



Deposited via The University of Leeds.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/111312/>

Version: Accepted Version

Article:

Kisil, VV (2013) Induced Representations and Hypercomplex Numbers. *Advances in Applied Clifford Algebras*, 23 (2). pp. 417-440. ISSN: 0188-7009

<https://doi.org/10.1007/s00006-012-0373-1>

© 2012 Springer Basel. This is an author produced version of a paper published in *Advances in Applied Clifford Algebras*. Uploaded in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

INDUCED REPRESENTATIONS AND HYPERCOMPLEX NUMBERS

VLADIMIR V. KISIL

ABSTRACT. In the search for hypercomplex analytic functions on the half-plane, we review the construction of induced representations of the group $G = \mathrm{SL}_2(\mathbb{R})$. Firstly we note that G -action on the homogeneous space G/H , where H is any one-dimensional subgroup of $\mathrm{SL}_2(\mathbb{R})$, is a linear-fractional transformation on hypercomplex numbers. Thus, we investigate various hypercomplex characters of subgroups H . The correspondence between the structure of the group $\mathrm{SL}_2(\mathbb{R})$ and hypercomplex numbers can be illustrated in many other situations as well. We give examples of induced representations of $\mathrm{SL}_2(\mathbb{R})$ on spaces of hypercomplex valued functions, which are unitary in some sense. Raising/lowering operators for various subgroup prompt hypercomplex coefficients as well.

CONTENTS

1. Introduction	2
2. The Group $\mathrm{SL}_2(\mathbb{R})$ and Its Subgroups	4
3. Action of $\mathrm{SL}_2(\mathbb{R})$ as a Source of Hypercomplex Numbers	5
4. Hypercomplex Characters—an Algebraic Approach	6
5. A Parabolic Wheel—a Geometrical Viewpoint	7
6. Rebuilding Algebraic Structures from Geometry	9
6.1. Modulus and Argument	9
6.2. Rotation as Multiplication	10
7. Invariant Linear Algebra	11
8. Induced Representations	12
9. Similarity and Correspondence: Ladder Operators	14
10. Open Problems	17
References	18
1. Введение	1
2. Группа $\mathrm{SL}_2(\mathbb{R})$ и её подгруппы	3
3. Действие $\mathrm{SL}_2(\mathbb{R})$ приводит к гиперкомплексным числам	4
4. Гиперкомплексные характеры—алгебраический подход	6
5. Параболическое колесо из геометрического действия	8
6. Восстановление алгебраической структуры из геометрии	10
6.1. Модуль и аргумент	10
6.2. Параболический поворот как умножение	11
7. Инвариантная линейная алгебра	12
7.1. Тропическая форма	12
7.2. Экзотическая форма	12
8. Индуцированные представления	13

2000 *Mathematics Subject Classification*. Primary 22D30; Secondary 08A99, 13A50, 15A04, 20H05, 51M10.

Key words and phrases. induced representation, unitary representations, $\mathrm{SL}_2(\mathbb{R})$, semisimple Lie group, complex numbers, dual numbers, double numbers, Möbius transformations, split-complex numbers, parabolic numbers, hyperbolic numbers, raising/lowering operators, creation/annihilation operators.

On leave from Odessa University.

9. Принцип сходства и соответствия: операторы повышения	16
10. Заключение: открытые вопросы	19
Благодарности	19
Список литературы	19
Appendix A. Output of Symbolic Calculations	23
Appendix B. Library for Dual Number Algebra	25
B.1. Class dual_number	25
B.2. Algebraic Subroutines	27
B.3. Library Outline	33
B.4. Header and Implementation of the dual_number Class	35
Appendix C. Program for Symbolic Calculations	43
C.1. Calculation and Tests	43
C.2. Induced Representations	46
C.3. Program Outline	48
C.4. Drawing Orbits	51

1. INTRODUCTION

Analytic functions of a complex variable form a beautiful theory with rich applications in many fields ranging from number theory to electrical engineering. Thus, it is natural to look for its analogs and generalisations in different directions. The most basic (or fundamental?) situation appears if we replace the complex imaginary unit $i^2 = -1$ with either the hyperbolic one $\epsilon^2 = +1$ or the nilpotent $\epsilon^2 = 0$.

Two-dimensional commutative associative algebra over reals generated by 1 and ϵ consists of numbers $x + \epsilon y$, where $x, y \in \mathbb{R}$. They are known as split-complex, duplex, hyperbolic or *double numbers* [4, 17, 48, 55]. The algebra has zero divisors $\epsilon_{\pm} = \frac{1}{2}(1 \pm \epsilon)$ with the properties $\epsilon_{\pm}^2 = \epsilon_{\pm}$ and $\epsilon_+ \epsilon_- = 0$. Thus, double numbers are isomorphic to $\mathbb{R} \oplus \mathbb{R}$ —the direct sum of two copies of the real line spanned by ϵ_+ and ϵ_- . This explains the names “split-complex” and “double”.

The analogous algebra associated to the nilpotent unit ϵ consists of elements $x + \epsilon y$, which are called *dual numbers* [6, 15, 59]. All zero divisors in the algebra are ϵy , $y \in \mathbb{R}$. Physical applications of hypercomplex numbers are scattered through classical mechanics [38, 59], non-linear dynamics [48, 49], relativity [4, 9, 55], cosmology [12, 15] and quantum mechanics [17, 37, 56].

Remark 1.1. Unfortunately, there are no common notations for hypercomplex units. Moreover, it would be difficult simply to list the whole variety of symbols employed for this. Even the complex imaginary unit i is oftenly written as j in engineering. The hyperbolic unit is denoted by ϵ in many papers starting at least from the foundational article [58]; although a different letter e is used in the remarkable book [59]. The symbol ι is used for the nilpotent unit in [13, 50], however we chose ϵ following Yaglom [59]. The later notation is also suggestive in light of the following remark.

Remark 1.2. The parabolic unit ϵ is a close relative to the infinitesimal number ϵ from non-standard analysis [8, 57]. The former has the property that its square is *exactly* zero, meanwhile the square of the latter is *almost* zero at its own scale. In fact, there is a version of non-standard analysis [3] employing the nilpotent unit ϵ as an infinitesimal¹. Also, some non-standard proofs of the main calculus theorems are given in [6]. A similar property allows to obtain classical mechanics from the representations of the Heisenberg group [37, 38].

¹I am grateful to the anonymous referee for pointing my attention to the book [3] by J.L. Bell.

What kind of “analytic” functions can be associated with dual and double numbers? Since this question is very natural it was addressed over a prolonged period of time by various researchers. Many of them were unaware of works of their predecessors, neither I can claim to possess the complete knowledge. Below is a brief summary of several works known to me.

For double numbers, a systematic study was already accomplished in [58], there are also numerous later investigations and surveys, see [4; 17; 22; 41, Part IV; 43; 44; 47; 48; 53] and further references therein. The existing consensus is based on the factorisation of the wave equation $\partial_x^2 - \partial_y^2 = (\partial_x - \epsilon\partial_y)(\partial_x + \epsilon\partial_y)$ into a product of two linear differential operators. This is an analog of the factorisation of the Laplacian $\partial_x^2 + \partial_y^2 = (\partial_x - i\partial_y)(\partial_x + i\partial_y)$ into the product of the Cauchy–Riemann operator and its adjoint. Thus, hyperbolic analytic functions are defined to be null solutions of the operator $\partial_x + \epsilon\partial_y$. However, the split of dual number in the basis ϵ_{\pm} reduces an “analytic” function $f(x, y) = \epsilon_+ f_+(x + y) - \epsilon_- f_-(x - y)$ to the sum, where f_{\pm} are two generic differentiable real-valued functions of a real variable. This is related to the representation of the generic solution of the wave equation on the infinite string as a sum of a wave travelling to the left and another travelling to the right.

For the nilpotent unit ϵ the situation is even more trivial. The above factorisation approach does not lead anywhere useful², since $(\partial_x - \epsilon\partial_y)(\partial_x + \epsilon\partial_y) = \partial_x^2$. An attempt to define analytic functions in terms of power series of $x + \epsilon y$ produces only functions of the form $f(x) + \epsilon y f'(x)$ for a real-analytic function $f(x)$ of a single variable [6].

In the series of previous works [21–24] we proposed to develop an analytic function theory in the spirit of the Erlangen Programme of F. Klein: from an appropriate group action. The present paper is a further step towards that goal. We observe that classical spaces of complex analytical functions—the Hardy and Bergman spaces—are irreducible moduli under certain representations of the group $SL_2(\mathbb{R})$. Moreover, those representations are induced by complex-valued characters of the compact subgroup $K \subset SL_2(\mathbb{R})$. Thus, we hope to find *other types of analytic function spaces among all irreducible (or primary) $SL_2(\mathbb{R})$ -moduli under induced representations.*

The paper outline is as follows. We recall the structure of $SL_2(\mathbb{R})$ and list its tree non-isomorphic one-dimensional continuous subgroups in Section 2. This information is employed to classify tree non-isomorphic two-dimensional $SL_2(\mathbb{R})$ -homogeneous spaces in Section 3. We discover that three kinds of hypercomplex numbers are perfectly suited to describe $SL_2(\mathbb{R})$ -action on those homogeneous spaces. To induce representations from one-dimensional subgroups we describe their characters (from the algebraic point of view) in Section 4. The next Section 5 provides an additional objects, which can be also viewed as characters with geometrical spirit. An induced representation can be linear if the inducing characters is also linear. For geometric characters this is not straightforward and we discuss various possibilities in Sections 6 and 7. Finally, we are able to write down various induced representations in Section 8. A convenient description of an irreducible $SL_2(\mathbb{R})$ -module can be produced in terms of a linear basis, consisting of eigenvectors of the subgroup inducing the representation. The Lie algebra \mathfrak{sl}_2 of $SL_2(\mathbb{R})$ acts transitively on this basis by means of ladder operators, which are important in quantum mechanics. We describe the respective structures in Section 9. The paper is concluded by an (incomplete) list of interesting directions for further research.

Acknowledgement: I am grateful to the anonymous referee who made many useful suggestions and comments, which helped to improve this paper.

²However, there is an interesting factorisation of a parabolic operator $\partial_x^2 + \partial_y$ which requires Clifford algebras of higher dimensions [7].

2. THE GROUP $SL_2(\mathbb{R})$ AND ITS SUBGROUPS

Let $SL_2(\mathbb{R})$ be **the group of 2×2 matrices** with real entries and of determinant one [42]. This is the smallest semisimple Lie group. Any matrix in $SL_2(\mathbb{R})$ admits a (unique) decomposition of the form [16, Exer. I.14]:

$$(1) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{pmatrix} \begin{pmatrix} 1 & \nu \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix},$$

for the following real values of the parameters:

$$\alpha = \sqrt{c^2 + d^2} \in (0, \infty), \quad \nu = ac + bd \in (-\infty, \infty), \quad \phi = -\arctan \frac{c}{d} \in (-\pi, \pi].$$

The formula (1) rewritten in a way $SL_2(\mathbb{R}) = ANK$ is known as Iwasawa decomposition [42, § III.1] and can be generalised to any semisimple Lie group.

Each out of the three types of matrices in the right-hand side of (1) forms a one-parameter subgroup A, N and K. They are obtained by the exponentiation of the respective zero-trace matrices:

$$(2) \quad A = \left\{ \begin{pmatrix} e^t & 0 \\ 0 & e^{-t} \end{pmatrix} = \exp \begin{pmatrix} t & 0 \\ 0 & -t \end{pmatrix}, t \in \mathbb{R} \right\},$$

$$(3) \quad N = \left\{ \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} = \exp \begin{pmatrix} 0 & t \\ 0 & 0 \end{pmatrix}, t \in \mathbb{R} \right\},$$

$$(4) \quad K = \left\{ \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} = \exp \begin{pmatrix} 0 & t \\ -t & 0 \end{pmatrix}, t \in (-\pi, \pi] \right\}.$$

The following simple result have an instructive proof.

Proposition 2.1. *Any continuous one-parameter subgroup of $SL_2(\mathbb{R})$ is conjugate to one of subgroups A, N or K.*

Proof. Any one-parameter subgroup is obtained through the exponentiation

$$(5) \quad e^{tX} = \sum_{n=0}^{\infty} \frac{t^n}{n!} X^n$$

of an element X of the Lie algebra \mathfrak{sl}_2 of $SL_2(\mathbb{R})$. Such X is a 2×2 matrix with the zero trace. The behaviour of the Taylor expansion (5) depends from properties of powers X^n . This can be classified by a straightforward calculation:

Lemma 2.2. *The square X^2 of a traceless matrix $X = \begin{pmatrix} a & b \\ c & -a \end{pmatrix}$ is the identity matrix times $a^2 + bc = -\det X$. The factor can be negative, zero or positive, which corresponds to the three different types of the Taylor expansion (5) of e^{tX} .*

It is a simple exercise on characteristic polynomials to see that through the matrix similarity we can obtain from X a generator

- of the subgroup K if $(-\det X) < 0$;
- of the subgroup N if $(-\det X) = 0$;
- of the subgroup A if $(-\det X) > 0$.

The determinant is invariant under the similarity, thus these cases are distinct. \square

Example 2.3. The following two subgroups are conjugated to A and N respectively:

$$(6) \quad A' = \left\{ \begin{pmatrix} \cosh t & \sinh t \\ \sinh t & \cosh t \end{pmatrix} = \exp \begin{pmatrix} 0 & t \\ t & 0 \end{pmatrix}, t \in \mathbb{R} \right\},$$

$$(7) \quad N' = \left\{ \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} = \exp \begin{pmatrix} 0 & 0 \\ t & 0 \end{pmatrix}, t \in \mathbb{R} \right\}.$$

3. ACTION OF $SL_2(\mathbb{R})$ AS A SOURCE OF HYPERCOMPLEX NUMBERS

Let H be a subgroup of a group G . Let $\Omega = G/H$ be the corresponding homogeneous space and $s : \Omega \rightarrow G$ be a smooth section [18, § 13.2], which is a left inverse to the natural projection $p : G \rightarrow \Omega$. The choice of s is inessential in the sense that by a smooth map $\Omega \rightarrow \Omega$ we can always reduce one to another.

Any $g \in G$ has a unique decomposition of the form $g = s(\omega)h$, where $\omega = p(g) \in \Omega$ and $h \in H$. Note that Ω is a left homogeneous space with the G -action defined in terms of p and s as follows:

$$(8) \quad g : \omega \mapsto g \cdot \omega = p(g * s(\omega)),$$

where $*$ is the multiplication on G . This is also illustrated by the following commutative diagram:

$$\begin{array}{ccc} G & \xrightarrow{g*} & G \\ s \downarrow & & \downarrow s \\ \Omega & \xrightarrow{g \cdot} & \Omega \end{array}$$

For $G = SL_2(\mathbb{R})$, as well as for other semisimple groups, it is common to consider only the case of H being the maximal compact subgroup K . However in this paper we admit H to be any one-dimensional subgroup. Due to the previous Proposition it is sufficient to take $H = K, N'$ or A' . Then Ω is a two-dimensional manifold and for any choice of H we define [22, Ex. 3.7(a)]:

$$(9) \quad s : (u, v) \mapsto \frac{1}{\sqrt{v}} \begin{pmatrix} v & u \\ 0 & 1 \end{pmatrix}, \quad (u, v) \in \mathbb{R}^2, v > 0.$$

A direct (or computer algebra [30]) calculation show that:

Proposition 3.1. *The $SL_2(\mathbb{R})$ action (8) associated to the map s (9) is:*

$$(10) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} : (u, v) \mapsto \left(\frac{(au + b)(cu + d) - \sigma cv^2}{(cu + d)^2 - \sigma (cv)^2}, \frac{v}{(cu + d)^2 - \sigma (cv)^2} \right),$$

where $\sigma = -1, 0$ and 1 for the subgroups K, N' and A' respectively.

The expression in (10) does not look very appealing, however an introduction of hypercomplex numbers makes it more attractive:

Proposition 3.2. *Let a hypercomplex unit ι be such that $\iota^2 = \sigma$, then the $SL_2(\mathbb{R})$ action (10) becomes:*

$$(11) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} : w \mapsto \frac{aw + b}{cw + d}, \quad \text{where } w = u + \iota v,$$

for all three cases parametrised by σ as in Prop. 3.1.

Remark 3.3. We wish to stress that the hypercomplex numbers were not introduced here by our intention, arbitrariness or “generalising attitude” [51, p. 4]. They were naturally created by the $SL_2(\mathbb{R})$ action.

Notably the action (11) is a group homomorphism of the group $SL_2(\mathbb{R})$ into transformations of the “upper half-plane” on hypercomplex numbers. Although dual and double numbers are algebraically trivial, the respective geometries in the spirit of **Erlangen programme** are refreshingly inspiring [19, 31, 36] and provide useful insights even in the elliptic case [27]. In order to treat divisors of zero, we need to consider Möbius transformations (11) of conformally completed plane [15, 29].

Now we wish to linearise the action (8) through the **induced representations** [18, § 13.2; 22, § 3.1]. We define a map $r : G \rightarrow H$ associated to p and s from the identities:

$$(12) \quad r(g) = (s(\omega))^{-1}g, \quad \text{where } \omega = p(g) \in \Omega.$$

Let χ be an irreducible representation of H in a vector space V , then it induces a representation of G in the sense of Mackey [18, § 13.2]. This representation has the realisation ρ_χ in the space of V -valued functions by the formula [18, § 13.2.(7)–(9)]:

$$(13) \quad [\rho_\chi(g)f](\omega) = \chi(r(g^{-1} * s(\omega)))f(g^{-1} \cdot \omega),$$

where $g \in G$, $\omega \in \Omega$, $h \in H$ and $r : G \rightarrow H$, $s : \Omega \rightarrow G$ are maps defined above; $*$ denotes multiplication on G and \cdot denotes the action (8) of G on Ω .

In our consideration H is always one-dimensional. Traditionally, an irreducible representation of such a subgroup is supposed to be a complex valued character. However, hypercomplex numbers naturally appeared in the $SL_2(\mathbb{R})$ action (11), why shall we admit only $i^2 = -1$ to deliver a character then?

4. HYPERCOMPLEX CHARACTERS—AN ALGEBRAIC APPROACH

As we already mentioned the typical discussion of induced representations of $SL_2(\mathbb{R})$ is centred around the case $H = K$ and a complex valued character of K . A linear transformation defined by a matrix (4) in K is a rotation of \mathbb{R}^2 by the angle t . After identification $\mathbb{R}^2 = \mathbb{C}$ this action is given by the multiplication e^{it} , with $i^2 = -1$. The rotation preserve the (elliptic) metric given by:

$$(14) \quad x^2 + y^2 = (x + iy)(x - iy).$$

Therefore the orbits of rotations are circles, any line passing the origin (a “spoke”) is rotated by the angle t , see Fig. 1(E).

Introduction of hypercomplex numbers produces the most straightforward adaptation of this result.

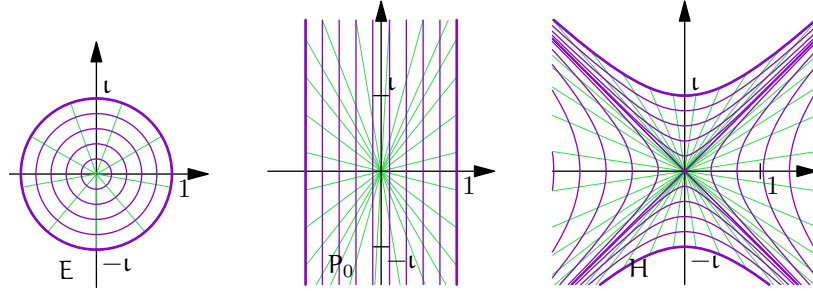


FIGURE 1. Rotations of algebraic wheels, i.e. the multiplication by e^{it} : elliptic (E), trivial parabolic (P_0) and hyperbolic (H). All blue orbits are defined by the identity $x^2 - i^2y^2 = r^2$. Green “spokes” (straight lines from the origin to a point on the orbit) are “rotated” from the real axis.

Proposition 4.1. *The following table show correspondences between three types of algebraic characters:*

<i>Elliptic</i>	<i>Parabolic</i>	<i>Hyperbolic</i>
$i^2 = -1$	$\epsilon^2 = 0$	$\epsilon^2 = 1$
$w = x + iy$	$w = x + \epsilon y$	$w = x + \epsilon y$
$\bar{w} = x - iy$	$\bar{w} = x - \epsilon y$	$\bar{w} = x - \epsilon y$
$e^{it} = \cos t + i \sin t$	$e^{\epsilon t} = 1 + \epsilon t$	$e^{\epsilon t} = \cosh t + \epsilon \sinh t$
$ w _e^2 = w\bar{w} = x^2 + y^2$	$ w _p^2 = w\bar{w} = x^2$	$ w _h^2 = w\bar{w} = x^2 - y^2$
$\arg w = \tan^{-1} \frac{y}{x}$	$\arg w = \frac{y}{x}$	$\arg w = \tanh^{-1} \frac{y}{x}$
<i>unit circle</i> $ w _e^2 = 1$	<i>"unit" strip</i> $x = \pm 1$	<i>unit hyperbola</i> $ w _h^2 = 1$

Geometrical action of multiplication by e^{it} is drawn on Fig. 1 for all three cases.

Explicitly parabolic rotations associated with $e^{\epsilon t}$ acts on dual numbers as follows:

$$(15) \quad e^{\epsilon x} : a + \epsilon b \mapsto a + \epsilon(ax + b).$$

This links the parabolic case with the Galilean group [59] of symmetries of the classic mechanics, with the absolute time disconnected from space.

The obvious algebraic similarity and the connection to classical kinematic is a wide spread justification for the following viewpoint on the parabolic case, cf. [14, 59]:

- the parabolic trigonometric functions are trivial:

$$(16) \quad \operatorname{cosp} t = \pm 1, \quad \operatorname{sinp} t = t;$$

- the parabolic distance is independent from y if $x \neq 0$:

$$(17) \quad x^2 = (x + \epsilon y)(x - \epsilon y);$$

- the polar decomposition of a dual number is defined by [59, App. C(30')]:

$$(18) \quad u + \epsilon v = u(1 + \epsilon \frac{v}{u}), \quad \text{thus} \quad |u + \epsilon v| = u, \quad \arg(u + \epsilon v) = \frac{v}{u};$$

- the parabolic wheel looks rectangular, see Fig. 1(P₀).

Those algebraic analogies are quite explicit and widely accepted as an ultimate source for parabolic trigonometry [14, 43, 59]. Moreover, those three rotations are all non-isomorphic symplectic linear transformations of the phase space, which makes them useful in the context of classical and quantum mechanics [34, 37].

However we will see shortly that there exists an alternative with geometric motivation and connection to equations of mathematical physics.

5. A PARABOLIC WHEEL—A GEOMETRICAL VIEWPOINT

We make another attempt to describe parabolic rotations. If multiplication (a linear transformation) is not sophisticated enough for this we can advance to the next level of complexity: linear-fractional.

Hypercomplex units do not need to be seen as abstract quantities. As follows from Lem. 2.2 the generators of subgroup K , N and A represent units of complex, dual and double numbers respectively. Their exponentiation to one-parameter subgroups K , N' and A' of $SL_2(\mathbb{R})$ produce matrix forms of the Euler identities from the fifth row of the table in Prop. 4.1.

Thus we attempt to define characters of subgroups K , N' and A' in term of geometric action of $SL_2(\mathbb{R})$ by Möbius transformations. The action (11) is defined on the upper half-plane and to relate it to unitary characters we wish to transfer it

to the unit disk. In the elliptic case this is done by the Cayley transform, its action on the subgroup K is:

$$(19) \quad \frac{1}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} = \begin{pmatrix} e^{it} & 0 \\ 0 & e^{-it} \end{pmatrix}.$$

The diagonal matrix in the right hand side define the Moebius transformation which reduces to multiplication by e^{2it} , i.e. the elliptic rotation.

A hyperbolic cousin of the Cayley transform is:

$$(20) \quad \frac{1}{2} \begin{pmatrix} 1 & \epsilon \\ -\epsilon & 1 \end{pmatrix} \begin{pmatrix} \cosh t & \sinh t \\ \sinh t & \cosh t \end{pmatrix} \begin{pmatrix} 1 & -\epsilon \\ \epsilon & 1 \end{pmatrix} = \begin{pmatrix} e^{\epsilon t} & 0 \\ 0 & e^{-\epsilon t} \end{pmatrix},$$

similarly produces a Moebius transformation which is the multiplication by $e^{2\epsilon t}$, which a unitary (Lorentz) transformation of two-dimensional Minkowski space-time.

In the parabolic case we use the similar pattern and define the Cayley transform from the matrix:

$$C_\epsilon = \begin{pmatrix} 1 & -\epsilon \\ -\epsilon & 1 \end{pmatrix}$$

The Cayley transform of matrices (3) from the subgroup N is:

$$(21) \quad \begin{pmatrix} 1 & -\epsilon \\ -\epsilon & 1 \end{pmatrix} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \epsilon \\ \epsilon & 1 \end{pmatrix} = \begin{pmatrix} 1 + \epsilon t & t \\ 0 & 1 - \epsilon t \end{pmatrix} = \begin{pmatrix} e^{\epsilon t} & t \\ 0 & e^{-\epsilon t} \end{pmatrix}.$$

This is not far from the previous identities (19) and (20), however, the off-diagonal (1, 2)-term destroys harmony. Nevertheless we will continue a unitary parabolic rotation to be the Möbius transformation with the matrix (21), which will not be a multiplication by a scalar anymore.

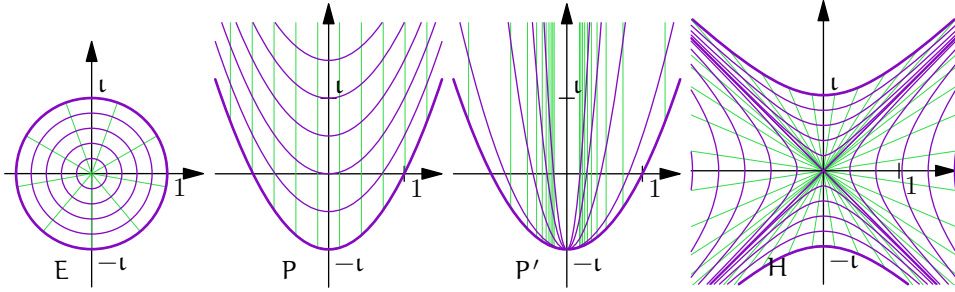


FIGURE 2. Rotation of geometric wheels: elliptic (E), two parabolic (P and P') and hyperbolic (H). Blue orbits are level lines for the respective moduli. Green straight lines join points with the same value of argument and are drawn with the constant "angular step" in each case.

Example 5.1. The parabolic rotations with the upper-triangular matrices from the subgroup N becomes:

$$(22) \quad \begin{pmatrix} e^{\epsilon t} & t \\ 0 & e^{-\epsilon t} \end{pmatrix} : -\epsilon \mapsto t + \epsilon(t^2 - 1).$$

This coincides with the *cyclic rotations* defined in [59, § 8]. A comparison with the Euler formula seemingly confirms that $\sin_p t = t$, but suggests a new expression for $\cos_p t$:

$$\cos_p t = 1 - t^2, \quad \sin_p t = t.$$

Therefore the parabolic Pythagoras' identity would be:

$$(23) \quad \text{sinp}^2 t + \text{cosp} t = 1,$$

which nicely fits in between the elliptic and hyperbolic versions:

$$\sin^2 t + \cos^2 t = 1, \quad \sinh^2 t - \cosh^2 t = -1.$$

The identity (23) is also less trivial than the version $\text{cosp}^2 t = 1$ from [14] (see also (16), (17)).

Example 5.2. There is the second option to define parabolic rotations for the lower-triangular matrices from the subgroup N' . The important difference now is: the reference point cannot be $-\varepsilon$ since it is a fixed point (as well as any point on the vertical axis). Instead we take ε^{-1} , which is an ideal element (a point at infinity [59, App. C]) since ε is a divisor of zero. The proper compactifications by ideal elements for all three cases were discussed in [29].

We get for the subgroup N' :

$$(24) \quad \begin{pmatrix} e^{-\varepsilon t} & 0 \\ t & e^{\varepsilon t} \end{pmatrix} : \frac{1}{\varepsilon} \mapsto \frac{1}{t} + \varepsilon \left(1 - \frac{1}{t^2}\right).$$

A comparison with (22) shows that this form is obtained by the change $t \mapsto t^{-1}$. The same transformation gives new expressions for parabolic trigonometric functions. The parabolic "unit circle" (or *cycle* [31, 36, 59]) is defined by the equation $x^2 - y = 1$ in both cases, see Fig. 2(P) and (P'). However other orbits are different and we will give their description in the next Section.

Fig. 2 illustrates Möbius actions of matrices (19), (20) and (21) on the respective "unit disk", which are images of the upper half-planes under respective Cayley transforms [31, § ??; 36, Ch. 10].

6. REBUILDING ALGEBRAIC STRUCTURES FROM GEOMETRY

We want induced representations to be linear, to this end the inducing character shall be linear as well. Rotations in elliptic and hyperbolic cases are given by products of complex or double numbers respectively and thus are linear. However non-trivial parabolic rotations (22) and (24) (Fig. 2(P) and (P')) are not linear. Can we find algebraic operations for dual numbers, which will linearise those Möbius transformations?

It is common in mathematics to "revert a theorem into a definition" and we will use this systematically to recover a compatible algebraic structure.

6.1. Modulus and Argument. In the elliptic and hyperbolic cases orbits of rotations are points with the constant norm (modulus): either $x^2 + y^2$ or $x^2 - y^2$. In the parabolic case we employ this point of view as well:

Definition 6.1. Orbits of actions (22) and (24) are contour lines for the following functions which we call respective moduli (norms):

$$(25) \quad \text{for } N : |u + \varepsilon v| = u^2 - v, \quad \text{for } N' : |u + \varepsilon v|' = \frac{u^2}{v + 1}.$$

Remark 6.2. (1) The expression $|(u, v)| = u^2 - v$ represents a parabolic distance from $(0, \frac{1}{2})$ to (u, v) , see [31, Lem. ??; 36, Ex. 10.6], which is in line with the "parabolic Pythagoras' identity" (23).

(2) Modulus for N' expresses the parabolic focal length from $(0, -1)$ to (u, v) as described in [31, Lem. ??; 36, Ex. 10.7].

The only straight lines preserved by both the parabolic rotations N and N' are vertical lines, thus we will treat them as “spokes” for parabolic “wheels”. Elliptic spokes in mathematical terms are “points on the complex plane with the same argument”, thus we again use this for the parabolic definition:

Definition 6.3. Parabolic arguments are defined as follows:

$$(26) \quad \text{for } N : \arg(u + \varepsilon v) = u, \quad \text{for } N' : \arg'(u + \varepsilon v) = \frac{1}{u}.$$

Both Definitions 6.1 and 6.3 possess natural properties with respect to parabolic rotations:

Proposition 6.4. Let w_t be a parabolic rotation of w by an angle t in (22) or in (24). Then:

$$|w_t|^{(\prime)} = |w|^{(\prime)}, \quad \arg^{(\prime)} w_t = \arg^{(\prime)} w + t,$$

where primed versions are used for subgroup N' .

All proofs in this and the following Sections were performed through symbolic calculations on a computer. Details can be found in the earlier (more heuristic) paper on this topic [30].

Remark 6.5. Note that in the commonly accepted approach [59, App. C(30')] parabolic modulus and argument are given by expressions (18), which are, in a sense, opposite to our agreements.

6.2. Rotation as Multiplication. We revert again theorems into definitions to assign multiplication. In fact, we consider parabolic rotations as multiplications by unimodular numbers thus we define multiplication through an extension of properties from Proposition 6.4:

Definition 6.6. The product of vectors w_1 and w_2 is defined by the following two conditions:

- (1) $\arg^{(\prime)}(w_1 w_2) = \arg^{(\prime)} w_1 + \arg^{(\prime)} w_2;$
- (2) $|w_1 w_2|^{(\prime)} = |w_1|^{(\prime)} \cdot |w_2|^{(\prime)}.$

We also need a special form of parabolic conjugation, which coincides with sign reversion of the argument.

Definition 6.7. Parabolic conjugation is given by

$$(27) \quad \overline{u + \varepsilon v} = -u + \varepsilon v.$$

Obviously we have the properties: $|\overline{w}|^{(\prime)} = |w|^{(\prime)}$ and $\arg^{(\prime)} \overline{w} = -\arg^{(\prime)} w$. A combination of Definitions 6.1, 6.3 and 6.6 uniquely determine expressions for products.

Proposition 6.8. The parabolic product of vectors is defined by formulae:

$$(28) \quad \text{for } N : \quad (u, v) * (u', v') = (u + u', (u + u')^2 - (u^2 - v)(u'^2 - v'));$$

$$(29) \quad \text{for } N' : \quad (u, v) * (u', v') = \left(\frac{uu'}{u + u'}, \frac{(v + 1)(v' + 1)}{(u + u')^2} - 1 \right).$$

Although both expressions look unusual they have many familiar properties:

Proposition 6.9. Both products (28) and (29) satisfy the following conditions:

- (1) They are commutative and associative;
- (2) The respective rotations (22) and (24) are given by multiplications with a dual number with the unit norm.
- (3) The product $w_1 \overline{w}_2$ is invariant under respective rotations (22) and (24).

(4) For any dual number w the following identity holds:

$$|w\bar{w}| = |w|^2.$$

In particular, the property (3) will be crucial below for an inner product (38), which makes induced representations unitary.

7. INVARIANT LINEAR ALGEBRA

Now we wish to define a linear structure on \mathbb{R}^2 which would be invariant under point multiplication from the previous Subsection (and thus under the parabolic rotations, cf. Prop. 6.9(2)). Multiplication by a real scalar is straightforward (at least for a positive scalar): it should preserve the argument and scale the norm of a vector. Thus we have formulae for $\alpha > 0$:

$$(30) \quad \alpha \cdot (u, v) = (u, \alpha v + u^2(1 - \alpha)) \quad \text{for } N,$$

$$(31) \quad \alpha \cdot (u, v) = \left(u, \frac{v+1}{\alpha} - 1 \right) \quad \text{for } N'.$$

On the other hand, the addition of vectors can be done in several different ways. One of them, related to tropical mathematics [5, 45], is outlined in Example 10.1. Here, we present another alternative with all due details.

Addition of vectors for both subgroups N and N' can be defined by the common rules, where subtle differences are hidden within corresponding Definitions 6.1 (norms) and 6.3 (arguments).

Definition 7.1. Parabolic addition of vectors is defined by the following formulae:

$$(32) \quad \arg^{(\prime)}(w_1 + w_2) = \frac{\arg^{(\prime)} w_1 \cdot |w_1|^{(\prime)} + \arg^{(\prime)} w_2 \cdot |w_2|^{(\prime)}}{|w_1 + w_2|^{(\prime)}},$$

$$(33) \quad |w_1 + w_2|^{(\prime)} = |w_1|^{(\prime)} \pm |w_2|^{(\prime)},$$

where primed versions are used for the subgroup N' .

The rule for the norm of sum (33) may look too trivial at the first glance. We should say in its defence that it nicely sits in between the elliptic $|w + w'| \leq |w| + |w'|$ and hyperbolic $|w + w'| \geq |w| + |w'|$ triangle inequalities for norms.

The rule (32) for argument of the sum is not arbitrary as well. From the Sine Theorem in the Euclidean geometry we can deduce that:

$$\sin(\phi - \psi') = \frac{|w| \cdot \sin(\psi - \psi')}{|w + w'|}, \quad \sin(\psi' - \phi) = \frac{|w'| \cdot \sin(\psi - \psi')}{|w + w'|},$$

where $\psi^{(\prime)} = \arg w^{(\prime)}$ and $\phi = \arg(w + w^{(\prime)})$. Using parabolic expression (16) for the sine $\sin \theta = \theta$ we obtain the arguments addition formula (32).

A proper treatment of zeros in denominator of (32) can be achieved through a representation of a dual number $w = u + \varepsilon v$ as a pair of homogeneous polar coordinates $[a, r] = [|w|^{(\prime)} \cdot \arg^{(\prime)} w, |w|^{(\prime)}]$ (dashed version for the subgroup N'). Then the above addition is defined component-wise in the homogeneous coordinates:

$$w_1 + w_2 = [a_1 + a_2, r_1 + r_2], \quad \text{where } w_i = [a_i, r_i].$$

The multiplication from Defn. 6.6 is given in the homogeneous polar coordinates by:

$$w_1 \cdot w_2 = [a_1 r_2 + a_2 r_1, r_1 r_2], \quad \text{where } w_i = [a_i, r_i].$$

Thus homogeneous coordinates linearise the addition (32)–(33) and multiplication by a scalar (30). A transition to other more transparent coordinates shall be treated withing birational geometry framework [39].

Both formulae (32)–(33) together uniquely define explicit expressions for addition of vectors. However those expressions are rather cumbersome and not really much needed. Instead we list properties of these operations:

Proposition 7.2. *Vector additions for subgroups N and N' defined by (32)–(33) satisfy the following conditions:*

- (1) *They are commutative and associative.*
- (2) *They are distributive for multiplications (28) and (29); consequently:*
- (3) *They are parabolic rotationally invariant;*
- (4) *They are distributive in both ways for the scalar multiplications (30) and (31) respectively:*

$$\mathbf{a} \cdot (\mathbf{w}_1 + \mathbf{w}_2) = \mathbf{a} \cdot \mathbf{w}_1 + \mathbf{a} \cdot \mathbf{w}_2, \quad (\mathbf{a} + \mathbf{b}) \cdot \mathbf{w} = \mathbf{a} \cdot \mathbf{w} + \mathbf{b} \cdot \mathbf{w}.$$

To complete the construction we need to define the zero vector and the inverse. The inverse of w has the same argument as w and the opposite norm.

Proposition 7.3. (N) *The zero vector is $(0, 0)$ and consequently the inverse of (u, v) is $(u, 2u^2 - v)$.*

(N') *The zero vector is $(\infty, -1)$ and consequently the inverse of (u, v) is $(u, -v - 2)$.*

Thereafter we can check that scalar multiplications by negative reals are given by the same identities (30) and (31) as for positive ones.

Remark 7.4. The irrelevance of the standard linear structure for parabolic rotations manifests itself in many different ways, e.g. in an apparent “non-conformality” of lengths from parabolic foci, that is with the parameter $\sigma = 0$ in [31, Prop. ??; 36, Ex. 7.14.iii]. An adjustment of notions to the proper framework restores the clear picture.

The initial definition of conformality [31, Defn. ??; 36, Defn. 7.13] considered the usual limit $y' \rightarrow y$ along a straight line, i.e. “spoke” in terms of Fig. 1. This is justified in the elliptic and hyperbolic cases. However in the parabolic setting the proper “spokes” are vertical lines, see Fig. 2(P) and (P'), so the limit should be taken along them [31, Prop. ??; 36, § 11.6.1].

8. INDUCED REPRESENTATIONS

We discussed above various implementations of hypercomplex unitary characters. Now we can return to consideration of induced representations. We can notice that only the subgroup K requires a complex valued character due to the fact of its compactness. For subgroups N' and A' we can consider characters of all three types—elliptic, parabolic and hyperbolic. Moreover a parabolic character can be taken either as algebraic (15) or any of two geometric (22) and (24). Therefore we have seven essentially different induced representations, which multiply types to eleven (counting flavours of parabolic characters).

Example 8.1. Consider the subgroup $H = K$, then we are limited to complex valued characters of K only. All of them are of the form χ_k :

$$(34) \quad \chi_k \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} = e^{-ikt}, \quad \text{where } k \in \mathbb{Z}.$$

Using the explicit form (9) of the map s we find the map r given in (12) as follows:

$$r \begin{pmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{pmatrix} = \frac{1}{\sqrt{c^2 + d^2}} \begin{pmatrix} \mathbf{d} & -\mathbf{c} \\ \mathbf{c} & \mathbf{d} \end{pmatrix} \in K.$$

Therefore:

$$r(g^{-1} * s(u, v)) = \frac{1}{\sqrt{(cu + d)^2 + (cv)^2}} \begin{pmatrix} cu + d & -cv \\ cv & cu + d \end{pmatrix}, \quad \text{where } g^{-1} = \begin{pmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{pmatrix}.$$

Substituting this into (34) and combining with the Möbius transformation of the domain (11) we get the explicit realisation ρ_k of the induced representation (13):

$$(35) \quad \rho_k(g)f(w) = \frac{|cw + d|^k}{(cw + d)^k} f\left(\frac{aw + b}{cw + d}\right), \quad \text{where } g^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad w = u + iv.$$

This representation acts on complex valued functions in the upper half-plane $\mathbb{R}_+^2 = \text{SL}_2(\mathbb{R})/K$ and belongs to the discrete series [42, § IX.2].

Proposition 8.2. *Let $f_k(w) = \frac{|w-i|^k}{(w-i)^k}$ for $k = 2, 3, \dots$, then*

- (1) f_k is an eigenvector for any operator $\rho_k(h)$, where $h \in K$, with the eigenvalue $\chi_k(h)$ [42, § IX.2].
- (2) The function $K(z, w) = \rho_k(s(z))f_k(w)$, where $s(z)$ is defined in (9), is the Bergman reproducing kernel up to the factor $|\frac{z-i}{w-i}|^k$ in the upper half-plane [22, § 3.2].

Similarly we can get the Cauchy kernel for the limiting case $k = 1$ of the mock discrete series [42, Ch. IX]. There are many other important connections of representation (35) with complex analysis and operator theory. For example, Möbius transformations of operators lead to Riesz-Dunford functional calculus and associated spectrum [25].

Example 8.3. In the case of the subgroup N there is a wider choice of possible characters.

- (1) Traditionally only complex valued characters of the subgroup N are considered, they are:

$$(36) \quad \chi_\tau^{\mathbb{C}} \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} = e^{i\tau t}, \quad \text{where } \tau \in \mathbb{R}.$$

A direct calculation shows that:

$$r \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{c}{d} & 1 \end{pmatrix} \in N'.$$

Thus:

$$(37) \quad r(g^{-1} * s(u, v)) = \begin{pmatrix} 1 & 0 \\ \frac{cv}{d+cu} & 1 \end{pmatrix}, \quad \text{where } g^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

A substitution of this value into the character (36) together with the Möbius transformation (11) we obtain the next realisation of (13):

$$\rho_\tau^{\mathbb{C}}(g)f(w) = \exp\left(i\frac{\tau cv}{cu + d}\right) f\left(\frac{aw + b}{cw + d}\right), \quad \text{where } w = u + \varepsilon v, \quad g^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

The representation acts on the space of *complex* valued functions on the upper half-plane \mathbb{R}_+^2 , which is subset of *dual* numbers as a homogeneous space $\text{SL}_2(\mathbb{R})/N$. The mixture of complex and dual numbers in the same expression is confusing.

- (2) The parabolic character χ_τ with the algebraic flavour is provided by multiplication (15) with the dual number:

$$\chi_\tau \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} = e^{\varepsilon\tau t} = 1 + \varepsilon\tau t, \quad \text{where } \tau \in \mathbb{R}.$$

As before substitute the value (37) into this character we receive the representation:

$$\rho_\tau(g)f(w) = \left(1 + \varepsilon\frac{\tau cv}{cu + d}\right) f\left(\frac{aw + b}{cw + d}\right),$$

where w , τ and g are as above. The representation is defined on the space of dual numbers valued functions on the upper half-plane of dual numbers. Thus expression contains only dual numbers with their usual algebraic operations. Thus it is linear with respect to them.

- (3) The geometric character χ_τ^g is given by the action (22). Then the corresponding representation acts again on the space of dual numbers valued functions on the upper half-plane of dual numbers as follows:

$$\rho_\tau^g(g)f(w) = \left(1 + \varepsilon \frac{2\tau cv}{cu + d}\right) f\left(\frac{aw + b}{cw + d}\right) + \frac{\tau cv}{cu + d} + \varepsilon \frac{(\tau cv)^2}{(cu + d)^2},$$

where w , τ and g are as above. This representation is linear with respect to operations (30), (32) and (33).

All characters in the previous Example are unitary, the first two in a conventional sense and the last one in the sense of Prop. 6.9. Then the general scheme of induced representations [18, § 13.2] implies their unitarity in proper senses.

Theorem 8.4. *All three representations of $SL_2(\mathbb{R})$ from Example 8.3 are unitary on the space of function on the upper half-plane \mathbb{R}_+^2 of dual numbers with the inner product:*

$$(38) \quad \langle f_1, f_2 \rangle = \int_{\mathbb{R}_+^2} f_1(w) \bar{f}_2(w) \frac{du dv}{v^2}, \quad \text{where } w = u + \varepsilon v,$$

and we use

- (1) the conjugation and multiplication of functions' values in algebras of complex and dual numbers for representations $\rho_\tau^{\mathbb{C}}$ and ρ_τ respectively;
- (2) conjugation (27) and multiplication (28) of functions' values for the representation ρ_τ^g .

The inner product (38) is positive defined for the representation $\rho_\tau^{\mathbb{C}}$ but is not for two others. The respective spaces are parabolic cousins of the Krein spaces [1], which are hyperbolic in our sense.

9. SIMILARITY AND CORRESPONDENCE: LADDER OPERATORS

From the above observation we can deduce the following empirical principle, which has a heuristic value.

Principle 9.1 (Similarity and correspondence). (1) *Subgroups K , N and A play the similar rôle in a structure of the group $SL_2(\mathbb{R})$ and its representations.*
 (2) *The subgroups shall be swapped simultaneously with the respective replacement of hypercomplex unit ι .*

The first part of the Principle (similarity) does not look sound alone. It is enough to mention that the subgroup K is compact (and thus its spectrum is discrete) while two other subgroups are not. However in a conjunction with the second part (correspondence) the Principle have received the following confirmations so far:

- The action of $SL_2(\mathbb{R})$ on the homogeneous space $SL_2(\mathbb{R})/H$ for $H = K$, N' or A' is given by linear-fractional transformations of complex, dual or double numbers respectively (Prop. 3.2).
- Subgroups K , N' or A' are isomorphic to the groups of unitary rotations of respective unit cycles in complex, dual or double numbers (Prop. 4.1).
- Representations induced from subgroups K , N' or A' are unitary if the inner product spaces of functions with values in complex, dual or double numbers (Thm. 8.4).

Remark 9.2. The principle of similarity and correspondence resembles supersymmetry between bosons and fermions in particle physics, but we have similarity between three different types of entities in our case.

Let us give another illustration to the Principle. Consider the Lie algebra \mathfrak{sl}_2 of the group $SL_2(\mathbb{R})$. Pick up the following basis in \mathfrak{sl}_2 [54, § 8.1]:

$$A = \frac{1}{2} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

The commutation relation between those elements are:

$$(39) \quad [Z, A] = 2B, \quad [Z, B] = -2A, \quad [A, B] = -\frac{1}{2}Z.$$

Let ρ be a representation of the group $SL_2(\mathbb{R})$ in a space V . Consider the derived representation $d\rho$ of the Lie algebra \mathfrak{sl}_2 [42, § VI.1] and denote $\tilde{X} = d\rho(X)$ for $X \in \mathfrak{sl}_2$. To see the structure of the representation ρ we can decompose the space V into eigenspaces of the operator \tilde{X} for some $X \in \mathfrak{sl}_2$, cf. Prop. 8.2 or the Taylor series in complex analysis.

Example 9.3. It would not be surprising that we are going to consider three cases:

- (1) Let $X = Z$ be a generator of the subgroup K (4). Since this is a compact subgroup the corresponding eigenspaces $\tilde{Z}v_k = ikv_k$ are parametrised by an integer $k \in \mathbb{Z}$. The raising/lowering (ladder) operators L^\pm [42, § VI.2; 54, § 8.2] are defined by the following commutation relations:

$$(40) \quad [\tilde{Z}, L^\pm] = \lambda_\pm L^\pm.$$

In other words L^\pm are eigenvectors for operators $\text{ad } Z$ of adjoint representation of \mathfrak{sl}_2 [42, § VI.2]. From the commutators (40) we deduce that $L^\pm v_k$ are eigenvectors of \tilde{Z} as well:

$$\begin{aligned} \tilde{Z}(L^+ v_k) &= (L^+ \tilde{Z} + \lambda_+ L^+) v_k = L^+(\tilde{Z} v_k) + \lambda_+ L^+ v_k = ikL^+ v_k + \lambda_+ L^+ v_k \\ &= (ik + \lambda_+) L^+ v_k. \end{aligned}$$

Thus action of ladder operators on respective eigenspaces can be visualised by the diagram:

$$\dots \begin{array}{c} \xrightarrow{L^+} \\ \xleftarrow{L^-} \end{array} V_{ik-\lambda} \begin{array}{c} \xrightarrow{L^+} \\ \xleftarrow{L^-} \end{array} V_{ik} \begin{array}{c} \xrightarrow{L^+} \\ \xleftarrow{L^-} \end{array} V_{ik+\lambda} \begin{array}{c} \xrightarrow{L^+} \\ \xleftarrow{L^-} \end{array} \dots$$

Assuming $L^+ = a\tilde{A} + b\tilde{B} + c\tilde{Z}$ from the relations (39) and defining condition (40) we obtain linear equations with unknown a , b and c :

$$c = 0, \quad 2a = \lambda_+ b, \quad -2b = \lambda_+ a.$$

The equations have a solution if and only if $\lambda_+^2 + 4 = 0$, and the raising/lowering operators are $L^\pm = \pm i\tilde{A} + \tilde{B}$.

- (2) Consider the case $X = 2B$ of a generator of the subgroup A' (6). The subgroup is not compact and eigenvalues of the operator \tilde{B} can be arbitrary, however raising/lowering operators are still important [16, § II.1; 46, § 1.1]. We again seek a solution in the form $L_h^+ = a\tilde{A} + b\tilde{B} + c\tilde{Z}$ for the commutator $[2\tilde{B}, L_h^+] = \lambda L_h^+$. We will get the system:

$$4c = \lambda a, \quad b = 0, \quad a = \lambda c.$$

A solution exists if and only if $\lambda^2 = 4$. There are obvious values $\lambda = \pm 2$ with the ladder operators $L_h^\pm = \pm 2\tilde{A} + \tilde{Z}$, see [16, § II.1; 46, § 1.1]. Each indecomposable \mathfrak{sl}_2 -module is formed by a one-dimensional chain of eigenvalues with a transitive action of ladder operators.

Admitting double numbers we have an extra possibility to satisfy $\lambda^2 = 4$ with values $\lambda = \pm 2\epsilon$. Then there is an additional pair of hyperbolic ladder operators $L_\epsilon^\pm = \pm 2\epsilon\tilde{A} + \tilde{Z}$, which shift eigenvectors in the ‘‘orthogonal’’ direction to the standard operators L_h^\pm . Therefore an indecomposable \mathfrak{sl}_2 -module can be parametrised by a two-dimensional lattice of eigenvalues on the double number plane:

$$\begin{array}{ccccccc}
 & & \dots & & \dots & & \dots \\
 & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} \\
 \dots & \xrightleftharpoons[L_h^-]{L_h^+} & V_{(n-2)+\epsilon(k-2)} & \xrightleftharpoons[L_h^-]{L_h^+} & V_{n+\epsilon(k-2)} & \xrightleftharpoons[L_h^-]{L_h^+} & V_{(n+2)+\epsilon(k-2)} \xrightleftharpoons[L_h^-]{L_h^+} \dots \\
 & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} \\
 \dots & \xrightleftharpoons[L_h^-]{L_h^+} & V_{(n-2)+\epsilon k} & \xrightleftharpoons[L_h^-]{L_h^+} & V_{n+\epsilon k} & \xrightleftharpoons[L_h^-]{L_h^+} & V_{(n+2)+\epsilon k} \xrightleftharpoons[L_h^-]{L_h^+} \dots \\
 & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} \\
 \dots & \xrightleftharpoons[L_h^-]{L_h^+} & V_{(n-2)+\epsilon(k+2)} & \xrightleftharpoons[L_h^-]{L_h^+} & V_{n+\epsilon(k+2)} & \xrightleftharpoons[L_h^-]{L_h^+} & V_{(n+2)+\epsilon(k+2)} \xrightleftharpoons[L_h^-]{L_h^+} \dots \\
 & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} & & \begin{array}{c} \uparrow \\ L_\epsilon^- \\ \downarrow \\ L_\epsilon^+ \end{array} \\
 & & \dots & & \dots & & \dots
 \end{array}$$

- (3) Finally consider the case of a generator $X = -B + Z/2$ of the subgroup N' (7). According to the above procedure we get the equations:

$$b + 2c = \lambda a, \quad -a = \lambda b, \quad \frac{a}{2} = \lambda c,$$

which can be resolved if and only if $\lambda^2 = 0$. If we restrict ourselves with the only real (complex) root $\lambda = 0$, then the corresponding operators $L_p^\pm = -\tilde{B} + \tilde{Z}/2$ will not affect eigenvalues and thus are useless in the above context. However the dual number roots $\lambda = \pm\epsilon$ lead to the operators $L_\epsilon^\pm = \pm\epsilon\tilde{A} - \tilde{B} + \tilde{Z}/2$. These operators are suitable to build an \mathfrak{sl}_2 -modules with a one-dimensional chain of eigenvalues.

Remark 9.4. It is noteworthy that:

- the introduction of complex numbers is a necessity for the *existence* of ladder operators in the elliptic case;
- in the parabolic case we need dual numbers to make ladder operators *useful*;
- in the hyperbolic case double numbers are required for neither existence nor usability of ladder operators, but do provide an enhancement.

We summarise the above consideration with a focus on the Principle of similarity and correspondence:

Proposition 9.5. *Let a vector $X \in \mathfrak{sl}_2$ generates the subgroup K , N' or A' , that is $X = Z$, $B - Z/2$, or B respectively. Let ι be the respective hypercomplex unit.*

Then raising/lowering operators L^\pm satisfying to the commutation relation:

$$[X, L^\pm] = \pm\iota L^\pm, \quad [L^-, L^+] = 2\iota X.$$

are:

$$L^\pm = \pm\iota\tilde{A} + \tilde{Y}.$$

Here $Y \in \mathfrak{sl}_2$ is a linear combination of B and Z with the properties:

- $Y = [A, X]$.

- $X = [A, Y]$.
- Killings form $K(X, Y)$ [18, § 6.2] vanishes.

Any of the above properties defines the vector $Y \in \text{span}\{B, Z\}$ up to a real constant factor.

It is natural to expect that the usability of the Principle of similarity and correspondence will not be limited to the considered examples only. For example, similar types of ladder operators appeared in relation to the Heisenberg group as well [34].

10. OPEN PROBLEMS

We start from an illustration, that the invariant linear algebra presented in Section 7 is not unique.

Example 10.1 (Tropical form). Let us introduce the lexicographic order on \mathbb{R}^2 :

$$(u, v) \prec (u', v') \quad \text{if and only if} \quad \begin{cases} \text{either} & u < u'; \\ \text{or} & u = u', v < v'. \end{cases}$$

One can define functions \min and \max of a pair of points on \mathbb{R}^2 respectively. Then an addition of two vectors can be defined either as their minimum or maximum. A similar definition is used in *tropical mathematics*, also known as Maslov dequantisation or \mathbb{R}_{\min} and \mathbb{R}_{\max} algebras, see [45] for an energetic survey and [5] for a comprehensive coverage. It is easy to check that such an addition is distributive with respect to scalar multiplications (30)—(31) and consequently is invariant under parabolic rotations. This approach looks promising and definitely deserves a further careful investigation.

There are many other interesting questions to be investigated for the induced representations built in this paper. Here is a sketchy list of some of them:

- All complex-valued irreducible unitary representations of $SL_2(\mathbb{R})$ split into the three series: discrete, principal and complementary [16, 42]. It is useful to find relations of new hypercomplex representations to these series.
- A generic irreducible \mathfrak{sl}_2 -module [16, 46] may not be unitarisable for any complex-valued inner product. How many of those non-unitarisable moduli become unitary for some hypercomplex-valued product?
- Most of the above irreducible hypercomplex-unitary $SL_2(\mathbb{R})$ -moduli can be realised as spaces of functions either on the real line or the upper half-plane [22; 35, § 5]. The concept of covariant transform [32; 35, § 4] provides the universal tool for a construction of associated integral formulae and reproducing kernels.
- The described spaces of functions are null-solutions to certain differential operators, which possess $SL_2(\mathbb{R})$ -symmetry. In particular, the covariant transform with a mother wavelet, which is annihilated by a ladder operator (or any its power), creates the space of null-solutions to an associated differential equation, see [35, § 5.3]. This shall be useful in analysis of solutions of partial differential equations [40, 48].
- Analytical spaces of complex-valued functions work as models for functional calculus of normal operators, e.g. Dunford–Riesz calculus. The newly build $SL_2(\mathbb{R})$ -moduli support covariant functional calculi and associated spectra [20; 25; 35, § 6] for non-selfadjoint operators.

These directions can be viewed as parts of the Erlangen programme at large [27, 35]. The Principle of similarity and correspondence may help to find most harmonious constructions.

REFERENCES

- [1] Damir Z. Arov and Harry Dym. *J-contractive matrix valued functions and related topics*. Encyclopedia of Mathematics and its Applications, vol. 116. Cambridge University Press, Cambridge, 2008. ↑14, 16
- [2] Christian Bauer, Alexander Frink, Richard Kreckel, and Jens Vollinga. *GiNaC is Not a CAS*, 2001. URL: <http://www.ginac.de/>. ↑23, 25, 43
- [3] John L. Bell. *A primer of infinitesimal analysis*. Cambridge University Press, Cambridge, Second, 2008. ↑2
- [4] Dino Boccaletti, Francesco Catoni, Roberto Cannata, Vincenzo Catoni, Enrico Nichelatti, and Paolo Zampetti. *The mathematics of Minkowski space-time and an introduction to commutative hypercomplex numbers*. Birkhäuser Verlag, Basel, 2008. ↑2, 3, 1, 5, 6
- [5] Peter Butkovič. *Max-linear systems: theory and algorithms*. Springer Monographs in Mathematics. Springer-Verlag London Ltd., London, 2010. ↑11, 17
- [6] Francesco Catoni, Roberto Cannata, and Enrico Nichelatti. The parabolic analytic functions and the derivative of real functions. *Adv. Appl. Clifford Algebras* 14 (2):185–190, 2004. ↑2, 3, 1, 5
- [7] P. Cerejeiras, U. Kähler, and F. Sommen. Parabolic Dirac operators and the Navier-Stokes equations over time-varying domains. *Math. Methods Appl. Sci.* 28 (14):1715–1724, 2005. ↑3, 2
- [8] Martin Davis. *Applied nonstandard analysis*. Wiley-Interscience [John Wiley & Sons], New York, 1977. ↑2
- [9] G.I. Garas'ko. Nachala finslerovoi geometrii dlya fizikov. [Elements of Finsler geometry for physicists]. TETRU, Moscow, 2009. 268 pp. URL: <http://hypercomplex.xpsweb.com/articles/487/ru/pdf/00-gbook.pdf>. ↑2
- [10] GNU. *General Public License (GPL)*. Free Software Foundation, Inc., Boston, USA, version 3, 2007. URL: <http://www.gnu.org/licenses/gpl.html>. ↑25, 43
- [11] N. A. Gromov. Kontraktsii i analiticheskie prodolzheniya klassicheskikh grupp. Edinyi podkhod. (Russian) [Contractions and analytic extensions of classical groups. Unified approach]. Akad. Nauk SSSR Ural. Otdel. Komi Nauchn. Tsentr, Syktyvkar, 1990. ↑5
- [12] N. A. Gromov and V. V. Kuratov. Noncommutative space-time models. *Czechoslovak J. Phys.* 55 (11):1421–1426, 2005. ↑2, 6
- [13] N. A. Gromov and V. V. Kuratov. Possible quantum kinematics. *J. Math. Phys.* 47 (1):013502, 9, 2006. ↑2, 6
- [14] Francisco J. Herranz, Ramón Ortega, and Mariano Santander. Trigonometry of spacetimes: a new self-dual approach to a curvature/signature (in)dependent trigonometry. *J. Phys. A* 33 (24):4525–4551, 2000. E-print: [arXiv:math-ph/9910041](https://arxiv.org/abs/math-ph/9910041). ↑7, 9, 8
- [15] Francisco J. Herranz and Mariano Santander. Conformal compactification of spacetimes. *J. Phys. A* 35 (31):6619–6629, 2002. E-print: [arXiv:math-ph/0110019](https://arxiv.org/abs/math-ph/0110019). ↑2, 5, 1, 6
- [16] Roger Howe and Eng-Chye Tan. *Nonabelian harmonic analysis. Applications of $SL(2, \mathbf{R})$* . Springer-Verlag, New York, 1992. ↑4, 15, 17, 3, 19
- [17] Andrei Khrennikov and Gavriel Segre. Hyperbolic quantization. In L. Accardi, W. Freudenberg, and M. Schürman (eds.) *Quantum probability and infinite dimensional analysis*, pages 282–287, World Scientific Publishing, Hackensack, NJ, 2007. ↑2, 3, 1, 5, 6
- [18] A. A. Kirillov. *Elements of the theory of representations*. Springer-Verlag, Berlin, 1976. Translated from the Russian by Edwin Hewitt, Grundlehren der Mathematischen Wissenschaften, Band 220. ↑5, 6, 14, 17, 4, 15, 18
- [19] Anastasia V. Kisil. Isometric action of $SL_2(\mathbb{R})$ on homogeneous spaces. *Adv. App. Clifford Algebras* 20 (2):299–312, 2010. E-print: [arXiv:0810.0368](https://arxiv.org/abs/0810.0368). MR2012b:32019. ↑5
- [20] Vladimir V. Kisil. Möbius transformations and monogenic functional calculus. *Electron. Res. Announc. Amer. Math. Soc.* 2 (1):26–33, 1996. On-line. ↑17, 19
- [21] Vladimir V. Kisil. How many essentially different function theories exist?. *Clifford algebras and their application in mathematical physics (Aachen, 1996)*, pages 175–184, Kluwer Academic Publishers, Dordrecht, 1998. ↑3
- [22] Vladimir V. Kisil. Analysis in $\mathbf{R}^{1,1}$ or the principal function theory. *Complex Variables Theory Appl.* 40 (2):93–118, 1999. E-print: [arXiv:funct-an/9712003](https://arxiv.org/abs/funct-an/9712003). ↑3, 5, 6, 13, 17, 2, 14, 19
- [23] Vladimir V. Kisil. Two approaches to non-commutative geometry. *Complex methods for partial differential equations (Ankara, 1998)*, pages 215–244, Kluwer Acad. Publ., Dordrecht, 1999. E-print: [arXiv:funct-an/9703001](https://arxiv.org/abs/funct-an/9703001). ↑3
- [24] Vladimir V. Kisil. Meeting Descartes and Klein somewhere in a noncommutative space. In A. Fokas, J. Halliwell, T. Kibble, and B. Zegarlinski (eds.) *Highlights of mathematical physics (London, 2000)*, pages 165–189, Amer. Math. Soc., Providence, RI, 2002. E-print: [arXiv:math-ph/0112059](https://arxiv.org/abs/math-ph/0112059). ↑3

- [25] Vladimir V. Kisil. Spectrum as the support of functional calculus. *Functional analysis and its applications*, pages 133–141, Elsevier, Amsterdam, 2004. E-print: [arXiv:math.FA/0208249](#). ↑13, 17, 14, 19
- [26] Vladimir V. Kisil. An example of Clifford algebras calculations with GiNaC. *Adv. Appl. Clifford Algebr.* 15 (2):239–269, 2005. E-print: [arXiv:cs.MS/0410044](#), [On-line](#). ↑43
- [27] Vladimir V. Kisil. Erlangen program at large=0: Starting with the group $SL_2(\mathbf{R})$. *Notices Amer. Math. Soc.* 54 (11):1458–1465, 2007. E-print: [arXiv:math/0607387](#), [On-line](#). ↑5, 17, 6, 19
- [28] Vladimir V. Kisil. Fillmore-Springer-Cnops construction implemented in GiNaC. *Adv. Appl. Clifford Algebr.* 17 (1):59–70, 2007. Updated full text and source files: E-print: [arXiv:cs.MS/0512073](#), [On-line](#). ↑23, 25, 43, 51
- [29] Vladimir V. Kisil. Two-dimensional conformal models of space-time and their compactification. *J. Math. Phys.* 48 (7):073506, 8, 2007. E-print: [arXiv:math-ph/0611053](#). ↑5, 9, 6
- [30] Vladimir V. Kisil. Erlangen program at large=2: Inventing a wheel. The parabolic one. *Trans. Inst. Math. of the NAS of Ukraine*, pages 89–98, 2010. E-print: [arXiv:0707.4024](#). ↑5, 10, 11, 25, 30, 44
- [31] Vladimir V. Kisil. Erlangen program at large=1: Geometry of invariants. *SIGMA, Symmetry Integrability Geom. Methods Appl.* 6 (076):45, 2010. E-print: [arXiv:math.CV/0512416](#). MR2011i:30044. ↑5, 9, 12, 6, 8, 10, 13
- [32] Vladimir V. Kisil. Covariant transform. *Journal of Physics: Conference Series* 284 (1):012038, 2011. E-print: [arXiv:1011.3947](#). ↑17, 19
- [33] Vladimir V. Kisil. Erlangen program at large=2 1/2: Induced representations and hypercomplex numbers. *Izvestiya Komi nauchnogo centra UrO RAN [Izvestiya Komi nauchnogo centra UrO RAN]* 1 (5):4–10, 2011. E-print: [arXiv:0909.4464](#). ↑11, 23, 25, 43
- [34] Vladimir V. Kisil. Erlangen Programme at Large 3.2: Ladder operators in hypercomplex mechanics. *Acta Polytechnica* 51 (4):44–53, 2011. E-print: [arXiv:1103.1120](#). ↑7, 17, 8, 19
- [35] Vladimir V. Kisil. Erlangen programme at large: an Overview. In S.V. Rogosin and A.A. Koroleva (eds.) *Advances in applied analysis*, pages 1–94, Birkhäuser Verlag, Basel, 2012. E-print: [arXiv:1106.1686](#). ↑17, 19
- [36] Vladimir V. Kisil. *Geometry of Möbius transformations: Elliptic, parabolic and hyperbolic actions of $SL_2(\mathbf{R})$* . Imperial College Press, London, 2012. Includes a live DVD. ↑5, 9, 12
- [37] Vladimir V. Kisil. Hypercomplex representations of the Heisenberg group and mechanics. *Internat. J. Theoret. Phys.* 51 (3):964–984, 2012. E-print: [arXiv:1005.5057](#). ↑2, 7, 6, 8
- [38] Vladimir V. Kisil. Is commutativity of observables the main feature, which separate classical mechanics from quantum?. *Izvestiya Komi nauchnogo centra UrO RAN [Izvestiya Komi nauchnogo centra UrO RAN]* 3 (11):4–9, 2012. E-print: [arXiv:1204.1858](#). ↑2
- [39] János Kollár and Shigefumi Mori. *Birational geometry of algebraic varieties*. Cambridge Tracts in Mathematics, vol. 134. Cambridge University Press, Cambridge, 2008. With the collaboration of C. H. Clemens and A. Corti. Paperback reprint of the hardback edition 1998. Zbl1143.14014. ↑11, 13
- [40] Nadiia Kononenko. Projective structures and algebras of their differential invariants. *Acta Applicandae Mathematicae* 109 (1):87–99, 2010. ↑17, 19
- [41] Vladislav V. Kravchenko. *Applied pseudoanalytic function theory*. Frontiers in Mathematics. Birkhäuser Verlag, Basel, 2009. With a foreword by Wolfgang Sproessig. ↑3, 2
- [42] Serge Lang. $SL_2(\mathbf{R})$. Graduate Texts in Mathematics, vol. 105. Springer-Verlag, New York, 1985. Reprint of the 1975 edition. ↑4, 13, 15, 17, 3, 14, 16, 19
- [43] M. A. Lavrent'ev and B. V. Shabat. Problemy gidrodinamiki i ikh matematicheskie modeli. [Problems of hydrodynamics and their mathematical models]. Izdat. “Nauka”, Moscow, Second, 1977. ↑3, 7, 2, 8
- [44] M. Libine. Hyperbolic Cauchy integral formula for the split complex numbers. *ArXiv e-prints*, December 2007. ↑3, 2
- [45] G. L. Litvinov. The Maslov dequantization, and idempotent and tropical mathematics: a brief introduction. *Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI)* 326 (Teor. Predst. Din. Sist. Komb. i Algoritm. Metody. 13):145–182, 282, 2005. E-print: [arXiv:math/0507014](#). ↑11, 17, 12
- [46] Volodymyr Mazorchuk. *Lectures on \mathfrak{sl}_2 -modules*. World Scientific, 2009. ↑15, 17, 19
- [47] A. E. Motter and M. A. F. Rosa. Hyperbolic calculus. *Adv. Appl. Clifford Algebras* 8 (1):109–128, 1998. ↑3, 2
- [48] Valery N. Pilipchuk. *Nonlinear dynamics. Between linear and impact limits*. Lecture Notes in Applied and Computational Mechanics, vol. 52. Springer, Berlin, 2010. ↑2, 3, 17, 1, 19
- [49] Valery N. Pilipchuk. Non-smooth spatio-temporal coordinates in nonlinear dynamics, January 2011. E-print: [arXiv:1101.4597](#). ↑2
- [50] R.I. Pimenov. Unified axiomatics of spaces with maximal movement group. *Litov. Mat. Sb.* 5:457–486, 1965. Zbl0139.37806. ↑2

- [51] L. S. Pontryagin. Obobshcheniya chisel [Generalisations of numbers]. Bibliotekha “Kvant” [Library “Kvant”], vol. 54. “Nauka”, Moscow, 1986. ↑5
- [52] Norman Ramsey. *Noweb — a simple, extensible tool for literate programming*. URL: <http://www.eecs.harvard.edu/~nr/noweb/>. ↑23, 25, 43
- [53] Garret Sobczyk. The hyperbolic number plane. *College Math Journal* 26 (4):268–280, 1995. ↑3, 2
- [54] Michael E. Taylor. *Noncommutative harmonic analysis*. Mathematical Surveys and Monographs, vol. 22. American Mathematical Society, Providence, RI, 1986. ↑15, 16, 17
- [55] S. Ulrych. Relativistic quantum physics with hyperbolic numbers. *Phys. Lett. B* 625 (3–4):313–323, 2005. ↑2, 1, 5, 6
- [56] S. Ulrych. Considerations on the hyperbolic complex Klein-Gordon equation. *J. Math. Phys.* 51 (6):063510, 8, 2010. ↑2
- [57] V. A. Uspenskii. CHto takoe nestandartnyi0 analiz? [What is non-standard analysis?]. “Nauka”, Moscow, 1987. With an appendix by V. G. Kanovei. ↑2
- [58] J. C. Vignaux and A. Durañona y Vedia. Sobre la teoría de las funciones de una variable compleja hiperbólica [On the theory of functions of a complex hyperbolic variable]. *Univ. nac. La Plata. Publ. Fac. Ci. fis. mat.* 104:139–183, 1935. Zbl62.1122.03. ↑2, 3
- [59] I. M. Yaglom. *A simple non-Euclidean geometry and its physical basis*. Heidelberg Science Library. Springer-Verlag, New York, 1979. Translated from the Russian by Abe Shenitzer, with the editorial assistance of Basil Gordon. ↑2, 7, 8, 9, 10, 1, 5, 6, 11

Индукцированных представлений группы $SL_2(\mathbb{R})$ и гиперкомплексные числа

В. В. Кисиль

Аннотация. В работе рассматривается конструкция индуцированных представлений для группы $G = SL_2(\mathbb{R})$. Оказывается, что действие этой группы на однородном пространстве G/H , где H является произвольной однопараметрической подгруппой $SL_2(\mathbb{R})$, является дробно-линейным преобразованием двумерной алгебры гиперкомплексных чисел. Это наблюдение может быть распространено на дальнейшие соответствия между структурными компонентами $SL_2(\mathbb{R})$ и гиперкомплексными системами. Соответственно мы рассматриваем вопрос о гиперкомплексных характеристиках подгруппы H . В частности мы приводим примеры индуцированных представлений группы $SL_2(\mathbb{R})$ в пространствах функций с гиперкомплексными значениями, которые являются унитарными в определённом смысле.

Ключевые слова: индуцированные представления, унитарные представления, полупростые группы Ли, комплексные числа, дуальные числа, двойные числа, преобразования Мёбиуса, дробно-линейные преобразования, операторы создания/уничтожения.

1. ВВЕДЕНИЕ

Теория аналитических функций комплексного переменного—красивая теория с многочисленными применениями в различных областях от теории чисел до электродинамики. Естественно, происходит поиск различных обобщений и аналогов такой успешной теории. Наиболее простая (или основная?) попытка заключается в замене мнимой единицы $i^2 = -1$ или на гиперболическую $\epsilon^2 = +1$ или нильпотентную $\epsilon^2 = 0$.

Двухмерная коммутативная ассоциативная алгебра порожденная 1 и ϵ состоит из выражений вида $x + \epsilon y$ с действительными параметрами x и y . Как правило в русскоязычной литературе, они называются двойными числами [59, Прил. Б], хотя из-за влияния англоязычной литературы встречаются и названия вроде **паракомплексные числа**, расщепляемые комплексные числа, комплексные числа гиперболического типа [4, 17, 48, 55]. Эта алгебра содержит делители нуля $\epsilon_{\pm} = \frac{1}{2}(1 \pm \epsilon)$ и выполняется тождества $\epsilon_{\pm}^2 = \epsilon_{\pm}$ и $\epsilon_+ \epsilon_- = 0$. Поэтому, алгебраически двойные числа изоморфны $\mathbb{R} \oplus \mathbb{R}$ —прямой сумме двух наборов действительных чисел натянутых на ϵ_+ и ϵ_- . Отсюда и происходит название «двойные» или «расщеплённые».

Сходная алгебра порождённая 1 и ϵ состоит из элементов $x + \epsilon y$, которые называются **дуальными числами** [6, 15, 59]. Среди них так же есть делители

нуля вида ϵy , $y \in \mathbb{R}$. Физические приложения дуальных чисел обнаруживаются в классической механике [38, 59], нелинейной динамике [48, 49], теории относительности [4, 9, 55], космологии [12, 15] и квантовой механике [17, 37, 56].

Замечание 1.1. К сожалению, не существует устоявшихся обозначений для образующих гиперкомплексных алгебр. Более того, было бы затруднительно просто перечислить всё многообразие использованных символов и мы упомянем лишь некоторые из них. Стоит отметить, что даже наиболее традиционная мнимая единица i обозначается как j во многих инженерных текстах. Гиперболическая единица обозначается как ϵ во многочисленных работах начиная, по крайней мере, с основополагающего текста [58]; однако, другое обозначение e употребляется в замечательной книге [59]. Для нильпотентной единицы часто используется символ ι [13, 50], но наш выбор ϵ позаимствован у Яглома [59] и его последователей. Нам это обозначение кажется удачным в силу следующего Замечания.

Замечание 1.2. Нильпотентная единица ϵ близко связана с инфинитезимальным числом ϵ используемом в нестандартном анализе [8, 57]. Квадрат нильпотентно единицы в точности равен нулю в то время, как квадрат инфинитезимального числа есть «практически» ноль. Это сходство использовалось в работе [6] для вывода основных теорем дифференциального исчисления на основе нильпотентной единицы. Более того, существует изложение нестандартного анализа основанное на нильпотентной единице [3]³. То же самое свойство нильпотентности стоит за построением классической механики из представлений группы Гейзенберга [37, 38].

Вполне естественен вопрос: какие «аналитические» функции связаны с двойными и дуальными числами? Не удивительно, что различные ответы на этот вопрос уже длительное время предлагаются различными исследователями. Так сложилось: что многие из них не были осведомлены о работах предшественников. Не претендуя на полноту, мы укажем известные автору работы, что бы повысить связанность в этой области.

Для двойных чисел систематическое исследование было предпринято в работе [58], среди последующих исследований и обзоров укажем [4; 17; 22; 41, Part IV; 43; 44; 47; 48; 53], а так же дальнейшие ссылки в этих работах. Доминирующий подход основан на факторизации волнового оператора $\partial_x^2 - \partial_y^2 = (\partial_x - \epsilon \partial_y)(\partial_x + \epsilon \partial_y)$ в произведение двух линейных операторов. Это аналогично разложению оператора Лапласа $\partial_x^2 + \partial_y^2 = (\partial_x - i \partial_y)(\partial_x + i \partial_y)$ в произведение оператора Коши–Римана и его сопряжённого. Соответственно, гиперболические аналитические функции определяются как нулевые решения оператора $\partial_x + \epsilon \partial_y$. Однако, разложение двойных чисел в базисе ϵ_{\pm} представляет любую «аналитическую» функцию $f(x, y) = \epsilon_+ f_+(x + y) - \epsilon_- f_-(x - y)$ в виде суммы двух произвольных функций f_{\pm} действительной переменной. Это, естественно, соответствует представлению общего решения волнового уравнения в виде суммы двух волн приближающихся справа и слева.

Для нильпотентной единицы ϵ ситуация ещё более тривиальна. Факторизация, сходная с предыдущими⁴ вырождается $(\partial_x - \epsilon \partial_y)(\partial_x + \epsilon \partial_y) = \partial_x^2$. Попытка определить аналитические функции через разложения в степенной ряд переменной $x + \epsilon y$ приводит к выражению $f(x) + \epsilon y f'(x)$ порождённой произвольной дифференцируемой функцией $f(x)$ действительной переменной [6].

³Я благодарен анонимному рецензенту за указание этой работы.

⁴Стоит, однако, отметить нетривиальную факторизацию волнового оператора $\partial_x^2 + \partial_y^2$ использующую высоко-размерные алгебры Клиффорда [7].

В наших предыдущих работах [21–24] мы предложили строить теорию аналитических функций в духе Эрлангенской программы Феликса Клейна, то есть исходя из некоторого действия группы. В данной работе мы продвигаемся в этом направлении. Заметим, что классические пространства комплексно-аналитических функций—пространства Харди и Бергмана—являются неприводимыми модулями некоторых унитарных представлений группы $SL_2(\mathbb{R})$. Далее, эти представления индуцированы комплекснозначным характером компактной подгруппы $K \in SL_2(\mathbb{R})$. Поэтому, мы надеемся найти другие типы пространств аналитических функций среди всех неприводимых (или примарных) $SL_2(\mathbb{R})$ -модулей для индуцированных представлений.

Композиция работы такова. Мы напоминаем структуру группы $SL_2(\mathbb{R})$ и перечисляем все её неизоморфные однопараметрические непрерывные подгруппы в разделе 2. Этот результат позволяет описать три неизоморфных типа $SL_2(\mathbb{R})$ -однородных пространств в разделе 3. Как оказалось, три типа гиперкомплексных чисел наилучшим образом описывают действие $SL_2(\mathbb{R})$ на этих однородных посредством дробно-линейных преобразований. Что бы индуцировать представления $SL_2(\mathbb{R})$ с одномерных подгрупп мы описываем их характеры в терминах алгебр гиперкомплексных чисел в разделе 4. В следующем разделе 5 мы описываем объекты, которые могут трактоваться как геометрические характеры однопараметрических подгрупп. Что бы индуцированное представление было линейным, индуцированный характер тоже должен быть линейным. Однако, линейность геометрических характеров требует дополнительных определений, которые обсуждаются в разделах 6 и 7. Далее, мы выписываем некоторые примеры индуцированных представлений в разделе 8. Удобное описание неприводимого $SL_2(\mathbb{R})$ -модуля достигается в линейном базисе состоящем из собственных векторов подгруппы индуцировавшей представление. Алгебра Ли \mathfrak{sl}_2 группы $SL_2(\mathbb{R})$ действует транзитивно на этом базисе с помощью операторов повышения/понижения (создания/уничтожения), которые широко известны в квантовой механике. Мы описываем соответствующую структуру в разделе 9. В заключении мы даем некоторые возможные направления дальнейших исследований.

2. ГРУППА $SL_2(\mathbb{R})$ И ЕЁ ПОДГРУППЫ

Группа $SL_2(\mathbb{R})$ состоит из квадратных 2×2 матриц с действительными элементами и единичным определителем. Групповая операция задаётся матричным умножением. $SL_2(\mathbb{R})$ является простейшей полупростой группой Ли. Произвольный элемент $SL_2(\mathbb{R})$ допускает единственное разложение вида [16, Exer. I.14]:

$$(1) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{pmatrix} \begin{pmatrix} 1 & \nu \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix},$$

с некоторыми параметрами $\alpha \in (0, \infty)$, $\nu \in (-\infty, \infty)$ и $\phi \in (-\pi, \pi]$. Эта формула (1) записанная в виде $SL_2(\mathbb{R}) = ANK$ известна как разложение Ивасавы [42, § III.1] и допускает обобщение на произвольную полупростую группу Ли.

Семейства матриц каждого из трёх видов в правой части (1) образуют однопараметрические подгруппы обычно обозначаемые A , N and K . Они получают

экспоненцированием соответствующих матриц с нулевым следом:

$$(2) \quad A = \left\{ \begin{pmatrix} e^t & 0 \\ 0 & e^{-t} \end{pmatrix} = \exp \begin{pmatrix} t & 0 \\ 0 & -t \end{pmatrix}, t \in \mathbb{R} \right\},$$

$$(3) \quad N = \left\{ \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} = \exp \begin{pmatrix} 0 & t \\ 0 & 0 \end{pmatrix}, t \in \mathbb{R} \right\},$$

$$(4) \quad K = \left\{ \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} = \exp \begin{pmatrix} 0 & t \\ -t & 0 \end{pmatrix}, t \in (-\pi, \pi] \right\}.$$

Следующий простой результат имеет поучительное доказательство:

Предложение 2.1. *Любая непрерывная однопараметрическая подгруппа $SL_2(\mathbb{R})$ матрично сопряжена к одной из подгрупп A , N или K .*

Доказательство. Любая непрерывная однопараметрическая подгруппа получается экспоненцированием

$$(5) \quad e^{tX} = \sum_{n=0}^{\infty} \frac{t^n}{n!} X^n$$

элемента X алгебры Ли \mathfrak{sl}_2 группы $SL_2(\mathbb{R})$. Такой элемент представим 2×2 матрицей с нулевым следом. Поведение ряда Тэйлора (5) зависит от свойств степеней X^n . Последние легко классифицируются непосредственным вычислением:

Лемма 2.2. *Квадрат X^2 матрицы $X = \begin{pmatrix} a & b \\ c & -a \end{pmatrix}$ с нулевым следом есть единичная матрица умноженная на $a^2 + bc = -\det X$. Этот множитель может быть отрицательным, нулевым или положительным, что соответствует трём различным типам ряда Тэйлора (5) для e^{tX} .*

Несложно видеть, что матричное подобие приводит элемент X к генератору

- подгруппы K если $(-\det X) < 0$;
- подгруппы N если $(-\det X) = 0$;
- подгруппы A если $(-\det X) > 0$.

Так как матричное подобие не меняет определителя матрицы, то эти три случая не сводятся друг к другу. \square

Пример 2.3. Следующие две подгруппы сопряжены к A и N соответственно посредством матричного сопряжения с фиксированной матрицей $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$:

$$(6) \quad A' = \left\{ \begin{pmatrix} \cosh t & \sinh t \\ \sinh t & \cosh t \end{pmatrix} = \exp \begin{pmatrix} 0 & t \\ t & 0 \end{pmatrix}, t \in \mathbb{R} \right\},$$

$$(7) \quad N' = \left\{ \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} = \exp \begin{pmatrix} 0 & 0 \\ t & 0 \end{pmatrix}, t \in \mathbb{R} \right\}.$$

В дальнейшем мы будем, как правило, рассматривать подгруппы N' и A' вместо N и A , так как это приведёт к более естественной геометрической картине.

3. ДЕЙСТВИЕ $SL_2(\mathbb{R})$ ПРИВОДИТ К ГИПЕРКОМПЛЕКСНЫМ ЧИСЛАМ

Пусть N является подгруппой группы G . Для соответствующего однородного пространства $\Omega = G/N$ определим гладкое сечение $s : \Omega \rightarrow G$ [18, § 13.2], которое является левым обратным для естественной проекции $p : G \rightarrow \Omega$. Выбор такого сечения непринципиален в том смысле, что наши дальнейшие построения определяются с точностью до гладкого отображения $\Omega \rightarrow \Omega$.

Каждый элемент $g \in G$ может быть представлен единственным образом в виде $g = s(\omega)h$, где $\omega = p(g) \in \Omega$ и $h \in H$. Соответственно, на Ω определяется действие группы G следующим образом:

$$(8) \quad g : \omega \mapsto g \cdot \omega = p(g * s(\omega)),$$

где $*$ обозначает групповое произведение. Это действие может быть также проиллюстрировано коммутативной диаграммой:

$$\begin{array}{ccc} G & \xrightarrow{g^*} & G \\ s \updownarrow p & & s \updownarrow p \\ \Omega & \xrightarrow{g \cdot} & \Omega \end{array}$$

Для группы $G = SL_2(\mathbb{R})$, как и для остальных полупростых групп, принято рассматривать случай, когда H является максимальной компактной подгруппой K . Однако в данной работе мы рассмотрим все три возможности для однопараметрической подгруппы H описанных в Предложении 2.1 и Примере 2.3, а именно $H = K$, N' или A' . В этом случае многообразие Ω является двумерным и для всех трёх возможных подгрупп H мы определим [22, Ex. 3.7(a)]:

$$(9) \quad s : (u, v) \mapsto \frac{1}{\sqrt{v}} \begin{pmatrix} v & u \\ 0 & 1 \end{pmatrix}, \quad (u, v) \in \mathbb{R}^2, v > 0.$$

Непосредственное вычисление или использование компьютерной программы [30] приводит к следующему результату:

Предложение 3.1. Действие (8) группы $SL_2(\mathbb{R})$ определено сечением s (9) есть:

$$(10) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} : (u, v) \mapsto \left(\frac{(au + b)(cu + d) - \sigma cv^2}{(cu + d)^2 - \sigma (cv)^2}, \frac{v}{(cu + d)^2 - \sigma (cv)^2} \right),$$

где $\sigma = -1, 0$ или 1 для подгрупп K, N' или A' соответственно.

Выражение (10) не выглядит привлекательно, однако, введение гиперкомплексных чисел существенно его улучшает:

Предложение 3.2. Пусть единица ι такова, что $\iota^2 = \sigma$, тогда действие группы $SL_2(\mathbb{R})$ (10) может быть записано как дробно-линейное (мёбиусово) преобразование:

$$(11) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} : w \mapsto \frac{aw + b}{cw + d}, \quad \text{где } w = u + \iota v,$$

для всех трёх возможных значений параметра σ из Предложения 3.1.

Замечание 3.3. Отметим что гиперкомплексные единицы не вводились нами умышленно, по произволу или из «чисто обобщательской попытки» [51, стр. 4]. Они естественно возникли из действия группы $SL_2(\mathbb{R})$.

Мнимая единица $i^2 = -1$ порождает хорошо знакомые комплексные числа. Двумерная алгебра выражений $x + \epsilon y$ с образующей $\epsilon^2 = 1$ имеет много имён: гиперболические, двойные числа [4, 17, 55] и др., это один из простейших примеров гиперкомплексных чисел. Параболические гиперкомплексные числа, часто так же называемые двойными имеют вид $x + \epsilon y$ с нильпотентной единицей такой, что $\epsilon^2 = 0$ [6, 11, 15, 59]. Мы используем символ ι для обозначения любой из трёх единиц i, ϵ или ϵ .

Действие (11) является групповым гомоморфизмом из $SL_2(\mathbb{R})$ в преобразование «верхней полуплоскости» гиперкомплексной алгебры. Несмотря на то,

что алгебраическая структура двойных и дуальных чисел достаточно вырождена, они интересны как однородные пространства для мёбиусовых преобразований. Построение соответствующих геометрий в духе **Эрлангенской программы** Ф. Клейна ободряюще необычно [31] и приводит к новым результатам даже в хорошо изученном случае геометрии Лобачевского [27].

Из-за делителей нуля корректное изучение мёбиусовых преобразований (11) должно рассматриваться на конформно пополненной плоскости [15, 29]. Физические приложения гиперкомплексных чисел простираются от классической механики [59] и специальной теории относительности [4, 55] до космологии [12, 15] и квантовой механики [13, 17, 37].

Стандартный способ линеаризации действия (8) заключается в переходе к **представлению индуцированному с подгруппы** H [18, § 13.2; 22, § 3.1]. Для этого мы определим отображение $r : G \rightarrow H$ порожденное отображениями p и s через соотношения:

$$(12) \quad r(g) = (s(\omega))^{-1}g, \quad \text{где } \omega = p(g) \in \Omega.$$

Пусть χ является неприводимым представлением подгруппы H в векторном пространстве V , тогда оно индуцирует представление G в смысле Макки [18, § 13.2]. Существует несколько реализаций индуцированного представления, мы будем рассматривать представление ρ_χ в пространстве V -значных функций по формуле [18, § 13.2.(7)–(9)]:

$$(13) \quad [\rho_\chi(g)f](\omega) = \chi(r(g^{-1} * s(\omega)))f(g^{-1} \cdot \omega),$$

где $g \in G$, $\omega \in \Omega$, $h \in H$ и $r : G \rightarrow H$, $s : \Omega \rightarrow G$ есть отображения определённые выше; $*$ обозначает групповую операцию на G и \cdot является действием (8) группы G на Ω .

Так как в нашем рассмотрении подгруппа H всегда одномерна, то её неприводимые представления обычно предполагаются комплексными характерами. Однако, если гиперкомплексные числа естественно появляются в действии группы $SL_2(\mathbb{R})$ на однородных пространствах (11), почему мы не должны также рассмотреть гиперкомплексные характеры?

4. ГИПЕРКОМПЛЕКСНЫЕ ХАРАКТЕРЫ—АЛГЕБРАИЧЕСКИЙ ПОДХОД

Как уже отмечалось, традиционно основное внимание при изучении представлений группы $SL_2(\mathbb{R})$ уделяется случаю $H = K$ и, соответственно, комплекснозначным характерам K . Линейное преобразование определённое матрицами (4) из K представляют собой вращения \mathbb{R}^2 на угол t . Идентификация $\mathbb{R}^2 = \mathbb{C}$ переводит это вращение в умножение на e^{it} , где $i^2 = -1$. Вращение является унитарным для эллиптической метрики:

$$(14) \quad x^2 + y^2 = (x + iy)(x - iy).$$

Каждая орбита вращения являются окружностью («ободом колеса»), и любая прямая проходящая через начало координат («спица») переходит в другую прямую повернутую на угол t , см. Рис. 3(E).

Для гиперкомплексных чисел наиболее прямолинейное применение этой модели для характеров выглядит так.

Предложение 4.1. *Нижеследующая таблица показывает алгебраическое соответствие между гиперкомплексными характерами:*

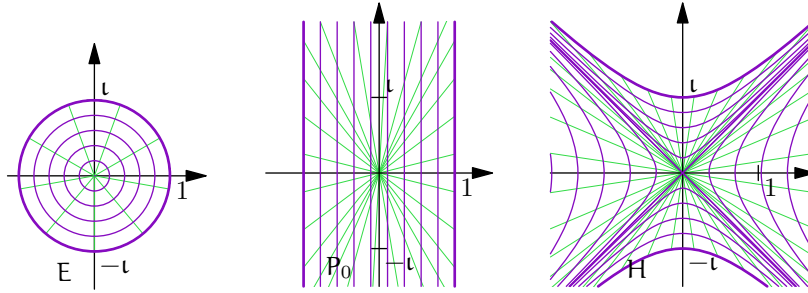


Рис. 3. Унитарные вращения, полученные алгебраическим способом, т.е. умножением на $e^{t\epsilon}$: эллиптические (E), параболические (P_0) и гиперболические (H). Все фиолетовые орбиты определяются уравнением $x^2 - \epsilon^2 y^2 = r^2$. Зелёные «спицы» (прямые линии из начала координат к точке на орбите) получаются вращением горизонтальной оси.

<i>Эллиптический</i>	<i>Параболический</i>	<i>Гиперболический</i>
<i>Subgroup K</i>	<i>Subgroup N'</i>	<i>Subgroup A'</i>
$i^2 = -1$	$\epsilon^2 = 0$	$\epsilon^2 = 1$
$w = x + iy$	$w = x + \epsilon y$	$w = x + \epsilon y$
$\bar{w} = x - iy$	$\bar{w} = x - \epsilon y$	$\bar{w} = x - \epsilon y$
$e^{it} = \cos t + i \sin t$	$e^{\epsilon t} = 1 + \epsilon t$	$e^{\epsilon t} = \cosh t + \epsilon \sinh t$
$ w _e^2 = w\bar{w} = x^2 + y^2$	$ w _p^2 = w\bar{w} = x^2$	$ w _h^2 = w\bar{w} = x^2 - y^2$
$\arg w = \arctan \frac{y}{x}$	$\arg w = \frac{y}{x}$	$\arg w = \tanh^{-1} \frac{y}{x}$
<i>окружность</i> $ w _e^2 = 1$	<i>две прямые</i> $x = \pm 1$	<i>гипербола</i> $ w _h^2 = 1$

Геометрическое действие умножения на $e^{t\epsilon}$ показано Рис. 3 для всех трёх случаев.

Явным образом параболическое «вращение» порождённое умножением на $e^{\epsilon t}$ действует на дуальные числа так:

$$(15) \quad e^{\epsilon x} : a + \epsilon b \mapsto a + \epsilon(ax + b).$$

Это связывает параболический случай с галилеевыми симметриями классической механики [59], в которой время абсолютно и независимо от пространства. Такое кинематическое обоснование предлагает следующие соответствия для параболического случая [14, 59]:

- Параболическая тригонометрия тривиальна:

$$(16) \quad \text{cosp } t = \pm 1, \quad \text{sinp } t = t;$$

- параболическое расстояние (при ненулевом x) не зависит от y :

$$(17) \quad x^2 = (x + \epsilon y)(x - \epsilon y);$$

- полярное разложение дуального числа [59, App. C(30')]:

$$(18) \quad u + \epsilon v = u(1 + \epsilon \frac{v}{u}), \quad \text{поэтому} \quad |u + \epsilon v| = u, \quad \arg(u + \epsilon v) = \frac{v}{u};$$

- параболическое «колесо» выглядит прямоугольным, см. Рис. 3(P_0).

Алгебраическая аналогия и кинематическое обоснование кажутся достаточно убедительными для единственно правильной параболической тригонометрии [14, 43, 59]. Более того, данные три типа вращений представляют собой все возможные линейные симплектические преобразования, что делает их и соответствующие мнимые единицы полезными в классической и квантовой механике [34, 37].

Однако геометрический подход и симметрии параболических уравнений математической физики предлагают менее вырожденную альтернативу изложенную ниже.

5. ПАРАБОЛИЧЕСКОЕ КОЛЕСО ИЗ ГЕОМЕТРИЧЕСКОГО ДЕЙСТВИЯ

Попробуем ввести нетривиальные параболические вращения другим способом. Если умножение в дуальных числах (линейные преобразования) слишком тривиальны, мы перейдём к следующему уровню сложности: дробно-линейным. linear-fractional.

Действительно, нет необходимости рассматривать гиперкомплексных единицы как абстрактные сущности. Мы видели в Лемме 2.2, что генераторы подгрупп K , N и A по сути представляют единицы для комплексных, дуальных и двойных чисел соответственно. Мы так же можем рассматривать выражения для однопараметрических подгрупп K , N' и A' как матричные версии тождества Эйлера из пятой строки таблицы в Предложении 4.1.

Поэтому мы можем определить характеры подгрупп K , N' и A' через геометрическое действие $SL_2(\mathbb{R})$ посредством преобразований Мёбиуса. Однако действие (11) определено в верхней полуплоскости, а было бы желательно получить его унитарную форму, что возможно при конформном отображении на единичный круг. В эллиптическом случае это достигается преобразованием Келли, которое переводит подгруппу K в:

$$(19) \quad \frac{1}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} = \begin{pmatrix} e^{it} & 0 \\ 0 & e^{-it} \end{pmatrix}.$$

Диагональная матрица справа задаёт преобразование Мёбиуса, которое сводится к умножению на e^{2it} , то есть унитарному вращению комплексной плоскости.

Гиперболическая версия преобразования Келли выглядит так [31, § ??]:

$$(20) \quad \frac{1}{2} \begin{pmatrix} 1 & \epsilon \\ -\epsilon & 1 \end{pmatrix} \begin{pmatrix} \cosh t & \sinh t \\ \sinh t & \cosh t \end{pmatrix} \begin{pmatrix} 1 & -\epsilon \\ \epsilon & 1 \end{pmatrix} = \begin{pmatrix} e^{\epsilon t} & 0 \\ 0 & e^{-\epsilon t} \end{pmatrix}.$$

Преобразование Мёбиуса с полученной матрицей так же сводится к умножению на $e^{2\epsilon t}$, что является унитарным преобразованием Лоренца двумерного пространства-времени Минковского.

В параболическом случае мы используем для преобразования Келли сходную матрицу [31, § ??]:

$$C_\epsilon = \begin{pmatrix} 1 & -\epsilon \\ -\epsilon & 1 \end{pmatrix}$$

Тогда образом матрицы (3) из подгруппы N будет:

$$(21) \quad \begin{pmatrix} 1 & -\epsilon \\ -\epsilon & 1 \end{pmatrix} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \epsilon \\ \epsilon & 1 \end{pmatrix} = \begin{pmatrix} 1 + \epsilon t & t \\ 0 & 1 - \epsilon t \end{pmatrix} = \begin{pmatrix} e^{\epsilon t} & t \\ 0 & e^{-\epsilon t} \end{pmatrix}.$$

Хотя этот результат не сильно отличается от двух предыдущих случаев (19) и (20), но присутствие ненулевого члена в позиции (1, 2) нарушает баланс. Тем не менее, мы по прежнему определим параболическое вращение, как преобразование Мёбиуса с матрицей (21), которое уже не будет простым умножением на скаляр.

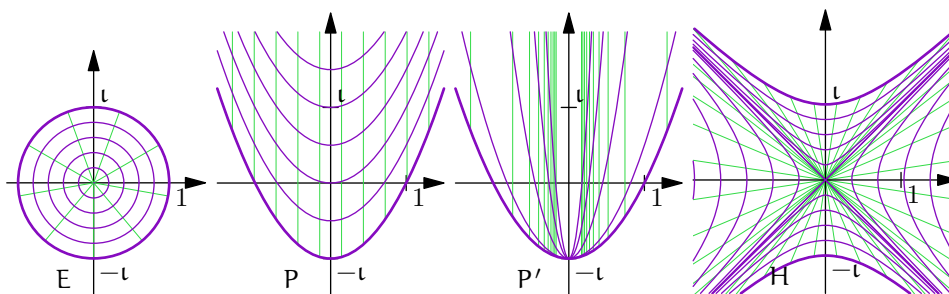


Рис. 4. Геометрическая версия унитарных вращений: эллиптические (E), два варианта параболических (P и P'), и гиперболические (H). Фиолетовые орбиты являются линиями уровня для соответствующих функций модуля. Зелёные прямые линии соединяют точки с одинаковым значением аргумента, они нарисованы с постоянным шагом приращения аргумента для каждого случая.

Пример 5.1. Параболическое вращение с верхне-треугольной матрицей из подгруппы N есть:

$$(22) \quad \begin{pmatrix} e^{\varepsilon t} & t \\ 0 & e^{-\varepsilon t} \end{pmatrix} : -\varepsilon \mapsto t + \varepsilon(t^2 - 1).$$

Это совпадает с *циклическим вращением* определённым в [59, § 8]. Сравнение с тождеством Эйлера по прежнему подтверждает определение параболического синуса как линейной функции $\text{sinp } t = t$, но предлагает новое значение для параболического косинуса (квадратичное вместо постоянного):

$$\text{cosp } t = 1 - t^2, \quad \text{sinp } t = t.$$

Соответственно параболическая теорема Пифагора выглядит так:

$$(23) \quad \text{sinp}^2 t + \text{cosp } t = 1,$$

что неплохо соседствует с эллиптической с одной стороны и гиперболической— с другой версиями:

$$\sin^2 t + \cos^2 t = 1, \quad \sinh^2 t - \cosh^2 t = -1.$$

Отметим также, что тождество (23) менее вырождено, чем выражение вообще без синуса $\text{cosp}^2 t = 1$ из [14] (см. так же (16), (17)).

Пример 5.2. Существует вторая возможность определить параболические вращения используя нижне-треугольные матрицы из подгруппы N'. Существенное отличие в том, что точка $-\varepsilon$ теперь неподвижна (как и все точки вертикальной оси) и для получения орбиты мы будем использовать ε^{-1} , который является идеальным элементом (точкой в бесконечности) [59, App. C] потому, что ε —делитель нуля. Вопросы замыкания гиперкомплексных чисел идеальными элементами в физическом контексте рассматривались в [29].

Для подгруппы N' мы получаем:

$$(24) \quad \begin{pmatrix} e^{-\varepsilon t} & 0 \\ t & e^{\varepsilon t} \end{pmatrix} : \frac{1}{\varepsilon} \mapsto \frac{1}{t} + \varepsilon \left(1 - \frac{1}{t^2} \right).$$

Сравнение с (22) показывает, что обе версии связаны отображением $t \mapsto t^{-1}$. Такое же соотношение связывает и параболические тригонометрические функции. Параболическая «единичная окружность» (или *цикл* [31, 59]) определяется уравнением $x^2 - y = 1$ в обоих случаях, см Рис. 4(P) и (P'). Однако остальные орбиты различаются и мы опишем их в следующей секции.

Рис. 4 показывает преобразования Мёбиуса определённых матрицами (19), (20) и (21) с соответствующими «единичными кругами», которые являются образами верхней полуплоскости под действием соответствующих преобразований Келли [31, § ??].

6. ВОССТАНОВЛЕНИЕ АЛГЕБРАИЧЕСКОЙ СТРУКТУРЫ ИЗ ГЕОМЕТРИИ

Мы хотим, чтобы индуцированные представления были линейными, для этого порождающий их характер должен также быть линейным. Вращения в эллиптическом и параболическом случае задаются произведениями с комплексными и двойными числами соответственно и потому — линейны. Однако, нетривиальные параболические вращения (22) и (24) (Рис. 4(P) и (P')) — нелинейны и мы введём новые алгебраические операции на дуальных числах, которые линеаризуют преобразования Мёбиуса. Для этого мы будем систематически использовать хорошо известный приём «обращения теорем в определения».

6.1. Модуль и аргумент. В эллиптическом и гиперболическом случае орбиты вращений являются точками с постоянной нормой (модулем): $x^2 + y^2$ или $x^2 - y^2$ соответственно. В параболическом случае мы используем этот же признак:

Определение 6.1. Орбиты преобразований (22) и (24) являются линиями уровня для следующих функций, которые мы будем называть соответствующими *модулем (нормой)*:

$$(25) \quad \text{для } N : |u + \varepsilon v| = u^2 - v, \quad \text{для } N' : |u + \varepsilon v|' = \frac{u^2}{v + 1}.$$

Замечание 6.2. (1) Выражение $|(u, v)| = u^2 - v$ равно параболическому расстоянию из $(0, \frac{1}{2})$ до (u, v) к точке (u, v) , см. [31, Lem. ??], и так же соответствует «параболической теореме Пифагора» (23).

(2) Модуль для подгруппы N' выражает параболическое фокальное расстояние из точки $(0, -1)$ к (u, v) , см. [31, Lem. ??].

Единственные прямые, которые переходят в прямые под действием параболических вращений с подгруппами N и N' , это вертикальные прямые. Поэтому мы их будем использовать как множества точек с постоянным значением аргумента и дадим следующее:

Определение 6.3. Параболический *аргумент* определяется как:

$$(26) \quad \text{для } N : \arg(u + \varepsilon v) = u, \quad \text{для } N' : \arg'(u + \varepsilon v) = \frac{1}{u}.$$

Оба определения 6.1 и 6.3 задают свойства хорошо согласующиеся с параболическими вращениями:

Предложение 6.4. Пусть точка w_t получается из w параболическим вращением на угол t в (22) или в (24). Тогда:

$$|w_t|^{(r)} = |w|^{(r)}, \quad \arg^{(r)} w_t = \arg^{(r)} w + t,$$

где выражения со штрихом используются для подгруппы N' .

Все доказательства в этой и следующей секциях производятся вычислением либо непосредственно, либо с помощью пакета символьческих расчётов [30, 33].

Замечание 6.5. Отметим, что в традиционном изложении [59, App. C(30')] параболический модуль и аргумент выражаются как в (18), что в некотором смысле обратно к нашим соглашениям.

6.2. Параболический поворот как умножение. Для определения параболического произведения мы снова будем использовать известные свойства комплексного произведения в качестве определения. Действительно, параболическое вращение можно рассматривать как умножение на унимодулярное скаляр, таким образом мы приходим к следующему

Определение 6.6. Произведение двух векторов w_1 и w_2 определяется такими двумя условиями:

$$(1) \arg^{(r)}(w_1 w_2) = \arg^{(r)} w_1 + \arg^{(r)} w_2;$$

$$(2) |w_1 w_2|^{(r)} = |w_1|^{(r)} \cdot |w_2|^{(r)}.$$

Параболическое сопряжение так же требует специального определения, которое основано на изменении знака аргумента.

Определение 6.7. Сопряжённый вектор задаётся равенством:

$$(27) \quad \overline{u + \varepsilon v} = -u + \varepsilon v.$$

Очевидно, мы имеем следующие свойства сопряжения: $|\overline{w}|^{(r)} = |w|^{(r)}$ и $\arg^{(r)} \overline{w} = -\arg^{(r)} w$. Определения 6.1, 6.3 и 6.6 совместно задают значение для произведения двух векторов единственным образом.

Предложение 6.8. Параболическое произведение двух векторов имеет следующие явные выражения:

$$(28) \quad \text{для } N: \quad (u, v) * (u', v') = (u + u', (u + u')^2 - (u^2 - v)(u'^2 - v'));$$

$$(29) \quad \text{для } N': \quad (u, v) * (u', v') = \left(\frac{uu'}{u + u'}, \frac{(v + 1)(v' + 1)}{(u + u')^2} - 1 \right).$$

Не смотря на необычный вид оба произведения имеют хорошо знакомые свойства:

Предложение 6.9. Оба произведения (28) и (29) удовлетворяют таким условиям:

- (1) Они коммутативны и ассоциативны;
- (2) Соответствующие параболические повороты (22) и (24) совпадают с умножением на унимодулярный вектор.
- (3) Значение произведения $w_1 \bar{w}_2$ не изменится если оба вектора w_1 и w_2 будут повернуты на тот же угол соответствующим вращением (22) или (24).
- (4) Для любого вектора w выполняется тождество:

$$|w \bar{w}| = |w|^2.$$

В частности, свойство (3) из этого Предложения будет важно для скалярного произведения (38), по отношению к которому наши индуцированные представления будут унитарны.

7. ИНВАРИАНТНАЯ ЛИНЕЙНАЯ АЛГЕБРА

Теперь мы хотим определить линейную структуру на \mathbb{R}^2 , которая будет совместима с параболическим умножением из предыдущей секции и, потому, с параболическими вращениями, см. Предложение 6.9(2)). Умножение на положительный действительный скаляр задаётся прямолинейно: оно не должно менять аргумент и соответственно масштабировать норму вектора. Поэтому для $\alpha > 0$ мы имеем:

$$(30) \quad \alpha \cdot (\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \alpha \mathbf{v} + \mathbf{u}^2(1 - \alpha)) \quad \text{для } \mathbf{N},$$

$$(31) \quad \alpha \cdot (\mathbf{u}, \mathbf{v}) = \left(\mathbf{u}, \frac{\mathbf{v} + 1}{\alpha} - 1 \right) \quad \text{для } \mathbf{N}'.$$

С другой стороны, сложение векторов может быть задано несколькими существенно отличными свойствами. Мы приведём две возможности: тропическую и экзотическую.

7.1. Тропическая форма. Введём лексиграфический порядок на \mathbb{R}^2 :

$$(\mathbf{u}, \mathbf{v}) \prec (\mathbf{u}', \mathbf{v}') \quad \text{тогда и только тогда,} \quad \begin{cases} \text{когда} & \mathbf{u} < \mathbf{u}'; \\ \text{или} & \mathbf{u} = \mathbf{u}', \mathbf{v} < \mathbf{v}'. \end{cases}$$

Соответственно мы можем задать функции \min и \max для любой пары векторов из \mathbb{R}^2 соответственно. Тогда сумма двух векторов определяется как их минимум (или максимум). Такое же определение используется в тропической математике, так же известной как «деквантизация» Маслова или алгебры \mathbb{R}_{\min} и \mathbb{R}_{\max} , см. превосходный обзор [45]. Легко проверить, что такое сложение совместимо с умножением (30)–(31) удовлетворяет дистрибутивному закону и потому будет инвариантно относительно параболических вращений. Это направление выглядит интересным и обещающим, однако мы больше не будем его рассматривать в данной работе.

7.2. Экзотическая форма. Сложение двух векторов может быть задано одной формулой для обеих подгрупп \mathbf{N} и \mathbf{N}' таким образом, что их отличие будет спрятано в соответствующих Определениях 6.1 (норма) и 6.3 (аргумент).

Определение 7.1. Параболическое сложение векторов задаётся в (экзотической) форме:

$$(32) \quad \arg^{(\prime)}(w_1 + w_2) = \frac{\arg^{(\prime)} w_1 \cdot |w_1|^{(\prime)} + \arg^{(\prime)} w_2 \cdot |w_2|^{(\prime)}}{|w_1 + w_2|^{(\prime)}},$$

$$(33) \quad |w_1 + w_2|^{(\prime)} = |w_1|^{(\prime)} \pm |w_2|^{(\prime)},$$

где штрихованные версии формул соответствуют подгруппе \mathbf{N}' .

На первый взгляд формула для нормы суммы (33) выглядит слишком тривиально. В её защиту мы можем отметить, что она располагается как раз между эллиптическим $|w + w'| \leq |w| + |w'|$ и гиперболическим $|w + w'| \geq |w| + |w'|$ неравенствами треугольника для норм.

Правило для сложения аргументов (32) тоже не является произвольным. Из теоремы синусов евклидовой геометрии мы можем получить

$$\sin(\phi - \psi') = \frac{|w| \cdot \sin(\psi - \psi')}{|w + w'|}, \quad \sin(\psi' - \phi) = \frac{|w'| \cdot \sin(\psi - \psi')}{|w + w'|},$$

где $\psi^{(\prime)} = \arg w^{(\prime)}$ и $\phi = \arg(w + w^{(\prime)})$. Используя значение (16) параболического синуса $\sin_p \theta = \theta$ мы получаем (32).

Аккуратность в обращении с нулями в знаменателе выражения (32) достигается при переходе от вектора $w = u + \varepsilon v$ к паре однородных полярных координат $[a, r] = [|w|^{(r)} \cdot \arg^{(r)} w, |w|^{(r)}]$ (штрихованная версия соответствует подгруппе N'). Тогда сложение задаётся покомпонентно в однородных координатах:

$$w_1 + w_2 = [a_1 + a_2, r_1 + r_2], \quad \text{где } w_i = [a_i, r_i].$$

Умножение из Определения 6.6 в однородных координатах есть:

$$w_1 \cdot w_2 = [a_1 r_2 + a_2 r_1, r_1 r_2], \quad \text{где } w_i = [a_i, r_i].$$

Таким образом, однородные координаты линеаризуют сложение (32)–(33) и умножение на скаляр (30). Переход к более прозрачным координатам может быть рассмотрен в контексте бирациональной геометрии [39].

Параболическое сложение определяется условиями (32)–(33) однозначно, но явные формулы достаточно сложны. Во многих случаях без них можно обойтись используя следующие свойства:

Предложение 7.2. *Векторное сложение для подгрупп N и N' заданное выражением (32)–(33) обладает следующими свойствами:*

- (1) Они коммутативны и ассоциативны.
- (2) Выполняются дистрибутивный закон для умножений (28) и (29); соответственно:
- (3) Сложения перестановочны с поворотами;
- (4) Выполняются дистрибутивные законы для умножения на скаляр (30) и (31) соответственно:

$$a \cdot (w_1 + w_2) = a \cdot w_1 + a \cdot w_2, \quad (a + b) \cdot w = a \cdot w + b \cdot w.$$

Что бы завершить построение действительного векторного пространства нам надо описать нулевой и обратный вектора. В частности, мы определим обратный элемент как имеющий тот же аргумент и противоположную норму.

Предложение 7.3. (N) *Нулевой вектор есть $(0, 0)$ и соответственно противоположный элемент для (u, v) есть $(u, 2u^2 - v)$.*

(N') *Нулевой вектор есть $(\infty, -1)$ и соответственно противоположный элемент для (u, v) есть $(u, -v - 2)$.*

Теперь используя равенство $-w = (-1) \cdot w$ можно убедиться, что та же формула (30) и (31) может быть использована для умножения на отрицательные действительные скаляры.

Замечание 7.4. Обычная линейная структура является неподходящей и в других вопросах, например, в кажущейся «неконформности» длин из параболического фокуса, то есть с параметром $\sigma = 0$ в [31, Prop. ??]. Однако подходящая выбор определений восстанавливает искомые свойства.

Первоначальное определение конформности [31, Defn. ??] рассматривало традиционный предел $y' \rightarrow y$ вдоль прямых линий, т.е. «спиц» изображённых на Рис. 3. Хорошо известно, что этот подход полностью работает в эллиптическом и гиперболическом случае. Однако, в параболическом случае вертикальные линии больше подходят на роль «спиц», см. Рис. 4(P) и (P'), таким образом обоснованно рассматривать конформный предел вдоль них [31, Prop. ??].

8. ИНДУЦИРОВАННЫЕ ПРЕДСТАВЛЕНИЯ

Теперь мы можем применить гиперкомплексные характеры построенные выше к индуцированным представлениям заданным формулой (13). Заметим, что только компактная подгруппа K требует комплексно-значных характеров, так

как для однозначности представления образ матрицы из (4) при $t = 2\pi$ должен быть единицей. Для подгрупп N' и A' такого ограничения нет и мы можем рассмотреть характеры всех трёх типов: эллиптические, параболические и гиперболические. Более того, параболические характеры могут быть любого типа: алгебраические (15) или одного из геометрических (22) и (24). Таким образом мы имеем семь существенно разных типов индуцированных представлений, или даже одиннадцати (если отличать разновидности параболических характеров).

Пример 8.1. Рассмотрим подгруппу $H = K$, как отмечено выше, в силу её компактности характер может быть только комплексно-значным. Тогда унитарный характер χ_k имеет вид:

$$(34) \quad \chi_k \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} = e^{-ikt}, \quad \text{где } k \in \mathbb{Z}.$$

Используя явный вид (9) отображения s вычисляем функцию r заданную в (12):

$$r \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{1}{\sqrt{c^2 + d^2}} \begin{pmatrix} d & -c \\ c & d \end{pmatrix} \in K.$$

Тогда:

$$r(g^{-1} * s(u, v)) = \frac{1}{\sqrt{(cu + d)^2 + (cv)^2}} \begin{pmatrix} cu + d & -cv \\ cv & cu + d \end{pmatrix}, \quad \text{где } g^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Подставляя это выражение в (34) и комбинируя с мёбиусовым преобразованием аргумента (11) получаем явную реализацию ρ_k индуцированного представления (13):

$$(35) \quad \rho_k(g)f(w) = \frac{|cw + d|^k}{(cw + d)^k} f\left(\frac{aw + b}{cw + d}\right), \quad \text{где } g^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad w = u + iv.$$

Это представление действует на комплексно-значных функциях в верхней полуплоскости $\mathbb{R}_+^2 = SL_2(\mathbb{R})/K$ и является унитарным из дискретной серии [42, § IX.2].

Предложение 8.2. Пусть $f_k(w) = \frac{|w-i|^k}{(w-i)^k}$ for $k = 2, 3, \dots$, тогда

- (1) f_k является собственным вектором операторов $\rho_k(h)$, для любого $h \in K$, с собственным значением $\chi_k(h)$ [42, § IX.2].
- (2) Функция $K(z, w) = \rho_k(s(z))f_k(w)$, где $s(z)$ определяется в (9), есть воспроизводящее ядро Бергмана в верхней полуплоскости [22, § 3.2] умноженное на $\left|\frac{z-i}{w-i}\right|^k$.

Сходным образом мы получаем ядро Коши если рассмотрим предельный случай $k = 1$ ложной дискретной серии [42, Ch. IX]. Существует много других связей представления (35) с комплексным анализом [22] и теорией операторов. К примеру, преобразования Мёбиуса операторов определяют функциональное исчисление Данфорда–Рисса и соответствующий спектр [25].

Пример 8.3. Для подгруппы N' возможен более широкий выбор характеров.

- (1) Традиционно рассматриваются комплекснозначные характеры подгруппы N' , т.е.:

$$(36) \quad \chi_\tau^{\mathbb{C}} \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} = e^{i\tau t}, \quad \text{где } \tau \in \mathbb{R}.$$

Непосредственные вычисления в этом случае показывают, что:

$$r \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{c}{d} & 1 \end{pmatrix} \in N'.$$

Поэтому:

$$(37) \quad \tau(g^{-1} * s(\mathbf{u}, \mathbf{v})) = \begin{pmatrix} 1 & 0 \\ \frac{c\mathbf{v}}{d+c\mathbf{u}} & 1 \end{pmatrix}, \quad \text{где } g^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Подставляя это значение в характер (36) и соединяя с мёбиусовым преобразованием (11) получаем соответствующую реализацию формулы (13):

$$\rho_{\tau}^{\mathbb{C}}(g)f(w) = \exp\left(i \frac{\tau c\mathbf{v}}{c\mathbf{u} + d}\right) f\left(\frac{a\mathbf{w} + b}{c\mathbf{w} + d}\right), \quad \text{где } w = \mathbf{u} + \varepsilon\mathbf{v}, \quad g^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Соответствующее индуцированное представление действует в пространстве комплекснозначных функций в верхней полуплоскости \mathbb{R}_+^2 , которая является подмножеством однородного пространства $SL_2(\mathbb{R})/N$ состоящего из дуальных чисел. Соседство комплексных и дуальных чисел в одном выражении является непривычным.

- (2) Возьмём параболический характер χ_{τ} алгебраического типа заданный умножением дуальных чисел (15):

$$\chi_{\tau} \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} = e^{\varepsilon\tau t} = 1 + \varepsilon\tau t, \quad \text{где } \tau \in \mathbb{R}.$$

По прежнему подставляем в него значение (37) и получаем такое выражение:

$$\rho_{\tau}(g)f(w) = \left(1 + \varepsilon \frac{\tau c\mathbf{v}}{c\mathbf{u} + d}\right) f\left(\frac{a\mathbf{w} + b}{c\mathbf{w} + d}\right),$$

где w , τ и g те же что и раньше.

Это индуцированное представление действует в пространстве функций определённых на верхней полуплоскости, как подмножестве дуальных чисел, и со значениями в дуальных числах. То есть, оно использует только дуальные числа и обычные алгебраические операции над ними. Естественным образом это представление является линейным.

- (3) Рассмотрим параболический характер χ_{τ}^g заданный геометрическим действием (22). Тогда представление действует в том же пространстве, что и в последнем случае, но формула будет другой:

$$\rho_{\tau}^g(g)f(w) = \left(1 + \varepsilon \frac{2\tau c\mathbf{v}}{c\mathbf{u} + d}\right) f\left(\frac{a\mathbf{w} + b}{c\mathbf{w} + d}\right) + \frac{\tau c\mathbf{v}}{c\mathbf{u} + d} + \varepsilon \frac{(\tau c\mathbf{v})^2}{(c\mathbf{u} + d)^2},$$

где w , τ и g описаны выше. Это представление линейно по отношению к операциям (30), (32) и (33).

Все характеры использованные в предыдущих примерах являются унитарными: два первых в традиционном смысле, последний—как объяснено в Предложении 6.9. Соответственно общая конструкция индуцированных представлений [18, § 13.2] гарантирует унитарность полученных представлений в соответствующих смыслах.

Теорема 8.1. *Все три индуцированных представления $\rho_{\tau}^{\mathbb{C}}$, ρ_{τ} и ρ_{τ}^g группы $SL_2(\mathbb{R})$ из Примера 8.3 являются унитарными в пространстве функций заданных на верхней полуплоскости \mathbb{R}_+^2 как подмножестве дуальных чисел с внутренним произведением:*

$$(38) \quad \langle f_1, f_2 \rangle = \int_{\mathbb{R}_+^2} f_1(w) \bar{f}_2(w) \frac{du dv}{v^2}, \quad \text{где } w = \mathbf{u} + \varepsilon\mathbf{v},$$

и мы используем

- (1) сопряжение и умножение значений функций в алгебре комплексных и дуальных чисел для представлений $\rho_{\tau}^{\mathbb{C}}$ и ρ_{τ} соответственно;

- (2) сопряжение (27) и умножение (28) значений функций для представления ρ_τ^g .

Внутреннее произведение (38) положительно определено для представления $\rho_\tau^{\mathbb{C}}$, но не для двух других. Соответствующие пространства с вырожденным внутренним произведением являются параболическими аналогами пространств Крейна [1]. Сами пространства Крейна в нашей классификации являются гиперболическими.

9. ПРИНЦИП СХОДСТВА И СООТВЕТСТВИЯ: ОПЕРАТОРЫ ПОВЫШЕНИЯ

Приведённые выше наблюдения позволяют сформулировать следующий эмпирический принцип, который должен представлять эвристический интерес:

- Принцип 9.1** (Сходство и соответствие). (1) Подгруппы K , N' и A' играют сходную роль в структуре группы $SL_2(\mathbb{R})$ и её представлений;
 (2) Перемена подгруппы должна производиться с одновременной соответствующей заменой гиперкомплексных единиц i , ε или ϵ .

Безусловно, первая часть Принципа (сходство) способна вызвать удивление у любого, кто знаком с группой $SL_2(\mathbb{R})$. Однако, к настоящему моменту мы видели, что совместно со второй частью (соответствие), принцип уже проявился следующим образом:

- Действие $SL_2(\mathbb{R})$ на однородном пространстве $SL_2(\mathbb{R})/H$ для $H = K$, N' или A' задаётся дробно-линейными преобразованиями комплексных, дуальных или двойных чисел соответственно (Предл. 3.2).
- Подгруппы A' , N' или K изоморфны группам унитарных вращений соответствующих единичных «окружностей» на плоскостях двойных, дуальных или комплексных чисел (Предл. 4.1).
- Представления индуцированные с подгруппы K , N' или A являются унитарными если скалярное произведение функций определяются посредством произведения и сопряжения значений функций как комплексных, дуальных или двойных чисел (Теор. 8.1).

Замечание 9.2. Принцип сходства и соответствия близок к суперсимметрии между бозонами и фермионами в физике, но в нашем случае сходство устанавливается между тремя различными типами объектов.

Приведём ещё одну иллюстрацию Принципа. Рассмотрим алгебру Ли \mathfrak{sl}_2 группы $SL_2(\mathbb{R})$. Один из возможных базисов в \mathfrak{sl}_2 таков [54, § 8.1]:

$$A = \frac{1}{2} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

Тогда коммутационные соотношения будут следующими:

$$(39) \quad [Z, A] = 2B, \quad [Z, B] = -2A, \quad [A, B] = -\frac{1}{2}Z.$$

Пусть у нас есть представление ρ группы $SL_2(\mathbb{R})$ в пространстве V . Рассмотрим производное представление $d\rho$ алгебры \mathfrak{sl}_2 [42, § VI.1] и будем использовать обозначение $\tilde{X} = d\rho(X)$ для $X \in \mathfrak{sl}_2$. Для анализа представления ρ оказывается полезным разложить V по собственным векторам оператора \tilde{X} для некоторого $X \in \mathfrak{sl}_2$, см. Предл. 8.2 и ряд Тайлора в комплексном анализе.

Пример 9.3. Уже не должно быть удивительным, что мы собираемся рассмотреть три случая:

- (1) Пусть $X = Z$ будет генератором подгруппы K (4). Так как эта подгруппа компактна, то собственные вектора $\tilde{Z}v_k = ikv_k$ параметризованы дискретным параметром $k \in \mathbb{Z}$. Особую роль играют операторы L^\pm повышения/понижения [42, § VI.2; 54, § 8.2], определённые коммутационными соотношениями:

$$(40) \quad [\tilde{Z}, L^\pm] = \lambda_\pm L^\pm.$$

Таким образом L^\pm являются собственными для операторов $\text{ad}Z$ присоединённого представления \mathfrak{sl}_2 [42, § VI.2]. Важно, что вектор L^+v_k так же является собственным для \tilde{Z} :

$$\begin{aligned} \tilde{Z}(L^+v_k) &= (L^+\tilde{Z} + \lambda_+L^+)v_k = L^+(\tilde{Z}v_k) + \lambda_+L^+v_k = ikL^+v_k + \lambda_+L^+v_k \\ &= (ik + \lambda_+)L^+v_k. \end{aligned}$$

Полагая $L^+ = a\tilde{A} + b\tilde{B} + c\tilde{Z}$ из коммутационного соотношений (39) и определяющего равенства (40) получаем систему уравнений:

$$c = 0, \quad 2a = \lambda_+b, \quad -2b = \lambda_+a.$$

Эта система совместима тогда и только тогда, когда $\lambda_+^2 + 4 = 0$. В этом случае операторы повышения/понижения $L^\pm = \pm i\tilde{A} + \tilde{B}$ действуют на одномерной цепочке собственных подпространств:

$$\dots \xrightleftharpoons[L^-]{L^+} V_{k-2} \xrightleftharpoons[L^-]{L^+} V_k \xrightleftharpoons[L^-]{L^+} V_{k+2} \xrightleftharpoons[L^-]{L^+} \dots$$

- (2) Рассмотрим случай $X = B$ генератора подгруппы A' (6). Эта подгруппа некомпактна и собственные значения для \tilde{B} могут быть произвольным числом, однако операторы повышения/понижения по прежнему могут играть важную роль [16, § II.1; 46, § 1.1]. Будем опять искать решение в форме $L_h^\pm = a\tilde{A} + b\tilde{B} + c\tilde{Z}$ для коммутатора $[\tilde{B}, L_h^\pm] = \lambda L^\pm$. Получаем систему:

$$2c = \lambda a, \quad b = 0, \quad \frac{a}{2} = \lambda c.$$

Она совместима только если $\lambda^2 = 1$. Очевидные значения $\lambda = \pm 1$ приводят к операторам $L_h^\pm = \pm \tilde{A} + \tilde{Z}/2$, которые и используются в [16, § II.1; 46, § 1.1]. В этом случае неприводимый \mathfrak{sl}_2 -модуль представляется одномерной цепочкой собственных значений.

Допуская двойные числа мы можем удовлетворить условие совместимости $\lambda^2 = 1$ дополнительными значениями $\lambda = \pm \epsilon$. Тогда возникает дополнительная пара операторов повышения/понижения $L_\epsilon^\pm = \pm \epsilon \tilde{A} + \tilde{Z}/2$, которая сдвигает собственные значения в «ортогональном» направлении к действию традиционных операторов L_h^\pm . Следовательно неприводимый \mathfrak{sl}_2 -модуль может параметризоваться уже двумерной

решёткой собственных значений на плоскости двойных чисел:

$$\begin{array}{ccccccc}
 \dots & & \dots & & \dots & & \dots \\
 & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} \\
 \dots & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{(n-2)+\epsilon(k-2)} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{n+\epsilon(k-2)} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{(n+2)+\epsilon(k-2)} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & \dots \\
 & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \\
 \dots & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{(n-2)+\epsilon k} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{n+\epsilon k} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{(n+2)+\epsilon k} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & \dots \\
 & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \\
 \dots & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{(n-2)+\epsilon(k+2)} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{n+\epsilon(k+2)} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & V_{(n+2)+\epsilon(k+2)} & \xrightleftharpoons[L_{h}^{-}]{L_{h}^{+}} & \dots \\
 & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \begin{array}{c} \updownarrow \\ L_{\epsilon}^{-} \quad L_{\epsilon}^{+} \\ \downarrow \end{array} & & \\
 & & \dots & & \dots & & \dots & &
 \end{array}$$

(3) Рассмотрим, наконец, случай $X = -B + Z/2$ генератора подгруппы N' (7). По описанной ранее процедуре получаем систему:

$$b + 2c = \lambda a, \quad -a = \lambda b, \quad \frac{a}{2} = \lambda c$$

с условием совместимости $\lambda^2 = 0$. Если мы ограничимся действительным (комплексным) корнем $\lambda = 0$, то соответствующие операторы $L_{\epsilon}^{\pm} = -\tilde{B} + \tilde{Z}/2$ не будут менять собственные значения и будут бесполезны в таком контексте. Однако дуальные значения $\lambda = \pm\epsilon$ позволяют использовать операторы $L_{\epsilon}^{\pm} = \pm\epsilon\tilde{A} - \tilde{B} + \tilde{Z}/2$ для построения \mathfrak{sl}_2 -модулей с одномерной цепочкой собственных значений в дуальных числах.

Замечание 9.4. Стоит отметить, что

- введение комплексных чисел необходимо для существования операторов повышения/понижения в эллиптическом случае;
- в параболическом случае введение дуальных чисел необходимо для полезности этих операторов;
- в гиперболическом случае двойственные числа выглядят необязательными.

Подытожим рассмотренный пример подчеркнув в нём роль принципа схождения и соответствия:

Предложение 9.5. Пусть вектор $X \in \mathfrak{sl}_2$ порождает подгруппу K , N' или A' , то есть векторами $X = Z$, $B - Z/2$, или B соответственно. Пусть ι есть соответствующая гиперкомплексная единица.

Тогда операторы повышения/понижения L^{\pm} удовлетворяющие коммутационным соотношениям:

$$[X, L^{\pm}] = \pm\iota L^{\pm}, \quad [L^{-}, L^{+}] = 2\iota X.$$

имеют вид:

$$L^{\pm} = \pm\iota\tilde{A} + \tilde{Y}.$$

Здесь $Y \in \mathfrak{sl}_2$ является линейной комбинацией B и Z а так же удовлетворяет следующим условиям:

- $Y = [A, X]$.
- $X = [A, Y]$.
- форма Киллинга $K(X, Y)$ [18, § 6.2] равна нулю.

Любое из приведённых выше условий совместно с $Y \in \text{span}\{B, Z\}$ определяет Y с точностью до действительного множителя.

Кажется правдоподобным, что применение Принципа сходства и соответствия не ограничивается приведёнными здесь случаями. Например, сходным образом могут быть рассмотрены операторы рождения/уничтожения для гармонического осциллятора и представлений группы Гейзенберга [34].

10. ЗАКЛЮЧЕНИЕ: ОТКРЫТЫЕ ВОПРОСЫ

Введенные индуцированные представления заслуживают пристального исследования. Среди важных вопросов можно отметить следующие:

- Унитарные комплексно-значные представления $SL_2(\mathbb{R})$ делятся на три серии: дискретную, непрерывную, дополнительную [16, 42]. Как эта классификация соотносится с введёнными гиперкомплексными представлениями группы $SL_2(\mathbb{R})$.
- Не каждый неприводимый $SL_2(\mathbb{R})$ -модуль допускает введение комплексного скалярного произведения, в котором представление становится унитарным [16, 46]. Возможно, некоторые из таких модулей унитаризируются в гиперкомплексных числах.
- Многие $SL_2(\mathbb{R})$ -модули, допускающие гиперкомплексную унитаризацию, могут быть реализованы как пространства функций на действительной прямой или верхней полуплоскости [22; 35, § 5]. Концепция ковариантного преобразования [32; 35, § 4] позволяет строить в этих пространствах интегральные формулы и находить воспроизводящие ядра.
- Упомянутые пространства функций состоят из нулевых векторов некоторых $SL_2(\mathbb{R})$ -инвариантных дифференциальных операторов, которые являются аналогами операторов Лапласа и Коши–Римана. В частности, ковариантное преобразование с материнским всплеском, которое исчезает под действием оператора уничтожения (или любой его степени), является решением соответствующего дифференциального уравнения [35, § 5.3]. Это может быть использовано для анализа решений уравнений в частных производных [40, 48].
- Аналитические функции используются в качестве модели функционального исчисления операторов, в исчислении Гельфанда–Данфорда–Риса. Новые неприводимые $SL_2(\mathbb{R})$ -модули являются подходящими моделями для исчисления и спектральной теории несамосопряжённых операторов [20; 25; 35, § 6].

Эти направления являются частями Эрлангенской программы в широком смысле [27, 35] и должны быть продолжены в последующих работах. Ожидается, что Принцип сходства и соответствия окажет направляющую роль в поисках наиболее гармоничных конструкций.

БЛАГОДАРНОСТИ

Автор благодарен проф. Н.А. Громову за полезное обсуждение данной работы и многочисленные ценные замечания.

СПИСОК ЛИТЕРАТУРЫ

- [1] Damir Z. Arov and Harry Dym. *J-contractive matrix valued functions and related topics*. Encyclopedia of Mathematics and its Applications, vol. 116. Cambridge University Press, Cambridge, 2008. ↑14, 16
- [2] Christian Bauer, Alexander Frink, Richard Kreckel, and Jens Vollinga. *GiNaC is Not a CAS*, 2001. URL: <http://www.ginac.de/>. ↑23, 25, 43

- [3] John L. Bell. *A primer of infinitesimal analysis*. Cambridge University Press, Cambridge, Second, 2008. ↑2
- [4] Dino Boccaletti, Francesco Catoni, Roberto Cannata, Vincenzo Catoni, Enrico Nichelatti, and Paolo Zampetti. *The mathematics of Minkowski space-time and an introduction to commutative hypercomplex numbers*. Birkhäuser Verlag, Basel, 2008. ↑2, 3, 1, 5, 6
- [5] Peter Butkovič. *Max-linear systems: theory and algorithms*. Springer Monographs in Mathematics. Springer-Verlag London Ltd., London, 2010. ↑11, 17
- [6] Francesco Catoni, Roberto Cannata, and Enrico Nichelatti. The parabolic analytic functions and the derivative of real functions. *Adv. Appl. Clifford Algebras* 14 (2):185–190, 2004. ↑2, 3, 1, 5
- [7] P. Cerejeiras, U. Kähler, and F. Sommen. Parabolic Dirac operators and the Navier-Stokes equations over time-varying domains. *Math. Methods Appl. Sci.* 28 (14):1715–1724, 2005. ↑3, 2
- [8] Martin Davis. *Applied nonstandard analysis*. Wiley-Interscience [John Wiley & Sons], New York, 1977. ↑2
- [9] G.I. Garas'ko. *Nachala finslerovoi geometrii dlya fizikov. [Elements of Finsler geometry for physicists]*. TETRU, Moscow, 2009. 268 pp. URL: <http://hypercomplex.xpsweb.com/articles/487/ru/pdf/00-gbook.pdf>. ↑2
- [10] GNU. *General Public License (GPL)*. Free Software Foundation, Inc., Boston, USA, version 3, 2007. URL: <http://www.gnu.org/licenses/gpl.html>. ↑25, 43
- [11] N. A. Gromov. *Kontraktsii i analiticheskie prodolzheniya klassicheskikh grupp. Edinyi podkhod. (Russian) [Contractions and analytic extensions of classical groups. Unified approach]*. Akad. Nauk SSSR Ural. Otdel. Komi Nauchn. Tsentr, Syktyvkar, 1990. ↑5
- [12] N. A. Gromov and V. V. Kuratov. Noncommutative space-time models. *Czechoslovak J. Phys.* 55 (11):1421–1426, 2005. ↑2, 6
- [13] N. A. Gromov and V. V. Kuratov. Possible quantum kinematics. *J. Math. Phys.* 47 (1):013502, 9, 2006. ↑2, 6
- [14] Francisco J. Herranz, Ramón Ortega, and Mariano Santander. Trigonometry of spacetimes: a new self-dual approach to a curvature/signature (in)dependent trigonometry. *J. Phys. A* 33 (24):4525–4551, 2000. E-print: [arXiv:math-ph/9910041](https://arxiv.org/abs/math-ph/9910041). ↑7, 9, 8
- [15] Francisco J. Herranz and Mariano Santander. Conformal compactification of spacetimes. *J. Phys. A* 35 (31):6619–6629, 2002. E-print: [arXiv:math-ph/0110019](https://arxiv.org/abs/math-ph/0110019). ↑2, 5, 1, 6
- [16] Roger Howe and Eng-Chye Tan. *Nonabelian harmonic analysis. Applications of $SL(2, \mathbf{R})$* . Springer-Verlag, New York, 1992. ↑4, 15, 17, 3, 19
- [17] Andrei Khrennikov and Gavriel Segre. Hyperbolic quantization. In L. Accardi, W. Freudenberger, and M. Schürman (eds.) *Quantum probability and infinite dimensional analysis*, pages 282–287, World Scientific Publishing, Hackensack, NJ, 2007. ↑2, 3, 1, 5, 6
- [18] A. A. Kirillov. *Elements of the theory of representations*. Springer-Verlag, Berlin, 1976. Translated from the Russian by Edwin Hewitt, Grundlehren der Mathematischen Wissenschaften, Band 220. ↑5, 6, 14, 17, 4, 15, 18
- [19] Anastasia V. Kasil. Isometric action of $SL_2(\mathbb{R})$ on homogeneous spaces. *Adv. App. Clifford Algebras* 20 (2):299–312, 2010. E-print: [arXiv:0810.0368](https://arxiv.org/abs/0810.0368). MR2012b:32019. ↑5
- [20] Vladimir V. Kasil. Möbius transformations and monogenic functional calculus. *Electron. Res. Announc. Amer. Math. Soc.* 2 (1):26–33, 1996. On-line. ↑17, 19
- [21] Vladimir V. Kasil. How many essentially different function theories exist? *Clifford algebras and their application in mathematical physics (Aachen, 1996)*, pages 175–184, Kluwer Academic Publishers, Dordrecht, 1998. ↑3
- [22] Vladimir V. Kasil. Analysis in $\mathbf{R}^{1,1}$ or the principal function theory. *Complex Variables Theory Appl.* 40 (2):93–118, 1999. E-print: [arXiv:funct-an/9712003](https://arxiv.org/abs/funct-an/9712003). ↑3, 5, 6, 13, 17, 2, 14, 19
- [23] Vladimir V. Kasil. Two approaches to non-commutative geometry. *Complex methods for partial differential equations (Ankara, 1998)*, pages 215–244, Kluwer Acad. Publ., Dordrecht, 1999. E-print: [arXiv:funct-an/9703001](https://arxiv.org/abs/funct-an/9703001). ↑3
- [24] Vladimir V. Kasil. Meeting Descartes and Klein somewhere in a noncommutative space. In A. Fokas, J. Halliwell, T. Kibble, and B. Zegarlini (eds.) *Highlights of mathematical physics (London, 2000)*, pages 165–189, Amer. Math. Soc., Providence, RI, 2002. E-print: [arXiv:math-ph/0112059](https://arxiv.org/abs/math-ph/0112059). ↑3
- [25] Vladimir V. Kasil. Spectrum as the support of functional calculus. *Functional analysis and its applications*, pages 133–141, Elsevier, Amsterdam, 2004. E-print: [arXiv:math.FA/0208249](https://arxiv.org/abs/math.FA/0208249). ↑13, 17, 14, 19
- [26] Vladimir V. Kasil. An example of Clifford algebras calculations with GiNaC. *Adv. Appl. Clifford Algebr.* 15 (2):239–269, 2005. E-print: [arXiv:cs.MS/0410044](https://arxiv.org/abs/cs.MS/0410044), On-line. ↑43

- [27] Vladimir V. Kisil. Erlangen program at large-0: Starting with the group $SL_2(\mathbf{R})$. *Notices Amer. Math. Soc.* 54 (11):1458–1465, 2007. E-print: [arXiv:math/0607387](#), [On-line](#). ↑5, 17, 6, 19
- [28] Vladimir V. Kisil. Fillmore-Springer-Cnops construction implemented in GiNaC. *Adv. Appl. Clifford Algebr.* 17 (1):59–70, 2007. Updated full text and source files: E-print: [arXiv:cs.MS/0512073](#), [On-line](#). ↑23, 25, 43, 51
- [29] Vladimir V. Kisil. Two-dimensional conformal models of space-time and their compactification. *J. Math. Phys.* 48 (7):073506, 8, 2007. E-print: [arXiv:math-ph/0611053](#). ↑5, 9, 6
- [30] Vladimir V. Kisil. Erlangen program at large—2: Inventing a wheel. The parabolic one. *Trans. Inst. Math. of the NAS of Ukraine*, pages 89–98, 2010. E-print: [arXiv:0707.4024](#). ↑5, 10, 11, 25, 30, 44
- [31] Vladimir V. Kisil. Erlangen program at large-1: Geometry of invariants. *SIGMA, Symmetry Integrability Geom. Methods Appl.* 6 (076):45, 2010. E-print: [arXiv:math.CV/0512416](#). MR2011i:30044. ↑5, 9, 12, 6, 8, 10, 13
- [32] Vladimir V. Kisil. Covariant transform. *Journal of Physics: Conference Series* 284 (1):012038, 2011. E-print: [arXiv:1011.3947](#). ↑17, 19
- [33] Vladimir V. Kisil. Erlangen program at large—2 1/2: Induced representations and hypercomplex numbers. *Izvestiya Komi nauchnogo centra UrO RAN [Izvestiya Komi nauchnogo centra UrO RAN]* 1 (5):4–10, 2011. E-print: [arXiv:0909.4464](#). ↑11, 23, 25, 43
- [34] Vladimir V. Kisil. Erlangen Programme at Large 3.2: Ladder operators in hypercomplex mechanics. *Acta Polytechnica* 51 (4):44–53, 2011. E-print: [arXiv:1103.1120](#). ↑7, 17, 8, 19
- [35] Vladimir V. Kisil. Erlangen programme at large: an Overview. In S.V. Rogosin and A.A. Koroleva (eds.) *Advances in applied analysis*, pages 1–94, Birkhäuser Verlag, Basel, 2012. E-print: [arXiv:1106.1686](#). ↑17, 19
- [36] Vladimir V. Kisil. *Geometry of Möbius transformations: Elliptic, parabolic and hyperbolic actions of $SL_2(\mathbf{R})$* . Imperial College Press, London, 2012. Includes a live DVD. ↑5, 9, 12
- [37] Vladimir V. Kisil. Hypercomplex representations of the Heisenberg group and mechanics. *Internat. J. Theoret. Phys.* 51 (3):964–984, 2012. E-print: [arXiv:1005.5057](#). ↑2, 7, 6, 8
- [38] Vladimir V. Kisil. Is commutativity of observables the main feature, which separate classical mechanics from quantum? *Izvestiya Komi nauchnogo centra UrO RAN [Izvestiya Komi nauchnogo centra UrO RAN]* 3 (11):4–9, 2012. E-print: [arXiv:1204.1858](#). ↑2
- [39] János Kollár and Shigefumi Mori. *Birational geometry of algebraic varieties*. Cambridge Tracts in Mathematics, vol. 134. Cambridge University Press, Cambridge, 2008. With the collaboration of C. H. Clemens and A. Corti. Paperback reprint of the hardback edition 1998. Zbl1143.14014. ↑11, 13
- [40] Nadiia Kononenko. Projective structures and algebras of their differential invariants. *Acta Applicandae Mathematicae* 109 (1):87–99, 2010. ↑17, 19
- [41] Vladislav V. Kravchenko. *Applied pseudoanalytic function theory*. Frontiers in Mathematics. Birkhäuser Verlag, Basel, 2009. With a foreword by Wolfgang Sproessig. ↑3, 2
- [42] Serge Lang. $SL_2(\mathbf{R})$. Graduate Texts in Mathematics, vol. 105. Springer-Verlag, New York, 1985. Reprint of the 1975 edition. ↑4, 13, 15, 17, 3, 14, 16, 19
- [43] M. A. Lavrent'ev and B. V. Shabat. *Problemy gidrodinamiki i ikh matematicheskie modeli. [Problems of hydrodynamics and their mathematical models]*. Izdat. “Nauka”, Moscow, Second, 1977. ↑3, 7, 2, 8
- [44] M. Libine. Hyperbolic Cauchy integral formula for the split complex numbers. *ArXiv e-prints*, December 2007. ↑3, 2
- [45] G. L. Litvinov. The Maslov dequantization, and idempotent and tropical mathematics: a brief introduction. *Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI)* 326 (Teor. Predst. Din. Sist. Komb. i Algoritm. Metody. 13):145–182, 282, 2005. E-print: [arXiv:math/0507014](#). ↑11, 17, 12
- [46] Volodymyr Mazorchuk. *Lectures on sl_2 -modules*. World Scientific, 2009. ↑15, 17, 19
- [47] A. E. Motter and M. A. F. Rosa. Hyperbolic calculus. *Adv. Appl. Clifford Algebras* 8 (1):109–128, 1998. ↑3, 2
- [48] Valery N. Pilipchuk. *Nonlinear dynamics. Between linear and impact limits*. Lecture Notes in Applied and Computational Mechanics, vol. 52. Springer, Berlin, 2010. ↑2, 3, 17, 1, 19
- [49] Valery N. Pilipchuk. Non-smooth spatio-temporal coordinates in nonlinear dynamics, January 2011. E-print: [arXiv:1101.4597](#). ↑2
- [50] R.I. Pimenov. Unified axiomatics of spaces with maximal movement group. *Litov. Mat. Sb.* 5:457–486, 1965. Zbl0139.37806. ↑2
- [51] L. S. Pontryagin. *Obobshcheniya chisel [Generalisations of numbers]*. Bibliotekha “Kvant” [Library “Kvant”], vol. 54. “Nauka”, Moscow, 1986. ↑5

- [52] Norman Ramsey. *Noweb* — a simple, extensible tool for literate programming. URL: <http://www.eecs.harvard.edu/~nr/noweb/>. ↑23, 25, 43
- [53] Garret Sobczyk. The hyperbolic number plane. *College Math Journal* 26 (4):268–280, 1995. ↑3, 2
- [54] Michael E. Taylor. *Noncommutative harmonic analysis*. Mathematical Surveys and Monographs, vol. 22. American Mathematical Society, Providence, RI, 1986. ↑15, 16, 17
- [55] S. Ulrych. Relativistic quantum physics with hyperbolic numbers. *Phys. Lett. B* 625 (3–4):313–323, 2005. ↑2, 1, 5, 6
- [56] S. Ulrych. Considerations on the hyperbolic complex Klein-Gordon equation. *J. Math. Phys.* 51 (6):063510, 8, 2010. ↑2
- [57] V. A. Uspenskiĭ. *CHto takoe nestandartnyi0 analiz? [What is non-standard analysis?]*. “Nauka”, Moscow, 1987. With an appendix by V. G. Kanoveĭ. ↑2
- [58] J. C. Vignaux and A. Durañona y Vedia. Sobre la teoría de las funciones de una variable compleja hiperbólica [On the theory of functions of a complex hyperbolic variable]. *Univ. nac. La Plata. Publ. Fac. Ci. fis. mat.* 104:139–183, 1935. Zbl62.1122.03. ↑2, 3
- [59] I. M. Yaglom. *A simple non-Euclidean geometry and its physical basis*. Heidelberg Science Library. Springer-Verlag, New York, 1979. Translated from the Russian by Abe Shenitzer, with the editorial assistance of Basil Gordon. ↑2, 7, 8, 9, 10, 1, 5, 6, 11

APPENDIX A. OUTPUT OF SYMBOLIC CALCULATIONS

Here are the results of our symbolic calculations. The source code can be obtained from this paper [33] source at <http://arXiv.org>. It uses Clifford algebra facilities [28] of the GiNaC library [2]. The source code is written in noweb [52] literature programming environment.

Calculations for subgroup N and straight spokes

Reference point: $(0 \ -1)$

Reference point arg: 0

Reference point norm: 1

Cayley of the matrix X: $\begin{pmatrix} 1 + e^0x & 1x \\ 0 & 1 - e^0x \end{pmatrix}$

Rotation by x: $(u + x \ 2ux + v + x^2)$

Rotation of (u_0, v_0) by x: $(x \ -1 + x^2)$

Parabolic norm: $u^2 - v$

Parabolic argument: u

Real number t_1 as a dual number: $(0 \ -t_1)$

Product: $(u' + u \ u'^2 + 2u'u + u^2 + v'u^2 - u'^2u^2 - v'v + u'^2v)$

Product by a scalar: $(u \ u^2 + av - au^2)$

Real part: $(0 \ -uv - u^2 + u^3 + v)$

Imag part: $(1 \ 1 + uv - u^3)$

Zero vector: $(0 \ 0)$

Negative vector: $(u \ 2u^2 - v)$

Dual number from argument 0 and norm n: $(0 \ -n)$

Dual number from argument a_1 and norm n: $(a_1 \ a_1^2 - n)$

Dual number from argument a_1 and norm n—norm: n

Dual number from argument a_1 and norm n—arg: a_1

Lin comb of two vectors $a*(1, 0)+b*(-1, 0)$: $\left(-\frac{b-a}{b+a} \ \frac{b^2-b^3-a^3-3b^2a+a^2-2ba-3ba^2}{(b+a)^2} \right)$

P is the sum $\Re(P)$ and $\Im(P)$: **true**

The real part of a real dual number is itself: **true**

norm is invariant under rotations: **true**

Product is invariant under rotations: **true**

Product $w\bar{w}$ is norm squared: **true**

Product $(u, v) * (u_0, v_0)$ is (u, v) : **true**

Add is commutative: **true**

Add is associative: **true**

S-mult is commutative: **true**

S-mult is associative: **true**

S-mult is distributive 1: **true**

S-mult is distributive 2: **true**

Product is symmetric (commutative): **true**

Prod is associative: **true**

Product is distributive: **true**

Calculations for subgroup N' and straight spokes

Reference point: $(\infty \ -1 + \infty^2)$

Reference point arg: 0

Reference point norm: 1

Cayley of the matrix X: $\begin{pmatrix} 1 - e^0x & 0 \\ 1x & 1 + e^0x \end{pmatrix}$

Rotation by x: $\left(\frac{u}{1+ux} \ -\frac{2ux+u^2x^2-v}{1+2ux+u^2x^2} \right)$

Rotation of (u_0, v_0) by x : $\left(\frac{1}{x} \quad -\frac{-1+x^2}{x^2} \right)$

Parabolic norm: $\frac{u^2}{1+v}$

Parabolic argument: $\frac{1}{u}$

Real number t_1 as a dual number: $\left(\infty \quad \frac{\infty^2 - t_1}{t_1} \right)$

Product: $\left(\frac{u' u}{u' + u} \quad \frac{1 + v' - u'^2 - 2u' u - u^2 + v' v + v}{(u' + u)^2} \right)$

Product by a scalar: $\left(u \quad -\frac{-1+a-v}{a} \right)$

Real part: $\left(\infty \quad \frac{\infty^2 - u^2 + \infty^2 v + u}{(-1+u)u} \right)$

Imag part: $\left(1 \quad -\frac{-1+u-v}{u} \right)$

Zero vector: $\left(\infty \quad -1 \right)$

Negative vector: $\left(u \quad -2 - v \right)$

Dual number from argument 0 and norm n : $\left(\infty \quad \frac{\infty^2 - n}{n} \right)$

Dual number from argument a_1 and norm n : $\left(\frac{1}{a_1} \quad -\frac{-1+a_1^2 n}{a_1^2 n} \right)$

Dual number from argument a_1 and norm n —norm: n

Dual number from argument a_1 and norm n —arg: a_1

Lin comb of two vectors $a * (1, 0) + b * (-1, 0)$: $\left(-\frac{b+a}{b-a} \quad \frac{b-b^2-a^2+2ba+a}{(b-a)^2} \right)$

P is the sum $\Re(P)$ and $\Im(P)$: **true**

The real part of a real dual number is itself: **true**

norm is invariant under rotations: **true**

Product is invariant under rotations: **true**

Product $w\bar{w}$ is norm squared: **true**

Product $(u, v) * (u_0, v_0)$ is (u, v) : **true**

Add is commutative: **true**

Add is associative: **true**

S-mult is commutative: **true**

S-mult is associative: **true**

S-mult is distributive 1: **true**

S-mult is distributive 2: **true**

Product is symmetric (commutative): **true**

Prod is associative: **true**

Product is distributive: **true**

Elliptic case of induced representations

map $r(M)$: $\left(\begin{array}{cc} \frac{d^2}{d^2+c^2} & -\frac{cd}{d^2+c^2} \\ \frac{cd}{d^2+c^2} & \frac{d^2}{d^2+c^2} \end{array} \right)$

map $s^{-1}(M)$: $\left(\begin{array}{cc} \frac{1}{d} & \frac{bc^2+c+bd^2}{d^2+c^2} \\ 0 & \frac{d^2}{d^2+c^2} \end{array} \right)$

character: $\left(\begin{array}{cc} \frac{2cdu+d^2+c^2u^2}{2cdu+d^2+c^2u^2+c^2v^2} & -\frac{c(cu+d)v}{2cdu+d^2+c^2u^2+c^2v^2} \\ \frac{c(cu+d)v}{2cdu+d^2+c^2u^2+c^2v^2} & \frac{(cu+d)^2+c^2v^2}{2cdu+d^2+c^2u^2+c^2v^2} \end{array} \right)$

Moebius map: $\left(\begin{array}{cc} \frac{cav^2+cau^2+bcu+bd+da u}{2cdu+d^2+c^2u^2+c^2v^2} & -\frac{bcv-dav}{2cdu+d^2+c^2u^2+c^2v^2} \end{array} \right)$

Moebius map is given by the imaginary unit: **true**

Parabolic (N') case of induced representations

map $r(M)$: $\left(\begin{array}{cc} 1 & 0 \\ \frac{c}{d} & 1 \end{array} \right)$

map $s^{-1}(M)$: $\left(\begin{array}{cc} \frac{1}{d} & b \\ 0 & d \end{array} \right)$

character: $\begin{pmatrix} 1 & 0 \\ \frac{cv}{cu+d} & 1 \end{pmatrix}$
 Moebius map: $\begin{pmatrix} \frac{b+au}{cu+d} & -\frac{bcv-dav}{(cu+d)^2} \\ \frac{1}{d} & -\frac{bc^2+c-bd^2}{d^2-c^2} \end{pmatrix}$
 Moebius map is given by the imaginary unit: **true**

Hyperbolic case of induced representations

map $r(M)$: $\begin{pmatrix} \frac{d^2}{d^2-c^2} & \frac{cd}{d^2-c^2} \\ \frac{cd}{d^2-c^2} & \frac{d^2}{d^2-c^2} \end{pmatrix}$
 map $s^{-1}(M)$: $\begin{pmatrix} \frac{1}{d} & -\frac{bc^2+c-bd^2}{d^2-c^2} \\ 0 & \frac{d}{d^2-c^2} \end{pmatrix}$
 character: $\begin{pmatrix} \frac{2cdu+d^2+c^2u^2}{2cdu+d^2+c^2u^2-c^2v^2} & \frac{c(cu+d)v}{2cdu+d^2+c^2u^2-c^2v^2} \\ \frac{c(cu+d)v}{2cdu+d^2+c^2u^2-c^2v^2} & \frac{(cu+d)^2}{2cdu+d^2+c^2u^2-c^2v^2} \end{pmatrix}$
 Moebius map: $\begin{pmatrix} -\frac{cav^2-cau^2-bcu-bd-dau}{2cdu+d^2+c^2u^2-c^2v^2} & -\frac{bcv-dav}{2cdu+d^2+c^2u^2-c^2v^2} \\ \frac{1}{d} & -\frac{bc^2+c-bd^2}{d^2-c^2} \end{pmatrix}$
 Moebius map is given by the imaginary unit: **true**

APPENDIX B. LIBRARY FOR DUAL NUMBER ALGEBRA

This is the second version of the library which perform non-standard computations with dual number described in this paper. The first version was published in [arXiv.org](https://arxiv.org) with the earlier more heuristic paper on this subject [30]. \LaTeX compilation of the file of this paper [33] will produces the file `dua1num.nw` in the current directory. This is a noweb [52] code of the program. It uses Clifford algebra facilities [28] of the GiNaC library [2].

This piece of software is licensed under [GNU General Public License](https://www.gnu.org/licenses/gpl-3.0.html) (Version 3, 29 June 2007) [10].

The part related to geodesic spokes is still unfinished.

B.1. Class `dual_number`.

B.1.1. *Public Methods.*

A dual number can be created simply by listing its two components.

25a `<Public methods 25a>` \equiv `dual_number(const ex & a, const ex & b);` (35d) 25b \triangleright

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Alternatively you can provide a 1×2 or 2×1 matrix, a list, another dual number P or a complex expression with a non-zero imaginary part to give two components. If P does not have two components and is a real-valued expression, it will be embedded into dual numbers with zero argument and norm equal to P .

25b `<Public methods 25a>` $+\equiv$ `dual_number(const ex & P);` (35d) \triangleleft 25a 25c \triangleright

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

We can also obtain the module and argument of a `dual_number`.

25c `<Public methods 25a>` $+\equiv$ `ex arg() const;`
`ex norm() const;` (35d) \triangleleft 25b 26a \triangleright

We define the conjugate of a `dual_number` as number with the same modulus but negative value of the argument.

26a (Public methods 25a)+≡ (35d) <25c 26b>
`ex conjugate() const;`

Real part may be defined out of the formula $\Re(u, v) = \frac{1}{2}((u, v) + \overline{(u, v)})$, but it simply reduces to the value of norm for the `dual_number`.

26b (Public methods 25a)+≡ (35d) <26a 26c>
`ex real_part() const;`
`ex imag_part() const;`

Negative of a `dual_number` and its power.

26c (Public methods 25a)+≡ (35d) <26b 26d>
`dual_number neg() const;`
`dual_number power(const ex & e) const;`

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

We can also convert a `dual_number` to a matrix.

26d (Public methods 25a)+≡ (35d) <26c 26e>
`matrix to_matrix() const { return matrix (1, 2, lst(u_comp, v_comp)); }`

We define the rule for parabolic norm of a sum, see (33).

26e (Public methods 25a)+≡ (35d) <26d 26f>
`ex add_norms(const dual_number & P) const;`

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Upper half-plane action of the group $SL_2(\mathbb{R})$ on the dual numbers.

26f (Public methods 25a)+≡ (35d) <26e 26g>
`dual_number sl2_moebius(const ex & a, const ex & b, const ex & c, const ex & d) const;`
`dual_number sl2_moebius(const ex & M) const;`

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Similar action in the form of the group SU on the unit disk.

26g (Public methods 25a)+≡ (35d) <26f 26h>
`dual_number su_moebius(const ex & a, const ex & b, const ex & c, const ex & d) const;`
`dual_number su_moebius(const ex & M) const;`

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

A particular, but important case of the above SU-action: unitary rotation by x for subgroups N and N' .

26h (Public methods 25a)+≡ (35d) <26g 27a>
`dual_number rot(const ex & x) const;`

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Algebraic operations are defined for `dual_number` in a way described in § 7. The standard C++ operators `+`, `-`, `*`, `÷` will be overloaded later in order to permit natural expressions with `dual_number`.

27a (Public methods 25a)+≡ (35d) <26h

```

dual_number add(const dual_number & P) const;
dual_number sub(const dual_number & P) const { return add(P.neg()); }
dual_number mul(const dual_number & P) const;
dual_number mul(const ex & a) const { return mul(dual_number(a)); }
dual_number div(const dual_number & P) const { return mul(P.power(-1)); }
dual_number div(const ex & a) const { return div(dual_number(a)); }

```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

We also define zero dual number.

27b (Additional routines declarations 27b)≡ (33c) 34d▷

```

dual_number zero_dual_number();
dual_number dn_from_arg_mod(const ex & a, const ex & n);

```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

B.2. Algebraic Subroutines. We need a couple of global variables which help to write uniformly algebraic rules for both cases of subgroups N and N' .

Firstly, we need to consider separately cases of subgroup N and N' , the following global variable keeps track on it.

27c (N-Nprime separation 27c)≡ (33c) 27d▷

```

bool subgroup_N=true;
bool straight_spoke=true;

```

In the case of the subgroup N' the reference point lies at infinity, the following `realsymbol` variable represents it in the symbolic calculations.

27d (N-Nprime separation 27c)+≡ (33c) <27c 27e▷

```

realsymbol Inf("Inf", "\\infty");

```

We define the “zero angle”, it is always zero.

27e (N-Nprime separation 27c)+≡ (33c) <27d

```

#define Arg0 0

```

Defines:

`Arg0`, used in chunks 30a, 34e, and 38b.

Here is the set of algebraic procedures representing definitions made in this paper.

B.2.1. Argument and u . We define the conjugate of a `dual_number` as number with the same modulus but negative value of the argument.

27f (Algebraic procedures 27f)≡ (34a) 29b▷

```

ex dual_number::conjugate() const { return dn_from_arg_mod(-arg(), norm()); }

```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

B.2.2. *Argument of a point.* The opposite task (finding argument of a point) is solved similarly (a direct formula).

28a (Dual number class further implementation 28a)≡ (34a) 28b▷

```

ex dual_number::arg() const {
    if (subgroup_N)
        return u_comp;

```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Again in the case of N' we need to consider cases of 0...

28b (Dual number class further implementation 28a)+≡ (34a) ◁28a 28c▷

```

    else {
        if (straight_spoke) {
            if (u_comp.normal().is_zero())
                return Inf;

```

...and ∞ . We try to replace $\frac{1}{\infty}$ by 0.

28c (Dual number class further implementation 28a)+≡ (34a) ◁28b 28d▷

```

    else
        try {
            realsymbol t("t");
            return pow(u_comp.subs(Inf ≡ pow(t, -1)), -1.normal().subs(t ≡ 0).normal());
        } catch (std::exception &p) {
            return pow(u_comp, -1);
        }

```

We now calculate argument for the case geodesic spokes (a direct formula).

28d (Dual number class further implementation 28a)+≡ (34a) ◁28c 28e▷

```

    } else
        return (1-pow(2*v_comp+1, half))÷u_comp;
    }

```

B.2.3. *Norm.* The corresponding value of the parabolic norm is calculated by the formulae (25) (a direct formula).

28e (Dual number class further implementation 28a)+≡ (34a) ◁28d 28f▷

```

ex dual_number::norm() const {
    if (subgroup_N)
        return pow(u_comp, 2)-v_comp;

```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

The case of subgroup N' require treatment of infinity. If the denominator is zero than the treatment is the same for straight and geodesic spokes.

28f (Dual number class further implementation 28a)+≡ (34a) ◁28e 29a▷

```

    else {
        ex denom=(straight_spoke?1:2)*v_comp+1.normal();
        if (denom.is_zero()) {
            if (u_comp.is_zero())
                return 1;
            else
                return Inf;

```

For a non-zero denominator we return the plain values.

29a `<Dual number class further implementation 28a>+≡ (34a) <28f 29e>`

```

    } else {
        if (straight_spoke)
            return (pow(u_comp,2)÷denom).normal();
        else
            return (u_comp÷pow(denom,halff)).normal();
    }
}

```

B.2.4. *The value of v from the argument and norm.* Often We need to find values of v such that for a given value of the first component u point (u, v) will have a given norm.

29b `<Algebraic procedures 27f>+≡ (34a) <27f 29c>`

```

ex v_from_u_and_norm(const ex & u, const ex & n) {
    realsymbol l("1");
    if (subgroup_N)
        return lsolve(dual_number(u, l).norm() ≡ n, l).normal();
}

```

Uses `dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.`

For subgroup N' we use powers to make equations linear.

29c `<Algebraic procedures 27f>+≡ (34a) <29b 29d>`

```

    else {
        if (straight_spoke)
            return lsolve(pow(dual_number(u, l).norm(), -1) ≡ pow(n, -1), l).normal();
        else
            return lsolve(pow(dual_number(u, l).norm(), -2).normal() ≡ pow(n, -2), l).normal();
    }
}

```

Uses `dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.`

29d `<Algebraic procedures 27f>+≡ (34a) <29c 30a>`

```

dual_number zero_dual_number() {
    return (subgroup_N ? dual_number(0, 0) : dual_number(Inf, -1));
}

```

Uses `dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.`

29e `<Dual number class further implementation 28a>+≡ (34a) <29a 30b>`

```

bool dual_number::is_zero() const {
    return is_equal(zero_dual_number());
}

```

Uses `dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.`

B.2.5. *Real and Imaginary Parts.* See [30] for a discussion of the real and imaginary parts of dual numbers.

30a `<Algebraic procedures 27f>+≡ (34a) <29d 31a>`

```

ex dual_number::real_part() const {
    return dn_from_arg_mod(Arg0, (1-arg())*norm());
}

ex dual_number::imag_part() const {
    return dn_from_arg_mod(1, arg()*norm());
}

```

Uses Arg0 27e and dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

B.2.6. *Product of Two Points.* We define now the product of two points according to the Definition 6.6. We also include a multiplication by a scalar: if a factor is a scalar it is replaced by a vector with the zero argument and norm equal to the scalar.

30b `<Dual number class further implementation 28a>+≡ (34a) <29e 30c>`

```

dual_number dual_number::mul(const dual_number & P) const {
    return dn_from_arg_mod(arg()+P.arg(), norm()*P.norm());
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

B.2.7. *Vector Addition of Two Points.* The rule for norm of a sum of two vectors

30c `<Dual number class further implementation 28a>+≡ (34a) <30b 30d>`

```

ex dual_number::add_norms(const dual_number & P) const {
    return (norm()+P.norm()).normal();
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

The sum is calculated from the expression (32).

30d `<Dual number class further implementation 28a>+≡ (34a) <30c 31b>`

```

dual_number dual_number::add(const dual_number & P) const {
    ex norms = add_norms(P);
    if (norms.normal().is_zero())
        return zero_dual_number();
    else
        return dn_from_arg_mod((arg()*norm()+P.arg()*P.norm())÷norms, norms);
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

To construct a dual number from its argument and norm, we first build a number with the given norm and then rotate it to obtain the desired argument.

31a \langle Algebraic procedures 27f $\rangle + \equiv$ (34a) \triangleleft 30a

```

dual_number dn_from_arg_mod(const ex & a, const ex & n) {
    dual_number P=dual_number(1, v_from_u_and_norm(1, n));
    ex angle=(a-P.arg()).normal();
    if (angle.is_zero())
        return P;
    else
        return P.rot(angle);
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

De Moivres Identity:

31b \langle Dual number class further implementation 28a $\rangle + \equiv$ (34a) \triangleleft 30d 31c \triangleright

```

dual_number dual_number::power(const ex & e) const {
    return dn_from_arg_mod(arg()*e, pow(norm(), e));
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Negative dual number for (u, v) has the same argument and the negative modulus.

31c \langle Dual number class further implementation 28a $\rangle + \equiv$ (34a) \triangleleft 31b 31d \triangleright

```

dual_number dual_number::neg() const {
    return dn_from_arg_mod(arg(), -norm());
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Action of the group $SL_2(\mathbb{R})$ on the upper halfplane of dual numbers (a direct formula).

31d \langle Dual number class further implementation 28a $\rangle + \equiv$ (34a) \triangleleft 31c 31e \triangleright

```

dual_number dual_number::sl2_moebius(const ex & a, const ex & b, const ex & c, const ex & d) const {
    return dual_number(((a*u_comp+b)/(c*u_comp+d)).normal(),
        (v_comp*(a*d-b*c)*pow(c*u_comp+d, -2)).normal());
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

The same transform, but given by a matrix

31e \langle Dual number class further implementation 28a $\rangle + \equiv$ (34a) \triangleleft 31d 32a \triangleright

```

dual_number dual_number::sl2_moebius(const ex & M) const {
    if (is_a<matrix>(M) & ex_to<matrix>(M).rows()≡2 & ex_to<matrix>(M).cols()≡2)
        return sl2_moebius(ex_to<matrix>(M)(0,0), ex_to<matrix>(M)(0,1),
            ex_to<matrix>(M)(1,0), ex_to<matrix>(M)(1,1));
    else
        throw(std::invalid_argument("dual_number.moebius(): parameter M"
            "should be a 2x2 matrix"));
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Action of the group $SL_2(\mathbb{R})$ conjugated with the Cayley transform on the unit disk of dual numbers (a direct formula).

32a (Dual number class further implementation 28a)+≡ (34a) <31e 32b>
`dual_number dual_number::su_moebius(const ex & a, const ex & b, const ex & c, const ex & d) const {
 ex denom=(c*u_comp+d).normal(), numer, summand=(straight_spoke? ex(1) : half);`

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a.

We now define two different expressions for numerator depending from the type of spokes we are using

32b (Dual number class further implementation 28a)+≡ (34a) <32a 32c>
`if (straight_spoke)
 numer = (v_comp-pow(u_comp, 2)+1)*(a*d-b*c)+pow(a*u_comp+b, 2);
else
 numer = half*((2*v_comp-pow(u_comp,2)+1)*(a*d-b*c)+pow(a*u_comp+b, 2));`

For zero denominator we replace it by Inf.

32c (Dual number class further implementation 28a)+≡ (34a) <32b 32d>
`if (denom.is_zero())
 return dual_number(((a*u_comp+b)*Inf, (numer*pow(Inf,2)-summand).normal()));
else`

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a.

A non-zero denominator may include infinity, which we treat separately.

32d (Dual number class further implementation 28a)+≡ (34a) <32c 32e>
`if (u_comp.has(Inf)) {
 realsymbol t("t");
 return dual_number((((a*u_comp+b)÷denom).subs(Inf ≡ pow(t, -1))
 .normal().subs(t ≡ 0).normal(),
 (numer*pow(denom, -2)-summand)
 .subs(Inf ≡ pow(t, -1)).normal().subs(t ≡ 0).normal());
 } else`

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a.

For non-zero determinant we return the plain value.

32e (Dual number class further implementation 28a)+≡ (34a) <32d 33a>
`return dual_number(((a*u_comp+b)÷denom).normal(),
 (numer*pow(denom, -2)-summand).normal());
}`

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a.

A variant of the function if the matrix is given as a whole.

```
33a (Dual number class further implementation 28a)+≡ (34a) <32e 33b>
    dual_number dual_number::su_moebius(const ex & M) const {
        if (is_a<matrix>(M) & ex_to<matrix>(M).rows()≡2 & ex_to<matrix>(M).cols()≡2)
            return su_moebius(ex_to<matrix>(M)(0,0), ex_to<matrix>(M)(0,1),
                               ex_to<matrix>(M)(1,0), ex_to<matrix>(M)(1,1));
        else
            throw(std::invalid_argument("dual_number.moebius(): parameter M should "
                                         "be a 2x2 matrix"));
    }
```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
41a 41a.

Parabolic rotations

```
33b (Dual number class further implementation 28a)+≡ (34a) <33a
    dual_number dual_number::rot(const ex & x) const {
        if (subgroup_N)
            return su_moebius(1, x, 0, 1);
        else
            return su_moebius(1, 0, x, 1);
    }
```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
41a 41a.

All algebraic routines are defined now.

B.3. Library Outline. Here is the outline how we use the above parts.

Routines for dual_number are collected in a separate library. We start from the definition dual_number class in the header file.

```
33c (dualnum.h 33c)≡
    (Header initialisation 33d)
    (N-Nprime separation 27c)
    (Dual number class declaration 35c)
    (Additional routines declarations 27b)
```

This is the initialisation part of the header file

```
33d (Header initialisation 33d)≡ (33c)
    #include <fstream>
    using namespace std;

    #include <ginac/ginac.h>
    using namespace GiNaC;
    #define GINAC_VERSION_ATLEAST( major, minor) \
        (GINACLIB_MAJOR_VERSION > major \
         ∨ (GINACLIB_MAJOR_VERSION ≡ major ∧ GINACLIB_MINOR_VERSION ≥ minor))
```

Defines:

GINAC_VERSION_ATLEAST, used in chunks 35–38.

Here is the file with the implementation.

```
34a <dualnum.cpp 34a>≡
    #include <dualnum.h>
    const numeric half=numeric(1,2);
    <Algebraic procedures 27f>
    <Dual number class implementation 37b>
    <Dual number class further implementation 28a>
    <Output routines 34b>
```

Defines:

numeric, used in chunks 52 and 53.

B.3.1. *Output routines.* We use standardised routines to output results of calculations.

```
34b <Output routines 34b>≡ (34a) 34c>
    void formula_out(string S, const ex & F, bool lineend) {
        cout << S << (latexout ? "\\(" : "(") << F << (latexout ? "\\)" : ")");
        if (lineend)
            cout << (latexout ? "\\\\" : "\\") << endl;
        else
            cout << "; ";
    }
```

Defines:

formula_out, used in chunks 35b, 43, 44, and 48.

This routine is used to check identities.

```
34c <Output routines 34b>+≡ (34a) <34b
    void test_out(string S, const ex & T) {
        cout << S << (latexout ? "\\textbf{" : "*" )
            << (is_a<dual_number>(T) ? ex_to<dual_number>(T).normal().is_zero() :
                T.evalm().normal().is_zero_matrix()) << (latexout ? "\\}" : "*")
            << endl;
    }
```

Defines:

test_out, used in chunks 35b, 45, 46, and 48e.

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40c 40c 40c 41a 41a 41a.

Here is declarations of additional routines for the header file.

```
34d <Additional routines declarations 27b>+≡ (33c) <27b 34e>
    bool latexout;
```

Function interface to global variables. Two work properly the library shall fix which kind rotations, either N or N' , is used.

```
34e <Additional routines declarations 27b>+≡ (33c) <34d 35a>
    ex arg0() { return Arg0; }
    bool is_subgroup_N() { return subgroup_N; }
    void set_subgroup(bool N);
```

Uses Arg0 27e and set_subgroup 42c.

We also define procedures to setup which spoke are used with the subgroup N' , either vertical lines or geodesics. This does not affect calculations with the subgroup N .

35a `<Additional routines declarations 27b>+≡ (33c) <34e 35b>`
`bool is_spoke_straight() { return straight_spoke; }`
`void set_straight_spoke(bool S);`

Uses `set_straight_spoke 42c`.

Nicer output for testing routines.

35b `<Additional routines declarations 27b>+≡ (33c) <35a>`
`void test_out(string S, const ex & T);`
`void formula_out(string S, const ex & F, bool lineend=true);`

Uses `formula_out 34b 43c 44g 44g 44g 44g 44h 54a` and `test_out 34c 45a 45b 45e 54b`.

B.4. Header and Implementation of the `dual_number` Class.

B.4.1. *Header File for `dual_number`*. We use the standard GiNaC machinery to define `dual_numbers` as derived of the class `basic`.

35c `<Dual number class declaration 35c>≡ (33c) 35d>`
`class dual_number : public basic`
`{`
`GINAC_DECLARE_REGISTERED_CLASS(dual_number, basic)`
`#if GINAC_VERSION_ATLEAST(1,5)`
`#else`
`static const tinfo_static_t return_type_tinfo_static[256];`
`#endif`

Uses `dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a`, `GINAC_VERSION_ATLEAST 33d`, and `tinfo_static_t 37b`.

Public methods (constructors, algebraic, etc.)

35d `<Dual number class declaration 35c>+≡ (33c) <35c 35e>`
`public:`
`<Public methods 25a>`
`<Technical methods 36d>`
`<Dual number operators 36a>`

We redefine protected methods for printing only.

35e `<Dual number class declaration 35c>+≡ (33c) <35d 35f>`
`protected:`
`void do_print(const print_context & c, unsigned level) const;`
`void do_print_latex(const print_latex & c, unsigned level) const;`

Private members: two components of a `dual_number`.

35f `<Dual number class declaration 35c>+≡ (33c) <35e 36c>`
`protected:`
`ex u_comp;`
`ex v_comp;`
`};`

We overload standard algebraic operations for `dual_number`.

```
36a (Dual number operators 36a)≡ (35d)
//const dual_number operator+(const dual_number & rh);
//const dual_number operator-(const dual_number & rh);
//const dual_number operator*(const dual_number & rh);
//const dual_number operator*(const ex & rh);
//const dual_number operator/(const ex & rh);
//const dual_number operator/(const dual_number & rh);
```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

We overload standard algebraic operations for `dual_number`.

```
36b (Dual number spare operators 36b)≡
const dual_number operator+(const dual_number & lh, const dual_number & rh);
const dual_number operator-(const dual_number & lh, const dual_number & rh);
const dual_number operator*(const dual_number & lh, const dual_number & rh);
const dual_number operator*(const dual_number & lh, const ex & rh);
const dual_number operator*(const ex & lh, const dual_number & rh);
const dual_number operator÷(const dual_number & lh, const ex & rh);
const dual_number operator÷(const ex & lh, const dual_number & rh);
const dual_number operator÷(const dual_number & lh, const ex & rh);
```

Defines:

`dual_number`, used in chunks 25–39, 41, and 42.

We overload standard algebraic operations for `dual_number`.

```
36c (Dual number class declaration 35c)+≡ (33c) <35f 37a>
//const dual_number operator*(const ex & lh, const dual_number & rh);
//const dual_number operator/(const ex & lh, const dual_number & rh);
```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

The following methods are needed for GiNaC to work properly.

```
36d (Technical methods 36d)≡ (35d)
dual_number normal() const { return dual_number(u_comp.normal(), v_comp.normal()); }
dual_number subs(const ex & e, unsigned options = 0) const;
bool is_zero() const;
bool is_equal(const ex & other) const;
size_t nops() const { return 2; }
ex op(size_t i) const;
ex & let_op(size_t i);
#if GINAC_VERSION_ATLEAST(1,5)
dual_number(const archive_node &n, lst &sym_lst);
void archive(archive_node &n) const;
ex unarchive(const archive_node &n, lst &sym_lst);
return_type_t return_type_tinfo() const {return make_return_type_t<dual_number>();}
#endif
```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a and `GINAC_VERSION_ATLEAST` 33d.

Additional routines.

37a (Dual number class declaration 35c)+≡ (33c) <36c
 //dual_number dn_from_arg_mod(const ex & a, const ex & n);
 ex v_from_u_and_norm(const ex & u, const ex & n);

 // End of "header"

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a.

B.4.2. *Standard Implementation Part.* The implementation uses standard GiNaC technique.

37b (Dual number class implementation 37b)≡ (34a) 37c▷

```
GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(dual_number, basic,
                                       print_func<print_context>(&dual_number::do_print),
                                       print_func<print_latex>(&dual_number::do_print_latex))

#if GINAC_VERSION_ATLEAST(1,5)
#else
const tinfo_static_t dual_number::return_type_tinfo_static[256] = {};
#endif
```

Defines:

tinfo_static_t, used in chunk 35c.

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a and GINAC_VERSION_ATLEAST 33d.

B.4.3. *Implementation of Constructors.* Default constructor.

37c (Dual number class implementation 37b)+≡ (34a) <37b 37d▷
 #if GINAC_VERSION_ATLEAST(1,5)
 dual_number::dual_number() : inherited(), u_comp(0), v_comp(0)
 #else
 dual_number::dual_number() : inherited(&dual_number::tinfo_static), u_comp(0), v_comp(0)
 #endif
 {
 setflag(status_flags::not_shareable);
 }

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a and GINAC_VERSION_ATLEAST 33d.

Constructor from two components.

37d (Dual number class implementation 37b)+≡ (34a) <37c 38a▷
 #if GINAC_VERSION_ATLEAST(1,5)
 dual_number::dual_number(const ex & a, const ex & b) : inherited(),
 u_comp(a), v_comp(b)
 #else
 dual_number::dual_number(const ex & a, const ex & b) : inherited(&dual_number::tinfo_static),
 u_comp(a), v_comp(b)
 #endif
 {
 }

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
 41a 41a and GINAC_VERSION_ATLEAST 33d.

Constructor from a single expression. It may contain two components...

```
38a (Dual number class implementation 37b)+≡ (34a) <37d 38b>
    #if GINAC_VERSION_ATLEAST(1,5)
    dual_number::dual_number(const ex & P) : inherited()
    #else
    dual_number::dual_number(const ex & P) : inherited(&dual_number::tinfo_static)
    #endif
    {
        if (is_a<lst>(P) ∨ is_a<matrix>(P) ∨ is_a<dual_number>(P)) {
            u_comp = P.op(0);
            v_comp = P.op(1);
        }
    }
```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a and `GINAC_VERSION_ATLEAST` 33d.

...if it is a real expression we embed it into `dual_number`...

```
38b (Dual number class implementation 37b)+≡ (34a) <38a 38c>
    } else if (P.imag_part().normal().is_zero()) {
        dual_number t=dn_from_arg_mod(Arg0, P);
        u_comp = t.op(0);
        v_comp = t.op(1);
    }
```

Uses `Arg0` 27e and `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

...or if its a complex expression we decompose it into the real and imaginary parts.

```
38c (Dual number class implementation 37b)+≡ (34a) <38b 38d>
    } else {
        u_comp = P.real_part();
        v_comp = P.imag_part();
    }
}
```

B.4.4. *Service methods for the GiNaC infrastructure.* Standard parts involving archiving, comparison and printing of the `dual_number` class

```
38d (Dual number class implementation 37b)+≡ (34a) <38c 38e>
    dual_number::dual_number(const archive_node &n, lst &sym_lst) : inherited()
    {
        n.find_ex("u_comp", u_comp, sym_lst);
        n.find_ex("v_comp", v_comp, sym_lst);
    }
```

Uses `dual_number` 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Archiving routine.

```
38e (Dual number class implementation 37b)+≡ (34a) <38d 39a>
    void dual_number::archive(archive_node &n) const
    {
        inherited::archive(n);
        n.add_ex("u_comp", u_comp);
        n.add_ex("v_comp", v_comp);
    }
```

Defines:

`dual_number`, used in chunks 25–39, 41, and 42.

Archiving routine.

39a (Dual number class implementation 37b)+≡ (34a) <38e 39b>
ex dual_number::unarchive(const archive_node &n, lst &sym_lst)
 {
 return (new dual_number(n, sym_lst))→setflag(status_flags::dynamallocated);
 }

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

Comparison routine.

39b (Dual number class implementation 37b)+≡ (34a) <39a 39c>
int dual_number::compare_same_type(const basic & other) const
 {
 GINAC_ASSERT(is_a<dual_number>(other));
 const dual_number &o = static_cast<const dual_number &>(other);

 int cmpval = u_comp.compare(o.op(0));
 if (cmpval≠0) return cmpval;

 return v_comp.compare(o.op(1));
 }

Defines:

dual_number, used in chunks 25–39, 41, and 42.

Equality of two dual numbers.

39c (Dual number class implementation 37b)+≡ (34a) <39b 39d>
bool dual_number::is_equal(const ex & other) const
 {
 GINAC_ASSERT(is_a<dual_number>(other));
 const dual_number &o = static_cast<const dual_number &>(other);

 return (u_comp-o.op(0)).normal().is_zero() ^ (v_comp-o.op(1)).normal().is_zero();
 }

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

B.4.5. *Printing.* Default printing.

39d (Dual number class implementation 37b)+≡ (34a) <39c 40a>
void dual_number::do_print(const print_context & c, unsigned level) const
 {
 c.s << "[";
 u_comp.print(c);
 c.s << ", ";
 v_comp.print(c);
 c.s << "]"
 }

Defines:

dual_number, used in chunks 25–39, 41, and 42.

L^AT_EX printing.

40a (Dual number class implementation 37b) +≡ (34a) <39d 40b>

```

void dual_number::do_print_latex(const print_latex & c, unsigned level) const
{
    c.s << "\\left(\\begin{array}{cc}";
    u_comp.print(c);
    c.s << "&";
    v_comp.print(c);
    c.s << "\\end{array}\\right)";
}

```

Defines:

dual_number, used in chunks 25–39, 41, and 42.

B.4.6. *Overloading algebraic operations.* Addition.

40b (Dual number class implementation 37b) +≡ (34a) <40a 40c>

```

const dual_number operator+(const dual_number & lh, const dual_number & rh)
{
    return lh.add(rh);
}

const dual_number operator-(const dual_number & lh, const dual_number & rh)
{
    return lh.sub(rh);
}

```

Defines:

dual_number, used in chunks 25–39, 41, and 42.

Multiplication.

40c (Dual number class implementation 37b) +≡ (34a) <40b 41a>

```

const dual_number operator*(const dual_number & lh, const ex & rh)
{
    return lh.mul(rh);
}

const dual_number operator*(const ex & lh, const dual_number & rh)
{
    return rh.mul(lh);
}

const dual_number operator*(const dual_number & lh, const dual_number & rh)
{
    return lh.mul(rh);
}

```

Defines:

dual_number, used in chunks 25–39, 41, and 42.

Division.

41a \langle Dual number class implementation 37b $\rangle + \equiv$ (34a) \langle 40c 41b \rangle

```

const dual_number operator÷(const dual_number & lh, const dual_number & rh)
{
    return lh.mul(rh.power(-1));
}

const dual_number operator÷(const ex & lh, const dual_number & rh)
{
    return rh.power(-1)*lh;
}

const dual_number operator÷(const dual_number & lh, const ex & rh)
{
    return lh.mul(pow(rh, -1));
}

```

Defines:

dual_number, used in chunks 25–39, 41, and 42.

B.4.7. *Component-related functions.*

41b \langle Dual number class implementation 37b $\rangle + \equiv$ (34a) \langle 41a 42a \rangle

```

ex dual_number::op(size_t i) const
{
    GINAC_ASSERT(i < nops());

    switch (i) {
    case 0:
        return u_comp;
    case 1:
        return v_comp;
    default:
        throw(std::invalid_argument("dual_number::op(): requested"
                                     " operand out of the range (2)"));
    }
}

```

Uses dual_number 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a
41a 41a.

42a (Dual number class implementation 37b) += (34a) <41b 42b>

```

dual_number dual_number::subs(const ex & e, unsigned options) const
{
    exmap m;
    if (e.info(info_flags::list)) {
        lst l = ex_to<lst>(e);
        for (lst::const_iterator i = l.begin(); i ≠ l.end(); ++i)
            m.insert(std::make_pair(i→op(0), i→op(1)));
    } else if (is_a<relational>(e)) {
        m.insert(std::make_pair(e.op(0), e.op(1)));
    } else
        throw(std::invalid_argument("dual_number::subs(): the parameter"
            " should be a relational or a lst"));

    return ex_to<dual_number>(inherited::subs(m, options));
}

```

Uses **dual_number** 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

42b (Dual number class implementation 37b) += (34a) <42a 42c>

```

ex & dual_number::let_op(size_t i)
{
    GINAC_ASSERT(i < nops());

    ensure_if_modifiable();
    switch (i) {
    case 0:
        return u_comp;
    case 1:
        return v_comp;
    default:
        throw(std::invalid_argument("dual_number::let_op(): requested operand"
            " out of the range (2)"));
    }
}

```

Uses **dual_number** 36b 36b 36b 36b 36b 36b 36b 36b 38e 39b 39d 40a 40b 40b 40c 40c 40c 41a 41a 41a.

42c (Dual number class implementation 37b) += (34a) <42b>

```

void set_subgroup(bool N) {
    subgroup_N = N;
}

void set_straight_spoke(bool S) {
    straight_spoke = S;
    if (not S)
        subgroup_N = false;
}

```

Defines:

set_straight_spoke, used in chunk 35a.
set_subgroup, used in chunk 34e.

This is the end of `dual_number` implementation.

APPENDIX C. PROGRAM FOR SYMBOLIC CALCULATIONS

This is a documentation for our symbolic calculations supporting this paper. You can obtain the program itself from the [source files](#) of this paper [33] at [arXiv.org](#); \LaTeX compilation of it will produces the file `parab-rotation.nw` in the current directory. This is a noweb [52] code of the program. It uses Clifford algebra facilities [28] of the GiNaC library [2].

This piece of software is licensed under [GNU General Public License](#) (Version 3, 29 June 2007) [10].

C.1. Calculation and Tests. This Subsection contains code for calculation of various expression. See [26] or GiNaCinfo for usage of Clifford algebra functions.

C.1.1. Calculation of Expressions. Firstly, we output the expression of the Cayley transform for a generic element from subgroups N and N' .

```
43a <Show expressions 43a>+≡ (50d) <43a 43b>
    ex XC=canonicalize_clifford((TC*X*TCI).evalm());
    formula_out("Cayley of the matrix X: ", XC.subs(sign≡0).normal());
```

Uses `formula_out` 34b 43c 44g 44g 44g 44h 54a.

Then we calculate Möbius action of those matrix on a point.

```
43b <Show expressions 43a>+≡ (50d) <43a 43c>
    dual_number W(clifford_moebius_map(XC, P.to_matrix(), e).subs(sign≡0).normal()),
    W1 = W.subs(1st(u≡u1, v≡v1));
    formula_out("Rotation by x: ", W);
    if (not W.is_equal(P.rot(x)))
        cout << "*** dualnumber::rot() gives wrong answer *** \\\(" <<
            P.rot(x) << "\\)" << endl << endl;
```

Uses `formula_out` 34b 43c 44g 44g 44g 44h 54a.

Next we specialise the above result to the reference point.

```
43c <Show expressions 43a>+≡ (50d) <43b 43d>
    formula_out("Rotation of \\\((u_0, v_0)\\) by \\\(x\\)": ",
    W.subs(1st(u ≡ u0, v ≡ v0)).subs(Inf ≡ pow(y, -1)).normal().subs(y ≡ 0).normal());
```

Defines:

`formula_out`, used in chunks 35b, 43, 44, and 48.

The expression for the parabolic norm.

```
43d <Show expressions 43a>+≡ (50d) <43c 43e>
    formula_out("Parabolic norm: ", P.norm());
    formula_out("Parabolic argument: ", P.arg());
```

Uses `formula_out` 34b 43c 44g 44g 44g 44h 54a.

Embedding of reals into dual numbers.

```
43e <Show expressions 43a>+≡ (50d) <43d 44a>
    possymbol t1("t1", "t_1"), a1("a1", "a_1"), n("n");
    formula_out("Real number \\\(t_1\\) as a dual number: ", dual_number(t1));
```

Uses `formula_out` 34b 43c 44g 44g 44g 44h 54a.

The expression for the product of two points.

44a (Show expressions 43a) +≡ (50d) <43e 44b>
`formula_out("Product: ", P*P1);`

Uses formula_out 34b 43c 44g 44g 44g 44h 54a.

The expression of the product of a point and a scalar.

44b (Show expressions 43a) +≡ (50d) <44a 44c>
`formula_out("Product by a scalar: ", a*P);`

Uses formula_out 34b 43c 44g 44g 44g 44h 54a.

Expressions for the real and imaginary parts.

44c (Show expressions 43a) +≡ (50d) <44b 44d>
`formula_out("Real part: ", P.real_part());`
`formula_out("Imag part: ", P.imag_part());`

Uses formula_out 34b 43c 44g 44g 44g 44h 54a.

The expression for a sum of two points is too cumbersome to be printed.

44d (Show expressions 43a) +≡ (50d) <44c 44e>
`//formula_out("Add is: ", (P+P1).normal());`

Uses formula_out 34b 43c 44g 44g 44g 44h 54a.

Expression for zero is

44e (Show expressions 43a) +≡ (50d) <44d 44f>
`formula_out("Zero vector: ", zero_dual_number());`

Uses formula_out 34b 43c 44g 44g 44g 44h 54a.

Expression for zero is

44f (Show expressions 43a) +≡ (50d) <44e 44g>
`formula_out("Negative vector: ", P.neg());`

Uses formula_out 34b 43c 44g 44g 44g 44h 54a.

Expression for zero is

44g (Show expressions 43a) +≡ (50d) <44f 44h>
`formula_out("Dual number from argument \\\(0\\\) and norm \\\(n\\\) : ",`
`$dn_from_arg_mod(0, n)$);`
`dual_number $PP=dn_from_arg_mod(a1, n)$;`
`$formula_out("Dual number from argument \\\(a_1\\\) and norm \\\(n\\\) : ",$`
`$PP.normal()$);`
`$formula_out("Dual number from argument \\\(a_1\\\) and norm \\\(n\\\)--norm: ",$`
`$PP.norm().normal()$);`
`$formula_out("Dual number from argument \\\(a_1\\\) and norm \\\(n\\\)--arg: ",$`
`$PP.arg().normal().normal()$);`

Defines:

`formula_out`, used in chunks 35b, 43, 44, and 48.

Linear combination of points (1, 0) and (-1, 0) with coefficients a and b, for the linearisation presented in [30].

44h (Show expressions 43a) +≡ (50d) <44g>
`formula_out("Lin comb of two vectors \\\(a*(1, 0)+b*(-1, 0)\\\) : ",`
`$(a*dual_number(1,0)+b*dual_number(-1,0)).normal()$);`

Defines:

`formula_out`, used in chunks 35b, 43, 44, and 48.

C.1.2. *Checking Algebraic Identities.* In this Subsection we verify basic algebraic properties of the defined operations.

A dual number is the sum of its real and imaginary parts.

45a `<Check identities 45a>≡ (50d) 45b▷
test_out("\(P\) is the sum \(\Re(P)\) and \(\Im(P)\): ",
P-(ex_to<dual_number>(P.real_part()+ex_to<dual_number>(P.imag_part())));`

Defines:

`test_out`, used in chunks 35b, 45, 46, and 48e.

A dual number made out of a real a has the norm of real part equal to a .

45b `<Check identities 45a>+≡ (50d) <45a 45c▷
test_out("The real part of a real dual number is itself: ",
ex_to<dual_number>(dual_number(a).real_part()).norm()-a);`

Defines:

`test_out`, used in chunks 35b, 45, 46, and 48e.

The norm is invariant under parabolic rotations, i.e. they are in agreement with Defn. 6.1.

45c `<Check identities 45a>+≡ (50d) <45b 45d▷
test_out("norm is invariant under rotations: ", P.norm()-W.norm());`

Uses `test_out` 34c 45a 45b 45e 54b.

The product $w_1 \bar{w}_2$ is invariant under rotations, Prop. 3.

45d `<Check identities 45a>+≡ (50d) <45c 45e▷
test_out("Product is invariant under rotations: ", P*P1.conjugate()-W*W1.conjugate());`

Uses `test_out` 34c 45a 45b 45e 54b.

Product $w \bar{w}$ is $(0, |w|^2)$, Prop. 4.

45e `<Check identities 45a>+≡ (50d) <45d 45f▷
test_out("Product \(\bar{w}\) is norm squared: ",
(P*P.conjugate()-dn_from_arg_mod(Arg0, pow(P.norm(), 2))));`

Defines:

`test_out`, used in chunks 35b, 45, 46, and 48e.

The reference point is unit under multiplication.

45f `<Check identities 45a>+≡ (50d) <45e 45g▷
test_out("Product \((u, v)*(u_0, v_0)\) is \((u, v)\): ", P*P0-P);`

Uses `test_out` 34c 45a 45b 45e 54b.

Addition is commutative, Prop. 1.

45g `<Check identities 45a>+≡ (50d) <45f 45h▷
test_out("Add is commutative: ", (P+P1)-(P1+P));`

Uses `test_out` 34c 45a 45b 45e 54b.

Addition is associative, Prop. 1.

45h `<Check identities 45a>+≡ (50d) <45g 46a▷
test_out("Add is associative: ", ((P+P1)+ P2)-(P+(P1+P2)));`

Uses `test_out` 34c 45a 45b 45e 54b.

Multiplication by a scalar is commutative.

46a \langle Check identities 45a $\rangle + \equiv$ (50d) \langle 45h 46b \rangle
`test_out("S-mult is commutative: ", P*a-a*P);`

Uses `test_out` 34c 45a 45b 45e 54b.

Multiplication by a scalar is associative.

46b \langle Check identities 45a $\rangle + \equiv$ (50d) \langle 46a 46c \rangle
`test_out("S-mult is associative: ", b*P*a-a*P*b);`

Uses `test_out` 34c 45a 45b 45e 54b.

Distributive law $a(w_1 + w_2) = aw_1 + aw_2$, Prop. 4.

46c \langle Check identities 45a $\rangle + \equiv$ (50d) \langle 46b 46d \rangle
`test_out("S-mult is distributive 1: ", a*(P+P1)-(a*P + a*P1));`

Uses `test_out` 34c 45a 45b 45e 54b.

Distributive law $(a + b)w = aw + bw$, Prop. 4.

46d \langle Check identities 45a $\rangle + \equiv$ (50d) \langle 46c 46e \rangle
`test_out("S-mult is distributive 2: ", P*(a+b)-(P*a + P*b));`

Uses `test_out` 34c 45a 45b 45e 54b.

Product is commutative, Prop. 1.

46e \langle Check identities 45a $\rangle + \equiv$ (50d) \langle 46d 46f \rangle
`test_out("Product is symmetric (commutative): ", P*P1-P1*P);`

Uses `test_out` 34c 45a 45b 45e 54b.

Product is associative, Prop. 1.

46f \langle Check identities 45a $\rangle + \equiv$ (50d) \langle 46e 46g \rangle
`test_out("Prod is associative: ", (P*P1)*P2-P*(P1*P2));`

Uses `test_out` 34c 45a 45b 45e 54b.

Product and addition are distributive, Prop. 2.

46g \langle Check identities 45a $\rangle + \equiv$ (50d) \langle 46f \rangle
`test_out("Product is distributive: ", (P+P1)*P2-(P*P2+P1*P2));`

Uses `test_out` 34c 45a 45b 45e 54b.

C.2. Induced Representations. Here we calculate the basic formulae for Section 3.

C.2.1. *Encoded formulae.* This routine encodes the map $s : \mathbb{R}^2 \rightarrow \text{SL}_2(\mathbb{R})$ (9).

46h \langle Induced representations routines 46h $\rangle \equiv$ (48g) 47a \rangle
`ex s_map(const ex & u, const ex & v) {
 return matrix(2, 2, lst(v,u,0,1));
}

ex s_map(const ex & P) {
 if (P.nops() == 2)
 return s_map(P.op(0), P.op(1));
 cerr << "s_map() error: parameter should have two operands" << endl;
 return s_map(P,1);
}`

This routine encodes the map $r : \text{SL}_2(\mathbb{R}) \rightarrow \text{H}$ (12). The first parameter is an element of $\text{SL}_2(\mathbb{R})$, the second—is a generic element of subgroup H. We look specific matrix of the form K which makes the product MK belonging to the image of $s_map()$, i.e. its (2, 1) element should be zero.

47a \langle Induced representations routines 46h $\rangle + \equiv$ (48g) \langle 46h 47b \rangle

```

ex r_map(const ex & M, const ex & K) {
    ex K1=K.evalm(), K2;
    lst vars = (is_a<symbol>(K1.op(2)) ? lst(K1.op(2)) : lst(K1.op(1)));
    if (is_a<symbol>(K1.op(3))) {
        vars = vars.append(K1.op(3));
        K2 = K1.subs(lsolve(lst((M*K1).evalm().op(2) $\equiv$ 0), vars)).subs(K1.op(3) $\equiv$ 1);
    } else
        K2 = K1.subs(lsolve(lst((M*K1).evalm().op(2) $\equiv$ 0), vars));
    return pow(K2, -1).evalm();
}

```

This is the inverse s^{-1} of the above map s.

47b \langle Induced representations routines 46h $\rangle + \equiv$ (48g) \langle 47a 47c \rangle

```

ex p_map(const ex & M, const ex & K) {
    ex x = K.evalm().op(3);
    ex MK=(M*pow(r_map(M,K),-1)).evalm();
    ex D=MK.op(3).subs(x $\equiv$ 1).normal();
    return matrix(1, 2, lst((MK.op(1).subs(x $\equiv$ 1).normal() $\div$ D).normal(),
        (MK.op(0).subs(x $\equiv$ 1).normal() $\div$ D).normal()));
}

```

This is a matrix form of the above inverse map $p_map()$.

47c \langle Induced representations routines 46h $\rangle + \equiv$ (48g) \langle 47b \rangle

```

ex p_map_m(const ex & M, const ex & K) {
    return (M*pow(r_map(M,K),-1)).evalm();
}

```

C.2.2. *Calculation of induced representation formulae.* Firstly we define a generic element M of $\text{SL}_2(\mathbb{R})$.

47d \langle Induced representations 47d $\rangle \equiv$ (51b) 47e \rangle

```

ex M=matrix(2,2, lst(a,b,c,d)), H;

```

We consider the three cases.

47e \langle Induced representations 47d $\rangle + \equiv$ (51b) \langle 47d 47f \rangle

```

string cases[]={"Elliptic", "Parabolic ( $\mathbb{N}^{\prime}$ )", "Hyperbolic"};

```

In the those cases *subgroups* holds a generic element of a subgroup H, see (4), (6) and (7).

47f \langle Induced representations 47d $\rangle + \equiv$ (51b) \langle 47e 48a \rangle

```

ex subgroups=lst(matrix(2, 2, lst(x,-y,y,x)),
    matrix(2, 2, lst(1,0,y,1)),
    matrix(2, 2, lst(x,y,y,x)));

```

Now we run a cycle over the three cases...

48a (Induced representations 47d)+≡ (51b) <47f 48b>

```
for(int i=0; i<3; i++) {
  H=subgroups[i];
  cout << cases[i] << " case of induced representations\\\\" << endl;
  //formula_out("M*H: ", (M*H).evalm());
```

Uses formula_out 34b 43c 44g 44g 44g 44h 54a.

...and output expression of r (12),...

48b (Induced representations 47d)+≡ (51b) <48a 48c>

```
formula_out("map \\(r(M)\\): ", r_map(M,H));
```

Uses formula_out 34b 43c 44g 44g 44g 44g 44h 54a.

...matrix form of the inverse s^{-1} (12),...

48c (Induced representations 47d)+≡ (51b) <48b 48d>

```
formula_out("map \\(s^{-1}(M)\\): ", p_map_m(M,H).subs(a≡(1+b*c)÷d).normal());
```

Uses formula_out 34b 43c 44g 44g 44g 44g 44h 54a.

...expression for the argument of the character in (13),...

48d (Induced representations 47d)+≡ (51b) <48c 48e>

```
formula_out("character: ", r_map(M*s_map(P),H));
```

Uses formula_out 34b 43c 44g 44g 44g 44g 44h 54a.

...and finally the action (8) of $SL_2(\mathbb{R})$ on the homogeneous space.

48e (Induced representations 47d)+≡ (51b) <48d>

```
formula_out("Moebius map: ", p_map(M*s_map(P.to_matrix()),H));
test_out("Moebius map is given by the imaginary unit: ", p_map(M*s_map(P),H) -
  clifford_moebius_map(a*one, b*one, c*one, d*one,P.to_matrix(),e).subs(sign≡i-1));
cout << (latexout ? "\\vspace{2mm}\\hrule" :
  "-----" ) << endl;
}
```

Uses formula_out 34b 43c 44g 44g 44g 44g 44h 54a and test_out 34c 45a 45b 45e 54b.

C.3. **Program Outline.** Here is the outline how we use the above parts.

C.3.1. *Test program outline.* Firstly we load `dual_number` support.

48f (* 48f)≡ 48g>

```
#include <cycle.h>
#include <dualnum.h>
```

The rest of the program makes all checks.

48g (* 48f)+≡ <48f>

```
(Definition of variables 48h)
(Test routine 49g)
(Induced representations routines 46h)
(Main procedure 50e)
```

C.3.2. *Variables.* These **realsymbols** are used in our calculations.

48h (Definition of variables 48h)≡ (48g) 49a>

```
const numeric half(1,2);
```

Defines:

```
numeric, used in chunks 52 and 53.
```

Variables v_s often appear under square roots of the form $\sqrt{1+2v}$. To facilitate the simplifications of the type $(\sqrt{1+2v})^2 = 1+2v$ we (falsely) define them to be positive symbols.

49a \langle Definition of variables 48h $\rangle + \equiv$ (48g) \langle 48h 49b \rangle
`possymbol v("v"), v1("v'"), v2("v'');`

Other real variables.

49b \langle Definition of variables 48h $\rangle + \equiv$ (48g) \langle 49a 49c \rangle
`realsymbol u("u"), u1("u'"), u2("u'"),
a("a"), b("b"), c("c"), d("d"), x("x"), y("y"),`

Finally this variable keeps the signature of the metric space.

49c \langle Definition of variables 48h $\rangle + \equiv$ (48g) \langle 49b 49d \rangle
`sign("s", "\sigma");`

This an index used for the definition of Clifford units.

49d \langle Definition of variables 48h $\rangle + \equiv$ (48g) \langle 49c 49e \rangle
`varidx mu(symbol("mu", "\mu"), 1), nu(symbol("nu", "\nu"), 2);`

Three generic points which are used in calculations.

49e \langle Definition of variables 48h $\rangle + \equiv$ (48g) \langle 49d 49f \rangle
`dual_number P(u, v), P1(u1, v1), P2(u2, v2);`

Here we define a parabolic Clifford units $one, e0$.

49f \langle Definition of variables 48h $\rangle + \equiv$ (48g) \langle 49e \rangle
`ex e = clifford_unit(mu, diag_matrix(lst(sign))),
e0 = e.subs(mu=0),
one = dirac_ONE(),
e2 = clifford_unit(nu, diag_matrix(lst(-1, sign)));`

C.3.3. *Test routine.* This routine make the same sequence of checks for both cases of subgroups N and N' .

First we define the reference point $(u0, v0)$.

49g \langle Test routine 49g $\rangle \equiv$ (48g) 50a \rangle
`void parab_rot_sub(bool N, bool S) {
cout << "Calculations for subgroup \\ \langle N"
<< (N ? "" : "'") << "\\ \langle " and "
<< (S ? "straight" : "geodesics") << " spokes\\\\"<<endl;
set_subgroup(N);
set_straight_spoke(S);
ex X,
W0=dn_from_arg_mod(Arg0, 1),
u0=W0.op(0),
v0=W0.op(1),
P0=matrix(1, 2, lst(u0, v0));`

Defines:

`parab_rot_sub`, used in chunk 51a.

These two matrices define the Cayley transform and its inverse.

```
50a <Test routine 49g>+≡ (48g) <49g 50b>
    cout << "Reference point: \\" << P0 << "\\)\\" << endl;
    cout << "Reference point arg: \\" << ex_to<dual_number>(W0).arg() << "\\)\\" << endl;
    cout << "Reference point norm: \\" << ex_to<dual_number>(W0).norm() << "\\)\\" << endl;
```

For the subgroup N we consider upper-triangular matrices, for N'—lower-triangular.

```
50b <Test routine 49g>+≡ (48g) <50a 50c>
    if (subgroup_N)
        X=matrix(2, 2, lst(one, one*x, 0, one));
    else
        X=matrix(2, 2, lst(one, 0, one*x, one));
```

Two different types of Cayley transforms.

```
50c <Test routine 49g>+≡ (48g) <50b 50d>
    ex TC, TCI;
    if (S) {
        TC=matrix(2, 2, lst(one, -e0, -e0, one));
        TCI=matrix(2, 2, lst(one, e0, e0, one));
    } else{
        TC=matrix(2, 2, lst(one, -e0*half, -e0*half, one));
        TCI=matrix(2, 2, lst(one, e0*half, e0*half, one));
    }
```

Common part of test routine.

```
50d <Test routine 49g>+≡ (48g) <50c>
    <Show expressions 43a>
    <Check identities 45a>
    cout << (latexout ? "\\vspace{2mm}\\hrule" :
        "-----" ) << endl;
    }
```

C.3.4. *Main procedure.* It just calls the test routine, calculates the induced representation and draws a few pictures.

We output formulae in L^AT_EX mode.

```
50e <Main procedure 50e>≡ (48g) 50f>
    int main(){
        latexout=true;
```

Defines:

`main`, never used.

Preparation of output stream.

```
50f <Main procedure 50e>+≡ (48g) <50e 51a>
    cout << boolalpha;
    if (latexout)
        cout << latex;
    <Drawing pictures 51d>
```

Now we call the test routine for both N and N' subgroups.

```
51a <Main procedure 50e>+≡ (48g) <50f 51b>
    parab_rot_sub(true, true);
    parab_rot_sub(false, true);
    // parab_rot_sub(false, false); To work with geodesic spokes
```

Uses `parab_rot_sub` 49g.

Calculation of induced representations formulae.

```
51b <Main procedure 50e>+≡ (48g) <51a 51c>
    <Induced representations 47d>
```

And we finishing by drawing several pictures for Figs. 1 and 2.

```
51c <Main procedure 50e>+≡ (48g) <51b>
    }
```

C.4. **Drawing Orbits.** To draw cycles we use `cycle` library [28].
Elliptic orbits (circles).

```
51d <Drawing pictures 51d>≡ (50f) 51e>
    ofstream asymptote("parab-rot-data.asy");
    asymptote << ";//This file is automatically created from the code in parab-rotation.nw"
        << endl << endl
        << "path[] K=";
    for(int i=0; i<6; i++)
        cycle2D(lst(0,0),e2.subs(sign≡-1),i*i*.04)
            .asy_path(asymptote, -1.5, 1.5, -2, 2, 0, (i>0));
    asymptote << ";" << endl;
    asymptote << "path[] Kb=";
    cycle2D(lst(0,0),e2.subs(sign≡-1),1)
        .asy_path(asymptote, -1.5, 1.5, -2, 2, 0, false);
    asymptote << ";" << endl;
```

Hyperbolic orbits.

```
51e <Drawing pictures 51d>+≡ (50f) <51d 52a>
    asymptote << "path[] A=";
    for(int i=0; i<6; i++) {
        cycle2D(lst(0,0),e2.subs(sign≡1),-i*i*.04)
            .asy_path(asymptote, -1.75, 1.75, -1.5, 2, 0, (i>0));
        cycle2D(lst(0,0),e2.subs(sign≡1),i*i*.12)
            .asy_path(asymptote, -1.75, 1.75, -1.5, 2, 0, true);
    }
    asymptote << ";" << endl;
    asymptote << "path[] Ab=";
    cycle2D(lst(0,0),e2.subs(sign≡1),-1)
        .asy_path(asymptote, -1.75, 1.75, -1.5, 2, 0, false);
    asymptote << ";" << endl;
```

Hyperbolic orbits for reflected orbits.

52a (Drawing pictures 51d) $\dagger \equiv$ (50f) \triangleleft 51e 52b \triangleright

```

asymptote << "path[] At=";
for(int i=0; i<6; i++) {
  cycle2D(lst(0,0),e2.subs(sign=1),i*i*.04)
  .asy_path(asymptote, -1.75, 1.75, -1.5, 2, 0, (i>0));
  cycle2D(lst(0,0),e2.subs(sign=1),-i*i*.12)
  .asy_path(asymptote, -1.75, 1.75, -1.5, 2, 0, true);
}
asymptote << ";" << endl;
asymptote << "path[] Abt=";
cycle2D(lst(0,0),e2.subs(sign=1),1)
.asy_path(asymptote, -1.75, 1.75, -1.5, 2, 0, false);
asymptote << ";" << endl;

```

Parabolic orbits, subgroup N.

52b (Drawing pictures 51d) $\dagger \equiv$ (50f) \triangleleft 52a 52c \triangleright

```

asymptote << "path[] N=";
for(int i=0; i<6; i++)
  cycle2D(1,lst(0,numeric(1,2)),numeric(i,2)-1,e2.subs(sign=0))
  .asy_path(asymptote, -1.5, 1.5, -2, 2, 0, (i>0));
asymptote << ";" << endl;
asymptote << "path[] Nb=";
cycle2D(1,lst(0,numeric(1,2)),-1,e2.subs(sign=0))
.asy_path(asymptote, -1.5, 1.5, -2, 2, 0, false);
asymptote << ";" << endl;

```

Uses numeric 34a 48h.

Parabolic orbits, subgroup N'.

52c (Drawing pictures 51d) $\dagger \equiv$ (50f) \triangleleft 52b 52d \triangleright

```

asymptote << "path[] N1=";
for(int i=0; i<5; i++)
  cycle2D(.5*i*i+1,lst(0,numeric(1,2)),-1,e2.subs(sign=0))
  .asy_path(asymptote, -1.5, 1.5, -1.5, 2, 0, (i>0));
asymptote << ";" << endl;
asymptote << "path[] N1b=";
cycle2D(1,lst(0,numeric(1,2)),-1,e2.subs(sign=0))
.asy_path(asymptote, -1.5, 1.5, -1.5, 2, 0, false);
asymptote << ";" << endl;

```

Uses numeric 34a 48h.

Parabolic orbits, subgroup N (geodesic version?).

52d (Drawing pictures 51d) $\dagger \equiv$ (50f) \triangleleft 52c 53a \triangleright

```

asymptote << "path[] Ng=";
for(int i=0; i<6; i++)
  cycle2D(0.5,lst(0,numeric(1,2)),numeric(i,2)-.5,e2.subs(sign=0))
  .asy_path(asymptote, -1.5, 1.5, -2, 2, 0, (i>0));
asymptote << ";" << endl;
asymptote << "path[] Ngb=";
cycle2D(0.5,lst(0,numeric(1,2)),-.5,e2.subs(sign=0))
.asy_path(asymptote, -1.5, 1.5, -2, 2, 0, false);
asymptote << ";" << endl;

```

Uses numeric 34a 48h.

Parabolic orbits, subgroup N' (geodesic version?).

53a (Drawing pictures 51d) \equiv (50f) \triangleleft 52d 53b \triangleright

```

asymptote << "path[] N1g=";
for(int i=0; i<5; i++)
    cycle2D(.25*i*i+.5, lst(0, numeric(1,2)), -.5, e2.subs(sign=0))
        .asy_path(asymptote, -1.5, 1.5, -1.5, 2, 0, (i>0));
asymptote << ";" << endl;
asymptote << "path[] N1gb=";
cycle2D(.5, lst(0, numeric(1,2)), -.5, e2.subs(sign=0))
    .asy_path(asymptote, -1.5, 1.5, -1.5, 2, 0, false);
asymptote << ";" << endl;

```

Uses numeric 34a 48h.

Parabolic orbits, subgroup N' elliptic geodesics.

53b (Drawing pictures 51d) \equiv (50f) \triangleleft 53a

```

numeric L[9] = {-2, -1.4, -.8, -.4, 0, .4, .8, 1.2, 2}, low, high;

asymptote << "path[] N1geoE=";
for(int i=0; i<9; i++){
    if (abs(L[i]) < .5) {
        low = min((4*L[i]-2) ÷ (4*L[i]*L[i]-1), (4*L[i]+2) ÷ (4*L[i]*L[i]-1));
        high = max((4*L[i]-2) ÷ (4*L[i]*L[i]-1), (4*L[i]+2) ÷ (4*L[i]*L[i]-1));
    } else if (L[i] > .5) {
        low = numeric(-3, 2);
        high = min((4*L[i]-2) ÷ (4*L[i]*L[i]-1), (4*L[i]+2) ÷ (4*L[i]*L[i]-1));
    } else {
        low = max((4*L[i]-2) ÷ (4*L[i]*L[i]-1), (4*L[i]+2) ÷ (4*L[i]*L[i]-1));
        high = numeric(3, 2);
    }
    cycle2D(-1+4+4*L[i]*L[i], lst(4*L[i], 2), 0, e2.subs(sign=0))
        .asy_path(asymptote, max(numeric(-3, 2), low), min(numeric(3, 2), high), -1.5, 2, 0, (i>0));
}
asymptote << ";" << endl;

asymptote << "path[] N1geoEf=";
for(int i=0; i<9; i++){
    cycle2D(-1+4+4*L[i]*L[i], lst(4*L[i], 2), 0, e2.subs(sign=0))
        .asy_path(asymptote, -1.5, 1.5, -1.5, 2, 0, (i>0));
}
asymptote << ";" << endl;

asymptote.close();

```

Uses numeric 34a 48h.

C.4.1. *Output routines.* We use standardised routines to output results of calculations.

```

54a <Output routines 54a>≡ 54b▷
    void formula_out(string S, const ex & F, bool lineend) {
        cout << S << (latexout ? "\\(" : "(") << F << (latexout ? "\\)" : ")");
        if (lineend)
            cout << (latexout ? "\\\\" : "\\") << endl;
        else
            cout << "; ";
    }

```

Defines:

`formula_out`, used in chunks 35b, 43, 44, and 48.

This routine is used to check identities.

```

54b <Output routines 54a>+≡ <54a
    void test_out(string S, const ex & T) {
        cout << S << (latexout ? "\\textbf{" : "*" )
            << (is_a<dual_number>(T) ? ex_to<dual_number>(T).normal().is_zero() :
                T.evalm().normal().is_zero_matrix()) << (latexout ? "\\}" : "*" )
            << endl;
    }

```

Defines:

`test_out`, used in chunks 35b, 45, 46, and 48e.

SCHOOL OF MATHEMATICS, UNIVERSITY OF LEEDS, LEEDS LS2 9JT, UK

E-mail address: kisilv@maths.leeds.ac.uk

URL: <http://www.maths.leeds.ac.uk/~kisilv/>