor parte del siglo XX, el análisis vectorial desarrollado por Gibbs y Heaviside fue dominante en el ámbito de las matemáticas y la física [3].

Recientemente, el álgebra geométrica se ha abierto camino en muchas áreas de la ciencia, principalmente gracias al esfuerzo de David Hestenes, quien tuvo el propósito de encontrar un lenguaje unificado para las matemáticas, y mostró las ventajas que podría obtenerse mediante el uso de álgebra geométrica en muchas áreas de la Física y la Geometría [4, 5]. Muchos otros investigadores siguieron sus pasos y demostraron que la aplicación del álgebra geométrica en su campo de investigación puede ser ventajoso [3].

Una generalización muy importante del Álgebra Geométrica fue propuesta por David Hestenes, quien propuso una extensión de la geometría proyectiva en 4 dimensiones, dando lugar al Álgebra Geométrica Conforme [6, 7], que pronto se convirtió en una herramienta muy poderosa para cálculos en geometría Euclidea. El Álgebra Geométrica Conforme se basa en el modelo conforme de 5 dimensiones. Este espacio de 5 dimensiones tiene la métrica de Minkowski, es decir, se trata de un espacio hiperbólico, y por esa razón al Álgebra Geométrica Conforme se le denota como $\mathbb{G}^{4,1}$.

El álgebra $\mathbb{G}^{4,1}$ desarrollada por Hestenes, tiene la ventaja de que las isometrías en tres dimensiones (rotaciones, reflexiones y traslaciones), pueden representarse como transformaciones ortogonales, además de que las primitivas geométricas como son: puntos, líneas, planos, círculos y esferas, pueden representarse

mediante identidades algebraicas simples y pueden ser manipuladas de forma sencilla, sin necesidad de especificar sus coordenadas, obteniendo una correspondencia inmediata entre objetos algebraicos y objetos geométricos con un gran nivel de abstracción [1]. Gracias a su capacidad para representar y operar sobre objetos geométricos de forma abstracta, el Álgebra Geométrica Conforme se ha convertido en una herramienta muy poderosa para la manipulación de objetos geométricos, con aplicaciones en diversas áreas [1, 3, 8, 9].

2.3 Estado del arte

Durante la última década, el Álgebra Geométrica (AG) se ha vuelto cada vez más popular, especialmente en problemas relacionados con la geometría, y ha encontrado aplicaciones en campos científicos como la robótica, la dinámica, los gráficos por computadora y la visión por computadora. Más recientemente, los desarrolladores de videojuegos se están adentrando en el Álgebra Geométrica, en busca de formas más simples y rápidas de describir y crear algoritmos para diversos gráficos y simulaciones físicas [1, 10].

Sin embargo, en contraste con el cálculo vectorial, el álgebra geométrica parece ser difícil de entender e inaccesible para muchos. Esto se debe principalmente a la cantidad de operaciones involucradas: el producto interno, el producto externo, la contracción, el producto geométrico y sus combinaciones, lo que da lugar a multitud de fórmulas, que son, aunque elegantes en apariencia, difíciles de recordar y no muy intuitivas. Como resultado, los libros de texto del álgebra geométrica se ven como una lista de fórmulas.

Para superar este problema, en la actualidad se ofrecen diferentes herramientas de *software*, algunas de las más utilizadas son CLUCalc [11] y GAViewer[12]. Actualmente, existe una gran variedad de ellas [13], para diversos lenguajes de programación. En estas herramientas de *software* los usuarios solo necesitan ingresar los datos requeridos y especificar la relación geométrica que quieren calcular, y estas herramientas ejecutan los cálculos de acuerdo con las reglas de operación del álgebra de Clifford [1]. La mayoría de los desarrolladores de estas herramientas hacen uso de lenguajes de programación, como C, C++, OpenCL o CUDA, que son lo suficientemente eficientes y abstractos para la mayoría de las necesidades del Álgebra Geométrica. Estas herramientas y sus tutoriales asociados fueron muy bien recibidos por la comunidad, y generaron que mucha gente de la comunidad gráfica se interesara en usar el álgebra geométrica [3].

Con respecto al Álgebra Geométrica Conforme, dada la importancia que ha adquirido, se han desarrollado diferentes plataformas para realizar cálculos en esta álgebra, entre las que destacan *Galoop* [14] y *CLUCalc*, un *toolbox* de *Matlab* llamado *GABLE*, y un paquete en *Mathematica* orientado al doblamiento de figuras de origami [15]. Este último, es una herramienta muy relacionada con el presente trabajo, pero hay que mencionar que no es de propósito general, sino que esta orientado a un problema particular.

2.4 Motivación

El Álgebra Geométrica Conforme, en la forma propuesta por David Hestenes [6, 7], actúa como una poderosa herramienta para la geometría Euclideana computacional. En esta álgebra de 5 dimensiones, los objetos geométricos tridimensionales, como líneas, planos, círculos y esferas, son representados por identidades algebraicas sencillas. Además, esta álgebra permite formulaciones simples, compactas, sin

coordenadas y ampliables a más dimensiones.

El Álgebra Geométrica Conforme proporciona un marco algebraico unificado y versátil que permite aprovechar las propiedades y operaciones algebraicas para abordar y resolver problemas geométricos de forma elegante y efectiva. Su aplicación se extiende a diversas disciplinas, destacando especialmente en áreas como la gráfica por computadora, la visión por computadora y la robótica.

El problema es que los cálculos en esta álgebra suelen ser tediosos y largos, razón por la cual el apoyo de herramientas gráficas y computaciones es importante para su aplicación. Actualmente se han desarrollado plataformas computacionales para facilitar los cálculos y el despliegue de gráficos. En este trabajo se busca estudiar, desarrollar y ampliar un paquete con estas características en el lenguaje de programación *Wolfram Language* para el programa *Wolfram Mathematica*, donde a diferencia de las otras plataformas existentes, se pueden realizar cálculos simbólicos.

Ya que el álgebra geométrica promete estimular nuevos métodos y conocimientos en todas las áreas de la ciencia que se ocupan de las propiedades geométricas, poder realizar estos cálculos de forma simbólica permitiría mantener su intuición geométrica. Esto nos convenció de que un paquete en el que se implementaran las ideas fundamentales del Álgebra Geométrica Conforme, usando el lenguaje de programación *Wolfram Language* sería muy útil para la comunidad.

2.5 Metodología

Para el desarrollo del paquete se usará el lenguaje de programación *Wolfram Language*, un lenguaje de programación que unifica varios paradigmas de la programación, como la programación procedural, simbólica, basada en reglas y funcional. En particular, para el presente trabajo se usará la programación basada en reglas, inspirados en la construcción del álgebra geométrica propuesta por A. Macdonald [16].

3. Álgebra Geométrica Conforme

En este capítulo se presentan conceptos y definiciones matemáticas fundamentales que servirán de base para introducir el Álgebra Geométrica Conforme.

3.1 El espacio $\mathbb{R}^{p,q}$

Las álgebras de Clifford, que constituyen el fundamento de este trabajo, pueden construirse de manera general sobre espacios vectoriales reales equipados con formas bilineales simétricas no degeneradas [17]. A pesar de que para los propósitos de este trabajo es suficiente considerar el espacio vectorial real de vectores, o más formalmente, arreglos ordenados de escalares (a_1, a_2, \dots, a_n) , equipado con el producto interno estándar o con una generalización que no es positiva definida, en esta sección se presenta al espacio $\mathbb{R}^{p,q}$ de manera general.

Definición 3.1.1 En un espacio vectorial real V , el mapeo $\mathcal{B} : V \times V \rightarrow \mathbb{R}$, tal que para todo $a, b, c \in V$, y $\alpha \in \mathbb{R}$, tiene las propiedades:

- $\mathcal{B}(\alpha a + b, c) = \alpha \mathcal{B}(a, c) + \mathcal{B}(b, c)$, y $\mathcal{B}(a, \alpha b + c) = \alpha \mathcal{B}(a, b) + \mathcal{B}(a, c)$,
- $\mathcal{B}(a, b) = \mathcal{B}(b, a)$,
- Para todo vector no nulo $a \in V$, existe $b \in V$, tal que $\mathcal{B}(a, b) \neq 0$,

A esto se le llama, forma bilineal (a), simétrica (b) y no degenerada (c). Además, a la dupla V, \mathcal{B} se le conoce como espacio con producto interno generalizado y, por definición, si $\mathcal{B}(a, b) = 0$, se dice que los vectores a y b son ortogonales.

Es inmediato probar que el producto interno estándar cumple con las condiciones de la Definición 3.1.1; si $a = (a_1, a_2, \dots, a_n)$ y $b = (b_1, b_2, \dots, b_n)$ son vectores en \mathbb{R}^n . Entonces,

$$\mathcal{B} = a \cdot b = \sum_{i=1}^n a_i b_i \quad (3.1)$$

es una forma bilineal que es, además, positiva definida. Al espacio \mathbb{R}^n equipado con un producto interno estándar 3.1 se conoce también como espacio Euclidiano de dimensión n .

La siguiente proposición es importante para la construcción del modelo conforme (para una demostración ver Ref. [18], Proposición 159.1).

Proposición 3.1.1 Sea \mathbb{R}^n un espacio vectorial equipado con un producto interno. Entonces existe una base ordenada (e_1, e_2, \dots, e_n) de \mathbb{R}^n , y enteros positivos p, q , tales que $p + q = n$, tales que:

- $\mathcal{B}(e_i, e_i) = 1$, para $i = 1, 2, \dots, p$.
- $\mathcal{B}(e_i, e_i) = -1$, para $i = p + 1, p + 2, \dots, p + q$.
- $\mathcal{B}(e_i \cdot e_j) = 0$, para $i \neq j$.

En este caso, la base (e_1, e_2, \dots, e_n) es una base ortogonal. Si $a = (a_1, a_2, \dots, a_n)$ y $a = (b_1, b_2, \dots, b_n)$ son vectores en \mathbb{R}^n , entonces, si p y q son enteros tales que $p + q = n$, la forma bilineal sobre \mathbb{R}^n , definida como

$$\mathcal{B}^* = a \cdot b = \sum_{i=1}^p a_i b_i - \sum_{i=p+1}^{p+q} a_i b_i, \quad (3.2)$$

es simétrica y no degenerada. La dupla $\mathbb{R}^n, \mathcal{B}$, se denota como $\mathbb{R}^{p,q}$.

Cualquier espacio con producto interno generalizado que satisface las condiciones 1, 2 y 3, de la Definición 3.1.1 es isomorfo a $\mathbb{R}^{p,q}$, y tal espacio se dice que tiene *característica* o *signatura* (p, q) . La importancia de estos espacios será visible más adelante porque el espacio conforme de 5 dimensiones es isomorfo a $\mathbb{R}^{4,1}$.

3.2 Álgebra de Hamilton

El matemático irlandés Sir William Rowan Hamilton (1805-1865) descubrió que los números complejos pueden ser extendidos introduciendo unidades imaginarias adicionales, j y k , lo que lleva a la aparición de los objetos matemáticos conocidos como cuaterniones [1].

Definición 3.2.1 El álgebra de Hamilton es un álgebra generada a partir del 1 y los símbolos i, j y k por una operación de multiplicación (producto de cuaternios) que es asociativa y está sujeta a las siguientes reglas:

$$\begin{aligned} i^2 = j^2 = k^2 &= -1, \\ jk = i, \quad ki = j, \quad ij = k, \\ kj = -i, \quad ik = -j, \quad ji = -k. \end{aligned}$$

El álgebra de Hamilton, además, contiene el conjunto de números complejos \mathbb{C} , al considerar únicamente el subespacio generado por 1 e i .

Definición 3.2.2 El conjunto de los números complejos es un álgebra generada a partir de 1 y el símbolo

i por una operación de multiplicación asociativa, sujeta a la regla;

$$i^2 = -1$$

3.3 Álgebra de Grassman

El álgebra de Grassmann fue introducida por el matemático alemán Hermann Günther Grassmann (1809–1877). En esta álgebra, se incluye una operación conocida como producto exterior \wedge , que resultó ser de fundamental importancia para definir el álgebra de Clifford. A partir de los trabajos de Grassmann surgió el cálculo vectorial actual, el cual es una reformulación del álgebra de Grassmann simplificada en términos del productos vectorial (o producto cruz) y el triple producto escalar. Sin embargo, el cálculo vectorial tiene una restricción crucial: solo se puede usar en \mathbb{R}^3 [1]. Esta restricción no se encuentra en el álgebra de Grassman, la cual puede ser usada en dimensiones mayores.

Definición 3.3.1 El álgebra de Grassmann es el álgebra generada por la identidad 1, los símbolos e_1, \dots, e_n y la operación \wedge , llamada producto exterior o producto de Grassman, que es una operación asociativa, distributiva y que cumple lo siguiente:

$$e_i \wedge e_i = 0 \tag{3.3a}$$

$$e_i \wedge e_j = -e_j \wedge e_i \tag{3.3b}$$

donde $i, j = 1, \dots, n$.

Por ejemplo para $i, j = 1, 2, 3$ la tabla del producto exterior se muestra en la Tabla 3.1.

\wedge	1	e_1	e_2	e_3
1	1	e_1	e_2	e_3
e_1	e_1	0	$e_1 \wedge e_2$	$-e_3 \wedge e_1$
e_2	e_2	$-e_1 \wedge e_2$	0	$e_2 \wedge e_3$
e_3	e_3	$e_3 \wedge e_1$	$-e_2 \wedge e_3$	0

Table 3.1: Tabla de producto exterior para el álgebra de Grassmann para la identidad 1 y los símbolos e_1, e_2 y e_3 , que identificamos con vectores.

El producto externo tiene una estrecha relación con el conocido producto vectorial, pero el primero puede extenderse a mas dimensiones. Así, el producto exterior de k vectores $\{a_1, a_2, \dots, a_k\}$ linealmente independientes es:

$$A_k = a_1 \wedge a_2 \wedge \dots \wedge a_k = \wedge_{i=1}^k a_i,$$

al elemento A_k se le llama k -blade y se dice que tiene grado k . Los k -blades, o simplemente blades, son los elementos computacionales básicos y entidades geométricas básicas del álgebra geométrica o álgebra de Clifford, que se definirá más adelante. Es claro que un 1-blade es un vector y cuando el grado es mayor, se llama *bivector* a un 2-blade, y *trivector* a un 3-blade (un 0-blade es un escalar). Estos objetos tienen la interpretación geométrica de líneas orientadas, áreas orientadas y volúmenes orientados [16]. Además, a una combinación lineal de blades de diferentes grados se le llama *multivector*.

De la Definición 3.3.1 pueden inferirse las siguientes propiedades del producto exterior:

Proposición 3.3.1 Si a, b y c son vectores, y α y β escalares, el producto exterior tiene las propiedades:

- Antisimetría: $a \wedge b = -b \wedge a$. En particular, $a \wedge a = 0$.
- Distributividad (o linealidad): $a \wedge (\alpha b + \beta c) = \alpha a \wedge b + \beta a \wedge c$.
- Asociatividad: $a \wedge (b \wedge c) = (a \wedge b) \wedge c$, que escribimos $a \wedge b \wedge c$.
- Operación escalar: $\alpha \wedge \beta = \alpha\beta$, $\alpha \wedge a = a \wedge \alpha = \alpha a$.

3.3.1 Subespacios

Un *subespacio vectorial* es un subconjunto de un espacio vectorial que, a su vez, es un espacio vectorial [19]. En el caso particular de \mathbb{R}^n , un subespacio es un subconjunto generado por vectores que parten del origen. Por ejemplo, en \mathbb{R}^3 , podemos definir subespacios de varias dimensiones:

- El origen es un subespacio de dimensión 0;
- Una línea que pasa por el origen es un subespacio de dimensión 1;
- Un plano que pasa por el origen es un subespacio de dimensión 2;
- El espacio completo \mathbb{R}^3 es un subespacio de si mismo.

A estos subespacios los llamaremos simplemente origen, líneas, planos y volúmenes, respectivamente. Mostraremos cómo el álgebra de Grassman permite manipular estos subespacios, además de que les asigna una magnitud y una orientación, como se detalla enseguida [1]:

- **Líneas**

Las líneas tienen una orientación, que es la dirección del vector que la genera y su magnitud es la longitud de este vector. En adelante, a una línea será referida por el vector que la genera.

- **Planos**

Los planos también tienen una orientación y una magnitud, que se definen de la siguiente manera. Consideremos dos vectores no colineales a y b , el plano queda definido por el bivector $a \wedge b$. Definir la orientación de un bivector es equivalente a distinguir su lado frontal de su lado posterior; el lado en el que la rotación de a hacia b es en el sentido contrario al de las manecillas del reloj, se define como el lado frontal, y la magnitud se define como el área del paralelogramo definido por a y b . De manera equivalente, el lado en el que la rotación de a hacia b es en el sentido de las manecillas del reloj, se define como el lado posterior, y su magnitud se define como el negativo del área del paralelogramo definido por a y b [1].

Proposición 3.3.2 Basados en las propiedades del producto exterior, los planos tienen las siguientes propiedades:

$$(\alpha a) \wedge (\beta b) = (\alpha\beta) a \wedge b, \quad (3.4a)$$

$$a \wedge b = -b \wedge a, \quad (3.4b)$$

$$a \wedge a = 0. \quad (3.4c)$$

- **Volúmenes**

También consideramos que todo el espacio tridimensional tiene una orientación y una magnitud. Dados tres vectores no coplanares a, b y c , este espacio se define por el trivector $a \wedge b \wedge c$ y su magnitud viene dada por el volumen del paralelepípedo definido por a, b y c ; positiva si a, b y c forman un sistema derecho y negativo si forman un sistema izquierdo [1].

Proposición 3.3.3 Utilizando las propiedades del producto exterior, los volúmenes cumplen con las siguientes propiedades:

$$\begin{aligned}(\alpha a) \wedge (\beta b) \wedge (\gamma c) &= (\alpha\beta\gamma)a \wedge b \wedge c, \\ a \wedge b \wedge c &= b \wedge c \wedge a = c \wedge a \wedge b, \\ a \wedge b \wedge c &= -b \wedge a \wedge c = -c \wedge b \wedge a = -a \wedge c \wedge b, \\ a \wedge c \wedge c &= a \wedge b \wedge a = a \wedge a \wedge c = 0.\end{aligned}$$

3.3.2 El pseudoescalar

El k -blade de mayor grado que se puede establecer en un espacio, se llama *pseudoescalar* de ese espacio. Por ejemplo, en \mathbb{R}^3 , el pseudoescalar es el volumen $a \wedge b \wedge c$, y en \mathbb{R}^n , el pseudoescalar es un n -blade; si se considera una base ortonormal $\{e_1, e_2, \dots, e_n\}$, se define el *pseudoescalar unitario* como:

$$I_n = e_1 \wedge e_2 \wedge \dots \wedge e_n.$$

3.3.3 Operando con el producto exterior en \mathbb{R}^3

Como un ejemplo de cómo se opera dentro de una álgebra de Grassmann, cuando se representan los vectores en \mathbb{R}^3 en términos de la base estándar, consideremos los vectores a y b que, en términos de la base e_1, e_2, e_3 , se escriben como $a = a_1e_1 + a_2e_2 + a_3e_3$ y $b = b_1e_1 + b_2e_2 + b_3e_3$. Entonces $a \wedge b$ puede calcularse utilizando la distributividad y la antisimetría del producto exterior:

$$\begin{aligned}a \wedge b &= (a_1e_1 + a_2e_2 + a_3e_3) \wedge (b_1e_1 + b_2e_2 + b_3e_3) \\ &= a_1b_1e_1 \wedge e_1 + a_1b_2e_1 \wedge e_2 + a_1b_3e_1 \wedge e_3 + a_2b_1e_2 \wedge e_1 + a_2b_2e_2 \wedge e_2 \\ &\quad + a_2b_3e_2 \wedge e_3 + a_3b_1e_3 \wedge e_1 + a_3b_2e_3 \wedge e_2 + a_3b_3e_3 \wedge e_3.\end{aligned}$$

Que, utilizando las ecuaciones (3.3) se reduce a:

$$\begin{aligned}&= a_1b_2e_1 \wedge e_2 + a_1b_3e_1 \wedge e_3 - a_2b_1e_1 \wedge e_2 + a_2b_3e_2 \wedge e_3 - a_3b_1e_1 \wedge e_3 - a_3b_2e_2 \wedge e_3 \\ &= (a_1b_2 - a_2b_1)e_1 \wedge e_2 + (a_1b_3 - a_3b_1)e_1 \wedge e_3 + (a_2b_3 - a_3b_2)e_2 \wedge e_3.\end{aligned}$$

Es interesante observar que, dada la interpretación de $a \wedge b$ como una área orientada, la expresión anterior se interpreta como que cualquier área orientada puede escribirse como combinación lineal de las áreas unitarias "base" $e_1 \wedge e_2$, $e_1 \wedge e_3$ y $e_2 \wedge e_3$, así como cualquier vector a puede representarse como una combinación lineal de las longitudes unitarias y ortogonales e_1 , e_2 y e_3 .

De manera similar, si consideramos un tercer vector $c = c_1e_1 + c_2e_2 + c_3e_3$, el trivector $a \wedge b \wedge c$ se simplifica así:

$$\begin{aligned}a \wedge b \wedge c &= (a_1e_1 + a_2e_2 + a_3e_3) \wedge (b_1e_1 + b_2e_2 + b_3e_3) \wedge (c_1e_1 + c_2e_2 + c_3e_3) \\ &= a_1b_2c_3e_1 \wedge e_2 \wedge e_3 + a_2b_3c_1e_2 \wedge e_3 \wedge e_1 + a_3b_1c_2e_3 \wedge e_1 \wedge e_2 \\ &\quad + a_1b_3c_2e_1 \wedge e_3 \wedge e_2 + a_2b_1c_3e_2 \wedge e_1 \wedge e_3 + a_3b_2c_1e_3 \wedge e_2 \wedge e_1 \\ &= (a_1b_2c_3 + a_2b_3c_1 + a_3b_1c_2 - a_1b_3c_2 - a_2b_1c_3 - a_3b_2c_1)e_1 \wedge e_2 \wedge e_3.\end{aligned}$$

Es decir, en \mathbb{R}^3 , cualquier volumen orientado $a \wedge b \wedge c$ es un múltiplo del volumen unitario "base" $e_1 \wedge e_2 \wedge e_3$.

3.3.4 Multivectores en \mathbb{R}^3

Dado el escalar α y los vectores a, b, c, d y f en \mathbb{R}^3 , el multivector más general en este espacio es

$$A = \alpha + a + b \wedge c + d \wedge e \wedge f$$

El conjunto de todos los multivectores forma un espacio vectorial de 8 dimensiones, con respecto a la suma/resta y la multiplicación escalar. Si los vectores se expresan en la base ortonormal e_1, e_2, e_3 , un multivector se reduce, debido a la antisimetría del producto exterior, a una combinación lineal de ocho elementos básicos:

$$\{1, e_1, e_2, e_3, e_2 \wedge e_3, e_3 \wedge e_1, e_1 \wedge e_2, e_1 \wedge e_2 \wedge e_3\}$$

3.3.5 Contracción

La contracción, denotada por un punto, es una operación definida a partir del producto interno, que reduce un subespacio a un subespacio de menor dimensión. Para ser específicos, si hacemos la contracción de un k -vector, que representa un subespacio de dimensión k , por algún j -vector con $j \leq k$, generaremos un $(k - j)$ -vector que representa un subespacio de dimensión $(k - j)$.

Algunos ejemplos de contracciones de subespacios son los siguientes:

- **Contracción de un vector**

Definición 3.3.2 La contracción de un vector $a = (a_1, a_2, \dots, a_n)$, por un vector $x = (x_1, x_2, \dots, x_n)$ es

$$x \cdot a = x_1 a_1 + x_2 a_2 + \dots + x_n a_n,$$

es decir, produce un escalar.

Podemos ver que el producto interior puede verse como la contracción de un vector con un vector. Geométricamente podemos interpretar esto como calcular la intersección del vector a , con el vector x (Fig. 3.3.5).

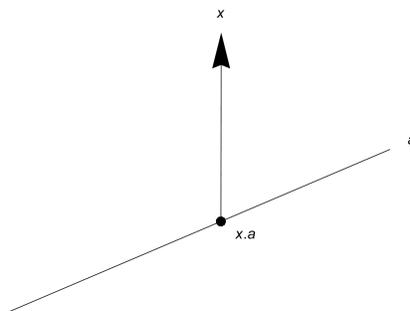


Figure 3.1: Contracción de un vector a por un vector x .

- **Contracción de un bivector**

Reducir la dimensionalidad de un bivector da como resultado un vector o un escalar:

- Contracción de un bivector por un vector.

Denotamos por $x \cdot (a \wedge b)$ al vector que se obtiene al contraer el bivector $a \wedge b$ por el vector x . En lo sucesivo, consideramos el producto exterior \wedge como una operación con mayor prioridad que la contracción y omitimos los paréntesis para escribir simplemente $x \cdot a \wedge b$ para $x \cdot (a \wedge b)$.

Definición 3.3.3 La contracción de un bivector por un vector se calcula alternando el signo:

$$x \cdot a \wedge b = (x \cdot a) \wedge b - a \wedge (x \cdot b) = (x \cdot a)b - (x \cdot b)a. \quad (3.5)$$

Dado que esta es una combinación lineal de a y b , el vector resultante está contenido en el plano $a \wedge b$ a menos que $(x \cdot a) = (x \cdot b) = 0$. Además, este vector es ortogonal a x :

$$x \cdot ((x \cdot a)b - (x \cdot b)a) = (x \cdot a)(x \cdot b) - (x \cdot b)(x \cdot a) = 0.$$

Por lo tanto, la contracción de un bivector $a \wedge b$ por el vector x puede interpretarse geométricamente como calcular la intersección del bivector $a \wedge b$ con el plano cuya normal es x (Fig. 3.3.5).

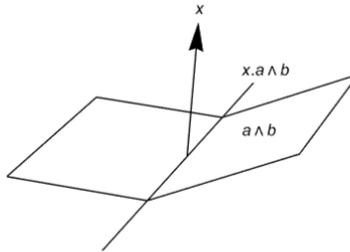


Figure 3.2: Contracción de un bivector $a \wedge b$ por un vector x .

- Contracción de un bivector por un bivector.

Denotamos por $x \wedge y \cdot a \wedge b$ al escalar obtenido al contraer el bivector $a \wedge b$ por el bivector $x \wedge y$.

Definición 3.3.4 La contracción de un bivector por un bivector se calcula de la siguiente forma:

$$x \wedge y \cdot a \wedge b = x \cdot (y \cdot a \wedge b) = x \cdot ((y \cdot a)b - (y \cdot b)a) = (y \cdot a)(x \cdot b) - (y \cdot b)(x \cdot a)$$

lo cual, es un escalar.

- **Contracción de un trivector**

Reducir la dimensionalidad de un trivector da como resultado un bivector, un vector o un escalar:

- Contracción de un trivector por un vector

Denotamos por $x \cdot a \wedge b \wedge c$ al bivector obtenido al contraer el trivector $a \wedge b \wedge c$ por el vector x .

Definición 3.3.5 La contracción de un trivector por un vector esta definida como:

$$\begin{aligned} x \cdot a \wedge b \wedge c &= (x \cdot a) \wedge b \wedge c - a \wedge (x \cdot b) \wedge c + a \wedge b \wedge (x \cdot c) \\ &= (x \cdot a)b \wedge c + (x \cdot b)c \wedge a + (x \cdot c)a \wedge b. \end{aligned} \quad (3.6)$$

La contracción de la Ec. (3.6) puede interpretarse geoméricamente como calcular la intersección del volumen $a \wedge b \wedge c$ con el plano cuya normal es x , ya que el bivector resultante es ortogonal a x . De hecho, si contraemos este bivector por el vector x , obtenemos de la Ec. (3.5):

$$(x \cdot a)((x \cdot b)c - (x \cdot c)b) + (x \cdot b)((x \cdot c)a - (x \cdot a)c) + (x \cdot c)((x \cdot a)b - (x \cdot b)a) = 0.$$

- Contracción de un trivector con un bivector

Denotamos por $x \wedge y \cdot a \wedge b \wedge c$ al vector que se obtiene contrayendo el trivector $a \wedge b \wedge c$ por el bivector $x \wedge y$.

Definición 3.3.6 La contracción del trivector por un bivector esta definida como:

$$\begin{aligned} x \wedge y \cdot a \wedge b \wedge c &= x \cdot (y \cdot a \wedge b \wedge c) + y \cdot (x \cdot a \wedge b \wedge c) \\ &= (y \cdot a)((x \cdot b)c - (x \cdot c)b) + (y \cdot b)((x \cdot c)a - (x \cdot a)c) + (y \cdot c)((x \cdot a)b - (x \cdot b)a) \\ &= ((x \cdot c)(y \cdot b) - (x \cdot b)(y \cdot c))a + ((x \cdot a)(y \cdot c) - (x \cdot c)(y \cdot a))b \\ &\quad + ((x \cdot b)(y \cdot a) - (x \cdot a)(y \cdot b))c. \end{aligned} \quad (3.7)$$

Geoméricamente, podemos interpretar la contracción de la Ec. (3.7) como calcular la intersección del volumen $a \wedge b \wedge c$ con la línea normal al plano $x \wedge y$, ya que este vector es ortogonal tanto a x como a y . De hecho, el producto interno de la Ec. (3.7) con x es:

$$\begin{aligned} ((x \cdot c)(y \cdot b) - (x \cdot b)(y \cdot c))(x \cdot a) + ((x \cdot a)(y \cdot c) - (x \cdot c)(y \cdot a))(x \cdot b) \\ + ((x \cdot b)(y \cdot a) - (x \cdot a)(y \cdot b))(x \cdot c) = 0. \end{aligned}$$

De manera similar, el producto interior con y es

$$\begin{aligned} ((x \cdot c)(y \cdot b) - (x \cdot b)(y \cdot c))(y \cdot a) + ((x \cdot a)(y \cdot c) - (x \cdot c)(y \cdot a))(y \cdot b) \\ + ((x \cdot b)(y \cdot a) - (x \cdot a)(y \cdot b))(y \cdot c) = 0. \end{aligned}$$

- Contracción de un trivector por un trivector

Denotamos por $x \wedge y \wedge z \cdot a \wedge b \wedge c$ al escalar obtenido al contraer el trivector $a \wedge b \wedge c$ por el trivector $x \wedge y \wedge z$.

Definición 3.3.7 La contracción de un trivector con un trivector esta definida como:

$$\begin{aligned}
 x \wedge y \wedge z \cdot a \wedge b \wedge c &= x \cdot (y \cdot (z \cdot a \wedge b \wedge c)) \\
 &= x \cdot ((y \cdot c)(z \cdot b) - (y \cdot b)(z \cdot c))a + \\
 &((y \cdot a)(z \cdot c) - (y \cdot c)(z \cdot a))b + ((y \cdot b)(z \cdot a) - (y \cdot a)(z \cdot b))c \\
 &= ((y \cdot c)(z \cdot b) - (y \cdot b)(z \cdot c))(x \cdot a) + ((y \cdot a)(z \cdot c) \\
 &\quad - (y \cdot c)(z \cdot a))(x \cdot b) + ((y \cdot b)(z \cdot a) - (y \cdot a)(z \cdot b))(x \cdot c) \\
 &= (x \cdot a)(y \cdot c)(z \cdot b) + (x \cdot b)(y \cdot a)(z \cdot c) + (x \cdot c)(y \cdot b)(z \cdot a) \\
 &\quad - (x \cdot a)(y \cdot b)(z \cdot c) - (x \cdot b)(y \cdot c)(z \cdot a) - (x \cdot c)(y \cdot a)(z \cdot b).
 \end{aligned}$$

Este escalar representa el origen, por lo que geoméricamente la contracción anterior se puede interpretar como calcular la intersección del espacio $a \wedge b \wedge c$ con el origen, que es ortogonal al espacio $x \wedge y \wedge z$.

En estas definiciones, podemos ver como contraer un subespacio por un k -vector, para $k = 1, 2, 3$, reduce la dimensionalidad por k .

Definición 3.3.8 Dado que ningún subespacio tiene dimensión menor que 0, la contracción de un escalar α por cualquier otro objeto que no sea un escalar es 0:

$$x \cdot \alpha = 0, \quad x \wedge y \cdot \alpha = 0, \quad x \wedge y \wedge z \cdot \alpha = 0.$$

Todos los resultados que hemos obtenido hasta el momento se resumen de la siguiente manera:

Proposición 3.3.4 Los subespacios se contraen por un k -vector, para $k = 1, 2, 3$, de la siguiente forma:

Contracción por un vector x :

$$\begin{aligned}
 x \cdot \alpha &= 0, \\
 x \cdot a &= (x \cdot a), \\
 x \cdot a \wedge b &= (x \cdot a)b - (x \cdot b)a, \\
 x \cdot a \wedge b \wedge c &= (x \cdot a)b \wedge c + (x \cdot b)c \wedge a + (x \cdot c)a \wedge b.
 \end{aligned}$$

Contracción por un bivector $x \wedge y$:

$$\begin{aligned}
 x \wedge y \cdot \alpha &= 0, \\
 x \wedge y \cdot a &= (a \cdot y)x - (a \cdot x)y, \\
 x \wedge y \cdot a \wedge b &= (x \cdot b)(y \cdot a) - (x \cdot a)(y \cdot b), \\
 x \wedge y \cdot a \wedge b \wedge c &= ((x \cdot c)(y \cdot b) - (x \cdot b)(y \cdot c))a \\
 &+ ((x \cdot a)(y \cdot c) - (x \cdot c)(y \cdot a))b + ((x \cdot b)(y \cdot a) - (x \cdot a)(y \cdot b))c.
 \end{aligned}$$

Contracción por un trivector $x \wedge y \wedge z$:

$$\begin{aligned}
 x \wedge y \wedge z \cdot \alpha &= 0, \\
 x \wedge y \wedge z \cdot a &= (a \cdot x)y \wedge z + (a \cdot y)z \wedge x + (a \cdot z)x \wedge y, \\
 x \wedge y \wedge z \cdot a \wedge b &= ((a \cdot z)(b \cdot y) - (a \cdot y)(b \cdot z))x \\
 &+ ((a \cdot x)(b \cdot z) - (a \cdot z)(b \cdot x))y + ((a \cdot y)(b \cdot x) - (a \cdot x)(b \cdot y))z, \\
 x \wedge y \wedge z \cdot a \wedge b \wedge c &= (x \cdot a)(y \cdot c)(z \cdot b) + (x \cdot b)(y \cdot a)(z \cdot c) + (x \cdot c)(y \cdot b)(z \cdot a) \\
 &- (x \cdot a)(y \cdot b)(z \cdot c) - (x \cdot b)(y \cdot c)(z \cdot a) - (x \cdot c)(y \cdot a)(z \cdot b).
 \end{aligned}$$

Proposición 3.3.5 La contracción de un subespacio por un k -vector está incluida en ese subespacio, tiene una dimensión menor por k , y es ortogonal al subespacio especificado por ese k -vector.

Proposición 3.3.6 La contracción de un subespacio por un subespacio es 0 si y solo si los dos subespacios son ortogonales:

$$(\dots) \cdot (\dots) = 0 \Leftrightarrow (\dots) \perp (\dots).$$

3.3.6 Norma de un k -blade

Definición 3.3.9 La *reversión* de un k -blade es una operación que arregla sus elementos en el orden inverso. Así, por ejemplo dado el k -blade $A_k = a_1 \wedge a_2 \wedge \dots \wedge a_k$, su reversión es:

$$\tilde{A}_k = a_k \wedge a_{k-1} \wedge \dots \wedge a_1 = (-1)^{\frac{1}{2}k(k-1)} A_k.$$

Definición 3.3.10 Definimos la norma de un k -blade, para $k = 0, 1, 2, 3$, como la contracción del k -blade con su reversión, es decir:

$$\|A_k\|^2 = A_k \cdot \tilde{A}_k.$$

De forma que:

- Si contraemos un escalar α por sí mismo, tenemos:

$$\|\alpha\|^2 = \alpha \cdot \alpha = \alpha^2.$$

Por lo tanto, la norma $\|\alpha\|$ es igual a $|\alpha|$

- Si contraemos un vector a por sí mismo, tenemos:

$$\|a\|^2 = a \cdot a,$$

lo cual coincide con la definición de la norma vectorial usual.

- Si contraemos un bivector $a \wedge b$ por su reversión $b \wedge a$, tenemos:

$$\|a \wedge b\|^2 = b \wedge a \cdot a \wedge b = (b \cdot b)(a \cdot a) - (b \cdot a)^2 = \|a\|^2 \|b\|^2 - (a \cdot b)^2.$$

Esto es igual al cuadrado del área del paralelogramo definido por a y b .

- Si contraemos un trivector $a \wedge b \wedge c$ por su reversión $c \wedge b \wedge a$, tendremos:

$$\begin{aligned} \|a \wedge b \wedge c\|^2 &= c \wedge b \wedge a \cdot a \wedge b \wedge c \\ &= (c \cdot a)(b \cdot c)(a \cdot b) + (c \cdot b)(b \cdot a)(a \cdot c) + (c \cdot c)(b \cdot b)(a \cdot a) \\ &\quad - (c \cdot a)(b \cdot b)(a \cdot c) - (c \cdot b)(b \cdot c)(a \cdot a) - (c \cdot c)(b \cdot a)(a \cdot b) \\ &= \|a\|^2 \|b\|^2 \|c\|^2 + 2(b \cdot c)(c \cdot a)(a \cdot b) - \|a\|^2 (b \cdot c)^2 - \|b\|^2 (c \cdot a)^2 - \|c\|^2 (a \cdot b)^2. \end{aligned}$$

Esto es igual al cuadrado del volumen del paralelepipedo definido por a , b y c .

Definición 3.3.11 Si definimos $a = \sum_{k=1}^3 a_k e_k$, $b = \sum_{k=1}^3 b_k e_k$ y $c = \sum_{k=1}^3 c_k e_k$ la norma será:

$$\begin{aligned} \|a\|^2 &= a_1^2 + a_2^2 + a_3^2 \\ \|a \wedge b\|^2 &= (a_2 b_3 - a_3 b_2)^2 + (a_3 b_1 - a_1 b_3)^2 + (a_1 b_2 - a_2 b_1)^2 \\ \|a \wedge b \wedge c\|^2 &= (a_1 b_2 c_3 + a_2 b_3 c_1 + a_3 b_1 c_2 - a_1 b_3 c_2 - a_2 b_1 c_3 - a_3 b_2 c_1)^2 \end{aligned}$$

3.3.7 Dualidad

Una operación importante en el Álgebra Geométrica Conforme es el *dual*, ya que proporciona una representación 'complementaria' de un objeto. En \mathbb{R}^n un k -vector especifica un subespacio de dimensión k , sin embargo, ese mismo subespacio puede ser especificado por su complemento ortogonal, un $(n - k)$ -vector conocido como el dual del k -vector, el cual se denota por un asterisco $*$.

La utilidad de tener esta otra representación del mismo objeto en el Álgebra Geométrica Conforme radica en las propiedades y aplicaciones de dicha representación dual. Al tener acceso a ambas representaciones, podemos analizar y manipular el objeto desde distintas perspectivas y realizar diversas operaciones algebraicas en cada una de ellas. Esto nos brinda la capacidad de abordar problemas geométricos de manera más flexible y eficiente, ya que ciertas operaciones pueden resultar más convenientes o intuitivas en una representación que en la otra.

Definición 3.3.12 El dual $(\dots)^*$ de un k -vector (\dots) se define como

$$(\dots)^* = -(\dots) \cdot I,$$

donde I es el ppseudoescalar.

Por ejemplo, en \mathbb{R}^3 con la base estándar $\{e_1, e_2, e_3\}$, y el ppseudoescalar definido en 3.3.2:

$$I_3 = e_1 \wedge e_2 \wedge e_3,$$

Tenemos que:

- Un escalar es un múltiplo de 1, y el dual de 1 es todo el espacio:
 $1^* = -1 \cdot I_3 = -e_1 \wedge e_2 \wedge e_3.$
- Un vector está dado por una combinación lineal de e_1, e_2 y e_3 , con los siguientes duales:
 $e_1^* = -e_1 \cdot I_3 = -e_1 \cdot e_1 \wedge e_2 \wedge e_3 = -(e_1 \cdot e_1)e_2 \wedge e_3 = -e_2 \wedge e_3,$
 $e_2^* = -e_2 \cdot I_3 = -e_2 \cdot e_1 \wedge e_2 \wedge e_3 = +(e_2 \cdot e_2)e_1 \wedge e_3 = +e_1 \wedge e_3,$
 $e_3^* = -e_3 \cdot I_3 = -e_3 \cdot e_1 \wedge e_2 \wedge e_3 = -(e_3 \cdot e_3)e_1 \wedge e_2 = -e_1 \wedge e_2,$
- Un bivector es una combinación lineal de $e_2 \wedge e_3, e_3 \wedge e_1, e_1 \wedge e_2$, con los duales:
 $(e_2 \wedge e_3)^* = -e_2 \wedge e_3 \cdot I_3 = -e_2 \cdot (e_3 \cdot e_1 \wedge e_2 \wedge e_3) = -e_2 \cdot (e_1 \wedge e_2) = e_1,$
 $(e_3 \wedge e_1)^* = -e_3 \wedge e_1 \cdot I_3 = -e_3 \cdot (e_1 \cdot e_1 \wedge e_2 \wedge e_3) = -e_3 \cdot (e_2 \wedge e_3) = e_2,$
 $(e_1 \wedge e_2)^* = -e_1 \wedge e_2 \cdot I_3 = -e_1 \cdot (e_2 \cdot e_1 \wedge e_2 \wedge e_3) = -e_1 \cdot (e_1 \wedge e_3) = e_3,$
- Un trivector es un múltiplo escalar de $e_1 \wedge e_2 \wedge e_3$, el cual tiene como dual a 1:
 $(e_1 \wedge e_2 \wedge e_3)^* = -e_1 \wedge e_2 \wedge e_3 \cdot e_1 \wedge e_2 \wedge e_3 = -e_1 \cdot (e_2 \cdot (e_1 \wedge e_2)) = e_1 \cdot e_1 = 1$

3.3.8 Representaciones directa y dual

Definición 3.3.13 En una ecuación de la forma:

$$x \wedge (\dots) = 0,$$

llamamos a (\dots) la representación directa del objeto. Mientras si la ecuación tiene la forma

$$x \cdot (\dots) = 0,$$

llamamos a (\dots) la representación dual del objeto.

De esta forma, a partir de la definición del producto exterior \wedge , la representación directa de líneas, planos, el espacio, y el origen es de la siguiente manera:

Definición 3.3.14 Un vector x se encuentra en la línea con dirección a si y solo si

$$x \wedge a = 0.$$

Por lo tanto, a es la representación directa de esa línea.

Definición 3.3.15 Un vector x se encuentra en el plano generado por a y b si

$$x \wedge a \wedge b = 0.$$

Por lo tanto, $a \wedge b$ es la representación directa de ese plano.

Definición 3.3.16 Cualquier vector x se encuentra el espacio generado por a , b y c si

$$x \wedge a \wedge b \wedge c = 0.$$

Por lo tanto, $a \wedge b \wedge c$ es la representación directa de ese espacio.

Definición 3.3.17 Finalmente, un vector x se encuentra en el origen si

$$x \wedge \alpha (= \alpha x) = 0,$$

para un α distinto de cero. Por lo tanto, un escalar $\alpha \neq 0$ es la representación directa del origen.

Mientras tanto, de la definición de contracción y siguiendo la proposición 3.3.4, la representación dual de rectas, planos, el espacio, y el origen es de la siguiente manera:

Definición 3.3.18 Un vector de posición x satisface

$$x \cdot a = 0$$

cuando x está en el complemento ortogonal de a (el plano perpendicular a a). Por lo tanto, a es la representación dual de un plano.

Definición 3.3.19 Un vector de posición x satisface

$$x \cdot a \wedge b = 0$$

si y sólo si $(x \cdot a) = (x \cdot b) = 0$, es decir, cuando x está en el complemento ortogonal de $a \wedge b$ (la normal al plano $a \wedge b$). Por lo tanto, $a \wedge b$ es la representación dual de una línea.

Definición 3.3.20 Un vector de posición x satisface

$$x \cdot a \wedge b \wedge c = 0$$

si y sólo si $(x \cdot a) = (x \cdot b) = (x \cdot c) = 0$, es decir, cuando x está en el origen, que es el complemento ortogonal de $a \wedge b \wedge c$. Por lo tanto, $a \wedge b \wedge c$ es la representación dual del origen.

Definición 3.3.21 De la primera ecuación de la Ec (3.8), un vector de posición x siempre satisface $x \cdot \alpha = 0$ para cualquier escalar α distinto de cero. Por lo tanto, un escalar α distinto de cero es la representación dual del espacio $a \wedge b \wedge c$.

Proposición 3.3.7 En las representación directa y dual se cumplen las siguientes relaciones de equivalencia:

$$x \wedge (\cdots) = 0 \Leftrightarrow x \cdot (\cdots)^* = 0,$$

$$x \cdot (\cdots) = 0 \Leftrightarrow x \wedge (\cdots)^* = 0.$$

Por tanto, el dual de la representación directa es la representación dual, y el dual de la representación dual es la representación directa. Ya que el dual especifica el complemento ortogonal, la ecuación $x \wedge (\cdots) = 0$ significa que x pertenece a ese subespacio, y la ecuación $x \cdot (\cdots) = 0$ significa que x es ortogonal a ese subespacio.

Para distinguir estas representaciones, usamos mayúsculas para representaciones directas y minúsculas para representaciones duales.

3.4 El Álgebra de Clifford

El álgebra de Clifford fue establecida por William Kingdon Clifford (1845–1879), un matemático británico, quien definió un álgebra general que contiene el álgebra de cuaterniones de Hamilton y el álgebra de productos externos de Grassmann.

Definición 3.4.1 El álgebra de Clifford en \mathbb{R}^n , denotada por C^n , es un álgebra generada a partir de 1 y los símbolos e_i , con una operación de multiplicación llamada producto geométrico o producto de Clifford que es asociativa y sujeta a las siguientes reglas:

$$e_i e_i = e_i^2 = 1, \tag{3.8a}$$

$$e_i e_j = -e_j e_i \tag{3.8b}$$

Así, por ejemplo, C^n ($i = 1, 2, 3$) es un espacio vectorial de 8 dimensiones, con respecto a sumas/restas y multiplicaciones escalares. Esto se debe a que el producto geométrico de 1, e_1 , e_2 y e_3 son objetos de la forma 1, e_1 , e_2 , e_3 , $e_2 e_3$, $e_3 e_1$, $e_1 e_2$ o $e_1 e_2 e_3$, como puede verse en la tabla 3.2.

	1	e_1	e_2	e_3	$e_2 e_3$	$e_3 e_1$	$e_1 e_2$	$e_1 e_2 e_3$
1	1	e_1	e_2	e_3	$e_2 e_3$	$e_3 e_1$	$e_1 e_2$	$e_1 e_2 e_3$
e_1	e_1	1	$e_1 e_2$	$-e_3 e_1$	$e_1 e_2 e_3$	$-e_3$	e_2	$e_2 e_3$
e_2	e_2	$-e_1 e_2$	1	$e_2 e_3$	e_3	$e_1 e_2 e_3$	$-e_2$	$e_3 e_1$
e_3	e_3	$e_3 e_1$	$-e_2 e_3$	1	$-e_2$	e_1	$e_1 e_2 e_3$	$e_1 e_2$
$e_2 e_3$	$e_2 e_3$	$e_1 e_2 e_3$	$-e_3$	e_2	-1	$-e_1 e_2$	$e_3 e_1$	$-e_1$
$e_3 e_1$	$e_3 e_1$	e_3	$e_3 e_1 e_3$	$-e_1$	$e_1 e_2$	-1	$-e_2 e_3$	$-e_2$
$e_1 e_2$	$e_1 e_2$	$-e_2$	e_1	$e_1 e_2 e_3$	$-e_3 e_1$	$e_2 e_3$	-1	$-e_3$
$e_1 e_2 e_3$	$e_1 e_2 e_3$	$e_2 e_3$	$e_3 e_1$	$e_1 e_2$	$-e_1$	$-e_2$	$-e_3$	-1

Table 3.2: Producto geométrico de los elementos del álgebra de Clifford de \mathbb{R}^3 .

Cuando hay de más de tres símbolos, por ejemplo, $e_1 e_2 e_3 e_1$, aplicando las propiedades incluidas en la Definición 3.4.1, se obtiene algunos de los elementos ya descritos:

$$e_1 e_2 e_3 e_1 = -e_1 e_2 e_1 e_3 = e_1^2 e_2 e_3 = e_2 e_3$$

En adelante daremos algunas definiciones considerando el caso particular de C^3 ; la generalización a C^n es inmediata.

3.4.1 Multivectores

Un elemento de C^3 tiene la forma:

$$C = \alpha + a_1e_1 + a_2e_2 + a_3e_3 + b_1e_2e_3 + b_2e_3e_1 + b_3e_1e_2 + ce_1e_2e_3,$$

lo que se denomina *multivector*. De los términos del multivector, α es un escalar, $a_1e_1 + a_2e_2 + a_3e_3$ un vector, $b_1e_2e_3 + b_2e_3e_1 + b_3e_1e_2$ un bivector, y $ce_1e_2e_3$ un trivector. En general, para el caso de C^n , se usa la notación 0-vector, 1-vector, 2-vector, ..., n -vector. El número de símbolos en el producto se llama *grado*, de manera que el grado de un n -vector es justamente n .

3.4.2 Paridad de Multivectores

Dado el grado de un elemento de C^n podemos definir la paridad de un multivector. Así, por ejemplo, en C^3 un multivector que consta de términos de grados impares

$$A = a_1e_1 + a_2e_2 + a_3e_3 + ce_1e_2e_3$$

se llama un multivector impar, y un multivector que consta de términos de grados pares

$$B = \alpha + b_1e_2e_3 + b_2e_3e_1 + b_3e_1e_2$$

se llama multivector par.

Puede verificarse que el producto de dos multivectores pares y el producto de dos multivectores impares produce un multivector par, y el producto de un multivector par por uno impar produce un multivector impar:

- El producto geométrico de dos multivectores impares

$$A = a_1e_1 + a_2e_2 + a_3e_3 + ce_1e_2e_3,$$

$$A' = a'_1e_1 + a'_2e_2 + a'_3e_3 + c'e_1e_2e_3,$$

produce un multivector par:

$$\begin{aligned} AA' = & a_1a'_1 + a_2a'_2 + a_3a'_3 - cc' + (a_2a'_3 - a_3a'_2 + ca_1 + c'a_1)e_2e_3 + \\ & (a_3a'_1 - a_1a'_3 + ca'_2 + c'a_2)e_3e_1 + (a_1a'_2 - a_2a'_3 + ca'_3 + c'a_3)e_1e_2. \end{aligned}$$

- El producto geométrico de dos multivectores pares

$$B = \alpha + b_1e_2e_3 + b_2e_3e_1 + b_3e_1e_2,$$

$$B' = \alpha' + b'_1e_2e_3 + b'_2e_3e_1 + b'_3e_1e_2,$$

produce un multivector par:

$$\begin{aligned} BB' = & \alpha\alpha' - b_1b'_1 - b_2b'_2 - b_3b'_3 + (\alpha b'_1 + \alpha' b_1 - b_2b'_3 + b_3b'_2)e_2e_3 + \\ & (\alpha' b_2 + \alpha b_2 - b_3b'_1 + b_1b'_3)e_3e_1 + (\alpha' b_3 + \alpha b_3 - b_1b'_2 + b_2b'_1)e_1e_2. \end{aligned}$$

- El producto geométrico del multivector impar A y el multivector par B produce un multivector impar:

$$\begin{aligned} AB = & (\alpha a_1 + a_3b_2 - a_2b_3 - cb_1)e_1 + (\alpha a_2 + a_1b_3 - a_3b_1 - cb_2)e_2 + \\ & (\alpha a_3 + a_2b_1 - a_1b_2 - cb_3)e_3 + (\alpha c + a_1b_1 + a_2b_2 + a_3b_3)e_1e_2e_3. \end{aligned}$$

- El producto geométrico del multivector par B y el multivector impar A produce un multivector impar:

$$BA = (\alpha a_1 + b_3 a_2 - b_2 a_3 - b_1 c) e_1 + (\alpha a_2 + b_1 a_3 - b_3 a_1 - b_2 c) e_2 + (\alpha a_3 + b_2 a_1 - b_1 a_2 - b_3 c) e_3 + (\alpha c + b_1 a_1 + b_2 a_2 + b_3 a_3) e_1 e_2 e_3.$$

Las sumas y los múltiplos escalares de multivectores pares son multivectores pares. Esto significa que el conjunto de multivectores pares forma en sí mismo un álgebra cerrada, es decir, una subálgebra de una álgebra de Clifford. En el caso de C^3 , la subálgebra par es el álgebra de Hamilton. Esto puede mostrarse si definimos:

$$i = e_3 e_2 (= -e_2 e_3), \quad j = e_1 e_3 (= -e_3 e_1), \quad k = e_2 e_1 (= -e_1 e_2),$$

que cumplen con las propiedades requeridas. En otras palabras, el álgebra de Hamilton es la subálgebra par de C^3 . Igualmente puede mostrarse que conjunto de números complejos \mathbb{C} es una subálgebra par de C^2 [4].

3.4.3 El Álgebra de Grassman en el Álgebra de Clifford y el producto externo

Hemos visto que C^3 incluye el álgebra de Hamilton. Ahora vamos a ver que el álgebra de Grassmann también es parte del álgebra de Clifford. Para esto consideremos los vectores $a = a_1 e_1 + a_2 e_2 + a_3 e_3$ y $b = b_1 e_1 + b_2 e_2 + b_3 e_3$ en \mathbb{R}^3 , y definimos el producto exterior $a \wedge b$ del vector a y b por la siguiente antisimetrización:

$$a \wedge b = \frac{1}{2}(ab - ba). \quad (3.9)$$

De esta definición vemos que

$$b \wedge a = -a \wedge b, \\ a \wedge a = 0.$$

Para los vectores a , b y c , definimos su producto exterior mediante la siguiente antisimetrización:

$$a \wedge b \wedge c = \frac{1}{6}(abc + bca + cab - cba - bac - acb). \quad (3.10)$$

Entonces, vemos que

$$a \wedge b \wedge c = b \wedge c \wedge a = c \wedge a \wedge b = -c \wedge b \wedge a = -b \wedge a \wedge c = -a \wedge c \wedge b.$$

Finalmente, el producto exterior de cuatro o más vectores $a \wedge b \wedge c \wedge d \wedge \dots$ se define como 0.

Vemos que se cumplen todos los axiomas del producto exterior de Grassmann, por lo que podemos identificar al producto (3.9) como el producto externo de Grassmann.

Como ejemplos concretos del tipo de cálculos que se realizan, consideremos los vectores $a = a_1 e_1 + a_2 e_2 + a_3 e_3$ y $b = b_1 e_1 + b_2 e_2 + b_3 e_3$. De forma que los productos geométricos ab y ba son:

$$ab = a_1 b_1 + a_2 b_2 + a_3 b_3 + (a_2 b_3 - a_3 b_2) e_2 e_3 + (a_3 b_1 - a_1 b_3) e_3 e_1 + (a_1 b_2 - a_2 b_1) e_1 e_2, \quad (3.11a)$$

$$ba = b_1 a_1 + b_2 a_2 + b_3 a_3 + (b_2 a_3 - b_3 a_2) e_2 e_3 + (b_3 a_1 - b_1 a_3) e_3 e_1 + (b_1 a_2 - b_2 a_1) e_1 e_2. \quad (3.11b)$$

Y el producto exterior $a \wedge b$ tiene la forma

$$a \wedge b = (a_2b_3 - a_3b_2)e_2e_3 + (a_3b_1 - a_1b_3)e_3e_1 + (a_1b_2 - a_2b_1)e_1e_2.$$

Además, si hacemos $c = c_1e_1 + c_2e_2 + c_3e_3$, podemos ver que:

$$a \wedge b \wedge c = (a_1b_2c_3 + a_2b_3c_1 + a_3b_1c_2 - a_1b_3c_2 - a_2b_1c_3 - a_3b_2c_1)e_1e_2e_3.$$

De donde se infiere que $e_1 \wedge e_2 = e_1e_2$, $e_2 \wedge e_3 = e_2e_3$, $e_3 \wedge e_1 = e_3e_1$ y $e_1 \wedge e_2 \wedge e_3 = e_1e_2e_3 = I_3$.

Con respecto a los elementos de la base, vemos que la única diferencia entre el producto exterior y el geométrico es que el producto exterior de los mismos símbolos es 0 mientras que su producto geométrico es 1. Fuera de esto, para diferentes símbolos, los productos geométricos y exteriores siguen la misma regla y, por lo tanto, los productos de diferentes símbolos se pueden identificar con sus productos exteriores [1].

3.4.3.1 Producto externo entre k -blades.

Las expresiones (3.9)-(3.10) son útiles cuando se trabaja en C^3 , sin embargo para dimensiones mayores hay que considerar productos externos entre k -blades. Para estos casos (incluyendo el de C^n), hay una definición más general [4]:

Definición 3.4.2 El producto externo entre dos k -blades A_l y B_m en C^n , se define como:

$$A_l \wedge B_m = \langle A_l B_m \rangle_{m+l}, \quad (3.12)$$

donde

$$\langle (\dots) \rangle_n, \quad (3.13)$$

se conoce como operador de grado y se encarga de extraer los elementos de grado n del multivector (\dots) .

3.4.4 El producto interno

De las Ecs. (3.11a) y (3.11b), se deduce que

$$a \cdot b = \frac{1}{2}(ab + ba). \quad (3.14)$$

En particular, si a y b son ortogonales, es decir, $a \cdot b = 0$, tenemos $ab = -ba$, es decir, los productos geométricos de vectores ortogonales son anticonmutativos.

3.4.4.1 Producto interno entre k -vectores.

El producto interno (3.14) se aplica a vectores. Para el caso más general del producto interno entre dos k -vectores, hay varias definiciones que son discutidas en la Ref. [20]. En el presente trabajo, adoptaremos una definición del producto interno que es equivalente a la contracción discutida en la Sección 3.3.5.

Definición 3.4.3 El producto interno (contracción) entre los k -blades A_l y B_m en C^n , se define como:

$$A_l \cdot B_m = \langle A_l B_m \rangle_{m-l}. \quad (3.15)$$

En el caso $l > m$ se define $A_k \cdot B_l = 0$

3.4.5 El producto geométrico

De los resultados anteriores, podemos verificar que para el caso de vectores, la ecuación (3.11a) también se puede escribir como

$$ab = (a \cdot b) + a \wedge b, \quad (3.16)$$

donde, de acuerdo con la Definición 3.4.3, el punto \cdot denota contracción.

De forma similar, el producto geométrico entre el vector a y el bivector $b \wedge c$ esta dado por:

$$a(b \wedge c) = a \cdot b \wedge c + a \wedge b \wedge c. \quad (3.17)$$

Y el producto geométrico del vector a y el trivector $b \wedge c \wedge d$ es dado por:

$$a(b \wedge c \wedge d) = a \cdot b \wedge c \wedge d + a \wedge b \wedge c \wedge d. \quad (3.18)$$

A manera de resumen, se tiene que para el caso de vectores:

Definición 3.4.4 La ecuación fundamental del álgebra de Clifford o geométrica es:

$$ab = a \cdot b + a \wedge b, \quad (3.19)$$

donde:

$$a \cdot b = \frac{1}{2}(ab + ba),$$

$$a \wedge b = \frac{1}{2}(ab - ba).$$

Una generalización de la identidad anterior es la siguiente:

El producto geométrico de un vector a y un k -vector (\dots) , se expresa en términos de la contracción y del producto externo de la siguiente manera:

$$a(\dots) = a \cdot (\dots) + a \wedge (\dots). \quad (3.20)$$

De esta forma, todos los productos geométricos de vectores con blades se reducen a contracciones y productos externos aplicando recursivamente esta regla [4].

La siguiente proposición resume las propiedades del producto geométrico para el caso general de multivectores:

Proposición 3.4.1 El producto geométrico es asociativo y distributivo pero, en general, no conmutativo. Es decir, para los multivectores A , B y C en C^n , tendremos:

$$AB \neq BA,$$

$$(AB)C = A(BC),$$

$$A(B+C) = (AB) + (AC),$$

$$(B+C)A = (BA) + (CA).$$

3.4.6 Norma de un multivector

Para el caso de vectores, considerando $a = b$ en (3.14), vemos que

$$||a||^2 = a^2. \quad (3.21)$$

En general, la magnitud de un multivector $A \in C^n$ se define como [4]:

$$|A| = \left\langle \widetilde{AA} \right\rangle_0^{1/2}, \quad (3.22)$$

donde \widetilde{A} es la reversión definida en 3.3.9. La operación reversión es distributiva [4], por lo que es posible calcular la reversión de cualquier multivector $A \in C^n$.

Además, la generalización de esta definición a productos geométricos es inmediata ya que si consideramos un blade arbitrario $e_1 e_2 \cdots e_p$, su reversión es:

$$(e_1 \widetilde{e_2 \cdots e_p}) = e_p \cdots e_2 e_1.$$

3.4.7 El inverso de un multivector

Si $||a|| \neq 0$, se tiene:

$$a \frac{a}{||a||^2} = 1 \implies a^{-1} = \frac{a}{||a||^2}.$$

La existencia de la inversa significa que si $ab = ac$, para $a \neq 0$, entonces $b = c$. Además, de la asociatividad del producto geométrico, se puede inferir que:

$$\begin{aligned} (ab)^{-1} &= b^{-1} a^{-1}, \\ (abc)^{-1} &= c^{-1} b^{-1} a^{-1}, \\ (abc \cdots)^{-1} &= \cdots c^{-1} b^{-1} a^{-1}. \end{aligned}$$

Lo que es evidente si se toma en cuenta que:

$$abc \cdots c^{-1} b^{-1} a^{-1} = \cdots = abc c^{-1} b^{-1} a^{-1} = abb^{-1} a^{-1} = aa^{-1} = 1.$$

Para el caso general, consideremos un k -vector $A_k \in C^n$, es decir, $A_k = \langle A \rangle_k$. Su inverso se define como:

$$A_k^{-1} = \frac{\widetilde{A}_k}{||a||^2}. \quad (3.23)$$

Definición 3.4.5 Si consideramos un k -vector $A_k \in C^n$, es decir, $A_k = \langle A \rangle_k$ y su inverso A_k^{-1} , la involución de grado (\dagger) estará definida por:

$$A_k^\dagger = (-1)^k A_k^{-1} \quad (3.24)$$

3.4.8 Espacios con signatura (p, q) .

Una álgebra de Clifford también puede construirse sobre espacios vectoriales dotados de una métrica no Euclideana con signatura (p, q) , donde $p + q = n$, que se denotan como $\mathbb{R}^{p,q}$. En este caso, la definición 3.4.1 se generaliza de la siguiente manera:

Definición 3.4.6 El álgebra de Clifford de $\mathbb{R}^{p,q}$ ($p+q=n$), denotada como $C^{p,q}$, es un álgebra generada a partir de 1 y los símbolos e_i , con una operación de multiplicación llamada producto geométrico o producto de Clifford que es asociativa y sujeta a la siguiente regla:

$$\begin{aligned} e_i^2 &= 1, & \text{para } i &= 1, 2, \dots, p, \\ e_i^2 &= -1, & \text{para } i &= p+1, p+2, \dots, p+q, \\ e_i e_j &= -e_j e_i, & \text{para } i, j &= 1, 2, \dots, n. \end{aligned}$$

Así, por ejemplo, el espacio $\mathbb{R}^{4,1}$ es el espacio de Minkowski de dimensión 5 y el álgebra $C^{4,1}$ será relevante para la construcción del Álgebra Geométrica Conforme, que se describe a continuación.

3.5 Álgebra Geométrica Conforme.

El modelo conforme en \mathbb{R}^3 equipado con un producto geométrico o, más generalmente, con una álgebra de Clifford, se denomina Álgebra Geométrica Conforme. El modelo conforme tiene sus orígenes en el trabajo de F. Wachter, que fue un estudiante de Gauss [7], y puede definirse de la siguiente manera.

Definición 3.5.1 El modelo o espacio conforme de \mathbb{R}^3 es un espacio de cinco dimensiones definido con la base $\{e_0, e_1, e_2, e_3, e_\infty\}$, y las relaciones:

$$e_i \cdot e_j = \delta_{ij}, \tag{3.25a}$$

$$e_0 \cdot e_i = 0, \tag{3.25b}$$

$$e_\infty \cdot e_i = 0, \tag{3.25c}$$

$$e_0 \cdot e_0 = e_\infty \cdot e_\infty = 0, \tag{3.25d}$$

$$e_0 \cdot e_\infty = -1, \tag{3.25e}$$

donde $i, j = 1, 2, 3$.

De aquí se obtiene que, si consideramos dos elementos de este espacio:

$$x = x_0 e_0 + x_1 e_1 + x_2 e_2 + x_3 e_3 + x_\infty e_\infty,$$

$$y = y_0 e_0 + y_1 e_1 + y_2 e_2 + y_3 e_3 + y_\infty e_\infty,$$

entonces

$$x \cdot y = x_1 y_1 + x_2 y_2 + x_3 y_3 - x_0 y_\infty - x_\infty y_0.$$

Y, si $x = y$ tendremos:

$$\|x\|^2 = x_1^2 + x_2^2 + x_3^2 - 2x_0 x_\infty.$$

Una ventaja importante del modelo conforme es que objetos geométricos como puntos, líneas, planos y esferas pueden ser representados por medio de identidades algebraicas, como veremos más adelante. Otra ventaja más, también muy importante, es que en el modelo conforme las isometrías (movimientos rígidos) son transformaciones lineales ortogonales, lo que no sucede en \mathbb{R}^3 , en donde las traslaciones no son transformaciones lineales. Esta propiedad puede usarse también para construir el espacio conforme, es decir, si nos preguntamos ¿en qué espacio las isometrías constituyen transformaciones lineales? la respuesta será en el espacio conforme de 5 dimensiones [21]. Este otro punto de vista lleva a una definición alternativa del modelo conforme que será de mucha utilidad en este trabajo. En la Ref. [21] se muestra que

las isometrías no constituyen transformaciones ortogonales en \mathbb{R}^3 ni en \mathbb{R}^4 , pero si tienen esta propiedad en cinco dimensiones, en el espacio de Minkowsky $\mathbb{R}^{4,1}$, que definimos en la Sección 3.4.8, siempre y cuando se definan dos nuevos vectores:

$$e_0 = \frac{e_4 + e_5}{2}, \quad y \quad e_\infty = e_5 - e_4. \quad (3.26)$$

Con una transformación inversa dada por:

$$e_4 = e_0 - \frac{e_\infty}{2}, \quad y \quad e_5 = e_0 + \frac{e_\infty}{2}. \quad (3.27)$$

Es decir, partiendo de $\mathbb{R}^{4,1}$, que tiene base $\{e_1, e_2, e_3, e_4, e_5\}$ (con la propiedad $e_4 \cdot e_4 = -1$), con el cambio de variable (4.1), se obtiene la base $\{e_0, e_1, e_2, e_3, e_\infty\}$ del espacio conforme.

Con esto, ya tenemos las herramientas para definir el Álgebra Geométrica Conforme:

Definición 3.5.2 El Álgebra Geométrica Conforme $\mathbb{G}^{4,1}$, sobre $\mathbb{R}^{4,1}$, es una álgebra generada por la identidad 1 y los símbolos $\{e_0, e_1, e_2, e_3, e_\infty\}$, por el *producto geométrico*, y las siguientes relaciones:

$$e_i e_j + e_j e_i = 0, \quad (3.28a)$$

$$e_i e_0 + e_0 e_i = 0, \quad (3.28b)$$

$$e_i e_\infty + e_\infty e_i = 0, \quad (3.28c)$$

$$e_i^2 = 1, \quad (3.28d)$$

$$e_0 e_\infty + e_\infty e_0 = -2, \quad (3.28e)$$

$$e_0^2 = e_\infty^2 = 0, \quad (3.28f)$$

donde $i, j = 1, 2, 3$.

El Álgebra Geométrica Conforme tiene la ventaja de que las primitivas geométricas como son: puntos, líneas, planos, círculos y esferas, pueden ser representadas directamente como “objetos” algebraicos que pueden ser comparados, intersecados y transformados (i.e., trasladados, rotados y escalados) sin necesidad de especificar sus coordenadas, obteniendo una correspondencia inmediata entre objetos algebraicos y objetos geométricos con un gran nivel de abstracción. Además, posee un álgebra de transformaciones conformes sobre la cual se puede aplicar Cálculo Geométrico.

El Álgebra Geométrica Conforme, además, tiene propiedades que lo hacen ventajoso para muchas aplicaciones de ingeniería. Algunas de estas propiedades son la unificación de muchos sistemas matemáticos en un sistema matemático fácil de entender, y un manejo intuitivo de objetos geométricos y operaciones geométricas. No solo las geometrías se describen de manera clara y compacta, sino que además, como esta álgebra es libres de coordenadas; se puede adaptar bien a procesos algorítmicos y computacionales [22].

Los tres productos principales del álgebra de Clifford, el producto exterior, el producto interno y el producto geométrico, tienen en el Álgebra Geométrica Conforme usos importantes. Usaremos el producto exterior principalmente para la construcción y la intersección de objetos geométricos, así como para la representación directa de objetos, mientras que el producto interior se utilizará para el cálculo de ángulos y distancias, y para la representación dual de objetos. Y por su parte, el producto geométrico se utilizará principalmente para la descripción de transformaciones [22].

Estos tres productos principales (producto exterior, interno y geométrico) se definen a continuación. Posteriormente, todo lo relacionado con los objetos geométricos, la geometría y las transformaciones conformes, se presentaran en el Capítulo 5, usando el paquete que se muestra en el Capítulo 4.

3.5.1 Producto geométrico

Si definimos $x = x_0e_0 + X + x_\infty e_\infty$ donde $X = x_1e_1 + x_2e_2 + x_3e_3$, y $y = y_0e_0 + Y + y_\infty e_\infty$ donde $Y = y_1e_1 + y_2e_2 + y_3e_3$, el producto geométrico xy es:

$$\begin{aligned} xy &= (x_0e_0 + X + x_\infty e_\infty)(y_0e_0 + Y + y_\infty e_\infty) \\ &= x_0y_0e_0^2 + x_0e_0Y + x_0y_\infty e_0e_\infty + y_0Xe_0 + XY + y_\infty Xe_\infty + x_\infty y_0e_\infty e_0 + x_\infty e_\infty Y + x_\infty y_\infty e_\infty^2 \\ &= (y_0X - x_0Y)e_0 + XY + x_0y_\infty e_0e_\infty + x_\infty y_0e_\infty e_0 + (y_\infty X - x_\infty Y)e_\infty, \end{aligned} \quad (3.29)$$

mientras tanto el producto geométrico yx es:

$$\begin{aligned} yx &= (y_0e_0 + Y + y_\infty e_\infty)(x_0e_0 + X + x_\infty e_\infty) \\ &= y_0x_0e_0^2 + y_0e_0X + y_0x_\infty e_0e_\infty + x_0Ye_0 + YX + x_\infty Ye_\infty + y_\infty x_0e_\infty e_0 + y_\infty e_\infty X + y_\infty x_\infty e_\infty^2 \\ &= (x_0Y - y_0X)e_0 + YX + y_0x_\infty e_0e_\infty + y_\infty x_0e_\infty e_0 + (x_\infty Y - y_\infty X)e_\infty. \end{aligned}$$

A partir de esto, usando (3.28) vemos que la simetrización del producto geométrico será:

$$\begin{aligned} \frac{1}{2}(xy + yx) &= \frac{1}{2}(XY + x_0y_\infty e_0e_\infty + x_\infty y_0e_\infty e_0 + YX + y_0x_\infty e_0e_\infty + y_\infty x_0e_\infty e_0) \\ &= \frac{1}{2}(XY + YX + x_0y_\infty (e_0e_\infty + e_\infty e_0) + x_\infty y_0 (e_0e_\infty + e_\infty e_0)) = X \cdot Y - x_0y_\infty - x_\infty y_0. \end{aligned} \quad (3.30)$$

Por otro lado, utilizando (3.28), la antisimetrización del producto geométrico es:

$$\begin{aligned} \frac{1}{2}(xy - yx) &= \frac{1}{2}(((2y_0X - 2x_0Y)e_0 + (2y_\infty X - 2x_\infty Y)e_\infty) \\ &\quad + (x_0y_\infty e_0e_\infty + x_\infty y_0e_\infty e_0) - (y_0x_\infty e_0e_\infty + y_\infty x_0e_\infty e_0)) \\ &= \frac{1}{2}(((2y_0X - 2x_0Y)e_0 + (2y_\infty X - 2x_\infty Y)e_\infty) \\ &\quad + (x_0y_\infty e_0e_\infty + x_\infty y_0(-2 - e_0e_\infty)) - (y_0x_\infty e_0e_\infty + y_\infty x_0(-2 - e_0e_\infty))) \\ &= \frac{1}{2}((2y_0X - 2x_0Y)e_0 + (2y_\infty X - 2x_\infty Y)e_\infty + x_\infty y_0(-2 - 2e_0e_\infty) + y_\infty x_0(2 + 2e_0e_\infty)) \\ &= (y_0X - x_0Y)e_0 + (y_\infty X - x_\infty Y)e_\infty + x_\infty y_0(-1 - e_0e_\infty) + y_\infty x_0(1 + e_0e_\infty) \\ &\quad (y_0X - x_0Y)e_0 + (y_\infty X - x_\infty Y)e_\infty + x_\infty y_0(1 + e_\infty e_0) + y_\infty x_0(1 + e_0e_\infty). \end{aligned} \quad (3.31)$$

3.5.2 Producto interno

Los productos internos entre los elementos básicos del espacio conforme se definen en 3.5.2. Entonces, si tomamos un elemento $x = x_0e_0 + X + x_\infty e_\infty$ donde $X = x_1e_1 + x_2e_2 + x_3e_3$, y $y = y_0e_0 + Y + y_\infty e_\infty$ donde $Y = y_1e_1 + y_2e_2 + y_3e_3$, el producto interno $x \cdot y$ será:

$$x \cdot y = X \cdot Y - x_0y_\infty - x_\infty y_0. \quad (3.32)$$

Lo cual coincide con la ecuación (3.30) de forma que tendremos:

$$x \cdot y = \frac{1}{2}(xy + yx) \quad (3.33)$$

3.5.2.1 Producto interno entre k-vectores

Definiendo S como un k-vector tendremos las siguientes relaciones:

$$S \cdot 1 = 1 \cdot S = 0 \quad (3.34)$$

$$e_0 \cdot S = \begin{cases} 0 & e_\infty \notin S \\ (-1)^{Posicion(e_\infty)}(S - e_\infty) & e_\infty \in S \end{cases} \quad (3.35)$$

$$e_\infty \cdot S = \begin{cases} 0 & e_0 \notin S \\ (-1)^{Posicion(e_0)}(S - e_0) & e_0 \in S \end{cases} \quad (3.36)$$

Donde $(S - e_0)$ significa retirar e_0 del conjunto de elementos de S , $(S - e_\infty)$ significa retirar e_∞ del conjunto de elementos de S y $Posicion(e_0)$ significa extraer la posición del elemento e_0 en S .

Por ejemplo, si S es el k-vector $e_0e_1e_2e_3$, tendremos:

$$\begin{aligned} e_\infty \cdot e_0e_1e_2e_3 &= (-1)^1 e_1e_2e_3 = -e_1e_2e_3 \\ e_0 \cdot e_0e_1e_2e_3 &= 0 \end{aligned}$$

O si S es el k-vector $e_0e_2e_\infty$, tendremos:

$$\begin{aligned} e_\infty \cdot e_0e_2e_\infty &= (-1)^1 e_2e_\infty = -e_2e_\infty \\ e_0 \cdot e_0e_2e_\infty &= (-1)^3 e_0e_2 = -e_0e_2 \end{aligned}$$

Si ahora definimos T como otro k-vector, tendremos:

$$T \cdot S = \begin{cases} T \cdot S & (e_0 \in T \& e_\infty \notin S) \\ T \cdot S & (e_0 \notin T \& e_\infty \in S) \\ T \cdot S & (e_0 \notin T \& e_\infty \notin S) \\ (-1)^{Posicion(e_0)+Posicion(e_\infty)+1}(T - e_0) \cdot (S - e_\infty) & (e_0 \in T \& e_\infty \in S) \end{cases}$$

Por ejemplo, si T es el k-vector e_0e_1 y S es el k-vector $e_1e_2e_3e_\infty$, tendremos:

$$e_0e_1 \cdot e_1e_2e_3e_\infty = (-1)^{1+3+1}(e_1) \cdot (e_1e_2e_3) = -e_2e_3$$

O si T es el k-vector e_1e_2 y S es el k-vector $e_1e_2e_3e_\infty$, tendremos:

$$e_1e_2 \cdot e_1e_2e_3e_\infty = (e_1e_2) \cdot (e_1e_2e_3e_\infty) = e_3e_\infty$$

3.5.3 Producto Exterior

El álgebra geométrica proporciona un producto externo \wedge con las siguientes propiedades:

$$e_i \wedge e_j = \begin{cases} 0 & i = j \\ e_i e_j & (i \neq j \& (i \neq 0 \parallel j \neq \infty)) \\ 1 + e_i e_j & (i = 0 \& j = \infty) \end{cases} \quad (3.37)$$

Lo que para $i, j \in 1, 2, 3$, genera la tabla 3.5.3.

\wedge	e_0	e_1	e_2	e_3	e_∞
e_0	0	e_0e_1	e_0e_2	e_0e_3	$1 + e_0e_\infty$
e_1	$-e_0e_1$	0	e_1e_2	e_1e_3	e_1e_∞
e_2	$-e_0e_2$	$-e_1e_2$	0	e_2e_3	e_2e_∞
e_3	$-e_0e_3$	$-e_1e_3$	$-e_2e_3$	0	e_3e_∞
e_∞	$-1 - e_0e_\infty$	$-e_0e_\infty$	$-e_2e_\infty$	$-e_3e_\infty$	0

Table 3.3: Tabla del producto exterior para los elementos básicos del espacio conforme.

Por ejemplo si queremos computar el producto exterior de dos vectores, $a = (e_0 + e_1)$ y $b = (e_1 - e_\infty)$, se puede utilizar la distributividad y la tabla 3.5.3:

$$\begin{aligned} a \wedge b &= (e_0 + e_1) \wedge (e_1 - e_\infty) = (e_0 \wedge e_1) - (e_0 \wedge e_\infty) + (e_1 \wedge e_1) - (e_1 \wedge e_\infty) \\ &= e_0e_1 - (1 + e_0e_\infty) + 0 - e_1e_\infty = e_0e_1 - 1 - e_0e_\infty - e_1e_\infty \end{aligned}$$

Como vimos anteriormente podemos definir el producto geométrico como:

$$xy = x \cdot y + x \wedge y \quad (3.38)$$

De forma que a partir de las ecuaciones (3.29) y (3.30) tendremos:

$$\begin{aligned} x \wedge y &= xy - x \cdot y \\ &= (y_0X - x_0Y)e_0 + XY + x_0y_\infty e_0e_\infty + x_\infty y_0 e_\infty e_0 + (y_\infty X - x_\infty Y)e_\infty - (XY - x_0y_\infty - x_\infty y_0) \\ &= (y_0X - x_0Y)e_0 + (y_\infty X - x_\infty Y)e_\infty + x_0y_\infty + x_0y_\infty e_0e_\infty + x_\infty y_0 + x_\infty y_0 e_\infty e_0 \\ &= (y_0X - x_0Y)e_0 + (y_\infty X - x_\infty Y)e_\infty + x_0y_\infty(1 + e_0e_\infty) + x_\infty y_0(1 + e_\infty e_0) \end{aligned}$$

Lo cual coincide con la ecuación (3.31) y tendremos:

$$x \wedge y = \frac{1}{2}(xy - yx) \quad (3.39)$$

3.5.3.1 Producto exterior entre k-vectores

Definiendo S como un k-vector, tendremos las siguientes relaciones:

$$e_0 \wedge S = \begin{cases} 0 & e_0 \in S \\ e_0 S & e_\infty \notin S \\ e_0 S + (-1)^{\text{Posicion}(e_\infty+1)}(S - e_\infty) & e_\infty \in S \end{cases} \quad (3.40)$$

$$e_\infty \wedge S = \begin{cases} e_\infty S & e_0 \notin S \\ e_\infty S + (-1)^{\text{Posicion}(e_0)}(S - e_0) & e_0 \in S \end{cases} \quad (3.41)$$

Por ejemplo, si queremos computar el producto exterior de dos multivectores $a = (e_0 + e_1e_2)$ y $b = (e_0e_1 + e_\infty)$, en base a la distributividad y las ecuaciones (3.40) y (3.41), tendremos:

$$\begin{aligned} a \wedge b &= (e_0 + e_1e_2) \wedge (e_0e_1 - e_\infty) \\ &= \cancel{e_0 \wedge e_0e_1} - e_0 \wedge e_\infty + e_1e_2 \wedge e_0e_1 - e_1e_2 \wedge e_\infty \\ &= -(1 + e_0e_\infty) + \cancel{e_1e_2e_0} - e_\infty \wedge e_1e_2 = -1 - e_0e_\infty - e_\infty e_1e_2 \end{aligned}$$

Por otro lado, si queremos computar el producto exterior de dos multivectores $a = (e_\infty + e_1 e_2)$ y $b = (e_0 e_1 + e_\infty)$, en base a la distributividad y las ecuaciones (3.40) y (3.41), tendremos:

$$\begin{aligned} a \wedge b &= (e_\infty + e_1 e_2) \wedge (e_0 e_1 - e_\infty) = e_\infty \wedge e_0 e_1 - e_\infty \wedge e_\infty + e_1 e_2 \wedge e_0 e_1 - e_1 e_2 \wedge e_\infty \\ &= (e_\infty e_0 e_1 - e_1) - 0 + e_1 e_2 e_0 e_1 - e_1 e_2 e_\infty \\ &= (e_\infty e_0 e_1 - e_1) - 0 + e_1 e_1 e_2 e_0 - e_1 e_2 e_\infty. \end{aligned}$$

3.5.4 Objetos geométricos y sus transformaciones

Como mencionamos, una de las mayores ventajas del AGC es que los objetos geométricos primitivos, puntos, líneas, planos, círculos y esferas, pueden ser representadas directamente como expresiones algebraicas que pueden ser comparados, intersecados y transformados. Esta es una parte importante de esta álgebra y nos ha parecido que la mejor manera de exponer este tema es por medio de ejemplos concretos, usando el paquete que de se desarrolló en este trabajo. Por esta razón, todo lo relacionado con representación de objetos geométricos, su manipulación y transformaciones, se presenta en el Capítulo 5, aquí sólo presentamos un resumen de los objetos, las transformaciones y su representación en CGA, mismo que se muestra en las Tablas 3.4 y 3.5.

Table 3.4: Objetos geométricos y su representación en CGA.

Objeto	Representación directa	Representación dual
Ecuación general	$p \wedge (\dots) = 0$	$p \cdot (\dots) = 0$
Puntos aislados	$p = e_0 + x + \ x\ ^2 e_\infty / 2$	$p = e_0 + x + \ x\ ^2 e_\infty / 2$
Líneas	$L = p_1 \wedge p_2 \wedge e_\infty = p \wedge u \wedge e_\infty$	$l = \pi_1 \wedge \pi_2$
Planos	$\Pi = p_1 \wedge p_2 \wedge p_3 \wedge e_\infty = p_1 \wedge p_2 \wedge u \wedge e_\infty$	$\pi = n + h e_\infty = p_1 - p_2$
Esferas	$\Sigma = p_1 \wedge p_2 \wedge p_3 \wedge p_4$	$\sigma = c - r^2 e_\infty / 2$
Círculos	$S = p_1 \wedge p_2 \wedge p_3$	$s = \sigma_1 \wedge \sigma_2 = \sigma \wedge \pi$
Par de puntos	$PP = p_1 \wedge p_2$	$pp = s \wedge \sigma = s \wedge \pi = \sigma_1 \wedge \sigma_2 \wedge \sigma_3$
Puntos planos	$FP = p \wedge e_\infty$	$fp = \pi \wedge l = \pi_1 \wedge \pi_2 \wedge \pi_3$

Table 3.5: Transformaciones conformes y su representación en CGA.

Transformación	Representación	Aplicación
Versor	$V = v_1 v_2 \dots v_k$	$V(\dots) V^\dagger$
Inversión	$V^{-1} = v_k^{-1} v_{k-1}^{-1} \dots v_1^{-1}$	
Involución	$V^\dagger = (-1)^k V^{-1}$	
Traslación	$T_t = 1 - \frac{1}{2} t e_\infty$	$T_t(\dots) T_t^{-1}$
Rotación	$R = ab$	$R(\dots) R^{-1}$

4. Implementación en *Mathematica*

En este Capítulo se presenta la implementación del Álgebra Geométrica Conforme en el lenguaje de programación *Mathematica*. Esta implementación proporciona una herramienta que permite realizar cálculos simbólicos en el Álgebra Geométrica Conforme. Con esta herramienta se facilita el cálculo de productos geométricos, productos exteriores e interiores, así como otros cálculos relacionados con la geometría de transformaciones. La implementación se realizó utilizando el paradigma de programación basado en reglas, para el que *Mathematica* constituye una plataforma adecuada. El paquete desarrollado se denominó CGAlgebra y se encuentra disponible en:

<https://github.com/jlaragonvera/Geometric-Algebra>

4.1 Programación en *Mathematica*.

Comúnmente considerado como un sistema de álgebra computacional, *Mathematica* es también un poderoso lenguaje de programación de propósito general utilizado en áreas científicas, computacionales, de ingeniería y de matemática. Originalmente fue concebido por Stephen Wolfram, quien continúa siendo el líder del grupo de matemáticos y programadores que desarrollan el producto en la empresa *Wolfram Research*.

Mathematica tiene una amplia gama de algoritmos incorporados, todos accesibles de forma automática por medio de su lenguaje simbólico unificado que se puede adaptar fácilmente a todo tipo de programas [23].

4.2 Programación basada en reglas.

La programación basada en reglas usa reglas de reemplazo en lugar de construcciones procedurales, es decir, funciona mediante la aplicación de reglas, primero realizando una comparación y, posteriormente, una aplicación de las reglas de acuerdo con la situación que se presente. Es decir, se basa en condiciones,

revisando la existencia o ausencia de cierta información y las relaciones entre datos, y acciones, donde se incluye o elimina información.

Las reglas llevan la coincidencia de patrones a un nuevo nivel de expresividad, lo que le permite realizar transformaciones en expresiones coincidentes. Pero es importante mencionar que en *Mathematica* el orden en que aparecen las reglas en la lista es importante, ya que cada regla se prueba por turno, comenzando por la primera. A veces podemos usar este hecho a nuestro favor colocando una regla más general después de una más específica. Esta es una construcción poderosa porque extiende el grado de control sobre el proceso [23].

4.3 Paquete CGAlgebra

El resultado principal de este trabajo es el paquete denominado CGAlgebra que contiene la implementación del Álgebra Geométrica Conforme en *Mathematica*. En este capítulo se describen las funciones que se desarrollaron para implementar las operaciones más importantes de esta álgebra.

4.3.1 Esquema general y metodología.

La implementación que se propone en este trabajo se basa en la definición del espacio conforme que consiste en considerar el espacio de Minkowsky $\mathbb{R}^{4,1}$, con los cambios de variable:

$$e_0 = \frac{e_4 + e_5}{2}, \quad \text{and} \quad e_\infty = e_5 - e_4, \quad (4.1)$$

con transformación inversa:

$$e_4 = e_0 - \frac{e_\infty}{2}, \quad \text{and} \quad e_5 = e_0 + \frac{e_\infty}{2}. \quad (4.2)$$

Con lo que la base del espacio conforme es $\{e_0, e_1, e_2, e_3, e_\infty\}$.

Es importante señalar que todos los cálculos se van a realizar en $C^{4,1}$, con la base estándar de $\mathbb{R}^{4,1}$, $\{e_1, e_2, e_3, e_4, e_5\}$. Esto conlleva realizar un doble cambio de variable, es decir, las entradas, que son dadas en la base del espacio conforme, se transforman a la base estándar de $\mathbb{R}^{4,1}$ usando (4.2), se realizan los cálculos solicitados, en $C^{4,1}$, y se regresa a la base del espacio conforme usando (4.1), para proporcionar la salida en este espacio. Este doble cambio de variable es necesario debido a que las fórmulas para calcular el producto exterior (3.12) e interior (3.15) de k -vectores no son válidas en el álgebra conforme, con la base $\{e_0, e_1, e_2, e_3, e_\infty\}$, sino que son válidas para $C^{4,1}$, con la base estándar $\{e_1, e_2, e_3, e_4, e_5\}$ de $\mathbb{R}^{4,1}$.

En resumen, el procedimiento general del paquete es:

1. Entradas en la base $\{e_0, e_1, e_2, e_3, e_\infty\}$.
2. Transformación de coordenadas $\{e_0, e_1, e_2, e_3, e_\infty\} \longrightarrow \{e_1, e_2, e_3, e_4, e_5\}$.
3. Cálculos en $C^{4,1}$.
4. Transformación de coordenadas $\{e_1, e_2, e_3, e_4, e_5\} \longrightarrow \{e_0, e_1, e_2, e_3, e_\infty\}$.
5. Salida en la base $\{e_0, e_1, e_2, e_3, e_\infty\}$.

Del anterior procedimiento se advierte que la base del paquete será el paso 3: los cálculos en $C^{4,1}$. De lo expuesto en la Sección 3.4, puede inferirse que la operación principal que se tiene que implementar en primer lugar es el producto geométrico entre multivectores, después es necesario implementar la operación de grado (3.13). Con esas dos operaciones básicas ya definidas, se pueden implementar el producto externo

(3.12) y el producto interno (3.15), con lo que se tiene la base para implementar todas las demás funciones que sean necesarias.

Para implementar el producto geométrico entre multivectores en $C^{4,1}$, hay que tomar en cuenta el hecho de que cualquier multivector A puede descomponerse como una suma de k -vectores:

$$A = \langle A \rangle_0 + \langle A \rangle_1 + \langle A \rangle_2 + \cdots. \quad (4.3)$$

En particular, en $C^{4,1}$ un multivector general A tiene la forma

$$A = a_0 + a_{1,1}e_1 + \cdots + a_{1,5}e_5 + a_{2,1}e_1e_2 + \cdots + a_{2,10}e_1e_5 + \cdots + a_5e_1e_2 \cdots e_5. \quad (4.4)$$

Entonces, por distributividad, para calcular el producto de dos multivectores A y B , basta con saber cómo calcular productos de blades

$$(e_1e_2 \cdots e_r)(e_1e_2 \cdots e_s), \quad (4.5)$$

donde, para el caso particular de $C^{4,1}$, $r, s \leq 5$. Esta es la base de la construcción del álgebra de Clifford que se propone en la Ref. [16], y que se implementará en este trabajo, usando el paradigma de la programación basada en reglas, ya que, junto con la asociatividad del producto geométrico se usarán las relaciones establecidas en la Definición 3.4.6 que, para el caso particular de $C^{4,1}$, son:

$$\begin{aligned} e_i^2 &= 1, \quad \text{para } i = 1, 2, \dots, 4, \\ e_5^2 &= -1, \\ e_ie_j &= -e_je_i, \quad \text{para } i, j = 1, 2, \dots, 5. \end{aligned}$$

Para $C^{4,1}$ definimos los vectores base en *Mathematica* como $e2[]$, de manera que $e_1 = e2[1]$, $e_1e_2e_3 = e2[1, 2, 3]$, etc. Estas funciones serán internas, ya que los vectores base que se usarán para interactuar con el usuario son las del Álgebra Geométrica Conforme: $e[0]$, $e[1]$, $e[2]$, $e[3]$ y $e[\infty]$.

Tomando en cuenta lo anteriormente expuesto, el código en *Mathematica* de las reglas para realizar el producto de blades de la forma (4.5) son:

```
e2[]:=1
e2[i_Integer,j_Integer] := e2[]/;i==j&&i<=4&&EvenQ[Length[{i,j}]]&&i>=0
e2[i_Integer,j_Integer] := e2[i]/;i==j&&i<=4&&OddQ[Length[{i,j}]]&&i>=0
e2[i_Integer,j_Integer] := (-e2[])^(Length[{i,j}]/2)/;i==j&&i>4&&i<=5&&
EvenQ[Length[{i,j}]] && i>=0
e2[i_Integer,j_Integer] :=(-e2[])^((Length[{i,j}]-1)/2) e2[i]/;i==j&&i>4
&&i<=5&&OddQ[Length[{i,j}]] && i>=0
e2[i_Integer,j_Integer]:=0/;i==j&&Max[{i,j}]>5&&AllTrue[{i,j},NonNegative]
e2[i_Integer,j_Integer] :=-e2[j,i]/;i!=j&&i>j&&AllTrue[{i,j},NonNegative]
e2[i_Integer] := Signature[Ordering[{i}]] Apply[e2,Sort[i]]/;!OrderedQ[{i}]
&&AllTrue[i,NonNegative]
e2[i_Integer] := (Times@@(Apply[e2,Gather[{i}],1]/.e2[_]->1))*e2@@(Cases[
Apply[e2,Gather[{i}],1],Except[_Integer]]/.(e2[x_] | a_ e2[x_]) -> x)
/;OrderedQ[{i}]&&!DuplicateFreeQ[{i}]&&AllTrue[{i},Positive]
```

Con esto se tiene la base del paquete para realizar los cálculos en $C^{4,1}$; lo siguiente es definir la operación de grado (3.13), el producto externo (3.12) y el interno (3.15), lo que se expondrá más adelante.

A pesar de que todos los cálculos se realizan en $C^{4,1}$, en el último paso se realiza el cambio de variable (4.2), y para simplificar las expresiones será necesario implementar las operaciones entre blades del tipo (4.5), pero con elementos de la base del Álgebra Geométrica Conforme $\{e_0, e_1, e_2, e_3, e_\infty\}$. Para ello es necesario usar las relaciones establecidas en la Definición 3.5.2. Como ya se mencionó, los vectores base del espacio conforme son $e[0]$, $e[1]$, $e[2]$, $e[3]$ y $e[\infty]$. El producto entre blades (4.5) para esta base se codifica como:

```
e[] := 1
e[i__Integer, j__] := 0 /; Count[{i}, 0] >= 2 || (DuplicateQ0Inf[{j}] &&
  (Max[Position[{j}, 0]] < FirstPosition[{j}, ∞][[1]] || Contiguous0Inf[{i}]))
e[i_, j_] := e[] /; i==j && EvenQ[Length[{i}, j]] && MemberQ[{1, 2, 3}, i]
e[i_, j_] := e[i] /; i==j && OddQ[Length[{i}, j]] && MemberQ[{1, 2, 3}, i]
e[∞, 0] := -2 - e[0, ∞]
e[∞, j_Integer] := -e[j, ∞] /; MemberQ[{1, 2, 3}, j]
e[i_, j_] := -e[j, i] /; i != j && i > j && SubsetQ[{0, 1, 2, 3}, {i}]
e[i__] := Signature[Ordering[{i}]] e @@ Sort[{i}] /; !OrderedQ[{i}] &&
  !DuplicateQ0Inf[{i}] && (!MemberQ[i, 0] || !MemberQ[{i}, ∞]
  || FirstPosition[{i}, 0][[1]] < FirstPosition[{i}, ∞][[1]])
e[i__] := Module[{es = Cases[Apply[e, Gather[{i}], 1], Except[_Integer]]},
  Return[(Times@@Join[Cases[Apply[e, Gather[i], 1], _Integer], Cases[{Times@@
  es}, _Integer]])e@@Cases[es, e[x_]:>x]] /; OrderedQ[{i}] && !DuplicateFreeQ[{i}]
e[i__Integer, ∞, j__Integer, k__] := Signature[Ordering[{j}]] (-2e@@Join[{i},
  DeleteCases[Sort[{j}], 0], {k}] - e@@Join[{i}, {0, ∞}, DeleteCases[Sort[{j}], 0],
  {k}]) /; MemberQ[{j}, 0]
e[i__Integer, ∞, j__Integer, k__] := 0 /; Count[{i}, 0] >= 2 && !MemberQ[j, 0]
  && FirstPosition[{k}, ∞][[1]] < FirstPosition[{k}, 0][[1]]
```

Una vez establecido el esquema general, a continuación se describen todas las funciones que componen el código del paquete.

4.3.2 Funciones internas auxiliares.

Las funciones del paquete que serán utilizadas internamente de apoyo a las funciones principales son:

- **DuplicateQ0Inf**, que regresa True si en una lista i dada los elementos $\{0\}$ o $\{\infty\}$ se encuentran repetidos.

```
DuplicateQ0Inf[i_] := Count[i, 0] >= 2 || Count[i, ∞] >= 2
```

- **Contiguous0Inf** que regresa True si los elementos $\{0\}$ o $\{\infty\}$ se encuentran repetidos y aparecen de forma consecutiva.

```
Contiguous0Inf[i_]:=SequenceCases[i,{0,0}]!={}|SequenceCases[i,{∞,∞}]!={}
```

- **GradeQ[m,r]** regresa True si m contiene solo términos de grado r .

```
GradeQ[a_, r_?NumberQ] := If[r === 0, True, False] /; FreeQ[a, e[___]]
GradeQ[e[i___], r_?NumberQ] := If[Length[i] === r, True, False]
GradeQ[(a_: 1) e[i___], r_?NumberQ] := GradeQ[e[i], r] /; FreeQ[a, e[___]]
GradeQ[x_Plus, r_?NumberQ] := And @@ (GradeQ[#, r] & /@ Apply[List, x])
```

4.3.3 Funciones de apoyo

Las siguientes funciones se usan durante los cálculos que se describen más adelante pero pueden ser útiles también para el usuario.

- **From45To0I** realiza la transformación (4.1).

```
From45To0I[a_] := a /; FreeQ[a, e2[___]]
From45To0I[x_ + y_] := From45To0I[x] + From45To0I[y]
From45To0I[a_ e2[i___]] := a From45To0I[e2[i]] /; FreeQ[a, e2[___]]
From45To0I[e2[i___]] := (Replace[e2[i], {e2[4]->e[0]-e[∞]/2, e2[5]->e[0]+
e[∞]/2, e2[1]->e[1], e2[2]->e[2], e2[3]->e[3]}] //Expand)/;Length[{i}]==1
From45To0I[e2[i___]]:=GeometricProduct@@((e2/@List@@e2[i])/.{e2[4]->e[0]-
e[∞]/2,e2[5]->e[0]+e[∞]/2,e2[1]->e[1],e2[2]->e[2], e2[3]->e[3]})//Expand
```

- **From0Ito45** realiza la transformación (4.2).

```
From0Ito45[a_] := a /; FreeQ[a, e[___]]
From0Ito45[x_ + y_] := From0Ito45[x] + From0Ito45[y]
From0Ito45[a_ e[i___]] := a From0Ito45[e[i]] /; FreeQ[a, e[___]]
From0Ito45[e[i___]] := (Replace[e[i], {e[0]->(e2[4]+e2[5])/2, e[∞]->e2[5]-
e2[4], e[1]->e2[1], e[2]->e2[2], e[3]->e2[3]})//Expand)/;Length[{i}]==1
From0Ito45[e[i___]] := (GeometricProduct2@@ ((e/@ List @@ e[i])/.{e[0]->
(e2[4]+e2[5])/2, e[∞]->e2[5]-e2[4], e[1]->e2[1], e[2]->e2[2],
e[3]->e2[3]})//Expand)/;Length[{i}]>1
```

- **MultiplicationTable** regresa la tabla de multiplicaciones de los productos de Clifford en símbolos e .

```
MultiplicationTable[e_] := Grid[Transpose[Insert[Transpose[Prepend[Table[
e[i, j],{i,{0,1,2,3,∞}}, {j,{0,1,2,3,∞}}], Table[e[i],{i,{0,1,2,3,∞}}]]],
{"" ,e[0],e[1],e[2],e[3],e[∞]}, 1]], Frame -> All]
```

- **GFactor** agrupa los términos con $e[___]$ común.

```
GFactor[x_] := Collect[Expand[x], e[___]]
```

- **ToVector** transforma el elemento del espacio conforme del tipo $v=xe[1]+ye[2]+ze[3]$ a la notación estandar de *Mathematica* $\{x,y,z\}$

```
ToVector[v_]:= Table[Coefficient[v,e[i]],{i,3}]/;GradeQ[v,1] &&
Coefficient[v,e[0]]==0&&Coefficient[v,e[∞]]==0
```

4.3.4 Funciones (internas) para el álgebra $C^{4,1}$

Como se discutió previamente, los cálculos en $C^{4,1}$ se basan en el producto de blades, como se muestra en la ecuación (4.5) y cuyo código se presenta en la Sección 4.3.1. Siguiendo esto, `GeometricProduct2[]` aplica la distributividad del producto geométrico para calcular el producto de dos multivectores, enseguida `Grade2[]` extrae el grado de un multivector, para, con esas dos funciones, calcular el producto interno (3.15) con la función `InnerProduct2[]` y externo (3.12) con `OuterProduct2[]`, estas cuatro funciones son las operaciones básicas del álgebra $C^{4,1}$.

- **GeometricProduct2** calcula el producto geométrico en $C^{4,1}$.

```
GeometricProduct2[_] := $Failed
GeometricProduct2[x_,y_] := GeometricProduct2[Expand[x],Expand[y]]/;
x!=Expand[x] || y!=Expand[y]
GeometricProduct2[x_,y_,z_] := Fold[GeometricProduct2,Expand[x],
{Expand[y],z}]/Simplify
GeometricProduct2[x_,y_+z_] := GeometricProduct2[x,y]+GeometricProduct2[x,z]
GeometricProduct2[x_+y_,z_] := GeometricProduct2[x,z]+GeometricProduct2[y,z]
GeometricProduct2[a_,b_] := a b e2[] /; FreeQ[a,e2[___]] && FreeQ[b,e2[___]]
GeometricProduct2[a_,(b_:1) e2[i_/_;SubsetQ[Range[5],{i}]]] := a b e2[i]
/;FreeQ[a,e2[___]] && FreeQ[b,e2[___]]
GeometricProduct2[(a_:1) e2[i_/_;SubsetQ[Range[5],{i}]],b_] := a b e2[i]
/;FreeQ[a,e2[___]] && FreeQ[b,e2[___]]
GeometricProduct2[(a_:1) e2[i_/_;SubsetQ[Range[5],i]],(b_:1) e2[j_/_;
SubsetQ[Range[5],j]]] := a b e2[i,j] /;FreeQ[a,e2[___]] && FreeQ[b,e2[___]]
```

- **Grade2** calcula el grado de un multivector.

```
Grade2[a_,r_/_;Element[r,Integers]] := If[r===0,a,0]/; FreeQ[a,e2[___]]
Grade2[x_,r_/_;Element[r,Integers]] := Grade2[Expand[x],r]/;x!=Expand[x]
Grade2[x_,r_/_;Element[r,Integers]] := 0 /;r<0
Grade2[x_+y_,r_/_;Element[r,Integers]] := Grade2[x,r]+Grade2[y,r]
Grade2[(a_:1) e2[i_/_;SubsetQ[Range[5],{i}]],r_/_;Element[r,Integers]] :=
If[Length[i]===r,a e2[i],0]
```

- **InnerProduct2** calcula el producto interno en $C^{4,1}$.

```
InnerProduct2[_] := $Failed
InnerProduct2[x_,y_] := InnerProduct2[Expand[x],Expand[y]]
/; x!=Expand[x] || y!=Expand[y]
InnerProduct2[x_,y_] := InnerProduct2[Expand[x],Expand[y]]
/; x!=Expand[x] || y!=Expand[y]
InnerProduct2[x_,y_+z_] := InnerProduct2[x,y]+InnerProduct2[x,z]
```

```

InnerProduct2[x_+y_,z_] := InnerProduct2[x,z]+InnerProduct2[y,z]
InnerProduct2[a_,b_] := 0 /; FreeQ[a,e2[___]] && FreeQ[b,e2[___]]
InnerProduct2[a_,(b_:1)e2[i___/;SubsetQ[Range[5],{i}]]]:=0/;FreeQ[a,e2[___]]
InnerProduct2[(b_:1)e2[i___/;SubsetQ[Range[5],{i}]],a_]:=0/;FreeQ[a,e2[___]]
InnerProduct2[(a_:1)e2[i___/;SubsetQ[Range[5],{i}]],(b_:1) e2[j___/;
SubsetQ[Range[5],{j}]]] := Grade2[ab e2[i,j],Abs[Length[{i}]-
Length[{j}]]]/;FreeQ[a,e2[___]] && FreeQ[b,e2[___]]

```

- **OuterProduct2** calcula el producto externo en $C^{4,1}$.

```

OuterProduct2[_] := $Failed
OuterProduct2[x_,y_] := OuterProduct2[Expand[x],Expand[y]]
/;x!=Expand[x]||y!=Expand[y]
OuterProduct2[x_,y_,z_] := Fold[OuterProduct2,x,{y,z}]/Simplify
OuterProduct2[x_,y_+z_] := OuterProduct2[x,y]+OuterProduct2[x,z]
OuterProduct2[x_+y_,z_] := OuterProduct2[x,z]+OuterProduct2[y,z]
OuterProduct2[a_,b_] := a b /; FreeQ[a,e2[___]] && FreeQ[b,e2[___]]
OuterProduct2[a_,(b_:1) e2[i___/;SubsetQ[Range[5],{i}]]] := a b e2[i]
/;FreeQ[a,e2[___]]
OuterProduct2[(b_:1) e2[i___/;SubsetQ[Range[5],{i}]],a_] := a b e2[i]
/;FreeQ[a,e2[___]]
OuterProduct2[(a_:1) e2[i___/;SubsetQ[Range[5],{i}]],(b_:1) e2[j___/;
SubsetQ[Range[5],{j}]]] := Grade2[a b e2[i,j],Length[{i}]+Length[{j}]]
/;FreeQ[a,e2[___]] && FreeQ[b,e2[___]]

```

Las siguientes funciones son de utilidad para simplificar cálculos:

- **GFactor2** agrupa términos comunes en $e2[___]$.

```
GFactor2[x_] := Collect[Expand[x],e2[___]]
```

- **Pseudoscalar2** es la definición del pseudoescalar.

```
Pseudoscalar2[n_/;Element[n,Integers]&&n>0] := e2@@(Range[n])
```

- **HomogeneousQ2** regresa True si el multivector es homogéneo.

```
HomogeneousQ2[x_,r_/;Element[r,Integers]&&NonNegative[r]] :=
Simplify[Expand[x]==Grade2[x,r]]
```

4.3.5 Funciones del Álgebra Geométrica Conforme

Las siguientes funciones constituyen la parte principal del paquete, en el sentido que son para uso del usuario y realizan cálculos en la base $\{e_0, e_1, e_2, e_3, e_\infty\}$, que se denotan como $e[0]$, $e[1]$, $e[2]$, $e[3]$ y $e[\infty]$.

El producto geométrico de blades, del tipo (4.5), ya se estableció en la Sección 4.3.1. Hay que recordar que, para calcular el producto geométrico entre multivectores, se utiliza la propiedad de la distributividad:

- **GeometricProduct** realiza el producto geométrico entre dos o mas multivectores.

```

GeometricProduct[x_,y_] := GeometricProduct[Expand[x], Expand[y]]/;
  x!=Expand[x] || y!=Expand[y]
GeometricProduct[x_,y_,z_] := Fold[GeometricProduct,Expand[x],{Expand[y],
  z}] // Simplify
GeometricProduct[x_,y_+ z_] := GeometricProduct[x,y]+GeometricProduct[x,z]
GeometricProduct[x_+y_,z_] := GeometricProduct[x,z]+GeometricProduct[y,z]
GeometricProduct[a_,b_] := ab e[] /; FreeQ[a,e[___]]&&FreeQ[b, e[___]]
GeometricProduct[a_,(b_:1) e[i___/;SubsetQ[{0,1,2,3,∞},{i}]]]:=ab e[i]
  /;FreeQ[a,e[___]]&&FreeQ[b,e[___]]
GeometricProduct[(a_:1)e[i___/;SubsetQ[{0,1,2,3,∞},{i}]],b_]:=ab e[i]
  /;FreeQ[a,e[___]]&&FreeQ[b,e[___]]
GeometricProduct[(a_:1)e[i___/;SubsetQ[{0,1,2,3,∞},{i}]],(b_:1) e[j___/;
  SubsetQ[{0,1,2,3,∞},{j}]]] :=ab e[i,j]/;FreeQ[a,e[___]]&&FreeQ[b,e[___]]

```

Para tener la base del paquete, sólo hace falta la operación de grado:

- **Grade[m,r]** extrae el termino de grado r de un multivector m.

```

Grade[a_, r_?NumberQ] := If[r === 0, a, 0] /; FreeQ[a, e[___]]
Grade[x_, r_?NumberQ] := Grade[Expand[x], r] /; x != Expand[x]
Grade[x_ + y_, r_?NumberQ] := Grade[x, r] + Grade[y, r]
Grade[(a_:1) e[i___/;SubsetQ[{0,1,2,3,∞},{i}]], r_?NumberQ] :=
  If[Length[{i}]===r,a e[i],0]

```

Con eso ya es posible definir los productos externos e internos:

- **OuterProduct** regresa el producto externo entre dos o mas multivectores.

```

OuterProduct[_] := $Failed
OuterProduct[x_,y_] := OuterProduct[Expand[x],Expand[y]]/; x!=Expand[x] ||
  y!=Expand[y]
OuterProduct[x_,y_] := From45To0I[OuterProduct2[From0Ito45[x],From0Ito45[y]]
  //Expand]//Expand
OuterProduct[x_,y_,z_] := Fold[OuterProduct,x,y,z] //Simplify

```

- **InnerProduct** regresa el producto externo entre dos o mas multivectores.

```

InnerProduct[_] := $Failed
InnerProduct[x_,y_] := InnerProduct[Expand[x],Expand[y]]
  /; x!=Expand[x] || y!=Expand[y]
InnerProduct[x_,y_] := From45To0I[InnerProduct2[From0Ito45[x],From0Ito45[y]]]

```

Una vez definidas estas funciones básicas, es posible definir un buen número de funciones del Álgebra Geométrica Conforme, como:

- **Reversión** regresa la reversión de un multivector.

```

Reversion[_] := $Failed
Reversion[x_] := Reversion[Expand[x]] /; x != Expand[x]
Reversion[a_] := a /; FreeQ[a, e[_]]
Reversion[(a_: 1) e[i_]; SubsetQ[0,1,2,3,∞,{i}]] := a e@@Reverse[{i}]/;
  FreeQ[a, e[_]]
Reversion[x_ + y_] := Reversion[x] + Reversion[y]

```

- **Involution** es la involución de grado (†) de un multivector.

```

Involution[_] := $Failed
Involution[x_] := Involution[Expand[x]]/;x != Expand[x]
Involution[a_] := a /; FreeQ[a, e[_]]
Involution[(a_:1) e[i_]; SubsetQ[0,1,2,3,∞,{i}]] := (-1)^Length[i] a e[i]
  /;FreeQ[a, e[_]]
Involution[x_ + y_] := Involution[x] + Involution[y]

```

- **Magnitude[v]** regresa la magnitud un multivector v de la forma (3.22).

```

Magnitude[x_] := Sqrt[Grade[GeometricProduct[Reversion[x], x], 0]]

```

- **MultivectorInverse** regresa el inverso de un multivector (3.23), si es que existe.

```

MultivectorInverse[_] := $Failed
MultivectorInverse[x_] := MultivectorInverse[Expand[x]] /; x != Expand[x]
MultivectorInverse[x_] := Simplify[Reversion[x] / Magnitude[x]^2]
  /; Magnitude[x] != 0

```

- El ppsudoescalar (**I5**) y el inverso del ppsudoescalar (**I5i**) del Álgebra Geométrica Conforme $\mathbb{G}^{4,1}$ están definidos siguiendo [24] de la forma:

```

I5:=OuterProduct[e[0],OuterProduct[e[1],e[2],e[3]],e[∞]]
I5i:=OuterProduct[e[0],MultivectorInverse[OuterProduct[e[1],e[2],e[3]]],e∞]]

```

- **Dual** regresa el dual de un multivector.

```

Dual[x_] := -InnerProduct[x, I5]

```

- **DDual** desdualiza un multivector.

```

DDual[x_] := InnerProduct[x, -e[1,2,3] + e[0,1,2,3,infty]]

```

- **Rotation[x,a,b,theta]** rota el vector x en un ángulo theta (en radianes), sobre el plano definido por a y b. El sentido de la rotación es de a a b. Si no se da el valor de theta, se toma el valor de default que es el ángulo entre a y b.

```
Rotation[x_, a_, b_, angle_:Automatic] := Module[{plane=OuterProduct[a, b]},
  If[angle===Automatic,theta=VectorAngle[ToVector[a],ToVector[b]],theta=angle];
  Return[GFactor[GeometricProduct[Cos[theta/2]-(plane/Magnitude[plane])*
  Sin[theta/2], x, Cos[theta/2] + (plane/Magnitude[plane])*Sin[theta/2]]]]];
  GradeQ[x,1] &&GradeQ[a,1] && GradeQ[b,1]
```

4.3.6 Funciones gráficas.

Finalmente se implementan diversas funciones que servirán para mostrar gráficamente objetos geométricos y de apoyo para su manipulación con el Álgebra Geométrica Conforme.

- **circle3D[centro,radio]** genera un círculo, dado un centro y un radio.

```
circle3D[centre_:{0,0,0}, radius_:1, normal_:{0,0,1}, angle_:{0,2 Pi}] :=
  Composition[Line,Map[RotationTransform[{{0,0,1},normal},centre],#]&, Map[
  Append[#, Last@centre] &, #] &, Append[DeleteDuplicates[Most@#], Last@#] &,
  Level[#,{ -2}]&,MeshPrimitives[#,1]&,DiscretizeRegion,If][First@Differences@
  angle>= 2Pi,Circle[Most@centre,radius], Circle[Most@centre,radius,angle]]
```

- **ToBasis** transforma vectores de la notación estándar de *Mathematica*, $\{x, y, z\}$, a la notación del espacio conforme $x e[1] + y e[2] + z e[3]$.

```
ToBasis[x_?VectorQ] := Dot[x, List@@e /@ Range[Length[x]]]
```

- **ToVector** transforma elementos de la notación del espacio conforme, $x e[1] + y e[2] + z e[3]$, a la notación estándar de *Mathematica* x, y, z .

```
ToVector[v_,d_:3] := Table[Coefficient[v,e[i]], {i,d}]
```

- **Centere** encuentra el centro de un círculo, representado en la notación del espacio conforme.

```
Centere[x_] := -1/2 ToVector[GeometricProduct[x, e[∞], x], 3]
```

- **Radio** encuentra el radio de un círculo, representado en la notación del espacio conforme.

```
Radio[x_] := Sqrt[Magnitude[x]^2/Magnitude[InnerProduct[e[∞], x]]^2]
```

- **Arco** genera un arco, dado un centro, un radio, una normal y un ángulo.

```
Arco[centre_:{0,0,0},radius_:1,normal_:{0,0,1},angle_:{0,2 Pi}, int_:{0,1}]
:=Module[{index},index=Flatten[Table[Position[{circle3D[centre,radius,normal,
  angle]][[1,1,;;,3]],Select[{circle3D[centre,radius,normal,angle]][[1,1,;;,3]]
  ,int[[1]]<#<int[[2]]&][[i]]], {i, 1, Length[Select[{circle3D[centre, radius,
  normal,angle]][[1,1,;;,3]],int[[1]]<#<int[[2]]&}}]];Line[{circle3D[centre,
  radius, normal, angle]][[1, 1, index]]]
```

- **Lines** encuentra la ecuación de una línea que pasa por dos puntos dados.

```

Lines[p1_, p2_] := Module[{X, p, L, s}, X = x e[1] + y e[2] + z e[3];
  p = e[0] + X + (Magnitude[X]^2/2) e[∞]; L=OuterProduct[p,p1,p2,e[∞]];
  {(ToVector[p1][[1]]-ToVector[p2][[1]])t,y,z}/.
  ((s=Solve[{Coefficient[L,e[0,1,2,∞]]==0, Coefficient[L,e[0,1,3,∞]]==0,
  Coefficient[L,e[0,2,3,∞]]==0,Coefficient[L,e[1,2,3,∞]]==0}, {x,y,z}]
  //Quiet)[[1]])/.x -> (ToVector[p1][[1]] - ToVector[p2][[1]]) t)

```

- **LinesPlot** gráfica una línea que pasa por dos puntos dados.

```

LinesPlot[p1_, p2_] := Show[{ParametricPlot3D[Lines[p1, p2], {t, -1, 2},
  Boxed->False, Axes->False], Graphics3D[{{PointSize[Large],
  Point[ToVector[p1]]}, {PointSize[Large], Point[ToVector[p2]]}}]}]

```

- **Plane** encuentra la ecuación de un plano que pasa por tres puntos dados.

```

Plane[p1_, p2_, p3_] := Module[{X,p,L,s}, X= xe[1]+ye[2]+ze[3];
  p= e[0]+X+(Magnitude[X]^2/2)e[∞]; L= OuterProduct[p,p1,p2,p3,e[∞]];
  c = Solve[Coefficient[plane, e[1, 2, 3]] == 0, z][[1]]; z /. c]

```

- **Sphere** encuentra la ecuación de una esfera que pasa por cuatro puntos dados.

```

Spheres[p1_, p2_, p3_, p4_] := Module[{X,p,L,S,s}, X = xe[1]+ye[2]+ze[3];
  p= e[0]+X+(Magnitude[X]^2/2)e[∞]; S= OuterProduct[p,p1,p2,p3,p4];
  s = Coefficient[S, e[1, 2, 3]]]

```

5. Aplicaciones de CGAlgebra

En este capítulo, tiene como objetivo ilustrar y ejemplificar los aspectos matemáticos fundamentales del Álgebra Geométrica. Se enfoca en la representación de objetos geométricos, su manipulación y sus transformaciones, utilizando el paquete CGAlgebra que se presentó en el Capítulo 4.

5.1 Empezando con CGAlgebra

CGAlgebra es de libre acceso y se encuentra disponible para su descarga en:

<https://github.com/jlaragonvera/Geometric-Algebra>

Una guía del paquete se encuentra en la Ref. [19].

En un notebook de *Mathematica*, el paquete se carga con el comando:

```
<<"DIR/CGAlgebra.m"
```

donde DIR es la ruta completa (*full path*) del directorio donde el paquete se localiza.

Como se mencionó en la Sección 4.3.1, la base de $\mathbb{G}^{4,1}$ es:

```
{e[0], e[1], e[2], e[3], e[∞]}
```

Para el producto geométrico de elementos base (blades) usamos $e[0, 1, \dots]$. Por ejemplo, tendremos que: $e_0 e_1 e_2 = e[0, 1, 2]$. De igual forma, los resultados de cualquier cálculo se dan en términos del producto geométrico de elementos base.

Otro ejemplo es el vector $e_0 + 2e_1 - ae_3 + e_\infty$, que se escribe como: $e[0] + 2 e[1] - a e[3] + e[\infty]$, y el multivector $a + 5e_1 + e_1 e_2 e_3$, que se escribe como: $a + 5 e[1] + e[1,2,3]$.

Para empezar, podemos probar algunas de las relaciones de la ecuación (3.28):

```

In[1]:= e[2,1]
Out[1]:= -e[1,2]
In[2]:= e[∞,0]
Out[2]:= -2-e[0,∞]
In[3]:= e[∞,∞]
Out[3]:= 0
In[4]:= e[0,0]
Out[4]:= 0
In[5]:= e[1,∞,2,0]
Out[5]:= 2e[1,2]+e[0,1,2,∞]

```

5.1.1 Producto Geométrico

El producto geométrico entre los multivectores

$$A = e_1 e_2 e_3 + a e_\infty e_3 e_2 \quad (5.1a)$$

$$B = a e_2 \quad (5.1b)$$

$$F = 2 \quad (5.1c)$$

$$G = 4 + e_1 e_3 \quad (5.1d)$$

se calcula como:

```

In[6]:=A=e[1,2,3] + a e[∞,3,2];
In[7]:=B=a e[2];
In[8]:=F=3;
In[9]:=G=4+e[1,3];
In[10]:= GeometricProduct[A,B,F,G]
Out[10]:= 3a - 12a e[1,3] + 3a^2 e[1,∞] - 12a^2 e[3,∞]

```

El pseudoscalar $I_5 = e_0 \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_\infty$ se calcula como:

```

In[11]:= OuterProduct[e[0], e[1], e[2], e[3], e[∞]]
Out[11]:= -e[1,2,3] + e[0,1,2,3,∞]

```

El ppseudoescalar, además, está predefinido en el paquete como I5:

```

In[12]:= I5
Out[12]:= -e[1,2,3] + e[0,1,2,3,∞]

```

5.1.2 Producto interno y externo

El producto interno entre los multivectores A y B en (5.1a) y (5.1b), respectivamente, se calcula como:

```

In[13]:=InnerProduct[A,B]
Out[13]:= -a e[1,3]- a^2 e[3, ∞]

```

Y el producto externo:

```
In[14] := OuterProduct[A, B]
Out[14] := 0
```

Las contracciones establecidas en la proposición 3.3.4, pueden probarse con GCAIgebra, definiendo:

```
In[15] := x = x1 e[1] + x2 e[2] + x3 e[3];
In[16] := y = y1 e[1] + y2 e[2] + y3 e[3];
In[17] := z = z1 e[1] + z2 e[2] + z3 e[3];
In[18] := a = a1 e[1] + a2 e[2] + a3 e[3];
In[19] := b = b1 e[1] + b2 e[2] + b3 e[3];
In[20] := c = c1 e[1] + c2 e[2] + c3 e[3];
```

De forma que, para la contracción por un vector x , tendremos:

```
In[21] := InnerProduct[x, a]
Out[21] := 0
```

```
In[22] := InnerProduct[x, OuterProduct[a, b]] ==
InnerProduct[x, a] b - InnerProduct[x, b] a
Out[22] := 0
```

```
In[23] := InnerProduct[x, OuterProduct[a, b, c]] ==
InnerProduct[x, a] OuterProduct[b, c] +
InnerProduct[y, b] OuterProduct[c, a] +
InnerProduct[x, c] OuterProduct[a, b] // FullSimplify
Out[23] := True
```

Para la contracción por un bivector $x \wedge y$ tendremos:

```
In[24] := InnerProduct[OuterProduct[x, y], a]
Out[24] := 0
```

```
In[25] := InnerProduct[OuterProduct[x, y], a]
== InnerProduct[a, y] x - InnerProduct[a, x] y
Out[25] := True
```

```
In[26] := InnerProduct[OuterProduct[x, y], OuterProduct[a, b]] == InnerProduct[x, b]
InnerProduct[y, a] - InnerProduct[x, a] InnerProduct[y, b] // FullSimplify
Out[26] := True
```

```
In[27] := InnerProduct[OuterProduct[x, y], OuterProduct[a, b, c]] ==
(InnerProduct[x, c] InnerProduct[y, b] - InnerProduct[x, b] InnerProduct[y, c]) a +
(InnerProduct[x, a] InnerProduct[y, c] - InnerProduct[x, c] InnerProduct[y, a]) b +
(InnerProduct[x, b] InnerProduct[y, a] - InnerProduct[x, a] InnerProduct[y, b]) c
Out[27] := True
```

Y para la contracción por un trivector $x \wedge y \wedge z$ tendremos:

```
In[28]:= InnerProduct[OuterProduct[x,y,z],a]
Out[28]:= 0
```

```
In[29]:= InnerProduct[OuterProduct[x,y,z],a]==InnerProduct[a,x]
OuterProduct[y,z]+InnerProduct[a,y] OuterProduct[z,x]+
InnerProduct[a,z] OuterProduct[x,y]//Expand
Out[29]:= True
```

```
In[30]:= InnerProduct[OuterProduct[x,y,z],OuterProduct[a,b]]==
(InnerProduct[a,z] InnerProduct[b,y]-InnerProduct[a,y] InnerProduct[b,z])x+
(InnerProduct[a,x] InnerProduct[b,z]-InnerProduct[a,z] InnerProduct[b,x])y+
(InnerProduct[a,y] InnerProduct[b,x]-InnerProduct[a,x] InnerProduct[b,y])z
Out[30]:= True
```

```
In[31]:= InnerProduct[OuterProduct[x,y,z], OuterProduct[a,b,c]]==
InnerProduct[x,a] InnerProduct[y,c] InnerProduct[z,b] +
InnerProduct[x,b] InnerProduct[y,a] InnerProduct[z,c] +
InnerProduct[x,c] InnerProduct[y,b] InnerProduct[z,a] -
InnerProduct[x,a] InnerProduct[y,b] InnerProduct[z,c] -
InnerProduct[x,b] InnerProduct[y,c] InnerProduct[z,a] -
InnerProduct[x,c] InnerProduct[y,a] InnerProduct[z,b]
Out[31]:= True
```

5.2 Representación de objetos geométricos

En el Álgebra Geométrica Conforme los objetos geométricos, como líneas, planos, círculos, esferas y pares de puntos en \mathbb{R}^3 , se representan de manera sencilla mediante identidades algebraicas, como veremos a continuación.

5.2.1 Representación de puntos

Definición 5.2.1 En $\mathbb{G}^{4,1}$ un punto p , en la posición (x, y, z) en \mathbb{R}^3 , está representado por el elemento

$$p = e_0 + P + \frac{1}{2} \|P\|^2 e_\infty, \quad (5.2)$$

donde $P = xe_1 + ye_2 + ze_3$.

Que se define de la siguiente forma:

```
In[32]:= P = x e[1] + y e[2] + z e[3];
In[33]:= p = e[0] + P + (GeometricProduct[P,P]/2) e[∞];
```

La representación de un punto en el Álgebra Geométrica Conforme está estrechamente vinculada al vector de coordenadas en el espacio euclidiano tridimensional. De hecho, el punto se expresa en términos de su vector de coordenadas, tal como se utilizaría en el espacio euclidiano tridimensional. Esta relación se deriva de la naturaleza de la representación conforme, donde el espacio euclidiano tridimensional se utiliza como base para construir el Álgebra Geométrica Conforme.

Se puede reescribir la ecuación (5.2) como:

$$\frac{p}{\|P\|^2/2} = \frac{e_0}{\|P\|^2/2} + \frac{P}{\|P\|^2/2} + e_\infty, \quad (5.3)$$

la cual representa el mismo punto en \mathbb{R}^3 . En el límite $\|P\|^2 \leftarrow \infty$, esta expresión tiende a e_∞ , por lo que este símbolo se identifica con un punto en el infinito en \mathbb{R}^3 ; y en cualquier dirección que nos movamos, llegaremos a un único punto: e_∞ .

Igualmente, si hacemos $P = 0$ en la ecuación 5.2 tenemos $p = e_0$. Por lo tanto, el símbolo e_0 se identifica con el origen en \mathbb{R}^3 .

Un punto en $\mathbb{G}^{4,1}$ tiene norma 0, es decir $\|p\|^2 = 0$, esto se puede comprobar con el paquete:

```
In[34]:= GeometricProduct[p, p] // Simplify
Out[34]:= 0
```

De la misma forma, de la ecuación (3.32) se obtiene que el producto interior de p , dado en (5.2), con otro punto definido por $q = e_0 + Q + \|Q\|^2 e_\infty / 2$ es

$$p \cdot q = (P \cdot Q) - \frac{1}{2}\|P\|^2 - \frac{1}{2}\|Q\|^2 = -\frac{1}{2} \langle P - Q, P - Q \rangle = -\frac{1}{2}\|P - Q\|^2, \quad (5.4)$$

lo que se puede verificar así:

```
In[35]:= Q = x1 e[1] + y1 e[2] + z1 e[3];
In[36]:= q = e[0] + Q + (GeometricProduct[Q,Q]/2) e[∞];
In[37]:= InnerProduct[p,q]//FullSimplify
Out[37]:= 1/2 (-(x-x1)^2 - (y-y1)^2-(z-z1)^2)
```

Por lo tanto, la distancia al cuadrado entre las posiciones en \mathbb{R}^3 de P y Q se expresa en la forma:

$$\|P - Q\|^2 = -2(p \cdot q). \quad (5.5)$$

El punto en In[33] puede mostrarse en forma gráfica, con $x = 1$, $y = 1$ y $z = 1$ con la función:

```
In[38]:=Graphics3D[{PointSize[Large],Point[ToVector[p]/.{x1->1,y->1,z->1}]}]
```

El resultado se muestra en la Fig. 5.1.

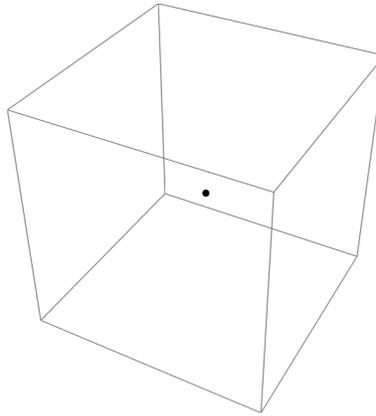


Figure 5.1: Representación gráfica de un punto.

5.2.2 Representación directas de líneas

Definición 5.2.2 Una línea L que pasa por 2 puntos, p_1 y p_2 , se representa como

$$L = p_1 \wedge p_2 \wedge e_\infty. \quad (5.6)$$

Entonces, un punto p en la línea satisface

$$p \wedge L = 0. \quad (5.7)$$

Para verificar esto, definimos:

$$P = (x, y, z), \quad P_1 = (x_1, y_1, z_1), \quad P_2 = (x_2, y_2, z_2),$$

$$p = e_0 + P + \frac{\|P\|^2}{2} e_\infty, \quad p_1 = e_0 + P_1 + \frac{\|P_1\|^2}{2} e_\infty, \quad p_2 = e_0 + P_2 + \frac{\|P_2\|^2}{2} e_\infty,$$

```
In[38]:= P = x e[1] + y e[2] + z e[3];
In[39]:= P1 = x1 e[1] + y1 e[2] + z1 e[3];
In[40]:= P2 = x2 e[1] + y2 e[2] + z2 e[3];
In[41]:= p = e[0] + P + (Magnitude[P]^2/2) e[∞];
In[42]:= p1 = e[0] + P1 + (Magnitude[P1]^2/2) e[∞];
In[43]:= p2 = e[0] + P2 + (Magnitude[P2]^2/2) e[∞];
```

Con lo que la ecuación de la línea que pasa por p_1 y p_2 se obtiene así:

```
In[44]:= line = OuterProduct[p, p1, p2, e[∞]] // FullSimplify
Out[44]:= (-x1 y + x2 y + x y1 - x2 y1 - x y2 + x1 y2) e[1,2] +
(-x1 z + x2 z + x z1 - x2 z1 - x z2 + x1 z2) e[1,3] +
(-y1 z + y2 z + y z1 - y2 z1 - y z2 + y1 z2) e[2, 3]
(x2 y+x y1-x2 y1-x y2+x1(-y+y2)) e[0,1,2,∞] +
(-x1 z+x2 z+x z1-x2 z1-x z2+x1 z2) e[0,1,3,∞] +
```

```
(-y1 z+y2 z+y z1-y2 z1-y z2+y1 z2) e[0,2,3,∞] +
(-x2 y1 z+x1 y2 z+x2 y z1-x y2 z1-x1 y z2+x y1 z2) e[1,2,3,∞]
```

Debido a (5.7), los coeficientes de los 2-vectores y de los 4-vectores deben ser cero:

```
In[45]:= Solve[ {Coefficient[line,e[1,2]]==0, Coefficient[line,e[1,3]]==0,
Coefficient[line,e[2,3]]==0, Coefficient[line,e[0,1,2,∞]] == 0,
Coefficient[line,e[0,1,3,∞]] == 0, Coefficient[line,e[0,2,3,∞]] == 0,
Coefficient[line,e[1,2,3,∞]] == 0},{x, y, z}]/Quiet
Out[45]:= {{y -> -((x (-y1 + y2))/(x1 - x2)) - (x2 y1 - x1 y2)/(x1 - x2),
z -> -((x (-z1 + z2))/(x1 - x2)) - (x2 z1 - x1 z2)/(x1 - x2)}
```

Esto es:

$$y = \frac{x(y_1 - y_2)}{x_1 - x_2} - \frac{x_2 y_1 - x_1 y_2}{x_1 - x_2},$$

$$z = \frac{x(z_1 - z_2)}{x_1 - x_2} - \frac{x_2 z_1 - x_1 z_2}{x_1 - x_2}.$$

Si definimos $x = (x_1 - x_2)t$, se obtiene que las ecuaciones paramétricas de las líneas son:

$$x = (x_1 - x_2)t,$$

$$y = (y_1 - y_2)t - \frac{x_2 y_1 - x_1 y_2}{x_1 - x_2},$$

$$z = (z_1 - z_2)t - \frac{x_2 z_1 - x_1 z_2}{x_1 - x_2}.$$

Esto también puede hacerse con el comando `Lines[]`:

```
In[46]:= Lines[p1, p2]
Out[46]:= {t (x1 - x2), -t (y2 - y1) - \frac{x2 y1 - x1 y2}{x1 - x2}, -t (z2 - z1) - \frac{x2 z1 - x1 z2}{x1 - x2}}
```

Por ejemplo si definimos $x_1 = 1$, $y_1 = z_1 = x_2 = 0$, $y_2 = 1$ y $z_2 = 0$ tendremos:

```
In[47]:= s=Lines[p1,p2]/. {x1->1,y1->0,z1->0,x2->0,y2->1,z2->0}
Out[47]:= {t, 1-t, 0}
```

El resultado se muestra en la Fig. 5.2:

```
In[48]:= LinesPlot[p1 /. {x1 -> 1, y1 -> 0, z1 -> 0},
p2 /. {x2 -> 0, y2 -> 1, z2 -> 0}]
```

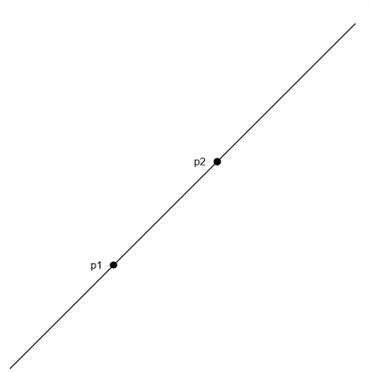


Figure 5.2: Representación gráfica de una línea que pasa por dos puntos.

Siguiendo una lógica similar, la línea que pasa a través de p y se extiende en la dirección u está representada por

$$L = p \wedge u \wedge e_{\infty}. \quad (5.8)$$

5.2.3 Representación directa de planos

Definición 5.2.3 La representación directa del plano Π , que pasa por tres puntos p_1 , p_2 y p_3 es

$$\Pi = p_1 \wedge p_2 \wedge p_3 \wedge e_{\infty}. \quad (5.9)$$

y su ecuación se escribe en términos de un punto p , que pertenece al plano, como

$$p \wedge \Pi = 0. \quad (5.10)$$

Como se hizo anteriormente, sean $P = (x, y, z)$, $P_1 = (x_1, y_1, z_1)$, $P_2 = (x_2, y_2, z_2)$, $P_3 = (x_3, y_3, z_3)$, y

$$p = e_0 + P + \frac{\|P\|^2}{2} e_{\infty}, \quad (5.11a)$$

$$p_1 = e_0 + P_1 + \frac{\|P_1\|^2}{2} e_{\infty}, \quad (5.11b)$$

$$p_2 = e_0 + P_2 + \frac{\|P_2\|^2}{2} e_{\infty}, \quad (5.11c)$$

$$p_3 = e_0 + P_3 + \frac{\|P_3\|^2}{2} e_{\infty}, \quad (5.11d)$$

```
In[49]:= P = x e[1] + y e[2] + z e[3];
In[50]:= P1 = x1 e[1] + y1 e[2] + z1 e[3];
In[51]:= P2 = x2 e[1] + y2 e[2] + z2 e[3];
In[52]:= P3 = x3 e[1] + y3 e[2] + z3 e[3];
In[53]:= p = e[0] + P + (Magnitude[P]^2/2) e[∞];
In[54]:= p1 = e[0] + P1 + (Magnitude[P1]^2/2) e[∞];
```

```
In[55]:= p2 = e[0] + P2 + (Magnitude[P2]^2/2) e[∞];
In[56]:= p3 = e[0] + P3 + (Magnitude[P3]^2/2) e[∞];
```

En \mathbb{R}^3 , la ecuación del plano puede ser obtenida de la siguiente forma:

```
In[57]:= plane = OuterProduct[p, p1, p2, p3, e[∞]] // FullSimplify
Out[57]:= -((-x1 y2 z + x1 y3 z + x y2 z1 - x y3 z1 + x1 y z2 -
x y1 z2 + x y3 z2 - x1 y3 z2 + x3 (-y1 z + y2 z + y z1 -
y2 z1 - y z2 + y1 z2) + (-x1 y + x y1 - x y2 + x1 y2) z3 +
x2 (-y3 z - y z1 + y3 z1 + y1 (z - z3) + y z3))(e[1,2,3]-e[0,1,2,3,∞]))
```

Tomando en cuenta (5.10), igualamos los coeficientes del factor común $e_1 e_2 e_3 - e_0 e_1 e_2 e_3 e_\infty (= I_5)$ a cero y obtenemos la ecuación del plano:

$$\begin{aligned} & ((z_3 - z_2)y_1 + (z_1 - z_3)y_2 + (z_2 - z_1)y_3)x + ((z_2 - z_3)x_1 + (z_3 - z_1)x_2 + (z_1 - z_2)x_3)y \\ & + ((y_3 - y_2)x_1 + (y_1 - y_3)x_2 + (y_2 - y_1)x_3)z + (x_2 y_3 - x_3 y_2)z_1 + (x_3 y_1 - x_1 y_3)z_2 + (x_1 y_2 - x_2 y_1)z_3 = 0. \end{aligned} \quad (5.12)$$

Por ejemplo, si definimos $x_1 = 1, y_1 = z_1 = x_2 = 0, y_2 = 1, z_2 = x_3 = y_3 = 0, z_3 = 1$, tendremos:

```
In[58]:= s=Plane[p1,p2,p3]/. {x1->1,y1->0,z1->0, x2->0,
y2->1,z2->0,x3->0,y3->0,z3->1}
Out[58]:= 1 - x - y
In[59]:= Show[{Plot3D[s,{x,-3,3},{y,-2,2},Axes->False,Boxed-> False]]]
```

El resultado se muestra en la Fig. 5.3.

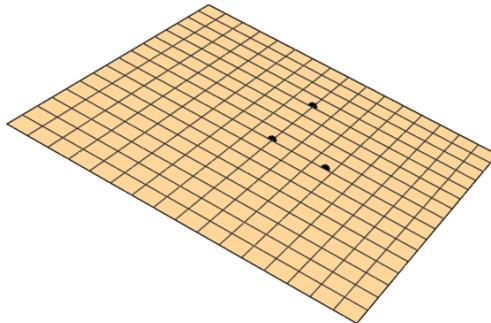


Figure 5.3: Representación gráfica de un plano que pasa por tres puntos.

El último término de la ecuación 5.9, $(\wedge e_\infty)$, indica que un plano pasa por el infinito e_∞ . En este caso, podemos observar que, por ejemplo, el plano que pasa por los puntos p_1 y p_2 y contiene la dirección u está representado por

$$\Pi = p_1 \wedge p_2 \wedge u \wedge e_\infty. \quad (5.13)$$

Esto también se puede escribir como $\pi = -L \wedge u$, en términos de la línea L de la ecuación (5.6).

Del mismo modo, el plano que pasa por el punto p y contiene las direcciones u y v tiene la representación:

$$\Pi = p \wedge u \wedge v \wedge e_\infty \quad (5.14)$$

5.2.4 Representación dual de planos

Definición 5.2.4 En la representación dual, la ecuación del plano es:

$$p \cdot \pi = 0. \quad (5.15)$$

Donde p es un punto que pertenece al plano con la forma (5.2), y

$$\pi = n + h e_\infty, \quad (5.16)$$

Donde $n = n_1 e_1 + n_2 e_2 + n_3 e_3$ es el vector normal del plano π y h es la distancia al origen.

Podemos verificar que (5.15) es la ecuación del plano:

$$p \cdot \pi = (e_0 + P + \frac{1}{2} \|P\|^2 e_\infty) \cdot (n + h e_\infty) = P \cdot n + e_0 \cdot (h e_\infty) = P \cdot n - h. \quad (5.17)$$

Por lo que la ecuación (5.16) es la representación dual del plano $\langle n, P \rangle = h$.

Esto es consecuencia del hecho de que e_0 y e_∞ son ortogonales a todos los elementos base, incluyéndose a sí mismos, claro, excepto por $e_0 \cdot e_\infty = -1$.

Como este espacio es homogéneo, cualquier múltiplo escalar $\alpha\pi$ de la ecuación (5.16), para $\alpha \neq 0$, representa el mismo plano. Haciendo $h = 0$, la ecuación (5.16) se reduce a $\pi = n$. Por lo tanto, un vector en \mathbb{R}^3 en la forma $n = n_1 e_1 + n_2 e_2 + n_3 e_3$, puede verse como la representación de la orientación del plano, independientemente de su posición. De esta forma, en lugar de la superficie normal n y la distancia h desde el origen e_0 , podemos definir un plano especificando dos puntos p_1 y p_2 , de forma que el plano se define como su bisectriz ortogonal. Esto se puede hacer definiendo:

$$\pi = p_1 - p_2. \quad (5.18)$$

Para $p_i = e_0 + P_i + \|P_i\|^2 e_\infty / 2$, con $i = 1, 2$ y p de la ecuación (5.2), vemos que:

$$p \cdot \pi = p \cdot p_1 - p \cdot p_2 = -\frac{1}{2} \|P - P_1\|^2 + \frac{1}{2} \|P - P_2\|^2.$$

Por lo tanto, la ecuación (5.15) significa $\|P - P_1\|^2 = \|P - P_2\|^2$, es decir, la bisectriz ortogonal de P_1 y P_2 , entonces la ecuación (5.18) es su representación dual.

Para comprobar esto, consideremos el plano Π pasando por los puntos $p_i = e_0 + P_i + \|P_i\|^2 e_\infty / 2$, para $i = 1, 2, 3$. En este caso tendremos:

$$\begin{aligned} \Pi &= p_1 \wedge p_2 \wedge p_3 \wedge e_\infty \\ &= (e_0 + P_1 + \frac{1}{2} \|P_1\|^2 e_\infty) \wedge (e_0 + P_2 + \frac{1}{2} \|P_2\|^2 e_\infty) \wedge (e_0 + P_3 + \frac{1}{2} \|P_3\|^2 e_\infty) \wedge e_\infty \\ &= (e_0 + P_1) \wedge (e_0 + P_2) \wedge (e_0 + P_3) \wedge e_\infty, \end{aligned} \quad (5.19)$$

que se puede escribir de la siguiente forma:

$$\Pi = (n_1 e_0 \wedge e_2 \wedge e_3 + n_2 e_0 \wedge e_3 \wedge e_1 + n_3 e_0 \wedge e_1 \wedge e_2 + h e_1 \wedge e_2 \wedge e_3) \wedge e_\infty. \quad (5.20)$$

De forma que tendremos una combinación lineal de $e_0 \wedge e_2 \wedge e_3 \wedge e_\infty$, $e_0 \wedge e_3 \wedge e_1 \wedge e_\infty$, $e_0 \wedge e_1 \wedge e_2 \wedge e_\infty$ y $e_1 \wedge e_2 \wedge e_3 \wedge e_\infty$, con duales dados por:

$$\begin{aligned} (e_0 \wedge e_2 \wedge e_3 \wedge e_\infty)^* &= -e_0 \wedge e_2 \wedge e_3 \wedge e_\infty \cdot e_0 \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_\infty \\ &= -e_0 \wedge e_2 \wedge e_3 \cdot (e_\infty \cdot e_0 \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_\infty) = -e_0 \wedge e_2 \wedge e_3 \cdot (e_\infty \cdot e_0) \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_\infty \\ &= e_0 \wedge e_2 \wedge e_3 \cdot e_1 \wedge e_2 \wedge e_3 \wedge e_\infty = e_0 \wedge e_2 \cdot (e_3 \cdot e_1 \wedge e_2 \wedge e_3 \wedge e_\infty) \\ &= e_0 \wedge e_2 \cdot e_1 \wedge e_2 \wedge (e_3 \cdot e_3) e_\infty = e_0 \wedge e_2 \cdot e_1 \wedge e_2 \wedge e_\infty = e_0 \cdot (e_2 \cdot e_1 \wedge e_2 \wedge e_\infty) \\ &= -e_0 \cdot e_1 \wedge (e_2 \cdot e_2) e_\infty = -e_0 \cdot e_1 \wedge e_\infty = e_1 (e_0 \cdot e_\infty) = -e_1. \end{aligned}$$

$$\begin{aligned} (e_1 \wedge e_2 \wedge e_3 \wedge e_\infty)^* &= -e_1 \wedge e_2 \wedge e_3 \wedge e_\infty \cdot e_0 \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_\infty \\ &= -e_1 \wedge e_2 \wedge e_3 \cdot (e_\infty \cdot e_0 \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_\infty) = -e_1 \wedge e_2 \wedge e_3 \cdot ((e_\infty \cdot e_0) e_1 \wedge e_2 \wedge e_3 \wedge e_\infty) \\ &= e_1 \wedge e_2 \wedge e_3 \cdot e_1 \wedge e_2 \wedge e_3 \wedge e_\infty = e_1 \wedge e_2 \cdot (e_3 \cdot e_1 \wedge e_2 \wedge e_3 \wedge e_\infty) \\ &= e_1 \wedge e_2 \cdot e_1 \wedge e_2 \wedge (e_3 \cdot e_3) e_\infty = e_1 \cdot (e_2 \cdot e_1 \wedge e_2 e_\infty) - e_1 \cdot e_1 \wedge (e_2 \cdot e_2) e_\infty \\ &= -e_1 \cdot e_1 \wedge e_\infty = -(e_1 \cdot e_1) e_\infty = -e_\infty, \end{aligned}$$

y

$$\begin{aligned} (e_0 \wedge e_3 \wedge e_1 \wedge e_\infty)^* &= -e_2, \\ (e_0 \wedge e_1 \wedge e_2 \wedge e_\infty)^* &= -e_3. \end{aligned}$$

De esta forma el dual de (5.20) es:

$$\Pi^* = -n_1 e_1 - n_2 e_2 - n_3 e_3 - h e_\infty = -(n + h e_\infty),$$

lo que concuerda con la representación dual π , excepto por el signo. Pero como el espacio conforme es homogéneo, el cambio de signo o la multiplicación escalar no afecta la representación.

Definición 5.2.5 La representación dual de un plano es:

$$\Pi^* = \pi = -(n + h e_\infty) = -n_1 e_1 - n_2 e_2 - n_3 e_3 - h e_\infty. \quad (5.21)$$

Para comprobar esto, consideremos el plano que pasa por los puntos p_1 , p_2 y p_3 , dados en (5.11)(b)-(d). Este plano se representa por el blade (5.19):

```
In[60]:= P1 = OuterProduct[p1, p2, p3, e[∞]] // GFactor
Out[60]:= (-x2 y1 + x3 y1 + x1 y2 - x3 y2 - x1 y3 + x2 y3) e[1, 2]
+(-x2 z1 + x3 z1 + x1 z2 - x3 z2 - x1 z3 + x2 z3) e[1, 3]
+(-y2 z1 + y3 z1 + y1 z2 - y3 z2 - y1 z3 + y2 z3) e[2, 3]
+(-x2 y1+x3 y1+x1 y2-x3 y2-x1 y3+x2 y3) e[0,1,2,∞]
+(-x2 z1+x3 z1+x1 z2-x3 z2-x1 z3+x2 z3) e[0,1,3,∞]
+(-y2 z1+y3 z1+y1 z2-y3 z2-y1 z3+y2 z3) e[0,2,3,∞]
```

```
+(-x3 y2 z1+x2 y3 z1+x3 y1 z2-x1 y3 z2-x2 y1 z3+x1 y2 z3) e[1,2,3,∞]
```

Donde GFactor[] es una función definida en el paquete que agrupa términos comunes e[. . .]'s.

El dual Π^* puede ser calculado usando la función Dual[]:

```
In[61]:= Pldual = Dual[P1]
Out[61]:= (y2 z1-y3 z1-y1 z2+y3 z2+y1 z3-y2 z3) e[1]
+(-x2 z1+x3 z1+x1 z2-x3 z2-x1 z3+x2 z3) e[2]
+(x2 y1-x3 y1-x1 y2+x3 y2+x1 y3-x2 y3) e[3]
+(x3 y2 z1-x2 y3 z1-x3 y1 z2+x1 y3 z2+x2 y1 z3-x1 y2 z3) e[∞]
```

Ahora, definiendo h y $\mathbf{n} = (n_1, n_2, n_3)$, donde

$$\begin{aligned} n_1 &= y_2 z_1 - y_3 z_1 - y_1 z_2 + y_3 z_2 + y_1 z_3 - y_2 z_3, \\ &= (z_3 - z_2)y_1 + (z_1 - z_3)y_2 + (z_2 - z_1)y_3 \\ n_2 &= x_3 z_1 - x_2 z_1 + x_1 z_2 - x_3 z_2 - x_1 z_3 + x_2 z_3, \\ &= (z_2 - z_3)x_1 + (z_3 - z_1)x_2 + (z_1 - z_2)x_3 \\ n_3 &= x_2 y_1 - x_3 y_1 - x_1 y_2 + x_3 y_2 + x_1 y_3 - x_2 y_3, \\ &= (y_3 - y_2)x_1 + (y_1 - y_3)x_2 + (y_2 - y_1)x_3 \\ h &= (x_3 y_2 z_1 - x_2 y_3 z_1 - x_3 y_1 z_2 + x_1 y_3 z_2 + x_2 y_1 z_3 - x_1 y_2 z_3), \\ &= (x_3 y_2 - x_2 y_3)z_1 + (x_1 y_3 - x_3 y_1)z_2 + (x_2 y_1 - x_1 y_2)z_3, \end{aligned}$$

tendremos:

$$\Pi^* = n_1 e_1 + n_2 e_2 + n_3 e_3 - h e_\infty = \mathbf{n} - h e_\infty. \quad (5.22)$$

Que es la representación dual del plano.

```
In[62]:= n=ToVector[Pldual]/. {x1->1,y1->0,z1->0,x2->0,y2->1,
z2->0,x3->0,y3->0,z3->1};
In[63]:= h=Coefficient[Pldual,e[∞ ]]/. {x1->1,y1->0,z1->0,x2->0,y2->1,
z2->0,x3->0,y3->0,z3->1};
```

Esto lo podemos visualizar en la Fig. 5.4.

```
In[64]:= Show[{ContourPlot3D[n.{x,y,z}-h==0,{x,-2,2},{y,-2,2},{z,-2,2},
ContourStyle->Opacity[0.5], Mesh->False, PlotRange->All,
Boxed->False, Axes->False], Graphics3D[{{PointSize[Large],
Point[ToVector[p1/. {x1->1, y1->0, z1->0}]}],{PointSize[Large],
Point[ToVector[p2/. {x2->0, y2->1, z2->0}]}], {PointSize[Large],
Point[ToVector[p3/. {x3->0, y3->0, z3->1}]}]}]}]
```

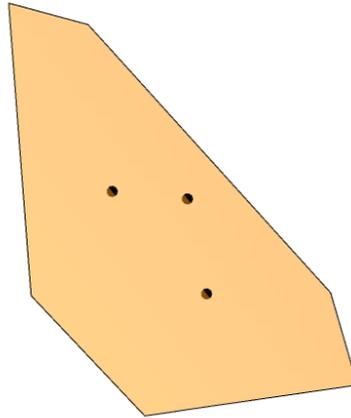


Figure 5.4: Representación gráfica de un plano que pasa por tres puntos.

Si $P = (x, y, z)$ es un punto en el plano, representado por p en (5.11 a), la ecuación del plano es

$$p \cdot \Pi^* = 0,$$

definido como:

```
In[65]:= pdual = n1 e[1] + n2 e[2] + n3 e[3] - h e[∞];
In[66]:= InnerProduct[p, pdual]
Out[66]:= -h + n1 x + n2 y + n3 z
```

Esto es $n_1x + n_2y + n_3z - h = 0$ que, como se esperaba, es igual a (5.12).

5.2.5 Representación dual de líneas

En vez de especificarse por dos puntos, una línea puede ser definida por la intersección de dos planos en su forma dual (5.16). De forma que consideramos dos planos, π_1 y π_2 , la intersección estará definida por el producto externo $\pi_1 \wedge \pi_2$. Por lo tanto:

$$l = \pi_1 \wedge \pi_2, \tag{5.23}$$

es la representación dual de la intersección de dos planos. Como lo podemos ver en la Fig. 5.5.

Usando (5.16) como la representación dual del plano, definimos $\pi_i = n_i + h_i e_\infty$, para $i = 1, 2$ y sea p un punto de la forma (5.2), la ecuación de la intersección será:

$$p \cdot l = p \cdot \pi_1 \wedge \pi_2 = 0 \tag{5.24}$$

También podemos usar $\pi = p_1 - p_2$ para definir representaciones duales de planos. Si hacemos $\pi_1 = p_1 - p_2$ y $\pi_2 = p_2 - p_3$, entonces (5.23) es la representación dual de su intersección.

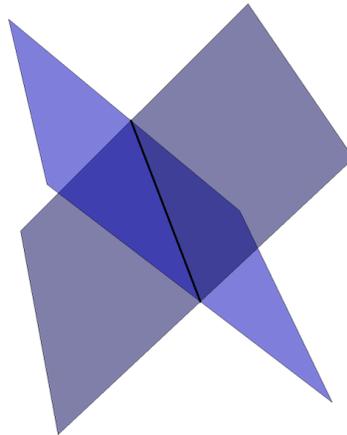


Figure 5.5: Representación gráfica de una línea obtenida mediante la representación dual.

5.2.6 Representación dual de puntos

Sea p un punto de la forma (5.2), y $q = e_0 + Q + \|Q\|^2 e_\infty / 2$, la igualdad $p \wedge q = 0$ implica que p es un múltiplo escalar de q . Por lo tanto, $p \wedge q = 0$ es la ecuación del punto q . Esto significa que $p = e_0 + P + \frac{1}{2}\|P\|e_\infty$ es la representación directa del punto p y, al mismo tiempo, es su representación dual.

Pero un punto también puede considerarse como la intersección de tres planos, por lo que podemos escribir su representación dual en la forma $\pi_1 \wedge \pi_2 \wedge \pi_3$, donde $\pi_i (= \Pi_i^*)$ con $i = 1, 2, 3$, son las representaciones duales de los tres planos Π_i . Esta representación la podemos observar en la Fig. 5.6

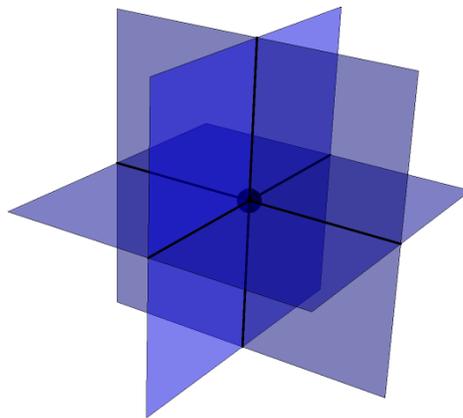


Figure 5.6: Representación gráfica de un punto obtenida mediante la representación dual.

5.2.7 Representación directa de esferas

Definición 5.2.6 Si $p_i = e_0 + P_i + \frac{1}{2}\|P_i\|^2 e_\infty$, con $i = 1, 2, 3, 4$, representa cuatro posiciones P_i en \mathbb{R}^3 , la expresión

$$\Sigma = p_1 \wedge p_2 \wedge p_3 \wedge p_4, \quad (5.25)$$

representa la esfera que pasa por esos puntos, y

$$p \wedge \Sigma = 0 \quad (5.26)$$

es su ecuación.

Esto se puede demostrar de la siguiente manera. Expandiendo el lado izquierdo, vemos que

$$\begin{aligned} p \wedge \Sigma &= (e_0 + P + \frac{1}{2}\|P\|^2 e_\infty) \wedge (e_0 + P_1 + \frac{1}{2}\|P_1\|^2 e_\infty) \wedge (e_0 + P_2 + \frac{1}{2}\|P_2\|^2 e_\infty) \\ &\wedge (e_0 + P_3 + \frac{1}{2}\|P_3\|^2 e_\infty) \wedge (e_0 + P_4 + \frac{1}{2}\|P_4\|^2 e_\infty) = (\dots) e_0 \wedge e_1 \wedge e_2 \wedge e_3 \wedge e_\infty, \end{aligned} \quad (5.27)$$

donde (\dots) son coeficientes que omitimos por su gran longitud. Si, de acuerdo con (5.26), igualamos a cero esta expresión, esto implica $(\dots) = 0$, lo que resulta en la ecuación de una esfera, como veremos enseguida. Así, Σ representa la esfera que pasa a través de los cuatro puntos p_i para $i = 1, 2, 3$.

Si consideramos $\mathbf{P}, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, p, p_1, p_2$ y p_3 , de los ejemplos anteriores, $P_4 = (x_4, y_4, z_4)$ y

$$p_4 = e_0 + \|P_4\| + \frac{\|P_4\|^2}{2} e_\infty \quad (5.28)$$

declarado como:

```
In[67]:= P4 = x4 e[1] + y4 e[2] + z4 e[3];
In[68]:= p4 = e[0] + P4 + (Magnitude[P4]^2/2) e[∞];
```

Entonces (5.28) es:

```
In[69]:= sphere = OuterProduct[p, p1, p2, p3, p4]
Out[69]:= (.....) (e[1, 2, 3] - e[0, 1, 2, 3, ∞])
```

donde (\dots) es una expresión larga que se omitió, y observamos que, de acuerdo con (5.27), $e_1 e_2 e_3 - e_0 e_1 e_2 e_3 e_\infty = I_5$. Si, por ejemplo, consideramos la esfera pasando por los puntos $\mathbf{P}_1 = (1, -1, 3)$, $\mathbf{P}_2 = (4, 1, -2)$, $\mathbf{P}_3 = (-1, -1, 1)$, $\mathbf{P}_4 = (1, 1, 1)$, entonces:

```
In[70]:= sphere2 = sphere/.{x1-> 1, y1-> -1, z1-> 3, x2-> 4, y2-> 1,
z2-> -2, x3-> -1, y3-> -1, z3-> 1, x4-> 1, y4-> 1, z4-> 1}
Out[70]:= -12 (-4+(-5+x) x+y (5+y)+z+ z^2) (e[1,2,3] - e[0,1,2,3,∞])
In[71] := sphere2 = Coefficient[sphere2, (e[1,2,3] - e[0,1,2,3,∞])]
Out[71] := -12 (-4+(-5+x) x+y (5+y)+z+ z^2)
```

Por lo que la ecuación de la esfera es:

$$(x - 5)x + (y + 5)y + (z + 1)z - 4 = 0.$$

Podemos visualizar esto así:

```

In[72]:= r=Sqrt[Abs[FindMaximum[sphere2,{x,y,z}][[1]])
/Coefficient[sphere2,x^2]];
In[73]:= c={x,y,z}/.FindMaximum[sphere2,{x,y,z}][[2]];
In[74]:= Show[{Graphics3D[{Opacity[.5], Sphere[c, r]}, Boxed -> False],
Graphics3D[{{PointSize[Large],Point[ToVector[p1/.{x1->1,y1->-1,z1->3}]}],
{PointSize[Large], Point[ToVector[p2 /. {x2 -> 4, y2 -> 1, z2 -> -2}]}],
{PointSize[Large], Point[ToVector[p3 /. {x3 -> -1, y3 -> -1, z3 -> 1}]}],
{PointSize[Large], Point[ToVector[p4 /. {x4 -> 1, y4 -> 1, z4 -> 1}]}]}]}]}

```

Lo que produce la esfera mostrada en la Fig.5.7.

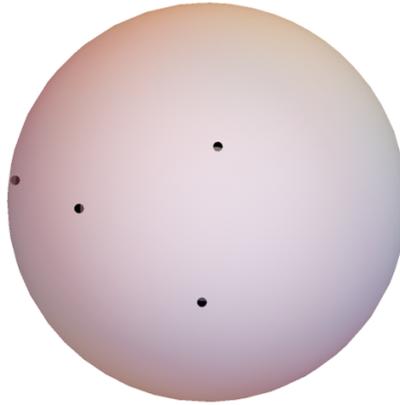


Figure 5.7: Representación gráfica de una esfera que pasa por cuatro puntos.

Además, hay que tomar en cuenta que si hacemos $p_4 = e_\infty$, entonces (5.26) se reduce a (5.9). Este hecho proporciona la interpretación de que un plano es una esfera de radio infinito.

5.2.8 Representación dual de esferas

Definición 5.2.7 Definiendo el punto $c = e_0 + P + \|P\|^2 e_\infty / 2$, la expresión

$$\sigma = c - \frac{r^2}{2} e_\infty \quad (5.29)$$

representa una esfera de radio r con centro en c .

Tomando un punto de la forma (5.2), esta representación se confirma de la siguiente manera:

$$p \cdot \sigma = \langle p, c - \frac{r^2}{2} e_\infty \rangle = (p \cdot c) - \frac{r^2}{2} (p \cdot e_\infty) = -\frac{1}{2} \|P - c\|^2 - \frac{r^2}{2} \quad (5.30)$$

En donde se tomó en cuenta que $(p \cdot e_\infty) = e_0 \cdot e_\infty = -1$.

De este modo, $p \cdot \sigma = 0$, es equivalente a $\|P - c\|^2 = r^2$, es decir, a la ecuación de la esfera. Confirmando así que (5.29) es la representación dual de esta esfera.

Para visualizar esto, consideremos c como el punto p_1 dado en (5.11b). Si $P = (x, y, z)$ es un punto en la esfera, con p dado en (5.11a), observamos que $p \cdot S$ es:

```
In[75]:= S = p1 - r^2/2 e[∞];
In[76]:= InnerProduct[p,S] // FullSimplify
Out[76]:= 1/2 (r^2 - (x-x1)^2 - (y-y1)^2 - (z-z1)^2)
```

Y si definimos $x_1 = 0$, $y_1 = 1$, $z_1 = 1$ y $r = 1$:

```
In[77]:= S2=S /. {x1-> 0,y1-> 1,z1-> 1,r-> 1}
In[78]:= Graphics3D[{Opacity[.5], Sphere[ToVector[S2],Coefficient[S2,e[∞]]]}]
```

Podemos observar la esfera tal y como se muestra en la Fig. 5.8.

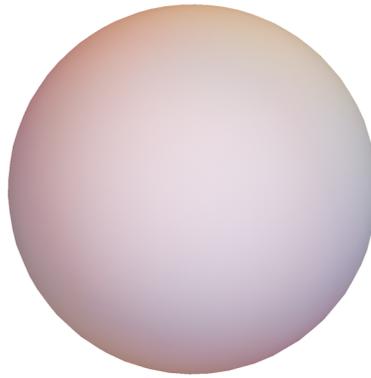


Figure 5.8: Representación gráfica de una esfera obtenida mediante la representación dual.

Definición 5.2.8 El radio de la esfera (5.29) esta dado por:

$$r^2 = \sigma^2, \quad (5.31)$$

y su centro esta dado por:

$$c = \sigma + \frac{r^2}{2} e_\infty. \quad (5.32)$$

Además, si definimos $r = 0$ entonces (5.29) se reduce a $\sigma = c$, que es el punto c , por lo que un punto en \mathbb{R}^3 se interpreta como una esfera de radio 0.

5.2.9 Representación directa de círculos

Definición 5.2.9 Para $p_i = e_0 + P_i + \|P_i\|^2 e_\infty / 2$, con $i = 1, 2, 3$, representando tres posiciones P_i en \mathbb{R}^3 , la expresión

$$S = p_1 \wedge p_2 \wedge p_3, \quad (5.33)$$

representa el círculo que pasa a través de ellos, y su ecuación sera:

$$p \wedge S = 0. \quad (5.34)$$

Esto lo podemos ver de la siguiente forma:

$$p \wedge S = (\dots)e_1 \wedge e_2 \wedge e_3 \wedge e_\infty + (\dots)e_0 \wedge e_2 \wedge e_3 \wedge e_\infty + (\dots)e_0 \wedge e_3 \wedge e_1 \wedge e_\infty \\ + (\dots)e_0 \wedge e_1 \wedge e_2 \wedge e_\infty + (\dots)e_0 \wedge e_1 \wedge e_2 \wedge e_3$$

donde (\dots) son coeficientes que se omiten por su longitud.

La ecuación $p \wedge S = 0$ significa $p \wedge p_1 \wedge p_2 \wedge p_3 = 0$, lo que implica que no existe ninguna esfera que pase por los cuatro puntos p, p_1, p_2 y p_3 . Por eso, el punto p debe estar en el plano que pasa por p_1, p_2 y p_3 . Por lo tanto, S está en un plano. Esto implica que S es una intersección entre una esfera y un plano, es decir, un círculo. Si hacemos $p_3 = e_\infty$ (5.33) se reduce a (5.6). Este hecho proporciona la interpretación de que una línea es un círculo de radio infinito, es decir, que pasa por e_∞ .

Se puede ver que el círculo $p_1 \wedge p_2 \wedge p_3$ tiene la orientación opuesta de $p_1 \wedge p_3 \wedge p_2$, pero la misma que $p_3 \wedge p_1 \wedge p_2$ [3]. Esto es:

```
In[79]:= OuterProduct[p1, p2, p3] == OuterProduct[p3, p1, p2]
Out[79]:= True
```

Podemos visualizar un círculo que pasa por $(1,0,0)$, $(0,1,0)$ y $(0,0,1)$, tal y como se muestra en la Fig. 5.9, de la siguiente forma:

```
In[80]:= c=OuterProduct[p,p1,p2,p3];
In[81]:= c2 = c/.{x1->1,y1->0,z1->0,x2->0,y2->1,z2->0,x3->0,y3->0,z3->1};
In[82]:= c3=Quiet[Solve[{Coefficient[c2,e[1,2,3,∞]]==0,
Coefficient[c2,e[0,2,3,∞]]==0,Coefficient[c2,e[0,1,3,∞]]==0,
Coefficient[c2,e[0,1,2,∞]]==0,Coefficient[c2,e[0,1,2,3]]==0},{x,y,z}]]
Out[82]:= {{y->1/2*(-sqrt(3)*x+2+2*x+1-x+1),z->1/2*(sqrt(3)*x^2+2*x+1-x+1)},
{y->1/2*(sqrt(3)*x^2+2*x+1-x+1),z->1/2*(-sqrt(3)*x^2+2*x+1-x+1)},
In[83]:= Show[{ParametricPlot3D[{x,y,z]/.c3,{x,-10,13},Boxed->False,
Axes->False],Graphics3D[{PointSize[0.03],Blue,
Point[{ToVector[p1],ToVector[p2],ToVector[p3]}]}]}]}
```

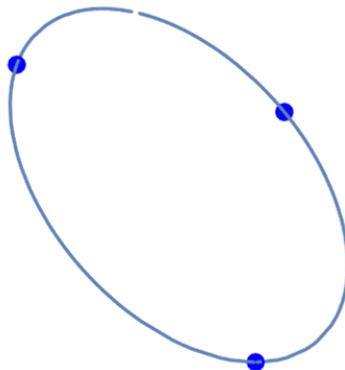


Figure 5.9: Representación gráfica de un círculo que pasa por tres puntos.

5.2.10 Representación directa de un par de puntos

Ahora, consideremos (5.33) pero con dos puntos:

$$PP = p_1 \wedge p_2. \quad (5.35)$$

Y podemos utilizar la siguiente formula [25] para extraer el par de puntos de PP :

$$t_{1,2} = \frac{\pm \sqrt{PP \cdot PP} + PP}{e_\infty \cdot PP}. \quad (5.36)$$

Este es un par de puntos, que puede considerarse como una esfera de baja dimensión. Esto se entiende por el razonamiento de que una esfera es “el conjunto de puntos equidistantes de un punto en \mathbb{R}^3 ” y un círculo es “el conjunto de puntos equidistantes de un punto en \mathbb{R}^2 ”, por tanto, un par de puntos, es decir, “el conjunto de puntos equidistantes de un punto en \mathbb{R}^1 ” es una esfera de dimensión 0. Si uno de los dos puntos se reemplaza por e_∞ , el par de puntos $p \wedge e_\infty$ representa un punto, que se distingue geoméricamente de un punto aislado p (= una esfera de radio 0) dado por (5.2) llamado punto plano.

Puede verse que el par de puntos $p_1 \wedge p_2$ tiene la orientación opuesta de $p_2 \wedge p_1$:

```
In[84] := OuterProduct[p1,p2]==-OuterProduct[p2,p1] // FullSimplify
Out[84] := True
```

5.2.11 Representación dual de círculos

Un círculo se puede definir como la intersección de dos esferas. Podemos escribir las representaciones duales de las dos esferas como $\sigma_i = c_i - r_i^2 e_\infty / 2$, para $i = 1, 2$. Después, obtenemos la representación dual del círculo de la intersección:

$$s = \sigma_1 \wedge \sigma_2. \quad (5.37)$$

Podemos verificar esto de la siguiente manera:

$$\begin{aligned} p \cdot s &= p \cdot \sigma_1 \wedge \sigma_2 = \langle p, \sigma_1 \rangle \sigma_2 - \langle p, \sigma_2 \rangle \sigma_1 \\ &= -\left(\frac{1}{2} \|P - c_1\|^2 - \frac{r_1^2}{2}\right) \left(c_2 - \frac{r_2^2}{2} e_\infty\right) - \left(-\frac{1}{2} \|P - c_2\|^2 + \frac{r_2^2}{2}\right) \left(c_1 - \frac{r_1^2}{2} e_\infty\right). \end{aligned}$$

Como los centros c_1 y c_2 de las dos esferas son distintos, son linealmente independientes y, por lo tanto, sus coeficientes son ambos 0, con lo que se obtiene $\|P - c_i\|^2 = r_i^2$, para $i = 1, 2$, lo que significa que P está en ambas esferas, definiendo así un círculo. Por lo tanto (5.37) es su representación dual. Esto se puede visualizar de la siguiente forma:

```
In[85] := S1=p1- r1^2/2 e[∞]; S2=p2- r2^2/2 e[∞];
In[86] := σ =OuterProduct[S1,S2]/. {x1->1,y1->0,z1->0,
  x2->0,y2->1,z2->0,r1->2,r2->1.5};
In[87] := aux=GeometricProduct[InnerProduct[e[∞],σ],InnerProduct[e[∞],σ]]
In[88] := Graphics3D[{{Opacity[.5],Sphere[Centere[S1 /. {x1->1,y1->0,z1->0,
  x2->0,y2->1,z2->0,r1->2,r2->1.5}], Radio[S1 /. {x1->1,y1->0,z1->0,
  x2->0,y2->1,z2->0,r1->2,r2->1.5}]]}, {Opacity[.5], Sphere[Centere[
  S2 /. {x1->1,y1->0,z1->0,x2->0,y2->1,z2->0,r1->2,r2->1.5}],
  Radio[S2 /. {x1->1,y1->0,z1->0,x2->0,y2->1,z2->0,r1->2,r2->1.5}]]}],
```

```
{circle3D[Centereσ]/aux, Radioσ], ToVector[S1-S2] /.{x1->1,
y1->0, z1->0, x2->0, y2->1, z2->0, r1->2, r2->1.5}}}], Boxed-> False]
```

El resultado se muestra en la Fig. 5.10.

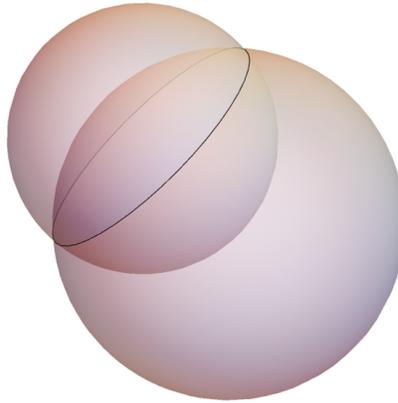


Figure 5.10: Representación gráfica de un círculo, como la intersección de dos esferas.

Otra forma de representar a un círculo es que en lugar de considerarlo como la intersección de dos esferas, lo consideremos como la intersección de una esfera σ y un plano π , para así definirlo por $\sigma \wedge \pi$.

5.2.12 Representación dual de pares de puntos

Un par de puntos puede ser representado como la intersección de 3 esferas:

$$pp = \sigma_1 \wedge \sigma_2 \wedge \sigma_3 \quad (5.38)$$

De manera similar, un par de puntos se considera como la intersección de un círculo S y una esfera σ o un plano π , por lo que su representación dual es $s \wedge \sigma$ o $s \wedge \pi$, donde $s = S^*$ es la representación dual del círculo S .

Usando la misma lógica, podemos considerar un *punto plano* cuya representación directa es $p \wedge e_\infty$ como la intersección de un plano Π y una línea L y por lo tanto su representación dual viene dada por $\pi \wedge l$, donde $\pi (= \Pi^*)$ y $l (= L^*)$ son las representaciones duales del plano Π y la recta L , respectivamente.

5.3 Transformaciones conformes

La geometría conforme es el estudio de las transformaciones conformes y, como ya se mencionó, una de las ventajas del modelo conforme es que las transformaciones conformes en \mathbb{R}^3 se representan por transformaciones ortogonales en $\mathbb{G}^{4,1}$. En lo que sigue se describen las principales transformaciones conformes y se ilustran con el uso de CGAlgebra.

5.3.1 Involución

Definición 5.3.1 Un *versor* es una expresión de la forma:

$$V = v_k v_{k-1} \cdots v_1. \quad (5.39)$$

Donde cada v_i tiene la forma $a_0 e_0 + a_1 e_1 + a_2 e_2 + a_3 e_3 + a_\infty e_\infty$.

El número k se llama grado del versor.

Definición 5.3.2 Si $V^{-1} = v_1^{-1} v_2^{-1} \cdots v_k^{-1}$ es la inversa del versor V , entonces la *involución* de V se define como

$$V^\dagger = (-1)^k V^{-1} = (-1)^k v_1^{-1} v_2^{-1} \cdots v_k^{-1}. \quad (5.40)$$

Definición 5.3.3 Los versores operan sobre elementos del espacio conforme en la siguiente forma:

$$V(\cdots) V^\dagger \quad (5.41)$$

5.3.2 Translaciones

Las traslaciones se obtienen como reflexiones por dos planos paralelos. La traslación de un punto \mathbf{p} en \mathbb{R}^3 por el vector $\mathbf{t} = t_1 e_1 + t_2 e_2 + t_3 e_3$ es

$$T_{\mathbf{t}} \mathbf{p} T_{\mathbf{t}}^{-1},$$

donde

$$T_{\mathbf{t}} = 1 - \frac{1}{2} \mathbf{t} e_\infty.$$

Podemos verificar fácilmente que $T_{\mathbf{t}}^{-1} = T_{-\mathbf{t}}$:

```
In[1]:= t = t1 e[1] + t2 e[2] + t3 e[3];
In[2]:= Tt = 1-GeometricProduct[t,e[∞]]/2
Out[2]:= 1 - (t1 e[1,∞]- t2 e[2,∞]- t3 e[3,∞])/2
In[3]:= Tti = MultivectorInverse[Tt]
Out[3]:= 1+ (t1 e[1,∞]+ t2 e[2,∞]+ t3 e[3,∞])/2
In[4]:= Tti == 1-1/2 GeometricProduct[-t,e[∞]] // Simplify
Out[4]:= True
```

donde la función `MultivectorInverse[]` definida en el paquete regresa el inverso (si existe) de un multivector. Algunos ejemplos simples son los que se muestran a continuación:

- Traslación del origen e_0 por los vectores \mathbf{t} :

```
In[5]:= GeometricProduct[Tt,e[0],Tti];
Out[5]:= e[0]+t1 e[1]+t2 e[2]+t3 e[3]+1/2(t1^2+t2^2+t3^2) e[∞]
```

entonces $T_{\mathbf{t}} e_0 T_{\mathbf{t}}^{-1} = e_0 + \mathbf{t} + \frac{1}{2} \mathbf{t}^2 e_\infty = t$, donde t es la representación de \mathbf{t} en $\mathbb{G}^{4,1}$.

- Traslación del infinito e_∞ por los vectores \mathbf{t} :

```
In[6]:= GeometricProduct[Tt,e[∞],Tti];
Out[6]:= e[∞]
```

- Translación de un vector $\mathbf{x} = (x_1, x_2, x_3)$:

```
In[7]:= x= x1 e[1] + x2 e[2] + x3 e[3];
In[8]:= GeometricProduct[Tt,x,Tti];
Out[8]:= x1 e[1]+x2 e[2]+x3 e[3]+(t1 x1+t2 x2+t3 x3) e[∞]
```

Esto es, $T_t \mathbf{x} T_t^{-1} = \mathbf{x} + (\mathbf{x} \cdot \mathbf{t}) e_\infty$.

5.3.3 Rotaciones

La rotación de un vector $\mathbf{x} \in \mathbb{R}^3$, alrededor de un eje ortogonal a los vectores \mathbf{a} y \mathbf{b} , esta dada por

$$\mathbf{x}' = R \mathbf{x} R^{-1}, \quad (5.42)$$

donde $R = \mathbf{a} \mathbf{b}$.

Podemos fácilmente verificar que e_0 y e_∞ no son afectados por la rotación:

```
In[9]:= a = a1 e[1] + a2 e[2] + a3 e[3];
In[10]:= b = b1 e[1] + b2 e[2] + b3 e[3];
In[11]:= R = GeometricProduct[a,b];
In[12]:= GeometricProduct[R,e[0],MultivectorInverse[R]]
Out[12]:= e[0]
In[13]:= GeometricProduct[R,e[∞],MultivectorInverse[R]]
Out[13]:= e[∞]
```

O también podemos usar la función especificada en el paquete:

```
In[14]:= Rotation[e[0], a, b]
Out[14]:= e[0]
In[15]:= Rotation[e[∞], a, b]
Out[15]:= e[∞]
```

El plano de rotación está especificado por los vectores \mathbf{a} y \mathbf{b} , pero si se especifica un ángulo de rotación ϕ , entonces la expresión de la rotación es:

$$\mathbf{x}' = \left(\cos(\phi/2) - \hat{A} \sin(\phi/2) \right) \mathbf{x} \left(\cos(\phi/2) + \hat{A} \sin(\phi/2) \right),$$

donde

$$\hat{A} = \frac{\mathbf{a} \wedge \mathbf{b}}{|\mathbf{a} \wedge \mathbf{b}|}.$$

Si quisiéramos especificar una rotación de $\theta = \pi$ sería de la siguiente forma:

```
In[16]:= Rotation[e[0], a, b, Pi]
Out[16]:= e[0]
In[17]:= Rotation[e[∞], a, b, Pi]
Out[17]:= e[∞]
```

5.3.4 Inversión por una esfera

Sea S una esfera con centro en p y radio r :

$$S = p - \frac{r^2}{2} e_\infty.$$

La inversión de un punto x en $\mathbb{G}^{4,1}$, con respecto a la esfera S es

$$-SxS^{-1}.$$

Podemos definir en *Mathematica* una función que constituya el operador inversión por una esfera, de la siguiente manera:

```
In[18]:= Clear[x,p,r]
In[18]:= Inversor[x_,p_,r_]:=
-GeometricProduct[p-r^2 e[∞]/2,x,MultivectorInverse[p-r^2 e[∞]/2]]
```

Entonces, la función definida por el usuario `Inversor[x,p,r]` invierte un punto x con respecto a una esfera centrada en p y con radio r . Ahora vamos a probar `Inversor[]` considerando inversiones con respecto a una esfera centrada en el origen, que es $p = e[0]$.

La inversión del origen e_0 es e_∞ :

```
In[19]:= Inversor[e[0],e[0],r]
Out[19]:= r^2 e[∞]/2
```

El infinito e_∞ es invertido al origen:

```
In[20]:= Inversor[e[∞],e[0],r]
Out[20]:= 2 e[0]/r^2
```

Y los vectores en \mathbb{R}^3 no son afectados por la inversión:

```
In[21]:= v = v1 e[1] + v2 e[2] + v3 e[3];
In[22]:= Inversor[v,e[0],r]
Out[22]:= v1 e[1] + v2 e[2] + v3 e[3]
```

5.3.5 Resumen

Como una referencia, las funciones más útiles definidas en `CGAlgebra`, se enlistan en la Tabla 5.1.

Table 5.1: Funciones complementarias de CGAlgebra.

Expression	Output
Magnitude[A]	A^2
Reversion[A]	La reversión de un multivector A
Involution[A]	La involución de grado (\dagger) de un multivector A .
MultivectorInverse[A]	La inversa (si existe) de un multivector A .
GradeQ[A,k]	True si A es un k -vector.
Dual[A]	El dual de un multivector A
Rotation[x,a,b, θ]	Rotación de un vector x en un ángulo θ El plano definido por los vectores a y b . El sentido de rotación es de a a b . El valor de default de θ es el ángulo entre a y b .
ToVector[v]	El \mathbb{R}^3 vector $v = x e[1] + y e[2] + z e[3]$ transformado al formato $\{x, y, z\}$.
GFactor[A]	Factoriza los términos de A según $e[i, j, k]$.
I5	El pseudoscalar I de $\mathbb{G}^{4,1}$
I5i	I^{-1}

Pero como observamos en la sección 5.3.4, con las funciones básicas y complementarias, cualquier otra función requerida puede ser construida para hacer el paquete aun mas completo usando las capacidades del lenguaje de programación de *Wolfram Mathematica* para definir nuestras propias funciones.

5.4 El problema de la geometría de distancias.

La geometría de distancia (DG) es la rama de las matemáticas que se ocupa de caracterizar y estudiar conjuntos de puntos basándose únicamente en valores dados de las distancias entre pares de puntos [26]. De esta forma, el problema de geometría de distancia (DGP) consiste en determinar las posiciones espaciales de un conjunto de puntos, dadas las distancias entre ellos [27]. Los problemas de geometría de distancia también surgen siempre que se necesita inferir la forma de una configuración de puntos (posiciones relativas) a partir de las distancias entre ellos.

Históricamente, el primer resultado en geometría de distancias es la fórmula de Heron, del siglo I d.C., que da el área de un triángulo a partir de las distancias entre sus 3 vértices. La fórmula de Brahmagupta, del siglo VII d. C., la generaliza a cuadriláteros cíclicos. Tartaglia, del siglo XVI dC, lo generalizó para dar el volumen del tetraedro a partir de las distancias entre sus 4 vértices. Por su parte, la teoría moderna de la geometría de distancias comenzó con Arthur Cayley y Karl Menger [28].

El interés de DGP reside en la riqueza de sus aplicaciones para la ciencia y la ingeniería. Como ejemplo, una aplicación interesante es la de ubicar sensores en redes de telecomunicaciones [29], en tal caso, se conocen las posiciones de algunos sensores y se conocen algunas de las distancias entre sensores: el problema es identificar las posiciones en el espacio para todos los sensores.

Además, surge una aplicación interesante en biología[30] y química [31, 32]: las técnicas experimentales

como la Resonancia Magnética Nuclear (RMN) son capaces de estimar distancias entre pares de átomos de una molécula dada, y el problema pasa a ser el de identificar la conformación tridimensional de la molécula a partir de esas distancias, es decir, las posiciones de todos sus átomos. En este campo, el principal interés se centra en las proteínas, ya que descubrir su conformación tridimensional nos permite obtener pistas sobre la función que son capaces de realizar.

5.4.1 Planteamiento del problema DMDGP

El problema de la geometría de la distancia molecular (MDGP) es una variante del problema de la geometría de la distancia (DGP) que se enfoca en el estudio de estructuras moleculares. El problema de geometría de distancia molecular discretizable (DMDGP) es un subconjunto particular del MDGP que se caracteriza por el hecho de que se puede resolver mediante una búsqueda discreta que ocurre en un espacio continuo.

El problema MDGP puede tener soluciones vacías, finitas o infinitamente incontables. Mientras tanto, en el problema DMDGP, gracias al ordenamiento, las posibles soluciones forman un conjunto finito, ya que esta "discretización" ayuda a garantizar un conjunto finito de soluciones.

Este problema (DMDGP) comúnmente se centra en las proteínas, ya que descubrir su conformación tridimensional nos permite obtener pistas sobre la función que son capaces de realizar. Como ya mencionamos, las técnicas experimentales como la RMN son capaces de estimar distancias entre pares de átomos de una molécula dada, y el problema pasa a ser el de identificar la conformación tridimensional de la molécula a partir de esas distancias. Las implicaciones en campos relacionados, como la biomedicina y el diseño de fármacos, son evidentes.

Para plantear formalmente el problema de la geometría de distancias, consideremos primero los siguientes tres conjuntos:

1. Un conjunto de n átomos: $V = \{v_1, v_2, \dots, v_n\}$.
2. Un conjunto de pares de átomos: $E = \{\{v_1, v_2\}, \{v_3, v_5\}, \dots, \{v_i, v_j\}\}$.
3. El conjunto de las distancias de los pares de átomos en E : $d = \{d_{1,2}, d_{3,5}, \dots, d_{i,j}\}$, donde $d_{i,j} = d(v_i, v_j)$.

Y también consideremos una función $\mathbf{x} : V \rightarrow \mathbb{R}^3$ definida como

$$\mathbf{x}(v_i) = (x_i, y_i, z_i) = \mathbf{x}_i, \quad (5.43)$$

donde \mathbf{x}_i son las coordenadas del átomo v_i .

Dado que V puede considerarse un conjunto de vértices y E un conjunto de aristas, la pareja ordenada (V, E) constituye un grafo. Aún más, el conjunto de distancias d puede considerarse un peso o costo de cada arista, entonces la tripleta (V, E, d) constituye un grafo pesado o ponderado. Es interesante mencionar que el mismo origen de la palabra "grafo" se debe a la representación de moléculas [33].

Con todo esto, el problema de la geometría de distancias puede plantearse así:

Problema de la geometría de distancias: Dado un grafo (V, E, d) , conectado y pesado, determinar una función $\mathbf{x} : V \rightarrow \mathbb{R}^3$ tal que para cada $\{v_i, v_j\} \in E$,

$$\|\mathbf{x}(v_i) - \mathbf{x}(v_j)\| = d_{i,j}. \quad (5.44)$$

Inicialmente se considera que todas las distancias en d son exactas y en ese caso el problema es hallar las coordenadas $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, de los átomos en V , de tal manera que se satisfagan todas las condiciones:

$$\|\mathbf{x}_i - \mathbf{x}_j\| = d_{i,j}. \quad (5.45)$$

Para evitar soluciones que difieran por traslaciones o rotaciones, los primeros tres átomos de la molécula se fijan en el espacio, con lo que se conoce $\mathbf{x}_1, \mathbf{x}_2$ y \mathbf{x}_3 . Entonces del conjunto $V = \{v_1, v_2, \dots, v_n\}$, ya se conocen las coordenadas de v_1, v_2 y v_3 , que deben satisfacer:

$$d_{1,2} + d_{2,3} > d_{1,3}.$$

De acuerdo con (5.45), para encontrar la posición \mathbf{x}_4 , de v_4 , hay que resolver

$$\|\mathbf{x}_4 - \mathbf{x}_3\|^2 = d_{3,4}^2, \quad (5.46a)$$

$$\|\mathbf{x}_4 - \mathbf{x}_2\|^2 = d_{2,4}^2, \quad (5.46b)$$

$$\|\mathbf{x}_4 - \mathbf{x}_1\|^2 = d_{1,4}^2, \quad (5.46c)$$

lo que puede dar lugar hasta a dos posibles soluciones. De forma que, para cada posición determinada de v_4 , obtenemos otras dos para v_5 , y así sucesivamente, lo que implica que DMDGP tiene 2^{n-3} posibles soluciones [34, 35].

El DMDGP es un problema computacionalmente complejo y se han propuesto varios métodos para resolverlo; los más importantes son revisados en la Ref. [27, 36].

No es el propósito de este trabajo abordar este problema, sólo mostrar que el Álgebra Geométrica Conforme y el paquete CGAlgebra, pueden ser una herramienta útil para los métodos que se usan para resolver el problema DGP. Para aclarar esto, observemos que, usando la métrica Euclideana, el problema (5.46a)-(5.46c) puede reescribirse así:

$$(x_4 - x_3)^2 + (y_4 - y_3)^2 + (z_4 - z_3)^2 = d_{3,4}^2,$$

$$(x_4 - x_2)^2 + (y_4 - y_2)^2 + (z_4 - z_2)^2 = d_{2,4}^2,$$

$$(x_4 - x_1)^2 + (y_4 - y_1)^2 + (z_4 - z_1)^2 = d_{1,4}^2,$$

de manera que hallar la solución (x_4, y_4, z_4) puede verse como encontrar la intersecciones de tres esferas. Dado que la intersección de dos esferas es un círculo, la intersección de tres esferas son dos puntos, que son las dos soluciones que se mencionaron arriba.

Y el proceso puede repetirse de manera interactiva. Para algún $i > 4$, se tiene:

$$\|\mathbf{x}_i - \mathbf{x}_{i-1}\|^2 = d_{i-1,i}^2,$$

$$\|\mathbf{x}_i - \mathbf{x}_{i-2}\|^2 = d_{i-2,i}^2,$$

$$\|\mathbf{x}_i - \mathbf{x}_{i-3}\|^2 = d_{i-3,i}^2,$$

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{j,i}^2.$$

Y se debe satisfacer

$$d_{i-3,i-2} + d_{i-2,i-1} > d_{i-3,i-1}.$$

Y el problema sigue siendo el mismo, es decir, el de la intersección de tres esferas [21][34]. Este proceso iterativo de intersección de tres esferas es la base del llamado algoritmo *Branch and Prune* (BP) para resolver el DMDGP [37].

Como mencionamos anteriormente, el DMDGP no está dentro de los objetivos de esta tesis, ni el estudio de los algoritmos que se usan para abordarlo. En esta sección de aplicaciones sólo mostraremos cómo el AGC, y en particular el paquete CGAlgebra, puede ser una herramienta de mucha utilidad para el algoritmo BP, ya que permite resolver el problema de la intersección de esferas de manera elegante y práctica.

5.4.2 Intersección de esferas

Usando la representación dual, consideremos dos esferas, de radio r_1 y r_2 , centradas en p_1 y p_2 , respectivamente:

$$S_1 = p_1 - \frac{r_1^2}{2} e_\infty. \quad (5.47)$$

$$S_2 = p_2 - \frac{r_2^2}{2} e_\infty. \quad (5.48)$$

La intersección de esas esferas es:

$$\sigma = S_1 \wedge S_2,$$

que se calcula así:

```
In[23]:= Y = x1 e[1] + y1 e[2] + z1 e[3];
In[24]:= Z = x2 e[1] + y2 e[2] + z2 e[3];
In[25]:= p1 = e[0] + Y + (Magnitude[Y]^2/2) e[∞];
In[26]:= p2 = e[0] + Z + (Magnitude[Z]^2/2) e[∞];
In[27]:= S1 = p1 - r1^2/2 e[∞];
In[28]:= S2 = p2 - r2^2/2 e[∞];
In[29]:= σ =OuterProduct[S1,S2]//GFactor
Out[29]:= .5(r1^2-r2^2-x1^2+x2^2-y1^2+y2^2-z1^2+z2^2)
+(x2-x1) e[0,1] +(-y1 + y2) e[0,2] +(z2-z1) e[0,3]
+.5(r1^2-r2^2-x1^2+x2^2-y1^2+y2^2-z1^2+z2^2) e[0,∞]
+(x1 y2 - x2 y1) e[1,2] +(x1 z2 - x2 z1) e[1,2] +(y1 z2 - y2 z1)e[2,3]
+.5(r1^2 x2-r2^2 x1-x1^2 x2+x1 x2^2+x1 y2^2+x1 z2^2-x2 y1^2-x2 z1^2)e[1,∞]
+.5(r1^2 y2-r2^2 y1-x1^2 y2+x2^2 y1-y1^2 y2+y1 y2^2+y1 z2^2-y2 z1^2)e[2,∞]
+.5(r1^2 z2-r2^2 z1-x1^2 z2+x2^2 z1-y1^2 z2+y2^2 z1-z1^2 z2+z1 z2^2)e[3,∞]
```

Como ejemplo, consideremos las esferas con centros $Y = (-4, 0, 0)$ y $Z = (-3, -1.1, 1.3)$, y radios $r_1 = 2.5$ y $r_2 = 1.3$, respectivamente.

```
In[30]:= σ2 = σ /. {x1->-4, y1->0, z1->0, x2->-3, y2->-1.1, z2->1.3,
r1->2.5, r2->1.6}
Out[30]:= -0.205 +e[0,1] -1.1 e[0,2] +1.3 e[0,3] -0.205 e[0,∞]
+4.4 e[1,2]-5.2 e[1,3] -4.055 e[1,∞] +5.3625 e[2,∞] -6.3375 e[3,∞]
```

La intersección de dos esferas es un círculo, y de acuerdo con [25], el centro y radio de este círculo esta dado por:

$$C_\sigma = -\frac{1}{2} \frac{\sigma e_\infty \sigma}{(e_\infty \cdot \sigma)^2}, \quad r_\sigma = \frac{(-1)^{(m+1)} \sigma^2}{(e_\infty \cdot \sigma)^2}, \quad (5.49)$$

entonces:

```
In[31]:= aux=GeometricProduct[InnerProduct[e[∞],σ2],InnerProduct[e[∞],σ2]]
In[32]:= c=-1/2 (GeometricProduct[σ2,e[∞],σ2])/aux// Chop // FullSimplify
Out[32]= e[0] - 3.02692 e[1] - 1.07038 e[2] + 1.265 e[3] + 5.95411 e[∞]
In[33]:= r=Sqrt[(-GeometricProduct[σ,σ]//Expand)/aux]//Chop
Out[33]= 1.59912
```

De donde se tiene que el círculo obtenido de la intersección de S_1 y S_2 tiene centro en $(-3.02692, -1.07038, 1.265)$, y radio 1.59912. Esto lo podemos visualizar así:

```
In[34]:= Graphics3D[{{Opacity[.5], Sphere[Centere[S1],Radio[S1]]},{Opacity[.5],
Sphere[Centere[S2],Radio[S2]]},{circle3D[ToVector[c],r,ToVector[S1-S2]]}}
/.{x1->-4,y1->0,z1->0,x2->-3,y2->-1.1,z2->1.3,r1->3,r2->1.3},Boxed->False]
```

El resultado se muestra en la Fig. 5.11.

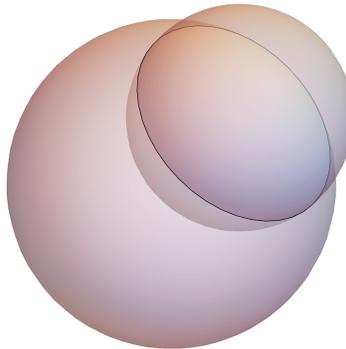


Figure 5.11: Intersección de dos esferas.

Consideramos ahora una tercera esfera, de radio r_3 centrada en p_3 :

$$S_3 = p_3 - \frac{r_3^2}{2} e_\infty \quad (5.50)$$

La intersección de las 3 esferas es $S_1 \wedge S_2 \wedge S_3$, que se calcula así:

```

In[35]:= V= x3 e[1] + y3 e[2] + z3 e[3];
In[36]:= p3= e[0] + V + (Magnitude[V]^2/2) e[∞];
In[37]:= S3= p3 - r3^2/2 e[∞];
In[38]:= pp= OuterProduct[S1,S2,S3]//GFactor
Out[38]:= (...)e[0,1,2] + (...)e[0,1,3] + (...)e[0,2,3] + (...)e[1,2,3]
+ (...) (e[1] -e[0,1,∞]) + (...) (e[2]-e[0,2,∞]) + (...) (e[3]-e[0,3,∞])
+ (...) e[1,2,∞]+ (...) e[1,3,∞] + (...) e[2,3,∞]

```

El resultado es la variable pp, en donde (...) representa un coeficiente grande que se omite. Como ejemplo, ahora considere las esferas con centros $Y = (-4, 0, 0)$, $Z = (-3, -1.1, 1.3)$ y $V = (0, 0, 0)$, y radios $r1 = 2.5$, $r2 = 1.6$ y $r3 = 2.3$.

```

In[39]:= pp2= pp /. {x1->-4, y1->0, z1->0, x2->-3, y2->-1.1, z2->1.3,
x3->0, y3->0, z3->0, r1->2.5, r2->1.6, r3->2.3}
Out[69]:= 4.4 e[0,1,2] -5.2 e[0,1,3] -11.638 e[1,2,∞]+13.754 e[1,3,∞]
+6.7 (e[1] -e[0,1,∞]) -8.272 (e[2]-e[0,2,∞]) +9.776 (e[3]-e[0,3,∞])

```

La intersección de tres esferas, pp2, es un par de puntos que, de acuerdo con (5.36), se calculan así:

```

In[40]:= t1 = GeometricProduct[(Dual[pp2]+Sqrt[InnerProduct[Dual[pp2],
Dual[pp2]]]),MultivectorInverse[-InnerProduct[e[∞],Dual[pp2]]]//Chop
Out[40]:= e[0] -1.88e[1] -1.31306 e[2] + 0.177411 e[3] + 2.645 e[∞]
In[41]:= t2 = GeometricProduct[(Dual[pp2]-Sqrt[InnerProduct[Dual[pp2],
Dual[pp2]]]),MultivectorInverse[-InnerProduct[e[∞],Dual[pp2]]]//Chop
Out[41]:= e[0] -1.88e[1] +0.0423699 e[2] + 1.32431 e[3] + 2.645 e[∞]

```

La intersección de las tres esferas, entonces, son los puntos $(-1.88, -1.31306, 0.177411)$ y $(-1.88, 0.0423699, 1.32431)$, los cuales son la solución a nuestro problema. Todo esto puede mostrarse gráficamente:

```

In[42]:= σ3=OuterProduct[S1,S2]/. {x1->-4,y1->0,z1->0,x2->-3,y2->-1.1,
z2->1.3,x3->0,y3->0,z3->0,r1->2.5,r2->1.6,r3->2.3};
In[43]:= aux=GeometricProduct[InnerProduct[e[∞],σ3],InnerProduct[e[∞],σ3]];
In[44]:= c=-1/2 (GeometricProduct[σ3,e[∞],σ3])/aux;
In[45]:= r=Sqrt[(-GeometricProduct[σ3,σ3]//Expand)/aux]//Chop;
In[46]:= Graphics3D[{{Opacity[.5], Sphere[Centere[S1],Radio[S1]]}, {Opacity[.5],
Sphere[Centere[S2],Radio[S2]]}, {Opacity[.5], Sphere[Centere[S3],Radio[S3]]},
{circle3D[ToVector[c],r,ToVector[S1-S2]]}, {PointSize[Large], Point[{ToVector
[t1],ToVector[t2]]}}]/. {x1->-4,y1->0,z1->0,x2->-3,y2->-1.1,z2->1.3,x3->0,
y3->0,z3->0,r1->2.5,r2->1.6,r3->2.3}, Boxed -> False]

```

El resultado se muestra en la Fig. 5.12.

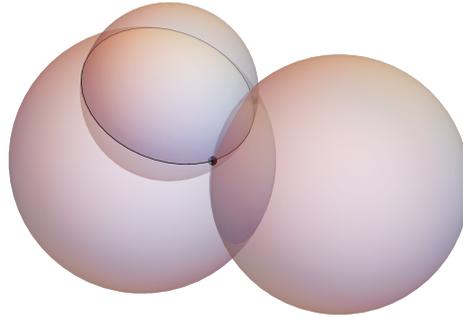


Figure 5.12: Intersección de tres esferas.

Este par de puntos también se pueden obtener al encontrar la intersección del círculo que resultó de la intersección de las esferas S_1 y S_2 (σ_2 en Out[30]) y la esfera S_3 :

```
In[47]:= Graphics3D[{{Opacity[.5], Sphere[Centere[S3], Radio[S3]]},
  {circle3D[ToVector[c], r, ToVector[S1-S2]]}, {PointSize[Large], Point[{
  ToVector[t1], ToVector[t2]}}]}/. {x1->-4, y1->0, z1->0, x2->-3, y2->-1.1,
  z2->1.3, x3->0, y3->0, z3->0, r1->2.5, r2->1.6, r3->2.3}, Boxed -> False]
```

El resultado se muestra en la Fig. 5.13.

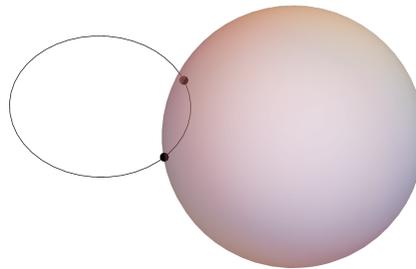


Figure 5.13: Intersección de un círculo y una esfera.

Se puede verificar fácilmente que la intersección $t_{1,2}$ de 3 esferas tiene alguna de las siguientes posibilidades:

- la intersección es un conjunto vacío, si $t_{1,2} < 0$;
- la intersección es un punto, si $t_{1,2} = 0$;
- la intersección son dos puntos, si $t_{1,2} > 0$.

Así que podemos verificar el signo de $t_{1,2}$ para saber si tendremos intersección:

```
In[48]:= Sign[t1]
Out[48]:= 1
```

```
In[49] := Sign[t2]
Out[49] := 1
```

Lo cual indica que el signo es positivo, y la intersección serán dos puntos.

5.4.2.1 Problema de Navegación Hiperbólica

La navegación hiperbólica es una clase de sistemas de radionavegación que se basa en la medición de la diferencia de tiempo de llegada de las ondas de radio emitidas por transmisores de navegación. Estos sistemas permiten determinar la ubicación utilizando instrumentos receptores de navegación que calculan la distancia a partir de la diferencia de fase de las ondas recibidas.

Los sistemas de localización hiperbólica se utilizaron por primera vez durante la Primera Guerra Mundial en sistemas de localización acústica para localizar artillería enemiga. Consistían en micrófonos distribuidos que registraban el sonido de los proyectiles y enviaban los tiempos de llegada a un centro de cálculo para calcular la ubicación. Estos sistemas también se utilizaron en la Segunda Guerra Mundial. El primer sistema hiperbólico de radionavegación fue el *Gee*, de la era de la Segunda Guerra Mundial, presentado por la Royal Air Force para uso del RAF Bomber Command. Esto fue seguido por el sistema Decca Navigator en 1944 por la Royal Navy, junto con LORAN por la Marina de los EE. UU. para la navegación de largo alcance en el mar. Ejemplos de posguerra incluyen el conocido Loran-C de la Guardia Costera de los EE.UU., el sistema internacional Omega y el Alpha y CHAYKA soviéticos. Todos estos sistemas tuvieron uso hasta que fueron reemplazados por sistemas de navegación por satélite como el Sistema de Posicionamiento Global (GPS) en la década de 1990 [38].

Como ejemplo vamos a considerar el problema de navegación hiperbólica con 3 receptores, el cual también se puede ver como un problema de intersección de esferas. Para resolver este problema vamos a considerar el grafo $G = (V, E, d)$, dado por:

$$V = v_1, v_2, v_3, v_4,$$

$$E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_1, v_4\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_4\}\}.$$

Con transmisores de navegación en Saltillo, Querétaro y Merida:

$$X_1 = \{-100.9892857142857, 25.37248302641369, 11\} \quad (5.51)$$

$$X_2 = \{-100.36071428571427, 20.540340169270838, 10\} \quad (5.52)$$

$$X_3 = \{-89.63571428571427, 20.97248302641369, 6\} \quad (5.53)$$

Con las distancias dadas por:

$$d_{1,4} = 20, \quad d_{2,4} = 18.8, \quad d_{3,4} = 13.3. \quad (5.54)$$

De forma que tendremos la configuración que se muestra en la Fig. 5.4.2.1.

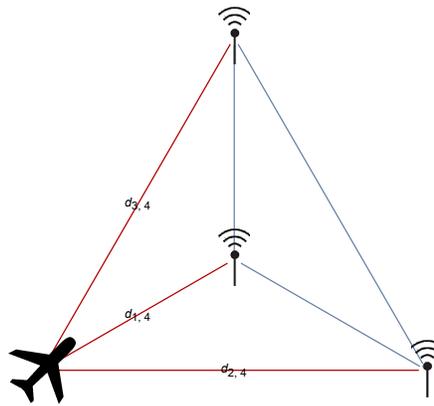


Figure 5.14: Problema de navegación hiperbólica.

El problema es encontrar la posición de v_4 . Para resolver este problema primero definimos las esferas:

$$S_1 = p_1 - \frac{d_{1,4}^2}{2} e_\infty$$

$$S_2 = p_2 - \frac{d_{2,4}^2}{2} e_\infty$$

$$S_3 = p_3 - \frac{d_{3,4}^2}{2} e_\infty$$

Donde p_i está dado de la forma (5.2). Esto lo podemos definir en CGAlgebra de la siguiente forma:

```
In[50]:= d14 = 20; d24 = 18.8; d34 = 13.3;
In[51]:= {x1, y1, z1} = {-100.9892857142857, 25.37248302641369, 11};
In[52]:= {x2, y2, z2} = {-100.36071428571427, 20.540340169270838, 10};
In[53]:= {x3, y3, z3} = {-89.63571428571427, 20.97248302641369, 6};

In[54]:= X1=x1 e[1] +y1 e[2] + z1 e[3];
In[55]:= p1= e[0] + X1 + (GeometricProduct[X1,X1]/2) e[∞]

In[56]:= X2=x2 e[1] +y2 e[2] + z2 e[3];
In[57]:= p2= e[0] + X2 + (GeometricProduct[X2,X2]/2) e[∞]

In[58]:= X3=x3 e[1]+ y3 e[2] + z3 e[3];
In[59]:= p3= e[0] + X3 + (GeometricProduct[X3,X3]/2) e[∞]

In[60]:= S1 = p1 - d14^2/2 e[∞];
In[61]:= S2 = p2 - d24^2/2 e[∞];
In[62]:= S3 = p3 - d34^2/2 e[∞];
```

De forma que la intersección de las esferas $S_1 \wedge S_2 \wedge S_3$ está dada por:

```
In[63]:=pp= OuterProduct[S1, S2, S3];
```

Como esto es un par de puntos en su forma dual, podemos utilizar la formula (5.36) para obtener ambos puntos:

```
In[64]:= t1 = GeometricProduct[-DDual[pp] + Sqrt[InnerProduct[
  DDual[pp], DDual[pp]]], InnerProduct[e[∞], DDual[pp]]];
In[65]:= t2 = GeometricProduct[-DDual[pp] - Sqrt[InnerProduct[DDual[pp],
  DDual[pp]]], InnerProduct[e[∞], DDual[pp]]];
```

Con lo que podemos obtener la posición como se puede observar en la Fig. 5.15, con:

```
In[66]:=Show[GeoListPlot[Entity["City", /"Merida", "Yucatan", "Mexico"/],
  Entity["City", /"Queretaro", "Queretaro", "Mexico"/], Entity["City",
  /"Saltillo", "Coahuila", "Mexico"/], GeoLabels->(Tooltip[#1,#2]&), PlotRange->
  {{-85, -107}, {15, 30}}, PlotMarkers->Red], GeoListPlot[GeoPosition[({
  ToVector[t1][[;;2]])//Reverse)/. {r1->20, r2->18.8, r3->13.3}, "ITRF00"],
  GeoPosition[({ ToVector[t2][[;;2]])//Reverse)/. {r1->20, r2->18.8, r3->13.3},
  "ITRF00"/], GeoLabels->(Tooltip[#1,#2]&), PlotRange->{{-85, -107}, {15, 30}},
  PlotMarkers->Black]]
```



Figure 5.15: Solución al problema de Navegación Hiperbolica.

5.4.3 Cáscaras esféricas

Como ya mencionamos, los datos de RMN ofrecen las medidas de distancia con errores e incertidumbres. Esto trae considerables dificultades extras al DMDGP [39] ya que ahora las esferas que se intersectan no tienen un radio preciso, sino que el radio se define por un intervalo, de manera que ahora la intersección de dos esferas (ahora cascarones) no produce un círculo sino que produce discos o capas esféricas[39].

El álgebra geometría conforme, también permite una forma eficiente de realizar la intersección entre esferas y capas esféricas que resultan de la incertidumbre en los radios. Enseguida mostramos cómo puede implementarse esto con CGAlgebra.

Primero, consideremos un DMDGP definido por el grafo $G=(V,E,d)$, dado por: [40]:

$$V = v_1, v_2, v_3, v_4, v_5,$$

$$E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_1, v_4\}, \{v_1, v_5\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_2, v_5\}, \{v_3, v_4\}, \{v_3, v_5\}, \{v_4, v_5\}\}$$

Con:

$$\begin{aligned} d_{1,2} = d_{1,3} = \sqrt{2}d_{1,4} = 1 \quad d_{1,5} &\in [0.8, 0.9] \\ d_{2,3} = \sqrt{2}d_{2,4} = 1 \quad d_{2,5} &\in [0.7, 1] \\ d_{2,4} = 1d_{3,5} = 1 \quad d_{4,5} &= 1 \end{aligned} \quad (5.55)$$

Y:

$$\begin{aligned} X_1 = \{0, 0, 1\} X_2 = \{0, 1, 0\} \\ X_3 = \{1, 0, 0\} X_4 = \{0, 0, 0\} \end{aligned} \quad (5.56)$$

De forma que tendremos la configuración que se muestra en la Fig. 5.4.3.

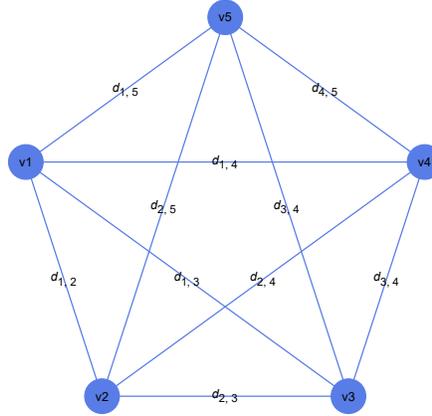


Figure 5.16: Problema DMDGP.

El problema es encontrar la posición de v_5 . Para resolver este problema lo haremos en dos partes, primero definimos las esferas:

$$\begin{aligned} S_{4,5} &= p_4 - \frac{d_{4,5}^2}{2} e_\infty \quad d_{4,5} = 1 \\ S_{3,5} &= p_3 - \frac{d_{3,5}^2}{2} e_\infty \quad d_{3,5} = 1 \\ S_{1,5} &= p_1 - \frac{d_{2,5}^2}{2} e_\infty \quad d_{2,5} \in [0.7, 1] \end{aligned}$$

Donde p_i esta dado de la forma (5.2). El proceso ahora puede ser visualizado como se muestra en la Fig. 5.17.

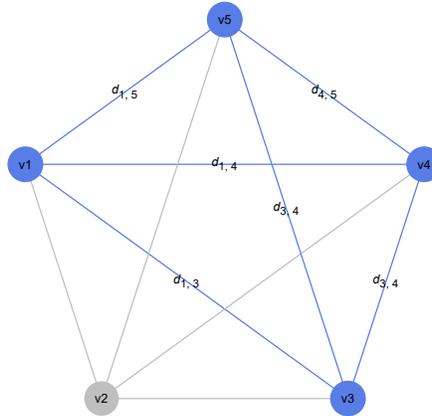


Figure 5.17: Problema de DMDGP, parte 1.

Esto lo podemos definir en CGAlgebra de la siguiente forma:

```
In[67]:= d12 = Sqrt[2]; d13 = Sqrt[2]; d14 = 1; d15 = {0.8, 0.9};
In[67]:= d23 = Sqrt[2]; d24 = 1; d25 = {0.7, 1}; d34 = 1; d35 = 1; d45 = 1;
In[68]:= {x1, y1, z1} = {0, 0, 1}; {x2, y2, z2} = {0, 1, 0};
In[69]:= {x3, y3, z3} = {1, 0, 0}; {x4, y4, z4} = {0, 0, 0};

In[70]:= X2=x2 e[1] +y2 e[2] + z2 e[3];
In[71]:= p2= e[0] + X2 + (GeometricProduct[X2,X2]/2) e[∞]

In[72]:= X3=x3 e[1]+ y3 e[2] + z3 e[3];
In[73]:= p3= e[0] + X3 + (GeometricProduct[X3,X3]/2) e[∞]

In[74]:= X4=x4 e[1] +y4 e[2] + z4 e[3];
In[75]:= p4= e[0] + X4 + (GeometricProduct[X4,X4]/2) e[∞]

In[76]:= S45 = p4 - d45^2/2 e[∞];
In[77]:= S35 = p3 - d35^2/2 e[∞];
In[78]:= S25d = p2 - d25[[1]]^2/2 e[∞];
In[79]:= S25u = p2 - d25[[2]]^2/2 e[ ∞];
```

De forma que la intersección de la las esferas $S_{2,5} \wedge S_{3,5} \wedge S_{4,5}$ está dada por:

```
In[80]:= Z= OuterProduct[S25d, S35, S45]
Out[80]= 0.755e[1]-0.5e[2]-1e[0,1,2]-0.755e[0,1,∞]+0.5e[0,2,∞]+0.5e[1,2,∞]
```

Como esto es un par de puntos en su forma dual, podemos utilizar la formula (5.36) para obtener ambos puntos:

```
In[81]:= t1 = GeometricProduct[-DDual[Z] + Sqrt[InnerProduct[
  DDual[Z], DDual[Z]]], InnerProduct[e[∞], DDual[Z]]];
In[82]:= t2 = GeometricProduct[-DDual[Z] - Sqrt[InnerProduct[DDual[Z],
  DDual[Z]]], InnerProduct[e[∞], DDual[Z]]];
```

Y la intersección de las esferas $S_{2,5} \wedge S_{3,5} \wedge S_{4,5}$ está dada por:

```
In[83]:= Z1 = OuterProduct[S25u, S35, S45]
Out[83]= 1/2 (e[1] -e[2] -2 e[0,1,2] -e[0,1,∞] + e[0,2,∞] +e[1,2,∞])
```

Donde también podemos utilizar la ecuación (5.36) para obtener ambos puntos:

```
In[84]:= t11 = GeometricProduct[-DDual[Z1] + Sqrt[InnerProduct[DDual[Z1],
  DDual[Z1]]], InnerProduct[e[∞], DDual[Z1]]] // N;
In[84]:= t21 = GeometricProduct[-DDual[Z1] - Sqrt[InnerProduct[DDual[Z1],
  DDual[Z1]]], InnerProduct[e[∞], DDual[Z1]]] // N;
```

Con estos dos puntos, podemos obtener el arco de intersección entre las dos esferas y el cascarón hueco, obteniendo el círculo de intersección $S_{4,5} \wedge S_{3,5}$:

```
In[85]:= σ = OuterProduct[S45, S35]
Out[85]= 1/2 + e[0, 1] + 1/2 e[0, ∞] + 1/2 e[1,∞]
```

Con centro y radio dados por (5.49):

```
In[86]:= aux=GeometricProduct[InnerProduct[e[∞],σ],InnerProduct[e[∞],σ]];
In[87]:= c=-1/2((GeometricProduct[σ,e[∞],σ])) 1/aux//FullSimplify;
In[88]:= r = Sqrt[(-GeometricProduct[σ, σ] // Expand)/aux] // Chop;
```

Con lo que podemos obtener los arcos, los cuales se puede observar en la Fig. 5.18, con:

```
In[89]:=Graphics3D[{{Opacity[.5],Sphere[Centere[S45],Radio[S45]]},
  {Opacity[.5],Sphere[Centere[S35],Radio[S35]]},{Opacity[.5],Sphere[
  Centere[S25u],Radio[S25u]]},{Opacity[.5],Sphere[Centere[S25d],Radio[
  S25d]]},Arco[ToVector[c],r,ToVector[S45-S35],{0,.5Pi},
  {ToVector[t1][[3]],ToVector[t12][[3]]}],Arco[ToVector[c],r,ToVector[
  S45-S35],{.5Pi,Pi},{ToVector[t22][[3]],ToVector[t2][[3]]}],{Point[
  {ToVector[t1], ToVector[t2]}]},Point[{ToVector[t12], ToVector[t22]}]}}
```

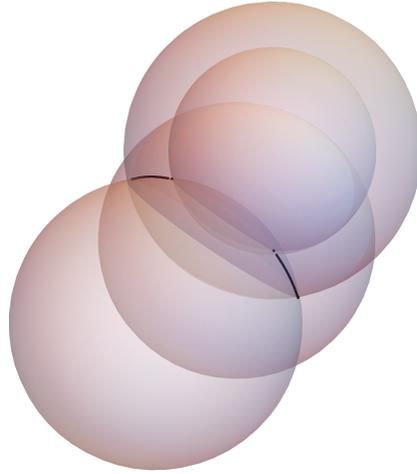


Figure 5.18: Intersección de dos esferas y un cascaron esférico.

Ahora vamos con la segunda parte, para esto definimos las esferas:

$$\begin{aligned}
 S_{4,5} &= p_4 - \frac{d_{4,5}^2}{2} e_\infty & d_{4,5} &= 1 \\
 S_{3,5} &= p_3 - \frac{d_{3,5}^2}{2} e_\infty & d_{3,5} &= 1 \\
 S_{1,5} &= p_1 - \frac{d_{1,5}^2}{2} e_\infty & d_{1,5} &\in [0.8, .9]
 \end{aligned}$$

Donde p_i esta dado de la forma (5.2). Esto puede ser visualizado como se muestra en la Fig. 5.19.

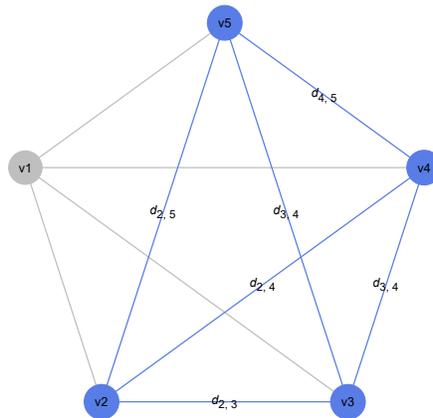


Figure 5.19: Problema DMDGP parte 2.

Esto lo podemos definir en CGAlgebra de la siguiente forma:

```

In[90]:= X1=x1 e[1] +y1 e[2] + z1 e[3];
In[91]:= p1= e[0] + X1 + (GeometricProduct[X1,X1]/2) e[∞]

In[92]:= S45 = p4 - d45^2/2 e[∞];
In[93]:= S35 = p3 - d35^2/2 e[∞];
In[94]:= S15d = p1 - d15[[1]]^2/2 e[∞];
In[95]:= S15u = p1 - d15[[2]]^2/2 e[∞];

```

De manera que la intersección de las esferas $S_{1,5} \wedge S_{3,5} \wedge S_{4,5}$ es:

```

In[96]:= Z2 = OuterProduct[S15d, S35, S45]
Out[96]= .68 e[1]-0.5e[2]-1e[0,1,2]-0.68 e[0,1,∞]+0.5e[0,2,∞]+0.5e[1,2,∞]

```

Usamos (5.36) para obtener ambos puntos (t3 y t4):

```

In[97]:= t3 = GeometricProduct[-DDual[Z2] + Sqrt[InnerProduct[
  DDual[Z2], DDual[Z2]]], InnerProduct[e[∞], DDual[Z2]]];
In[98]:= t4 = GeometricProduct[-DDual[Z2] - Sqrt[InnerProduct[DDual[Z2],
  DDual[Z2]]], InnerProduct[e[∞], DDual[Z2]]];

```

Ahora, la intersección de las esferas $S_{1,5} \wedge S_{3,5} \wedge S_{4,5}$ está dada por:

```

In[99]:= Z3 = OuterProduct[S15u, S35, S45]
Out[99]:=0.595e[1]-0.5e[3]-e[0,1,3]-0.595 e[0,1,∞]+.5e[0,3,∞]+0.5e[1,3,∞]

```

Donde también podemos utilizar (5.36) para obtener ambos puntos (t31 y t41):

```

In[100]:= t31 = GeometricProduct[-DDual[Z3] + Sqrt[InnerProduct[DDual[Z3],
  DDual[Z3]]], InnerProduct[e[∞], DDual[Z3]]] // N;
In[101]:= t41 = GeometricProduct[-DDual[Z3] - Sqrt[InnerProduct[DDual[Z3],
  DDual[Z3]]], InnerProduct[e[∞], DDual[Z3]]] // N;

```

A partir de estos dos puntos, podemos obtener el arco de intersección entre las dos esferas y el cascaron hueco, obteniendo el círculo de intersección $S_{4,5} \wedge S_{3,5}$

```

In[102]:= σ = OuterProduct[S45, S35]
Out[102]= 1/2 + e[0, 1] + 1/2 e[0, ∞] + 1/2 e[1,∞]

```

Con centro y radio dados por (5.49):

```

In[103]:=aux=GeometricProduct[InnerProduct[e[∞],σ],InnerProduct[e[∞],σ]];
In[104]:=c=-1/2((GeometricProduct[σ, e[∞],σ])) 1/aux//FullSimplify//Chop;
In[105]:= r = Sqrt[(-GeometricProduct[σ, σ] // Expand)/aux] // Chop;

```

Con lo que podemos obtener los arcos, que se muestran en en la Fig. 5.20, con:

```

In[106]:=Graphics3D[{{Opacity[.5], Sphere[Centere[S45],Radio[S45]]},{Opacity
[.5], Sphere[Centere[S35],Radio[S35]]},{Opacity[.5], Sphere[Centere[S15u],Radio
[S15u]]},{Opacity[.5], Sphere[Centere[S15d],Radio[S15d]]},{Thick,Arco[ToVector
[c],r,ToVector[S45-S35],{0,Pi},{ToVector[t31][[3]],ToVector[t3][[3]]}},
{Red,Arco[ToVector[c],r,ToVector[S45-S35],{-Pi,2Pi},{ToVector[t4][[3]],
ToVector[t41][[3]]}}, {Thick,Line[{ToVector[t3],ToVector[t31]}]}, {Black,

```

```
Point[{ToVector[t3],ToVector[t4]}],{Point[{ToVector[t31],ToVector[t41]}]}
```

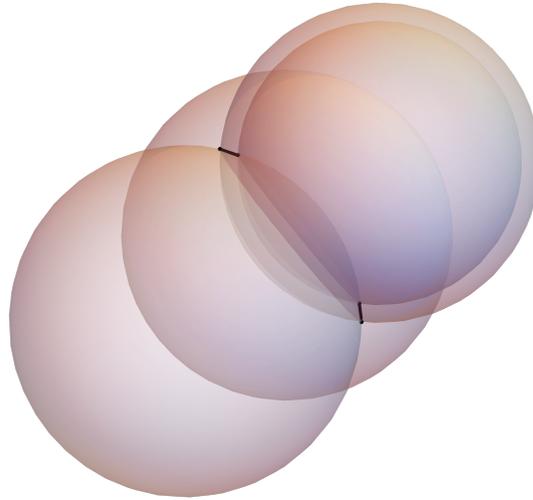


Figure 5.20: Intersección de dos esferas y un cascaron esférico.

Ya que la solución debe de cumplir con las distancias (5.55), la solución se encuentra en la intersección de los arcos $S_{1,5} \wedge S_{3,5} \wedge S_{4,5}$ (en color rojo) y $S_{2,5} \wedge S_{3,5} \wedge S_{4,5}$ (en color negro), que se muestran en la Fig. 5.21.

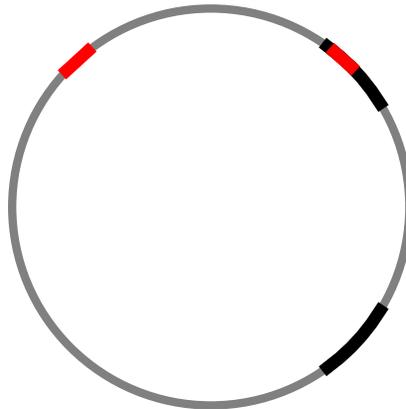


Figure 5.21: Intersección de dos esferas y un cascaron esférico.

Esta intersección entre $S_{1,5} \wedge S_{3,5} \wedge S_{4,5}$ y $S_{2,5} \wedge S_{3,5} \wedge S_{4,5}$ sera el arco $S_{1,5} \wedge S_{3,5} \wedge S_{4,5}$ que va del punto t_3 en $\text{In}[97]$ al punto t_{31} en $\text{In}[100]$, la cual sera la solución a este problema.

6. Conclusiones

En la última década, el Álgebra Geométrica (AG) ha recibido más atención por parte de la comunidad académica, perfilándose como alternativa del álgebra vectorial, tensorial, compleja y cuaterniónica y al cálculo vectorial, análisis complejo, geometría diferencial y otras teorías avanzadas. La razón de ello es que el Álgebra Geométrica se trata de una teoría matemática que unifica varias teorías matemáticas de forma consistente y es por lo tanto un lenguaje unificado para la física y la ingeniería; esto la hace inmensamente poderosa. El Álgebra Geométrica es, además, un lenguaje que hace accesibles las investigaciones en ciencias para los no especialistas en matemáticas, ya que codifica los fundamentos matemáticos subyacentes de diversas teorías matemáticas y facilita relacionarlos y transportarlos hacia nuevas áreas de estudio. Asimismo, basada en el Álgebra Geométrica se propuso el Álgebra Geométrica Conforme (CGA) como una herramienta para la manipulación de objetos geométricos, la cual, es el tema principal de este trabajo.

Algunas de las aplicaciones de AG y del AGC mas recientes son:

- **Ingeniería eléctrica y fibras ópticas:** El GA se ha convertido en una herramienta esencial para el análisis de corriente alterna [41]. La idea fundamental es reemplazar el fasor (un número complejo) por un vector que luego se trata en términos de GA o de CGA.
- **Robótica y control:** Donde es natural modelar movimientos euclidianos en forma compacta y libre de coordenadas en GA o en CGA [42].
- **Estimación de poses:** La estimación de pose es una tarea de visión por computadora que infiere la pose de una persona u objeto en una imagen o vídeo. Esto incluye intersecciones de líneas y planos, rotaciones y traslaciones; por lo que GA y CGA entran en acción [43].
- **Gráficos y modelado por computadora** En gráficos por computadora, el CGA contribuye al modelado y la manipulación de objetos gráficos [44].
- **Sistemas de Información Geográfica:** Para la ciencia de la información geográfica (SIG), el CGA es una herramienta para la representación multidimensional y geométrica unificada [45, 46, 47].
- **Aplicaciones en Física:** En el libro [34] se introducen varias aplicaciones del GA a la física del espacio-tiempo (electromagnetismo de Maxwell y ecuación de Dirac), robótica (cinemática directa e inversa y problema de singularidad para robots en serie), y geometría molecular.
- **Imágenes médicas:** El CGA ha surgido como un nuevo enfoque de la computación geométrica que

ofrece una representación simple y eficiente de objetos geométricos y transformaciones en tareas de imágenes médicas, como segmentación, modelado 3D y registro de datos médicos [48].

Pero es importante mencionar que el trasfondo de la propagación del AG es el desarrollo de herramientas de software para ejecutar automáticamente operaciones algebraicas que son sistemáticas pero tediosa y largas para los humanos.

6.1 Resultados

En esta tesis se presentan las ideas básicas del AG. Además, se presenta el paquete *CGAlgebra* para realizar cálculos con AGC en el lenguaje de programación *Wolfram Language*. Este paquete implementa las operaciones básicas del AGC y permite explorar problemas de aplicación en el contexto del cálculo de estructuras molecular [19, 25, 40] y navegación hiperbólica. Además, actúa como un lenguaje matemático eficiente en un vasto dominio de la física [49].

Wolfram Language es un lenguaje de programación que unifica varios paradigmas de la programación, como la programación procedural, simbólica, basada en reglas y funcional. Y, ya que el álgebra geométrica promete estimular nuevos métodos y conocimientos en todas las áreas de la ciencia que se ocupan de las propiedades geométricas, poder realizar estos cálculos de forma simbólica permitirá mantener su intuición geométrica. Es por esto que un paquete en el que se implementaran las ideas fundamentales del Álgebra Geométrica Conforme, usando el lenguaje de programación *Wolfram Language*, puede representar una útil herramienta que genera una solución fácil para manejar implementaciones del AG para *Mathematica*.

Este paquete también permite a los usuarios de *Mathematica* contar con una interfaz intuitiva para el desarrollo, prueba y visualización de algoritmos del AGC, combinando la intuición geométrica del AG con el desarrollo eficiente de algoritmos que proporciona *Wolfram Mathematica*, lo cual contribuye en permitir el uso del AG y hacerlo disponible a una amplia audiencia, lo cual es el propósito de este trabajo [50].

En esta tesis, además, se aplicó el paquete al problema de la geometría de distancias y se mostró que es una herramienta útil para quienes trabajan en este tipo de problemas, que tienen aplicaciones importantes. Vimos que algunos ejemplos de estos problemas se encuentran en la conformación molecular y la navegación hiperbólica. Adicionalmente, *CGAlgebra* permite a los usuarios definir sus propias funciones, de manera que puede ser adaptado a distintos problemas que se aborden con AGC.

6.2 Discusión

Es importante mencionar que existen otros paquetes y programas especializados para cálculos en AGC tales como:

- GAViewer: Diseñado para computación y visualización de objetos en AGC [12].
- GAALOP: Diseñado para producir implementaciones robustas y rápidas en el AG y AGC.[14]
- Clucalc: Un herramienta de software interactivo que visualiza grupos espaciales en 3D y admite cálculos de AG [11].
- GMac: Con un diseño centrado en la optimización de los algoritmos de AG, en términos de tiempo de ejecución y robustez [51].
- Ganja.js: Un paquete basado en javascript para la web, el cual permite el cálculo de algoritmos de álgebra geométrica, y proporciona también una visualización en 3D [52].

- GrassmannAlgebra [53], un paquete *Mathematica* para realizar cálculos en álgebra de Grassman, que ya no cuenta con soporte para las nuevas versiones.
- MathematicaCGA, es un paquete *Mathematica* que podría considerarse el más cercano al que proponemos, su formulación está orientada específicamente a resolver problemas de origami [15].
- La biblioteca de álgebra geométrica para *Mathematica* de Terje Vold, que puede realizar cálculos del espacio tridimensional y del espacio-tiempo tetradimensional [54].

Estos paquetes y herramientas requieren un conocimiento más especializado de CGA, la interface para el usuario es poco intuitiva y no son fáciles de usar, requiriendo una curva de aprendizaje. Por su parte el paquete presentado aquí es intuitivo, fácil de usar, y puede ser utilizado por no matemáticos o especialistas en AGC, ya que permite mantener la intuición geométrica durante su uso. Otra desventaja de estos paquetes en comparación con el que se presenta en esta tesis es que solo permiten cálculos numéricos. Por su parte, CGAlgebra permite realizar operaciones numéricas y simbólicas. Todo esto permite que el paquete sea compatible con el usuario de *Mathematica* con una interfaz intuitiva para el desarrollo, las pruebas y la visualización de algoritmos que utilizan CGA, permitiendo combinar la intuición geométrica del álgebra geométrica con un desarrollo eficiente de algoritmos en *Wolfram mathematica* [49].

6.3 En perspectiva

Dentro de las posibilidades para dar continuidad al trabajo destaca la oportunidad de extender el paquete para manipular objetos geométricos en más de tres dimensiones. Se ha demostrado en [7] que para manipular objetos en \mathbb{R}^n , se construye un ACG en $\mathbb{R}^{n+1,1}$. Esta generalización es una continuación interesante de este trabajo.

Bibliografía

- [1] Kenichi Kanatani. *Understanding geometric algebra: Hamilton, Grassmann, and Clifford for computer vision and graphics*. CRC Press, 2015.
- [2] William K Clifford. “On the classification of geometric algebras”. In: *Mathematical Papers* (1882), pages 397–401.
- [3] Dietmar Hildenbrand et al. “Geometric Algebra and its Application to Computer Graphics.” In: *Eurographics (Tutorials)*. 2004.
- [4] David Hestenes and Garret Sobczyk. *Clifford algebra to geometric calculus: a unified language for mathematics and physics*. Volume 5. Springer Science & Business Media, 2012.
- [5] David Hestenes and Rensus Ziegler. “Projective geometry with Clifford algebra”. In: *Acta Applicandae Mathematica* 23.1 (1991), pages 25–63.
- [6] David Hestenes. “Old wine in new bottles: A new algebraic framework for computational geometry”. In: *Geometric Algebra with Applications in Science and Engineering*. Springer, 2001, pages 3–17.
- [7] Hongbo Li, David Hestenes, and Alyn Rockwood. “Generalized homogeneous coordinates for computational geometry”. In: *Geometric Computing with Clifford Algebras*. Springer, 2001, pages 27–59.
- [8] Leo Dorst, Daniel Fontijne, and Stephen Mann. *Geometric Algebra for Computer Science (Revised Edition)*. Elsevier Inc., 2009.
- [9] Christian Perwass et al. *Geometric algebra with applications in engineering*. Volume 4. Springer, 2009.
- [10] O Faugeras. *Panoramic vision: sensors, theory, and applications*. Springer Science & Business Media, 2001.
- [11] Eckhard Hitzer and Christian Perwass. “Interactive 3D space group visualization with CLUCalc and the Clifford Geometric Algebra description of space groups”. In: *Advances in applied Clifford algebras* 20.3 (2010), pages 631–658.

- [12] Leo Dorst and Daniël Fontijne. *3d euclidean geometry through conformal geometric algebra (a gviewer tutorial)*. 2003.
- [13] Eduardo Bayro-Corrochano. *Geometric computing: for wavelet transforms, robot vision, learning, control and action*. Springer, 2010.
- [14] Dietmar Hildenbrand, Joachim Pitt, and Andreas Koch. “Gaalop—high performance parallel computing based on conformal geometric algebra”. In: *Geometric Algebra Computing*. Springer, 2010, pages 477–494.
- [15] Mitsuhiro Kondo et al. “A Mathematica module for Conformal Geometric Algebra and Origami Folding.” In: SCSS. 2016, pages 68–80.
- [16] Alan Macdonald. “An elementary construction of the geometric algebra”. In: *Advances in applied Clifford algebras* 12.1 (2002), pages 1–6.
- [17] M.A. Rodríguez-Andrade et al. “An algorithm for the Cartan–Dieudonné theorem on generalized scalar product spaces”. In: *Linear Algebra and its Applications* 434.5 (2011), pages 1238–1254.
- [18] E. Snapper and R.J. Troyer. *Metric Affine Geometry*. Academic Press, 1971.
- [19] E Alejandra Ortiz-Durán and José L Aragón. “CGAlgebra: a Mathematica package for conformal geometric algebra. v. 2.0”. In: *arXiv e-prints* (2022), arXiv–1711.02513.
- [20] Leo Dorst. “The Inner Products of Geometric Algebra”. In: *Applications of Geometric Algebra in Computer Science and Engineering*. Edited by Leo Dorst, Chris Doran, and Joan Lasenby. Boston, MA: Birkhäuser Boston, 2002, pages 35–46. ISBN: 978-1-4612-0089-5. DOI: 10.1007/978-1-4612-0089-5_2. URL: https://doi.org/10.1007/978-1-4612-0089-5_2.
- [21] Carlile Lavor, Michael Souza, and José Luis Aragón. “Orthogonality of isometries in the conformal model of the 3D space”. In: *Graphical Models* 114 (2021), page 101100.
- [22] Dietmar Hildenbrand. “Foundations of geometric algebra computing”. In: *AIP Conference Proceedings*. Volume 1479. 1. American Institute of Physics. 2012, pages 27–30.
- [23] Sal Mangano. *Mathematica Cookbook: Building Blocks for Science, Engineering, Finance, Music, and More*. " O’Reilly Media, Inc.", 2010.
- [24] Leo Dorst, Daniel Fontijne, and Stephen Mann. *Geometric algebra for computer science: an object-oriented approach to geometry*. Elsevier, 2010.
- [25] Marcelo Carielo. “Intersection of spheres and geometric algebra”. PhD thesis. Universidade Estadual de Campinas (Brésil), 2019.
- [26] Yechiam Yemini. *The positioning problem—a draft of an intermediate summary*. Technical report. UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST, 1978.
- [27] Antonio Mucherino et al. *Distance geometry: theory, methods, and applications*. Springer Science & Business Media, 2012.
- [28] Leo Liberti and Carlile Lavor. “Six mathematical gems from the history of distance geometry”. In: *International Transactions in Operational Research* 23.5 (2016), pages 897–920.
- [29] Pratik Biswas et al. “Semidefinite programming approaches for sensor network localization with noisy distance measurements”. In: *IEEE transactions on automation science and engineering* 3.4 (2006), pages 360–371.

- [30] Jack D Dunitz. *Distance geometry and molecular conformation*, by GM Crippen and TF Havel, Research Studies Press, Taunton, England, John Wiley and Sons, New York, 1988. 1990.
- [31] Gordon M Crippen, Timothy F Havel, et al. *Distance geometry and molecular conformation*. Volume 74. Research Studies Press Taunton, 1988.
- [32] Leonard Mascot Blumenthal. *Theory and applications of distance geometry*. Volume 242. Chelsea Publishing Company, Incorporated, 1970.
- [33] John Joseph Sylvester. “Chemistry and algebra”. In: *Nature* 17.432 (1878), page 284.
- [34] Carlile Lavor, Sebastià Xambó-Descamps, and Isiah Zaplana. *A geometric algebra invitation to space-time physics, robotics and molecular geometry*. Springer, 2018.
- [35] Leo Liberti et al. “On the Number of Solutions of the Discretizable Molecular Distance Geometry Problem.” In: *COCOA* 11 (2011), pages 322–342.
- [36] Thérèse E Malliavin, Antonio Mucherino, and Michael Nilges. “Distance geometry in structural biology: new perspectives”. In: *Distance Geometry*. Springer, 2013, pages 329–350.
- [37] Leo Liberti, Carlile Lavor, and Nelson Maculan. “A Branch-and-Prune algorithm for the Molecular Distance Geometry Problem”. In: *International Transactions in Operational Research* 15.1 (2008), pages 1–17. doi: <https://doi.org/10.1111/j.1475-3995.2007.00622.x>.
- [38] Robert L Frank. “History of Loran-C”. In: *Navigation* 29.1 (1982), pages 1–6.
- [39] Michael Souza et al. “Solving the molecular distance geometry problem with inaccurate distance data”. In: *BMC bioinformatics* 14.9 (2013), pages 1–6.
- [40] Carlile Lavor et al. “NMR protein structure calculation and sphere intersections”. In: *Computational and Mathematical Biophysics* 8.1 (2020), pages 89–101.
- [41] Francisco G Montoya et al. “Geometric algebra for teaching AC circuit theory”. In: *International Journal of Circuit Theory and Applications* 49.11 (2021), pages 3473–3487.
- [42] Eduardo Bayro-Corrochano. *Geometric algebra applications vol. II: Robot modelling and control*. Springer Nature, 2020.
- [43] Xin Huang and Lei Gao. “Reconstructing three-dimensional human poses: a combined approach of iterative calculation on skeleton model and conformal geometric algebra”. In: *Symmetry* 11.3 (2019), page 301.
- [44] Charles G Gunn and Steven De Keninck. “Geometric algebra and computer graphics”. In: *ACM SIGGRAPH 2019 Courses*. 2019, pages 1–140.
- [45] Yingzhi Wang and Feng Zhang. “An unified CGA-based formal expression of spatio-temporal topological relations for computation and analysis of geographic objects”. In: *Advances in Applied Clifford Algebras* 29 (2019), pages 1–17.
- [46] Zhifeng Shi et al. “Calculation for multidimensional topological relations in 3D cadastre based on geometric Algebra”. In: *ISPRS International Journal of Geo-Information* 8.11 (2019), page 469.
- [47] Uzair Aslam Bhatti et al. “Geometric algebra applications in geospatial artificial intelligence and remote sensing image processing”. In: *IEEE Access* 8 (2020), pages 155783–155796.
- [48] Silvia Franchini et al. “Implementation and evaluation of medical imaging techniques based on conformal geometric algebra”. In: *International Journal of Applied Mathematics and Computer Science* 30.3 (2020), pages 415–433.

-
- [49] G Aragon-Camarasa et al. “Clifford algebra with mathematica”. In: *arXiv preprint arXiv:0810.2412* (2008).
- [50] Geometric Algebra Computing and Dietmar Hildenbrand. “Geometric Algebra enhanced Precompiler for C++ and OpenCL”. In: (2012).
- [51] Ahmad H Eid et al. “Gmac: A novel code generator for geometric modeling based on geometric algebra”. In: *The 19th International Conference on Computer Theory and Applications. ICCTA, Alexandria, Egypt. 2009.*
- [52] Stephane Breuils, Kanta Tachibana, and Eckhard Hitzer. “New applications of Clifford’s geometric algebra”. In: *Advances in Applied Clifford Algebras* 32.2 (2022), pages 1–39.
- [53] John Browne. “Grassmann algebra”. In: *Mebourne, Australia: Quantica Publishing* (2009).
- [54] Rafael Alves et al. “Efficient development of competitive Mathematica solutions based on geometric algebra with GAALOPWeb”. In: *Advances in Applied Clifford Algebras* 30.4 (2020), pages 1–18.