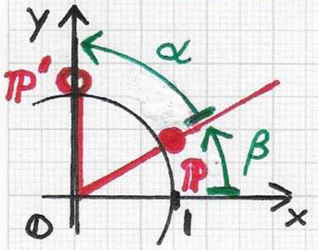
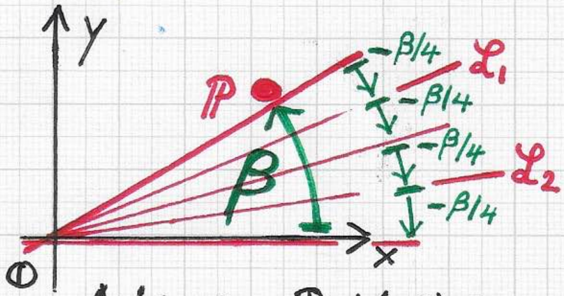
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ... For completeness, we briefly consider the general rotation of a point $\mathbb{P}$ around the origin in the 2D plane by an angle α — when the (signed) angle between the (oriented) x-axis and the positional vector of $\mathbb{P}$ is an angle $\beta \neq 0$. The goal is to perform this rotation via the concatenation of six reflections. For this purpose,



we perform a rotation by $-\beta$, done via two reflections; a rotation by α, done via two reflections; and a rotation by $+\beta$, done via two reflections.

Even though this sequence of transformations might seem to be "too complicated and redundant" to achieve the desired general rotation, this approach shows that the desired general rotation can be performed purely by concatenating six "elementary reflections" — as defined on page 8 (1-6-2024). By using the relationships



Achieving Rot(−β) via two reflections with respect to lines $\mathcal{L}_1$ and $\mathcal{L}_2$.

described on page 9 (1-6-2024) for rotations and reflections, we can perform Rot−β as

$$\mathrm{Rot}_{-\beta} = \mathrm{Ref}_{\mathcal{L}_2 \frac{1}{4}\beta} \cdot \mathrm{Ref}_{\mathcal{L}_1 \frac{3}{4}\beta} = \begin{bmatrix} c(\frac{1}{2}\beta) & s(\frac{1}{2}\beta) \\ s(\frac{1}{2}\beta) & -c(\frac{1}{2}\beta) \end{bmatrix} \cdot \begin{bmatrix} c(\frac{3}{2}\beta) & s(\frac{3}{2}\beta) \\ s(\frac{3}{2}\beta) & -c(\frac{3}{2}\beta) \end{bmatrix},$$

see left figures.

• • •

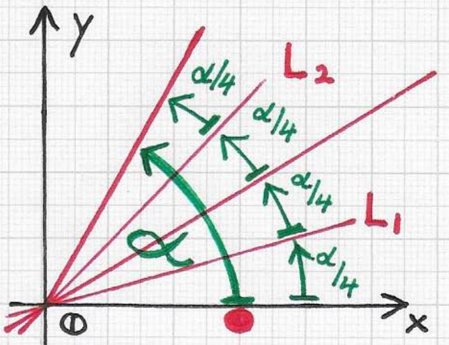
OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• **Laplacian eigenfunctions and neural networks:...** The multiplication of these two matrices yields the result

$$\begin{bmatrix} c(\tfrac{1}{2}\beta)c(\tfrac{3}{2}\beta)+s(\tfrac{1}{2}\beta)s(\tfrac{3}{2}\beta) & -(-c(\tfrac{1}{2}\beta)s(\tfrac{3}{2}\beta)+s(\tfrac{1}{2}\beta)c(\tfrac{1}{2}\beta)) \\ s(\tfrac{1}{2}\beta)c(\tfrac{3}{2}\beta)-c(\tfrac{1}{2}\beta)s(\tfrac{3}{2}\beta) & s(\tfrac{1}{2}\beta)s(\tfrac{3}{2}\beta)+c(\tfrac{1}{2}\beta)c(\tfrac{3}{2}\beta) \end{bmatrix} =$$

$$= \begin{bmatrix} c(\tfrac{1}{2}\beta-\tfrac{3}{2}\beta) & -s(\tfrac{1}{2}\beta-\tfrac{3}{2}\beta) \\ s(\tfrac{1}{2}\beta-\tfrac{3}{2}\beta) & c(\tfrac{1}{2}\beta-\tfrac{3}{2}\beta) \end{bmatrix} = \begin{bmatrix} c-\beta & -s-\beta \\ s-\beta & c-\beta \end{bmatrix} = Rot\,-\beta,$$

using the trigonometric identities on page 9 (1-6-2024). Next, one can perform the "elementary rotation," as described on page 9 (1-6-2024). One obtains

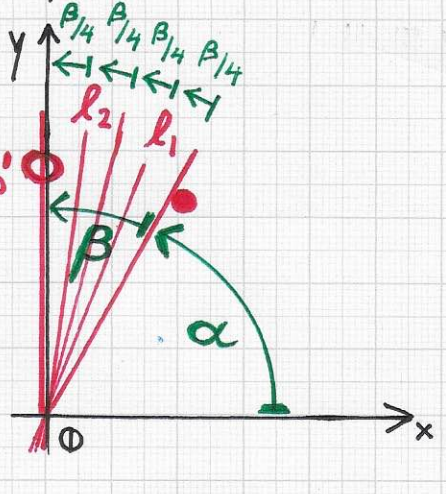


$$Rot\,\alpha = Ref_{L_2}\tfrac{3}{4}\alpha \cdot Ref_{L_1}\tfrac{1}{4}\alpha =$$

$$= \begin{bmatrix} c(\tfrac{3}{2}\alpha) & s(\tfrac{3}{2}\alpha) \\ s(\tfrac{3}{2}\alpha) & -c(\tfrac{3}{2}\alpha) \end{bmatrix} \cdot \begin{bmatrix} c(\tfrac{1}{2}\alpha) & s(\tfrac{1}{2}\alpha) \\ s(\tfrac{1}{2}\alpha) & -c(\tfrac{1}{2}\alpha) \end{bmatrix} = \ldots$$

$$= \ldots = \begin{bmatrix} c(\tfrac{3}{2}\alpha-\tfrac{1}{2}\alpha) & -s(\tfrac{3}{2}\alpha-\tfrac{1}{2}\alpha) \\ s(\tfrac{3}{2}\alpha-\tfrac{1}{2}\alpha) & c(\tfrac{3}{2}\alpha-\tfrac{1}{2}\alpha) \end{bmatrix} = \begin{bmatrix} c\alpha & -s\alpha \\ s\alpha & c\alpha \end{bmatrix} =$$

$= Rot\,\alpha$, see left figure.



Finally, one executes the rotation $Rot\,\beta$ via two reflections with respect to the lines ℓ_1 and ℓ_2, see left figure. Thus,

$$Rot\,\beta = Ref_{\ell_2}(\alpha+\tfrac{3}{4}\beta)\cdot Ref_{\ell_1}(\alpha+\tfrac{1}{4}\beta) =$$

$$= \begin{bmatrix} c(2\alpha+\tfrac{3}{2}\beta) & s(2\alpha+\tfrac{3}{2}\beta) \\ s(2\alpha+\tfrac{3}{2}\beta) & -c(2\alpha+\tfrac{3}{2}\beta) \end{bmatrix} \cdot \begin{bmatrix} c(2\alpha+\tfrac{1}{2}\beta) & s(2\alpha+\tfrac{1}{2}\beta) \\ s(2\alpha+\tfrac{1}{2}\beta) & -c(2\alpha+\tfrac{1}{2}\beta) \end{bmatrix} =$$

$$= \ldots = \begin{bmatrix} c\beta & -s\beta \\ s\beta & c\beta \end{bmatrix} = Rot\,\beta.$$

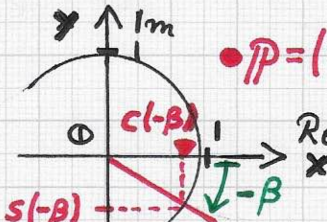
Thus, the general rotation $Rot\,\alpha = Rot\,\beta \cdot Rot\,\alpha \cdot Rot\,-\beta$ equals

$$Rot\,\alpha = Ref_{\ell_2}(\alpha+\tfrac{3}{4}\beta)\cdot Ref_{\ell_1}(\alpha+\tfrac{1}{4}\beta)\cdot Ref_{L_2}(\tfrac{3}{4}\alpha)\cdot Ref_{L_1}(\tfrac{1}{4}\alpha)\cdot$$
$$\cdot Ref_{\mathcal{L}_2}(\tfrac{1}{4}\beta)\cdot Ref_{\mathcal{L}_1}(\tfrac{3}{4}\beta).$$ ...

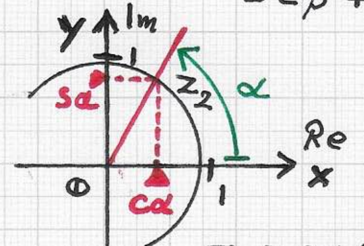
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... It is possible to derive the same result for the general rotation via the use of complex numbers (and "rotors") to perform the three rotations $Rot{-\beta}$, $Rot\,\alpha$ and $Rot\,\beta$ by multiplication. We rotate the point $\mathbb{P} = (x, y)^T$ – written in complex representation as $p = x + iy$ – by the rotation matrix $Rot\,\alpha = Rot\,\beta \cdot Rot\,\alpha \cdot Rot{-\beta}$. (Again, this concatenation involves redundant terms and operations. The purpose of this example is to explain the general principles defining the relationship between geometric transformations and complex numbers.) The left figures show the three "rotors", i.e., the three (unit) complex numbers z_1, z_2 and z_3, that act on the given point / complex number p.

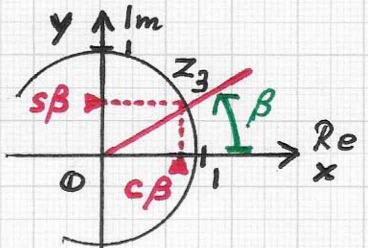
• $\mathbb{P} = \binom{x}{y} \triangleq p = x + iy$



$z_1 = c(-\beta) + i\,s(-\beta) = c\beta + i(-s\beta)$



$z_2 = c\alpha + i s\alpha$



$z_3 = c\beta + i s\beta$

"Rotors" z_1, z_2 and z_3 used to perform the concatenation $Rot\,\alpha = Rot\,\beta \cdot Rot\,\alpha \cdot Rot{-\beta}$.

! Thus, one must show that $Rot\,\beta \cdot Rot\,\alpha \cdot Rot{-\beta} \cdot \mathbb{P}$ is equivalent to the product $z_3 \cdot (z_2 \cdot (z_1 \cdot p))$.

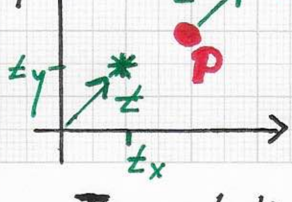
...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ... To demonstrate the equivalence of these products, we perform the necessare multiplications of the complex numbers:

$$
\begin{aligned}
Rot\,\alpha\cdot \mathbb{p} \mathrel{\hat{=}} z_3\cdot(z_2\cdot(z_1\cdot p)) &= z_3\cdot(z_2\cdot([c\beta + i(-s\beta)]\cdot[x+iy])) = \\
&= z_3\cdot(z_2\cdot([xc\beta + ys\beta] + i[yc\beta - xs\beta])) = \\
&= z_3\cdot((c\alpha + is\alpha)\cdot([xc\beta + ys\beta] + i[yc\beta - xs\beta])) = \\
&= z_3\cdot([xc\alpha c\beta + yc\alpha s\beta - ys\alpha c\beta + xs\alpha s\beta] \\
&\qquad + i[xs\alpha c\beta + ys\alpha s\beta + yc\alpha c\beta - xc\alpha s\beta]) = \\
&= (c\beta + is\beta)\cdot([x(c\alpha c\beta + s\alpha s\beta) + y(c\alpha s\beta - s\alpha c\beta)] \\
&\qquad + i[x(s\alpha c\beta - c\alpha s\beta) + y(s\alpha s\beta + c\alpha c\beta)]) = \\
&= (c\beta + is\beta)\cdot([x\,c(\alpha-\beta) - y\,s(\alpha-\beta)] \\
&\qquad + i[x\,s(\alpha-\beta) + y\,c(\alpha-\beta)]) = \\
&= [xc\beta c(\alpha-\beta) - yc\beta s(\alpha-\beta) - xs\beta s(\alpha-\beta) - ys\beta c(\alpha-\beta)] \\
&\quad + i[xs\beta c(\alpha-\beta) - ys\beta s(\alpha-\beta) + xc\beta s(\alpha-\beta) + yc\beta c(\alpha-\beta)] \\
&= [x(c\beta c(\alpha-\beta) - s\beta s(\alpha-\beta)) - y(c\beta s(\alpha-\beta) + s\beta c(\alpha-\beta))] \\
&\quad + i[x(s\beta c(\alpha-\beta) + c\beta s(\alpha-\beta)) + y(-s\beta s(\alpha-\beta) + c\beta c(\alpha-\beta))] \\
&= [x\,c\alpha - y\,s\alpha] + i[x\,s\alpha + y\,c\alpha] \\
&= (c\alpha + is\alpha)\cdot(x+iy) = z_2\cdot p \mathrel{\hat{=}} Rot\,\alpha\cdot\mathbb{p}.
\end{aligned}
$$

Following the general idea of performing geometrical transformations by equivalent algebraic operations applied to complex numbers, we briefly consider translation and scaling, see left figures.



Translation

Scaling.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks:... With the use of complex numbers (in the 2D complex plane), we can perform these two transformations by simply adding a complex number ($\hat{=}$ translation vector) to the complex number $p = x + iy$, representing the point $\mathbb{p} = (x, y)^T$, or by multiplying p by a real number ($\hat{=}$ scaling factor). The complex number used for translation is $t = t_x + it_y$, and the real number for scaling – written as a complex number – is $s = s + i0$. Thus, the equivalent transformations/operations are:

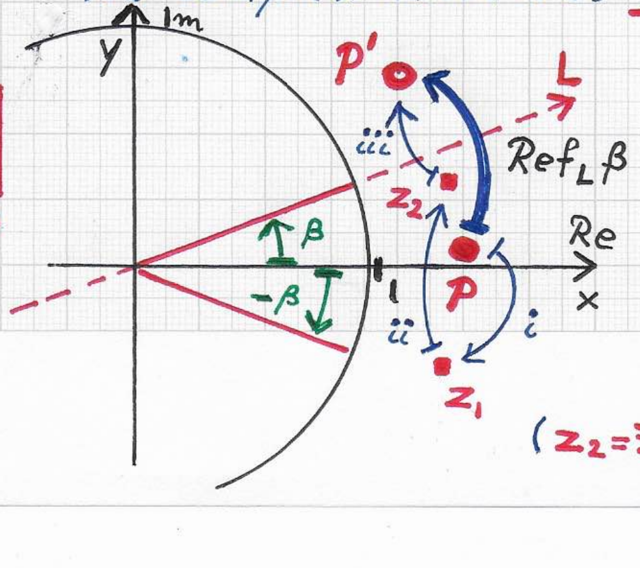
- Translation – by $\mathbb{t} = (t_x, t_y)^T \hat{=} t_x + it_y$

$$\mathbb{p}' = (x', y')^T = \mathbb{p} + \mathbb{t} = (x, y)^T + (t_x, t_y)^T = (x + t_x, y + t_y)^T$$
$$\hat{=}\ p' = x' + iy' = p + t = (x + iy) + (t_x + it_y) = (x + t_x) + i(y + t_y)$$

- Scaling – by $s \hat{=} s + i \cdot 0$

$$\mathbb{p}' = (x', y')^T = s \cdot \mathbb{p} = s \cdot (x, y)^T = (sx, sy)^T$$
$$\hat{=}\ p' = x' + iy' = s \cdot p = (s + i \cdot 0) \cdot (x + iy) = (sx + isy)$$

The remaining transformation to be represented via complex numbers is reflection, with respect to



an (oriented) Line L, having an angle β with the (oriented) Re-axis (x-axis), see left figure. The reflection maps p to p', done by i) mapping p to z_1 (Rot $-\beta$); ii) mapping z_1 to z_2 ($z_2 = \bar{z}_1$); and iii) mapping z_2 to p' (Rot β). ...