■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:... Complex numbers, quaternions and more general "hyper-complex numbers" have also become relevant for the design of computational neural networks (CNNs). More specifically, concepts from geometric algebra can be adapted for the processing and analysis of "inherently geometrical data" as seen in 2D photographs, 2D video sequences or 3D volumetric object/material scans. (Image convolution is an operation originating in traditional digital image processing that has been applied successfully to multi-layer/multi-resolution CNN designs — to mention another example for employing a traditional geometry processing method to CNN-based data classification.) Therefore, we discuss some of the most important representations and algorithms used in geometric algebra at this point. A very good understanding of the approaches reviewed in the following is essential for their effective and efficient incorporation in CNNs, especially when the data to be analyzed and classified can be processed geometrically. In this context, the relationships between rotations and reflections, and corresponding representations via complex numbers, quaternions and so-called "hyper-complex numbers," are particularly relevant.

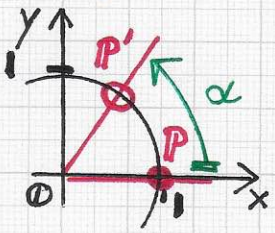
papersnake.com

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

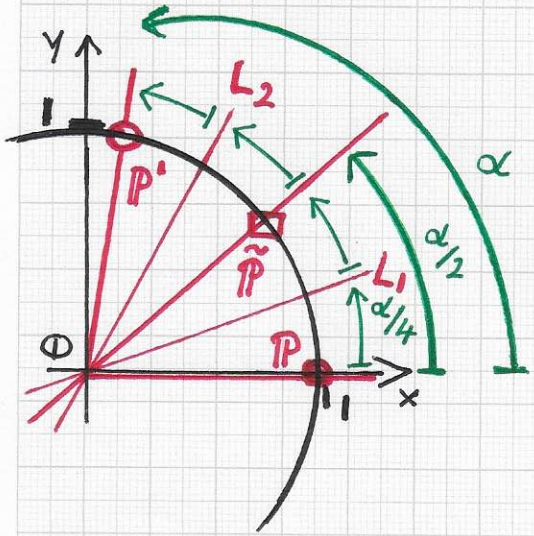
• Laplacian eigenfunctions and neural networks: ...

We review rotation and reflection in the 2D plane again, to fully appreciate fundamental derivations.

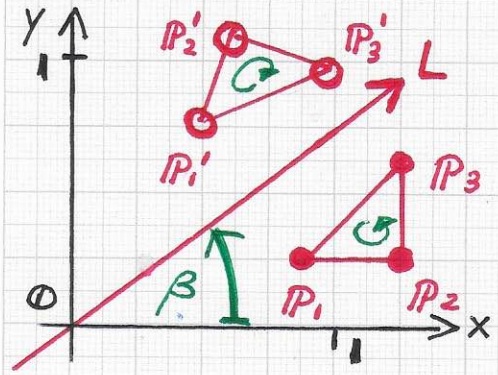


$$Rot(\alpha) = Rot\alpha = \begin{bmatrix} c\alpha & -s\alpha \\ s\alpha & c\alpha \end{bmatrix}$$

A rotation around the origin O maps the point P to P', where $P' = Rot\alpha \cdot P$, see left figure. The same effect can be achieved by a concatenation of two reflections, see left figure. Here, the point P is first reflected with respect to line L_1, yielding the "intermediate result" $\tilde{P}$ that is subsequently reflected with respect to line L_2, yielding the desired result point P'.



Before deriving the matrix representation for this concatenation of reflections, we must understand a single reflection with respect to a single line L that passes through the origin, where the (signed) angle between the x-axis unit basis vector and the (oriented) line L is β, see figure above. This general reflection can be accomplished by: rotating by -β; reflecting with respect to the x-axis; rotating by β. ...



■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ... We can multiply the associated transformation matrices of these three transformations to obtain the matrix for this general reflection. We call the matrix defining the reflection with respect to the line L "$Ref_L\beta$," where β is the signed angle between the oriented x-axis and oriented line L. Further, "Ref_x" is the special reflection matrix for a reflection with respect to the x-axis:

$$Ref_L\beta = Rot\beta \cdot Ref_x \cdot Rot(-\beta) = \begin{bmatrix} c\beta & -s\beta \\ s\beta & c\beta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} c-\beta & -s-\beta \\ s-\beta & c-\beta \end{bmatrix} =$$

$$= \begin{bmatrix} c\beta & s\beta \\ s\beta & -c\beta \end{bmatrix} \begin{bmatrix} c\beta & s\beta \\ -s\beta & c\beta \end{bmatrix} = \begin{bmatrix} c\beta c\beta - s\beta s\beta & c\beta s\beta + s\beta c\beta \\ s\beta c\beta + c\beta s\beta & s\beta s\beta - c\beta c\beta \end{bmatrix}$$

$$= \begin{bmatrix} c^2\beta - s^2\beta & 2s\beta c\beta \\ 2s\beta c\beta & -(c^2\beta - s^2\beta) \end{bmatrix} = \begin{bmatrix} c(2\beta) & s(2\beta) \\ s(2\beta) & -c(2\beta) \end{bmatrix}.$$

(• Note. Double-angle identities for the sin and cos functions are: $\sin(2\beta) = \sin\beta \cdot \cos\beta + \cos\beta \cdot \sin\beta$ and $\cos(2\beta) = \cos^2\beta - \sin^2\beta$. They have been used here.)

• Note. The determinant of this reflection matrix is $\det Ref_L\beta = -(c^2(2\beta) + s^2(2\beta)) = -1$. The value −1 implies that a (closed) polygon's orientation changes from clock-wise to counter-clockwise (or counter-clockwise to clock-wise), as indicated in the bottom figure on the previous page for the reflection of a triangle.

papersnake.com

• • •

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

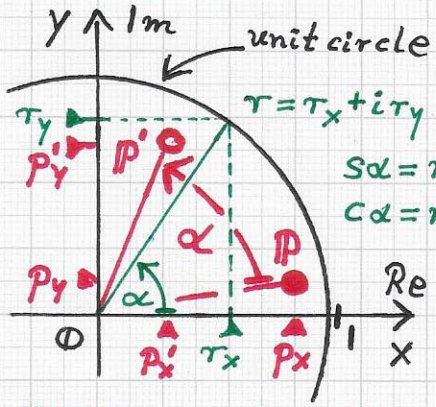
• Laplacian eigenfunctions and neural networks:... It is now possible to perform the rotation around the origin by the angle α (as depicted in the middle figure on page 7, 1-5-2024) via the concatenation of two reflections, with respect to line L_1 (first) and line L_2 (second). One obtains:

$$Ref_{L_2}\left(\tfrac{3}{4}\alpha\right)\cdot Ref_{L_1}\left(\tfrac{1}{4}\alpha\right)=\begin{bmatrix} c\left(\frac{3}{2}\alpha\right) & s\left(\frac{3}{2}\alpha\right)\\ s\left(\frac{3}{2}\alpha\right) & -c\left(\frac{3}{2}\alpha\right)\end{bmatrix}\begin{bmatrix} c\left(\frac{1}{2}\alpha\right) & s\left(\frac{1}{2}\alpha\right)\\ s\left(\frac{1}{2}\alpha\right) & -c\left(\frac{1}{2}\alpha\right)\end{bmatrix}=$$

$$=\begin{bmatrix} c\left(\frac{3}{2}\alpha\right)c\left(\frac{1}{2}\alpha\right)+s\left(\frac{3}{2}\alpha\right)s\left(\frac{1}{2}\alpha\right) & c\left(\frac{3}{2}\alpha\right)s\left(\frac{1}{2}\alpha\right)-s\left(\frac{3}{2}\alpha\right)c\left(\frac{1}{2}\alpha\right)\\ s\left(\frac{3}{2}\alpha\right)c\left(\frac{1}{2}\alpha\right)-c\left(\frac{3}{2}\alpha\right)s\left(\frac{1}{2}\alpha\right) & s\left(\frac{3}{2}\alpha\right)s\left(\frac{1}{2}\alpha\right)+c\left(\frac{3}{2}\alpha\right)c\left(\frac{1}{2}\alpha\right)\end{bmatrix}=$$

$$=\begin{bmatrix} c\left(\frac{3}{2}\alpha-\frac{1}{2}\alpha\right) & -s\left(\frac{3}{2}\alpha-\frac{1}{2}\alpha\right)\\ s\left(\frac{3}{2}\alpha-\frac{1}{2}\alpha\right) & c\left(\frac{3}{2}\alpha-\frac{1}{2}\alpha\right)\end{bmatrix}=\begin{bmatrix} c\alpha & -s\alpha\\ s\alpha & c\alpha\end{bmatrix}=Rot\,\alpha\,.$$

(• Note. Sum (difference) identities for the sin and cos functions are: $\sin(\alpha_1\pm\alpha_2)=s\alpha_1 c\alpha_2\pm c\alpha_1 s\alpha_2$ and $\cos(\alpha_1\pm\alpha_2)=c\alpha_1 c\alpha_2\mp s\alpha_1 s\alpha_2$. They have been used.)



ROTOR r: $r_x^2+r_y^2=1$. At this point, it is appropriate to recall the possible use of complex numbers (in the complex plane) to define and perform rotations around the origin via the multiplication of complex numbers – and avoiding the evaluation of trigonometric functions. The above figure shows the rotation of a point (complex number) P via the "rotor" (complex number) r, generating P'. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

(• Note. In the context of our driving applications, i.e., geometrical data processing for data analysis and classification, we need to understand the specific advantages of complex numbers, quaternions and "hyper-complex numbers" — making possible "simpler, more elegant, numerically robust and more efficient processing" of inherently geometric data.)

We now discuss the "rotor" example from the previous page in detail. The "rotor" is the complex number $r = r_x + i\,r_y$, with $r_x^2 + r_y^2 = 1$, defining a point on the unit circle in the complex plane. Thus, $r_x = \cos\alpha$, $r_y = \sin\alpha$, where the (signed) angle α is the angle between the basis vector of the (oriented) x-axis (Re-axis) and the positional vector of r in the plane. The "rotor r" acts as a "quasi-rotation operator" on $\mathbb{P}$ by multiplying r and $\mathbb{P}$:

$$r \cdot \mathbb{P} = (r_x + i r_y) \cdot (p_x + i p_y) = r_x p_x - r_y p_y + i(r_x p_y + r_y p_x)$$

$$\hat{=} \begin{bmatrix} r_x p_x - r_y p_y \\ r_x p_y + r_y p_x \end{bmatrix} = \begin{bmatrix} r_x & -r_y \\ r_y & r_x \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} c\alpha & -s\alpha \\ s\alpha & c\alpha \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} =$$

$$= \mathrm{Rot}\,\alpha \cdot \mathbb{P}. \Rightarrow \mathrm{Rot}\,\alpha \cdot \mathbb{P} \hat{=} r \cdot \mathbb{P}.$$

papersnake.com

Thus, a rotation $\mathrm{Rot}\alpha \cdot \mathbb{P}$, involving trigonometric functions, can be done via complex number multiplication.

...