Exprimiendo el silicio [opcional]**.** El siguiente código implementa en `sagemath` la función `proj_Wv` que calcula la proyección ortogonal sobre un subespacio a partir de una base suya utilizando la fórmula anterior. Su uso está ilustrado con dos de los ejemplos anteriores.

```
def proj_Wv(S,v):
        G = S.conjugate_transpose()*S
        M = S*G.inverse()*S.conjugate_transpose()
        return M*v

# S = Base de W en columna
# v = vector a proyectar

S = matrix(4,2,[-1,2,0,1,1,0,0,1])
v = vector([4,8,-4,12])
print(proj_Wv(S,v))
print('----------------')

S = matrix(2,1,[3/2+3/2*I, 1])
v = vector([2+I, 1+7*I])
print(proj_Wv(S,v))
```

La salida es `(9, 5, 1, 5)` y `(3*I, I + 1)`, en consonancia con los resultados obtenidos.

Ya puestos a la solución automática de los ejemplos, la siguiente variante `proj_Wv2` (que llama a `proj_Wv`) permite introducir las ecuaciones del subespacio en vez de una base suya.

```
def proj_Wv2(A, v):
        W = A.right_kernel()
        S = matrix([item for item in W.basis()]).transpose()
        return proj_Wv(S,v)

# A = matrix con las ecuaciones de W
# v = vector a proyectar

A = matrix(QQ,2,4,[1,-2,1,0, 1,-3,1,1])
v = vector([4,8,-4,12])
print(proj_Wv2(A,v))
print('----------------')

A = matrix(1,2,[1+I, -3*I])
v = vector([2+I, 1+7*I])
print(proj_Wv2(A,v))
print('----------------')
```

4.4. Matrices ortogonales y unitarias

Una pregunta recurrente en matemáticas es el tipo de funciones que preservan ciertas cantidades o estructuras. En este sentido, a pesar de que no ha sido exactamente nuestro enfoque, las aplicaciones lineales son las funciones que preservan la estructura de espacio vectorial.

En esta línea e incluso dentro de la geometría básica, cabe preguntarse qué funciones preservan las distancias en el plano o el espacio. Claramente los giros lo hacen, la distancia de Barcelona a Madrid no cambia a lo largo del día con la rotación de la Tierra. Un prurito matemático lleva a plantearse una demostración que determine cuáles son exactamente. Aquí restringiremos nuestro análisis a las aplicaciones linea-

les, a cambio no nos quedaremos en el plano o el espacio sino que estudiaremos la situación en $\mathbb{R}^n$ y en $\mathbb{C}^n$ para cualquier dimensión.

Proposición 4.4.1. *Sea $f : K^n \longrightarrow K^n$ con $K = \mathbb{R}, \mathbb{C}$ una aplicación lineal que preserva las longitudes, es decir, tal que $\|f(\vec{x})\| = \|\vec{x}\|$ para todo $\vec{x} \in K^n$, entonces f también preserva el producto escalar, es decir $f(\vec{x}) \cdot f(\vec{y}) = \vec{x} \cdot \vec{y}$ para todo $\vec{x} \in K^n$.*

Demostración. Por hipótesis y por la linealidad de f

$$\|\vec{x} + \vec{y}\|^2 = \|f(\vec{x} + \vec{y})\|^2 = \|f(\vec{x}) + f(\vec{y})\|^2.$$

Utilizando que $\|\vec{z}\|^2 = \vec{z} \cdot \vec{z}$ al desarrollar se sigue

$$\|\vec{x}\|^2 + \|\vec{y}\|^2 + 2\vec{x} \cdot \vec{y} = \|f(\vec{x})\|^2 + \|f(\vec{y})\|^2 + 2f(\vec{x}) \cdot f(\vec{y}).$$

Los dos primeros sumandos de ambos miembros se cancelan porque f preserva las distancias y se obtiene el resultado buscado. □

En términos geométricos se puede interpretar este resultado a la luz de (4.1) diciendo que preservar longitudes implica preservar ángulos, algo que no es tan intuitivo.

Ya habíamos visto (Proposición 2.3.1) que en K^n todas las aplicaciones lineales venían dadas por multiplicar por una matriz. En nuestro caso, tratamos con endomorfismos y esta matriz es necesariamente cuadrada. El siguiente resultado resuelve totalmente el problema que nos habíamos planteado.

Proposición 4.4.2. *Un endomorfismo $f : K^n \longrightarrow K^n$ con $K = \mathbb{R}$ o $\mathbb{C}$ dado por $f(\vec{x}) = A\vec{x}$ preserva el producto escalar usual en K^n si y solo si*

$$1)\ A^tA = I \quad \textit{cuando } K = \mathbb{R}, \qquad 2)\ A^\dagger A = I \quad \textit{cuando } K = \mathbb{C}.$$

Multiplicando a la derecha por A^{-1} y a la derecha por A, estas condiciones equivalen a $AA^t = I$ y $AA^\dagger = I$. Ambas condiciones en realidad se pueden resumir en $A^\dagger A = I$ (o $AA^\dagger = I$) porque $A^\dagger = A^t$ cuando $K = \mathbb{R}$.

Demostración. Basta considerar el caso $K = \mathbb{C}$. Pensando los vectores como matrices columna, el producto escalar usual es $\vec{x} \cdot \vec{y} = \vec{x}^t\overline{\vec{y}}$ y se tiene $f(\vec{x}) \cdot f(\vec{y}) = \vec{x} \cdot \vec{y}$ si y solo si $\vec{x}^t A^t \overline{A}\,\overline{\vec{y}} = \vec{x}^t I \overline{\vec{y}}$. Al ser $\vec{x}$ e $\vec{y}$ arbitrarios esto equivale a $A^t\overline{A} = I$ y conjugando a $A^\dagger A = I$. □

Las matrices $A \in \mathcal{M}_n(\mathbb{R})$ que cumplen lo primero se llaman *matrices ortogonales* y las matrices $A \in \mathcal{M}_n(\mathbb{C})$ que cumplen lo segundo se llaman *matrices unitarias*. Se mire como se mire, los nombres no son muy afortunados pero están demasiado asentados como para tomar iniciativas en su contra. Quizá sería más adecuado llamar a ambas "matrices ortonormales" gracias al siguiente resultado que es una mera observación si uno tiene claro el procedimiento para multiplicar matrices.

Proposición 4.4.3. *Una matriz $A \in \mathcal{M}_n(K)$ es ortogonal (cuando $K = \mathbb{R}$) o unitaria (cuando $K = \mathbb{C}$) si y solo si sus columnas forman una base ortonormal de K^n.*

Demostración. De nuevo, basta considerar el caso $K = \mathbb{C}$, y las condiciones para ser unitaria son $\sum_k b_{ik}a_{kj} = \delta_{ij}$ con δ_{ij} los elementos de la matriz identidad y $b_{ik} = \overline{a}_{ki}$. La igualdad con el sumatorio significa que los vectores $(a_{ki})_{k=1}^n$ y $(a_{kj})_{k=1}^n$ son ortogonales para $i \neq j$ y con $i = j$ tenemos un vector unitario. Estos vectores son las columnas i-ésima y j-ésima de A. □

Antes de seguir, aquí van algunos ejemplos 2×2 de matrices ortogonales y unitarias:

$$\begin{pmatrix} 3/5 & 4/5 \\ -4/5 & 3/5 \end{pmatrix}, \qquad \frac{1}{13}\begin{pmatrix} 5 & 12 \\ 12 & -5 \end{pmatrix}, \qquad \frac{1}{2}\begin{pmatrix} \sqrt{2}+i & -i \\ i & -\sqrt{2}+i \end{pmatrix}, \qquad \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$

Las dos primeras son ortogonales si las consideramos en $A \in \mathcal{M}_n(\mathbb{R})$ y unitarias si las consideramos en $A \in \mathcal{M}_n(\mathbb{C})$. Las dos últimas son unitarias y no tiene sentido plantearse si son ortogonales porque no son matrices reales. El cálculo detallado de que la tercera matriz es unitaria sería:

$$A^\dagger A = \frac{1}{4}\begin{pmatrix} \sqrt{2}-i & -i \\ i & -\sqrt{2}-i \end{pmatrix}\begin{pmatrix} \sqrt{2}+i & -i \\ i & -\sqrt{2}+i \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 3+1 & 0 \\ 0 & 3+1 \end{pmatrix} = I_2.$$

Gracias a la Proposición 4.4.3, con cualquiera de las bases ortonormales de ejemplos anteriores obtendremos matrices ortogonales y unitarias. Recuperando dos de ellos, de las siguientes matrices la primera es ortogonal ($K = \mathbb{R}$) y unitaria ($K = \mathbb{C}$) y la segunda es unitaria.

$$\frac{1}{9}\begin{pmatrix} 8 & 4 & 1 \\ 4 & -7 & -4 \\ 1 & -4 & 8 \end{pmatrix}, \qquad \frac{1}{9}\begin{pmatrix} 4+3i & 4 & -2+6i \\ 4 & 4-3i & -2-6i \\ -2+6i & -2-6i & 1 \end{pmatrix}$$

Ambas son además simétricas pero eso es casualidad[6], rompemos tal simetría intercambiando dos filas sin violar la propiedad de ser ortogonal o unitaria.

No es difícil parametrizar todas las matrices 2×2 ortogonales y unitarias. Veámoslo en el caso ortogonal con la excusa de entender sus significado geométrico.

Proposición 4.4.4. *Todas las matrices $A \in \mathcal{M}_2(\mathbb{R})$ ortogonales son de la forma*

$$A = \begin{pmatrix} \cos\alpha & -\operatorname{sen}\alpha \\ \operatorname{sen}\alpha & \cos\alpha \end{pmatrix} \qquad \textit{o de la forma} \quad A = \begin{pmatrix} \cos\alpha & \operatorname{sen}\alpha \\ \operatorname{sen}\alpha & -\cos\alpha \end{pmatrix}.$$

En el primer caso la aplicación lineal $f(\vec{x}) = A\vec{x}$ corresponde a un giro de ángulo α alrededor del origen y en el segundo a una simetría por la recta que forma un ángulo $\alpha/2$ con el eje X.

[6] En realidad no tanto: hay un truco para inventarse rápidamente bases ortonormales con números exactos y siempre da matrices simétricas. Este truco consiste en tomar un vector $\vec{u}$ con $\|\vec{u}\|^2 = 2$ y considerar las columnas de la llamada *matriz de Householder* $I - \vec{u}\vec{u}^t$.

Demostración. Por la Proposición 4.4.3, las columnas de A son vectores unitarios así que existen α y β tales que $a_{11} = \cos\alpha$, $a_{21} = \operatorname{sen}\alpha$ y $a_{12} = \cos\beta$, $a_{22} = \operatorname{sen}\beta$. La ortogonalidad de ambas columnas implica

$$0 = \cos\alpha\cos\beta + \operatorname{sen}\alpha\operatorname{sen}\beta = \cos(\alpha - \beta).$$

Entonces salvo múltiplos enteros de 2π se tiene $\alpha - \beta = -\pi/2$ o $\alpha - \beta = \pi/2$. Despejando β se obtienen ambos tipos de matrices.

Para estudiar el significado geométrico de f analizamos su efecto sobre una base. Para el primer tipo, notamos que $f(\vec{e}_1)$ y $f(\vec{e}_2)$ son los vectores de la base canónica $\{\vec{e}_1, \vec{e}_2\}$ girados un ángulo α. Por otro lado, usando la base ortonormal $\{\vec{u}_1, \vec{u}_2\}$ con $\vec{u}_1 = \big(\cos(\alpha/2), \operatorname{sen}(\alpha/2)\big)^t$, que es vector director de la recta del enunciado, y $\vec{u}_2 = \big(\operatorname{sen}(\alpha/2), -\cos(\alpha/2)\big)^t$, que es vector normal, se tiene $f(\vec{u}_1) = \vec{u}_1$ y $f(\vec{u}_2) = -\vec{u}_2$. Así pues actúa como la simetría mencionada sobre esta base. □

La segunda parte de la proposición anterior es un primer paso para resolver el problema planteado al inicio de la sección de determinar todas las funciones que preservan las distancias en el plano y en el espacio. Estas funciones se llaman *movimientos*. En el plano están las traslaciones, los giros por un punto (no necesariamente el origen), las simetrías y las simetrías deslizantes (aplicar una simetría y después trasladar la imagen en la dirección de su eje). En el espacio es un poco más complicado [13] pero en definitiva todo lo que hay son giros, simetrías, traslaciones y combinaciones de ellos.

Exprimiendo el silicio [opcional]**.** Gracias a la Proposición 4.4.3 las matrices ortogonales y unitarias están estrechamente ligadas a las bases ortonormales. El siguiente código muestra cómo generar matrices ortogonales y unitarias con `matlab/octave` ortonormalizando las columnas de matrices aleatorias reales o complejas mediante `orth`.

```
1  N = 3
2  % Matriz aleatoria NxN
3  A = rand(N);
4
5  % Matriz ortogonal
6  M = orth(A)
7
8  % Matriz aleatoria compleja
9  A = A + i*rand(N);
10
11 % Matriz unitaria
12 U = orth(A)
```

Se cumple que `M*M'` y `U*U'` son aproximaciones de la identidad.

Las cinco primeras líneas del siguiente código `sagemath` crean una matriz ortogonal partiendo de un vector arbitrario $\vec{v}$, usando el truco de las *matrices de Householder* que consiste en que si $\|\vec{u}\|^2 = 2$ entonces $I - \vec{u}\vec{u}^t$ es ortogonal y simétrica. La línea 4 toma $\vec{u} = \sqrt{2}\vec{v}/\|\vec{v}\|$ de una manera enrevesada para forzar el cálculo simbólico. Con el vector de partida indicado, se obtiene uno de los ejemplos que habíamos dado.

```
1  # Vector arbitrario
2  v = matrix([1,-4,-1])
```

```
3  N = v.ncols()
4  v *= sqrt( 2/(v*v.transpose())[0][0] )
5  H = identity_matrix(N)-v.transpose()*v
6
7  # Matriz ortogonal (de Householder) correspondiente
8  print(H)
9  print('--------------------')
10
11 # Cambios arbitrarios de filas
12 for k in range(N):
13         H.swap_rows(randint(0,N-1),randint(0,N-1))
14 print(H)
15 print('--------------------')
16
17 # ortogonal*unitaria*ortogonal = unitaria
18 D = diagonal_matrix( [I^randint(0,3) for _ in range(N)])
19 U = H*D*H
20 print(U)
```

El resto del código permuta algunas filas para romper la simetría (líneas 12–13) y genera una matriz unitaria usando que el producto de matrices unitarias es unitaria y que una ortogonal es unitaria. Concretamente se genera una matriz diagonal unitaria aleatoria muy sencilla y se multiplica por delante y por detrás por la matriz ortogonal antes hallada (líneas 18–19).

Si efectuamos `H*H.transpose()` y `U*U.conjugate_transpose()` obtendremos matrices identidad.