T2.0 Recordatorio de algebra lineal (opcional)

Índice

1 Notación

1.1 Vectores

Vector $\boldsymbol{x} \in \mathbb{R}^n$: suponemos que es vector columna

$$\boldsymbol{x} = (x_1, x_2, \dots, x_n)^t$$

Vector de todo unos y todo ceros: $\mathbf{1}$ $\mathbf{0}$

Vector unitario $\boldsymbol{e}_i$ o one-hot: todo ceros salvo un uno en la posición i

$$\boldsymbol{e}_i = (0, \dots, 0, 1, 0, \dots, 0)^t$$

1.2 Matrices

Matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$: **cuadrada** si $m = n$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Entrada de la i-ésima fila y j-ésima columna: A_{ij} o $A_{i,j}$

Fila i-ésima y columna j-ésima: $\mathbf{A}_{i,:}$ y $\mathbf{A}_{:,j}$:

Matriz vista por filas: $\mathbf{A}=[\mathbf{A}_{1,:};\ \mathbf{A}_{2,:};\ \ldots;\ \mathbf{A}_{n,:}]$

Matriz vista por columnas: $\mathbf{A}=[\mathbf{A}_{:,1},\ \mathbf{A}_{:,2},\ \ldots,\ \mathbf{A}_{:,n}]$

Transpuesta de $\mathbf{A}\in\mathbb{R}^{m\times n}$: $\mathbf{A}^t\in\mathbb{R}^{n\times m}$ tal que $(\mathbf{A}^t)_{ij}=A_{ji}$, cumple

$$(\mathbf{A}^t)^t=\mathbf{A}$$
$$(\mathbf{A}\mathbf{B})^t=\mathbf{B}^t\mathbf{A}^t$$
$$(\mathbf{A}+\mathbf{B})^t=\mathbf{A}^t+\mathbf{B}^t$$

Matriz simétrica: matriz cuadrada $\mathbf{A}$ tal que $\mathbf{A}=\mathbf{A}^t$; $\mathbb{S}^n$ denota todas las matrices simétricas de talla n

Reformato de matriz a vector: $\operatorname{vec}(\mathbf{A})=[\mathbf{A}_{:,1};\ \mathbf{A}_{:,2};\ \cdots;\ \mathbf{A}_{:,n}]\in\mathbf{R}^{mn\times 1}$

Reformato de vector a matriz: puede ser por filas (python, C++) o por columnas (julia, octave, R, fortran)

1.3 Tensores

Tensor: generalización de matriz a tres o más dimensiones

Orden o **rango:** dimensiones (concepto distinto en matrices)

Ejemplo: vector 8-dimensional reformatado como matriz y tensor

In [1]:
```
import numpy as np
a = np.arange(8)
print(a, a.shape)
a2d = a.reshape([2, -1])
print(a2d, a2d.shape)
a3d = a.reshape([2, 2, -1])
print(a3d, a3d.shape)
```

```
[0 1 2 3 4 5 6 7] (8,)
[[0 1 2 3]
 [4 5 6 7]] (2, 4)
[[[0 1]
  [2 3]]

 [[4 5]
  [6 7]]] (2, 2, 2)
```

2 Espacios vectoriales

2.1 Suma y escalado de vectores

Espacio vectorial: conjunto de **vectores** que pueden sumarse entre ellos y escalarse (multiplicarse) por **escalares**

2.2 Independencia lineal, span y base

Independencia lineal: $\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$ es **linealmente independiente** si ninguno puede representarse como combinación lineal del resto

Span de $\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}$: $\quad\operatorname{span}(\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\})=\left\{\boldsymbol{v}:\boldsymbol{v}=\sum_{i=1}^n\alpha_i\boldsymbol{x}_i,\;\alpha_i\in\mathbb{R}\right\}$

Base $\mathcal{B}$: $\quad\operatorname{span}(\mathcal{B})=\mathbb{R}^n$

Base estándar o canónica: $\quad\mathcal{B}=\{\boldsymbol{e}_1=(1,0,\dots,0)^t,\dots,\boldsymbol{e}_n=(0,0,\dots,1)^t\}$

2.3 Transformaciones lineales y matrices

Transformación lineal: $f:\mathcal{V}\to\mathcal{W}$ tal que $f(\boldsymbol{v}+\boldsymbol{w})=f(\boldsymbol{v})+f(\boldsymbol{w})$ y $f(a\boldsymbol{v})=af(\boldsymbol{v})$ para todo $\boldsymbol{v},\boldsymbol{w}\in\mathcal{V}$

Representación matricial: $f:\mathbb{R}^n\to\mathbb{R}^m$ con $\mathbf{A}\in\mathbb{R}^{m\times n}$ tal que $\boldsymbol{y}=\mathbf{A}\boldsymbol{x}$

2.4 Rango y espacio nulo de una matriz

Rango, espacio columna o imagen de $\mathbf{A}^{m\times n}$: $\operatorname{range}(\mathbf{A})=\{\boldsymbol{v}\in\mathbb{R}^m:\boldsymbol{v}=\mathbf{A}\boldsymbol{x},\ \boldsymbol{x}\in\mathbb{R}^n\}$

Espacio nulo o núcleo de $\mathbf{A}^{m\times n}$: $\operatorname{nullspace}(\mathbf{A})=\{\boldsymbol{x}\in\mathbb{R}^n:\mathbf{A}\boldsymbol{x}=\mathbf{0}\}$

2.5 Proyección lineal

Proyección de $\boldsymbol{y}\in\mathbb{R}^m$ en el span de $\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\in\mathbb{R}^m\}$:

$$\operatorname{Proj}(\boldsymbol{y};\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\})=\operatorname*{argmin}_{\boldsymbol{v}\in\operatorname{span}(\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\})}\|\boldsymbol{y}-\boldsymbol{v}\|_2$$

Proyección de $\boldsymbol{y}\in\mathbb{R}^m$ en el span de $\mathbf{A}\in\mathbb{R}^{m\times n}$:

$$\operatorname{Proj}(\boldsymbol{y};\mathbf{A})=\operatorname*{argmin}_{\boldsymbol{v}\in\operatorname{range}(\mathbf{A})}\|\boldsymbol{y}-\boldsymbol{v}\|_2=\operatorname{Proj}(\mathbf{A})\,\boldsymbol{y}\quad\text{con}\quad\operatorname{Proj}(\mathbf{A})=\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t$$

3 Normas vectoriales y matriciales

3.1 Normas vectoriales

Norma de un vector $\boldsymbol{x}$: $\|\boldsymbol{x}\|$ es cualquier $f:\mathbb{R}^n\to\mathbb{R}$ tal que

$$f(\boldsymbol{x})\geq 0\qquad f(\boldsymbol{x})=0\Leftrightarrow\boldsymbol{x}=\mathbf{0}\qquad f(t\boldsymbol{x})=|t|f(\boldsymbol{x})\qquad f(\boldsymbol{x}+\boldsymbol{y})\leq f(\boldsymbol{x})+f(\boldsymbol{y})$$

p-norma o Lp: $\|\boldsymbol{x}\|_p=\left(\sum_{i=1}^n|x_i|^p\right)^{1/p}\quad p\in\mathbb{R}^{\geq 1}$

2-norma, Euclídea o L2: $\|\boldsymbol{x}\|_2=\sqrt{\sum_{i=1}^n x_i^2}=\sqrt{\boldsymbol{x}^t\boldsymbol{x}}$

1-norma, taxicab, Manhattan o L1: $\|\boldsymbol{x}\|_1=\sum_{i=1}^n|x_i|$

Max-norma, infinito o L∞: $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$

0-pseudo-norma: $\|\boldsymbol{x}\|_0 = \sum_{i=1}^n \mathbb{I}(|x_i| > 0) \overset{0^0=0}{=} \sum_{i=1}^n x_i^0$

In [1]:
```python
import numpy as np
a = np.array([-3, 4])
for p in (2, 1, np.inf, 0):
    print(f'L{p}({a}) = {np.linalg.norm(a, p)}')
```

```
L2([-3  4]) = 5.0
L1([-3  4]) = 7.0
Linf([-3  4]) = 4.0
L0([-3  4]) = 2.0
```

3.2 Normas matriciales

Suponemos $\mathbf{A} \in \mathbb{R}^{m\times n}$ y la interpretamos como función lineal $f(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x}$

Norma inducida por Lp: $\|\mathbf{A}\|_p = \max\limits_{\boldsymbol{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p} = \max\limits_{\|x\|_p=1} \|\mathbf{A}\boldsymbol{x}\|_p$ (máxima ganancia en la dirección de $\boldsymbol{x}$)

Norma inducida por L2: $\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^t\mathbf{A})} = \sigma_{\max}(\mathbf{A})$ (máximo valor singular de $\mathbf{A}$)

Norma nuclear o traza: $\|\mathbf{A}\|_* = \operatorname{tr}(\sqrt{\mathbf{A}^t\mathbf{A}}) = \sum_i \sigma_i = \sum_i |\sigma_i| = \|\boldsymbol{\sigma}\|_1$

p-norma de Schatten: $\|\mathbf{A}\|_p = \left(\sum_i \sigma_i^p(\mathbf{A})\right)^{1/p} \quad p \in \mathbb{R}^{\geq 1}$

Norma vectorial: $\|\mathbf{A}\| = \|\operatorname{vec}(\mathbf{A})\|$

Norma de Frobenius: $\|\mathbf{A}\|_F = \|\operatorname{vec}(\mathbf{A})\|_2 = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2} = \sqrt{\operatorname{tr}(\mathbf{A}^t\mathbf{A})}$

In [ ]:
```python
import numpy as np
A = np.array([[1, 0], [0, 1]])
for p in (2, 1, np.inf, 'fro'):
    print(f'p = {p}: {np.linalg.norm(A, p)}')
```

```
p = 2: 1.0
p = 1: 1.0
p = inf: 1.0
p = fro: 1.4142135623730951
```

4 Propiedades de una matriz

4.1 Traza de una matriz cuadrada

Traza de $\mathbf{A}\in\mathbb{R}^{n\times n}$: $\quad \operatorname{tr}(\mathbf{A})=\sum_{i=1}^n A_{ii}$

Propiedades: $\quad c\in\mathbb{R}\quad \mathbf{A},\mathbf{B}\in\mathbb{R}^{n\times n}$

$$\begin{aligned}
\operatorname{tr}(\mathbf{A})&=\operatorname{tr}(\mathbf{A}^t)\\
\operatorname{tr}(\mathbf{A}+\mathbf{B})&=\operatorname{tr}(\mathbf{A})+\operatorname{tr}(\mathbf{B})\\
\operatorname{tr}(c\mathbf{A})&=c\operatorname{tr}(\mathbf{A})\\
\operatorname{tr}(\mathbf{A}\mathbf{B})&=\operatorname{tr}(\mathbf{B}\mathbf{A})\\
\operatorname{tr}(\mathbf{A})&=\sum_{i=1}^n\lambda_i\quad\text{(suma de valores propios)}
\end{aligned}$$

Propiedad de la permutación cíclica: $\quad \operatorname{tr}(\mathbf{ABC})=\operatorname{tr}(\mathbf{BCA})=\operatorname{tr}(\mathbf{CAB})\qquad \mathbf{ABC}$ cuadrada

Truco de la traza: $\quad \boldsymbol{x}^t\mathbf{A}\boldsymbol{x}=\operatorname{tr}(\boldsymbol{x}^t\mathbf{A}\boldsymbol{x})=\operatorname{tr}(\boldsymbol{x}\boldsymbol{x}^t\mathbf{A})$

Estimador traza de Hutchinson: aproximación Monte Carlo a $\operatorname{tr}(\mathbf{A})$ con vectores aleatorios $\boldsymbol{v}$ tal que $\mathbb{E}[\boldsymbol{v}\boldsymbol{v}^t]=\mathbf{I}$

$$\operatorname{tr}(\mathbf{A})=\operatorname{tr}(\mathbf{A}\mathbb{E}[\boldsymbol{v}\boldsymbol{v}^t])=\mathbb{E}[\operatorname{tr}(\mathbf{A}\boldsymbol{v}\boldsymbol{v}^t)]=\mathbb{E}[\operatorname{tr}(\boldsymbol{v}^t\mathbf{A}\boldsymbol{v})]$$

In [1]:
```python
import numpy as np
A = np.array([[2, 1], [1, 2]])
print(np.trace(A))
```

```
4
```

4.2 Determinante de una matriz cuadrada

Determinante de $\mathbf{A} \in \mathbb{R}^{n\times n}$: $|\mathbf{A}|$ mide cuánto cambia un volumen unitario viendo $\mathbf{A}$ como una transformación lineal

Propiedades: $\quad \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$

$$\begin{aligned}
|\mathbf{A}| &= |\mathbf{A}^t| \\
|c\mathbf{A}| &= c^n|\mathbf{A}| \\
|\mathbf{A}\mathbf{B}| &= |\mathbf{A}||\mathbf{B}| \\
|\mathbf{A}| &= 0 \quad \text{sii} \quad \mathbf{A} \text{ singular} \\
|\mathbf{A}^{-1}| &= \frac{1}{|\mathbf{A}|} \quad \text{si} \quad \mathbf{A} \text{ es no singular} \\
|\mathbf{A}| &= \prod_{i=1}^{n} \lambda_i \qquad \{\lambda_i\} \text{ valores propios de } \mathbf{A}
\end{aligned}$$

Si $\mathbf{A}$ es definida positiva, su **descomposición de Cholesky** es $\mathbf{A} = \mathbf{L}\mathbf{L}^t$ con $\mathbf{L}$ triangular inferior y:

$$|\mathbf{A}| = |\mathbf{L}||\mathbf{L}^t| = |\mathbf{L}|^2 = \left(\prod_i L_{ii}\right)^2$$

$$\log|\mathbf{A}| = 2\log|\mathbf{L}| = 2\log\prod_i L_{ii} = 2\operatorname{tr}(\log(\operatorname{diag}(\mathbf{L})))$$

In [2]:
```
import numpy as np
A = np.array([[2, 1], [1, 2]])
print(round(np.linalg.det(A)))
```

3

4.3 Rango de una matriz

Rango de $\mathbf{A}\in\mathbb{R}^{m\times n}$: $\operatorname{rank}(\mathbf{A})$ es la dimensión del espacio generado por las columnas o filas de $\mathbf{A}$ (pues coincide para toda $\mathbf{A}$)

Propiedades: $\quad\mathbf{A},\mathbf{B}\in\mathbb{R}^{m\times n},\ \mathbf{C}\in\mathbb{R}^{n\times p}$

$$\begin{aligned}\operatorname{rank}(\mathbf{A})&\leq\min(m,n)\\ \operatorname{rank}(\mathbf{A})&=\operatorname{rank}(\mathbf{A}^t)=\operatorname{rank}(\mathbf{A}^t\mathbf{A})=\operatorname{rank}(\mathbf{A}\mathbf{A}^t)\\ \operatorname{rank}(\mathbf{A}\mathbf{C})&\leq\min(\operatorname{rank}(\mathbf{A}),\operatorname{rank}(\mathbf{C}))\\ \operatorname{rank}(\mathbf{A}+\mathbf{B})&\leq\operatorname{rank}(\mathbf{A})+\operatorname{rank}(\mathbf{B})\end{aligned}$$

$\mathbf{A}$ **es de rango completo:** si $\operatorname{rank}(\mathbf{A})=\min(m,n)$; si no, es de **rango deficiente**

Invertibilidad: $\mathbf{A}$ es **invertible** sii es de rango completo

In [3]:
```python
import numpy as np
A = np.array([[1, 0, 1], [-2, -3, 1], [3, 3, 0]])
print(np.linalg.matrix_rank(A))
```

```
2
```

4.4 Números de condición

Número de condición de $\mathbf{A}$: $\kappa(\mathbf{A})$ mide la inestabilidad numérica de los cálculos con $\mathbf{A}$ (mejor que el determinante)

$$\kappa(\mathbf{A})=\|\mathbf{A}\|\|\mathbf{A}^{-1}\|\qquad\kappa(\mathbf{A})\geq 1$$

$\mathbf{A}$ **bien condicionada:** si $\kappa(\mathbf{A})\approx 1$; si no, $\mathbf{A}$ está **mal condicionada** y, si $\kappa(\mathbf{A})$ es grande, $\mathbf{A}$ es **casi-singular**

Número de condición con L2: $\quad\kappa(\mathbf{A})=\dfrac{\sigma_{\max}}{\sigma_{\min}}=\sqrt{\dfrac{\lambda_{\max}}{\lambda_{\min}}}\qquad$ σ valor singular, λ valor propio

In [4]:
```python
import numpy as np
A = 0.1 * np.eye(100)
```

```
print(np.linalg.cond(A), np.linalg.det(A))
```

```
1.0 1.00000000000033e-100
```

5 Tipos de matrices especiales

5.1 Matriz diagonal

Matriz diagonal: $\mathbf{D}=\operatorname{diag}(d_1,\dots,d_n)\in\mathbb{R}^{n\times n}$ con ceros fuera de la diagonal

Construcción y extracción de una matriz diagonal: $\mathbf{D}=\operatorname{diag}(\boldsymbol{d})$ y $\boldsymbol{d}=\operatorname{diag}(\mathbf{D})$

Matriz identidad: $\mathbf{I}=\operatorname{diag}(1,\dots,1)\in\mathbb{R}^{n\times n}$ con unos en diagonal y ceros fuera

Matriz diagonal por bloques: $\begin{pmatrix}\mathbf{A}&\mathbf{0}\\\mathbf{0}&\mathbf{B}\end{pmatrix}$

Matriz banda-diagonal: con elementos no nulos en una banda de ancho k centrada en la diagonal; **tridiagonal** si $k=3$

In [1]:
```
import numpy as np
print("Construcción:\n", np.diag([1, 2]))
print("Extracción: ", np.diag(np.array([[1, 0], [0, 2]])))
print("Identidad:\n", np.eye(2))
```

```
Construcción:
 [[1 0]
 [0 2]]
Extracción:  [1 2]
Identidad:
 [[1. 0.]
 [0. 1.]]
```

5.2 Matrices triangulares

Matriz triangular superior: solo tiene elementos no nulos en la diagonal y por encima

Matriz triangular inferior: solo tiene elementos no nulos en la diagonal y por debajo

Propiedades: valores propios en la diagonal, por lo que $|\mathbf{A}| = \prod_i A_{ii}$

In [2]:
```
import numpy as np
A = np.arange(1, 5).reshape((2, 2))
print(A, "\n", np.tril(A), "\n", np.triu(A))
```

```
[[1 2]
 [3 4]]
 [[1 0]
 [3 4]]
 [[1 2]
 [0 4]]
```

5.3 Matrices definidas positivas

Forma cuadrática asociada a una matriz real y simétrica, $\mathbf{A} \in \mathbb{R}^{n\times n}$:

$$f(\boldsymbol{x}) = \boldsymbol{x}^t \mathbf{A} \boldsymbol{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j$$

$\mathbf{A}$ **definida positiva** ($\mathbf{A} \succ 0$ o $\mathbf{A} > 0$) sii $f(\boldsymbol{x}) > 0$ para todo $\boldsymbol{x}$ no nulo

$\mathbf{A}$ **semi-definida positiva** ($\mathbf{A} \succeq 0$ o $\mathbf{A} \geq 0$) sii $f(\boldsymbol{x}) \geq 0$ para todo $\boldsymbol{x}$ no nulo

$\mathbf{A}$ **definida negativa** ($\mathbf{A} \prec 0$ o $\mathbf{A} > 0$) sii $f(\boldsymbol{x}) < 0$ para todo $\boldsymbol{x}$ no nulo

$\mathbf{A}$ **semi-definida negativa** ($\mathbf{A} \preceq 0$ o $\mathbf{A} \leq 0$) sii $f(\boldsymbol{x}) \leq 0$ para todo $\boldsymbol{x}$ no nulo

$\mathbf{A}$ **indefinida** si no es semi-definida positiva ni semi-definida negativa

Relación entre $\mathbf{A}$ y $-\mathbf{A}$: $\mathbf{A} \succ 0$ sii $-\mathbf{A} \prec 0$ $\mathbf{A} \succeq 0$ sii $-\mathbf{A} \preceq 0$ $\mathbf{A}$ indefinida sii $-\mathbf{A}$ indefinida

Condición suficiente para $\mathbf{A} \succ 0$: $\mathbf{A}$ diagonalmente dominante, $|a_{ii}| > \sum_{j\neq i} |a_{ij}|$ para todo i, y diagonal positiva

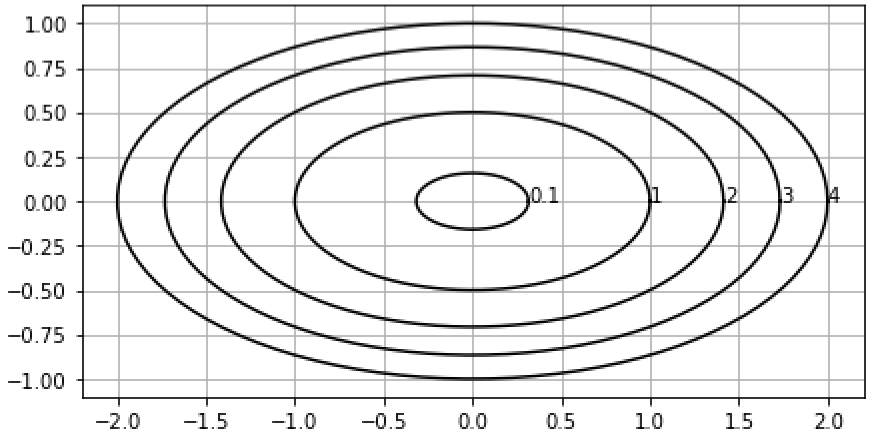
Condición necesaria y suficiente para $\mathbf{A} \succ 0$: valores propios positivos; por ejemplo, si $\mathbf{A} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ con $\{\lambda_i > 0\}$

$$f(\boldsymbol{x}) = \boldsymbol{x}^t \mathbf{A} \boldsymbol{x} = \sum_{i=1}^{n} \lambda_i x_i^2 > 0$$

Matriz de Gram de $\mathbf{A} \in \mathbb{R}^{m \times n}$: $\mathbf{G} = \mathbf{A}^t \mathbf{A}$ en general semi-definida positiva; definida positiva si $m \geq n$ y de rango completo

Ejemplo: si $\mathbf{A} = \operatorname{diag}(\lambda_1, \lambda_2) = \operatorname{diag}(1, 4)$, el conjunto de nivel C_k para f, $C_k = \{\boldsymbol{x} : f(\boldsymbol{x}) = k\}$, es una elipse centrada en el origen de semiejes alineados con los ejes canónicos y longitudes $\{\sqrt{k/\lambda_i}\}$

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
A = np.diag([1, 4])
t = np.linspace(0, 2.0 * np.pi, 100)
circ = np.array([np.cos(t), np.sin(t)]).T
fig, ax = plt.subplots(figsize=(7, 7))
ax.set(aspect='equal'); ax.grid()
for k in (.1, 1, 2, 3, 4):
    semiaxes = np.sqrt(k * np.linalg.inv(A))
    C = circ @ semiaxes # C @ A @ C.T = k
    plt.plot(C[:, 0], C[:, 1], color='black')
    plt.annotate(k, xy=(C[0]))
```



5.4 Matrices ortogonales

Vectores ortogonales: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ son ortogonales si $\boldsymbol{x}^t \boldsymbol{y} = 0$

Vector normalizado: $\boldsymbol{x} \in \mathbb{R}^n$ está normalizado si $\|\boldsymbol{x}\|_2 = 1$

Conjunto de vectores ortonormal: si son ortogonales dos a dos y normalizados

Matriz ortogonal $\mathbf{U} \in \mathbb{R}^{n\times n}$: si el conjunto de sus columnas (o filas) es ortonormal

Caracterización de matriz ortogonal: $\mathbf{U}$ ortogonal sii $\mathbf{U}^t\mathbf{U} = \mathbf{I} = \mathbf{U}\mathbf{U}^t$ sii $\mathbf{U}^t = \mathbf{U}^{-1}$

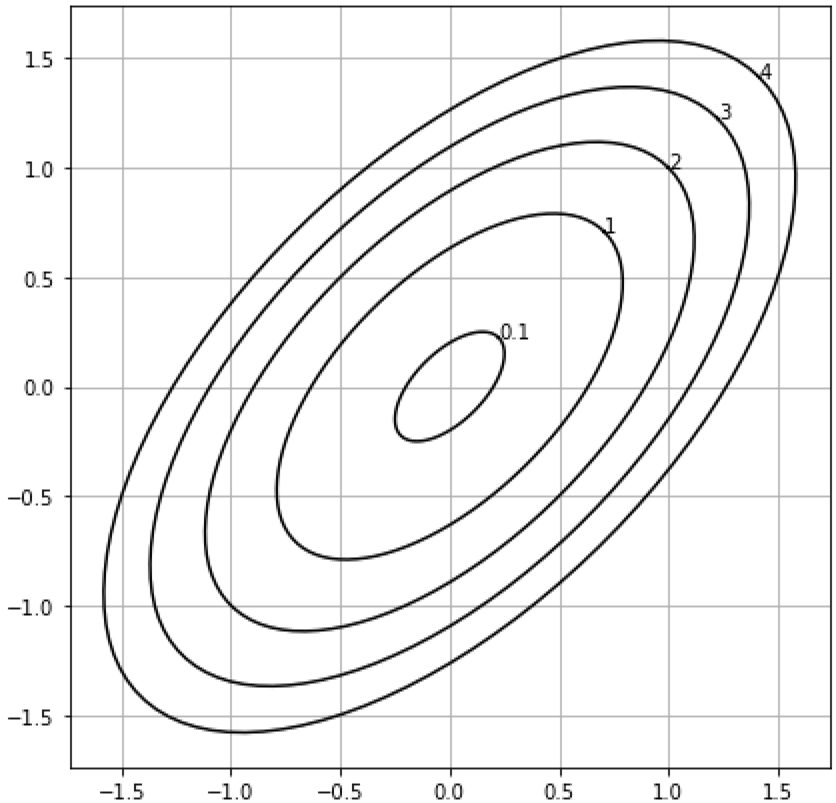
Propiedades de las transformaciones ortogonales: preservan longitudes y ángulos

$$\|\mathbf{U}\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_2$$
$$\cos(\alpha(\mathbf{U}\boldsymbol{x}, \mathbf{U}\boldsymbol{y})) = \frac{(\mathbf{U}\boldsymbol{x})^t(\mathbf{U}\boldsymbol{y})}{\|\mathbf{U}\boldsymbol{x}\|\|\mathbf{U}\boldsymbol{y}\|} = \frac{\boldsymbol{x}^t\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} = \cos(\alpha(\boldsymbol{x}, \boldsymbol{y}))$$

Ejemplo (cont.): matriz ortogonal de rotación 2d de α radianes en sentido antihorario,

$$\mathbf{R}(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
A = np.diag([1, 4])
t = np.linspace(0, 2.0 * np.pi, 100)
circ = np.array([np.cos(t), np.sin(t)]).T
alpha = np.pi / 4
R = np.array([ [np.cos(alpha), -np.sin(alpha)], [np.sin(alpha), np.cos(alpha)] ])
fig, ax = plt.subplots(figsize=(7, 7))
ax.set(aspect='equal'); ax.grid()
for k in (.1, 1, 2, 3, 4):
    semiaxes = np.sqrt(k * np.linalg.inv(A))
    C = circ @ semiaxes # C @ A @ C.T = k
    C = C @ R.T
    plt.plot(C[:, 0], C[:, 1], color='black')
    plt.annotate(k, xy=(C[0]))
```



6 Multiplicación de matrices

6.1 Definición y propiedades

Producto de dos matrices: $\mathbf{A}\in\mathbb{R}^{m\times n}$ y $\mathbf{B}\in\mathbb{R}^{n\times p}$

$$\mathbf{C}=\mathbf{A}\mathbf{B}\in\mathbb{R}^{m\times p}\qquad\text{con}\qquad C_{ij}=\sum\nolimits_{k=1}^{n}A_{ik}B_{kj}$$

Coste: $O(mnp)$, aunque mucho menor con GPUs

Propiedad asociativa: $\quad(\mathbf{A}\mathbf{B})\mathbf{C}=\mathbf{A}(\mathbf{B}\mathbf{C})$

Propiedad distributiva: $\quad\mathbf{A}(\mathbf{B}+\mathbf{C})=\mathbf{A}\mathbf{B}+\mathbf{A}\mathbf{C}$

No propiedad commutativa: $\quad\mathbf{A}\mathbf{B}\neq\mathbf{B}\mathbf{A}$

In [1]:
```python
import numpy as np
A = np.array([[0, 1], [1, 2]]); B = A; A @ B
```

Out[1]:
```
array([[1, 2],
       [2, 5]])
```

6.2 Productos vector-vector

6.2.1 Producto escalar, interno o punto

$$\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n:\quad\langle\boldsymbol{x},\boldsymbol{y}\rangle=\boldsymbol{x}^t\boldsymbol{y}=\sum_{i=1}^{n}x_iy_i=\boldsymbol{y}^t\boldsymbol{x}$$

In [1]:
```python
import numpy as np
x = np.array([1, 2]); y = np.array([2, 1])
print(np.dot(x, y))
```

4

6.2.2 Producto vectorial, externo o cruz

$$\boldsymbol{x} \in \mathbb{R}^m,\ \boldsymbol{y} \in \mathbb{R}^n : \quad \boldsymbol{x}\boldsymbol{y}^t = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & x_2y_2 & \cdots & x_2y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_my_1 & x_my_2 & \cdots & x_my_n \end{bmatrix} \in \mathbb{R}^{m\times n}$$

In [2]:
```
import numpy as np
x = np.array([1, 2]); y = np.array([2, 1])
print(np.outer(x, y));
```

```
[[2 1]
 [4 2]]
```

6.3 Productos matriz-vector

Producto matriz-vector:

$$\mathbf{A} \in \mathbb{R}^{m\times n}, \boldsymbol{x} \in \mathbb{R}^n : \quad \boldsymbol{y} = \mathbf{A}\boldsymbol{x} \in \mathbb{R}^m$$

6.3.1 Interpretación por filas

La i-ésima entrada de $\boldsymbol{y}$ es el producto escalar de la i-ésima fila de $\mathbf{A}$ y $\boldsymbol{x}$:

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1^t \\ \boldsymbol{a}_2^t \\ \vdots \\ \boldsymbol{a}_m^t \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1^t\boldsymbol{x} \\ \boldsymbol{a}_2^t\boldsymbol{x} \\ \vdots \\ \boldsymbol{a}_m^t\boldsymbol{x} \end{bmatrix}$$

6.3.2 Interpretación por columnas

$\boldsymbol{y}$ es una combinación lineal de las columnas de $\mathbf{A}$, las cuales pueden verse como una **base** de un **subespacio lineal:**

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \boldsymbol{a}_1 x_1 + \boldsymbol{a}_2 x_2 + \cdots + \boldsymbol{a}_n x_n$$

6.4 Productos matriz-matriz

6.4.1 Interpretaciones

Dadas $\mathbf{A} \in \mathbb{R}^{m \times n}$ y $\mathbf{B} \in \mathbb{R}^{n \times p}$, existen cuatro maneras diferentes de interpretar $\mathbf{C} = \mathbf{A}\mathbf{B} \in \mathbb{R}^{m \times p}$:

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{a}_1^t \\ \boldsymbol{a}_2^t \\ \vdots \\ \boldsymbol{a}_m^t \end{bmatrix} [\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_p] = \begin{bmatrix} \boldsymbol{a}_1^t\boldsymbol{b}_1 & \boldsymbol{a}_1^t\boldsymbol{b}_2 & \cdots & \boldsymbol{a}_1^t\boldsymbol{b}_p \\ \boldsymbol{a}_2^t\boldsymbol{b}_1 & \boldsymbol{a}_2^t\boldsymbol{b}_2 & \cdots & \boldsymbol{a}_2^t\boldsymbol{b}_p \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_m^t\boldsymbol{b}_1 & \boldsymbol{a}_m^t\boldsymbol{b}_2 & \cdots & \boldsymbol{a}_m^t\boldsymbol{b}_p \end{bmatrix}$$

$$\mathbf{C} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n] \begin{bmatrix} \boldsymbol{b}_1^t \\ \boldsymbol{b}_2^t \\ \vdots \\ \boldsymbol{b}_n^t \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{b}_i^t$$

$$\mathbf{C} = \mathbf{A}[\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_p] = [\mathbf{A}\boldsymbol{b}_1, \mathbf{A}\boldsymbol{b}_2, \dots, \mathbf{A}\boldsymbol{b}_p]$$

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{a}_1^t \\ \boldsymbol{a}_2^t \\ \vdots \\ \boldsymbol{a}_m^t \end{bmatrix} \mathbf{B} = \begin{bmatrix} \boldsymbol{a}_1^t\mathbf{B} \\ \boldsymbol{a}_2^t\mathbf{B} \\ \vdots \\ \boldsymbol{a}_m^t\mathbf{B} \end{bmatrix}$$

6.4.2 Notación

Cuadrado matricial: $\mathbf{A}^2 = \mathbf{A}\mathbf{A}$

Cuadrado elemental: $\mathbf{A}^{\odot 2} = [A_{ij}^2]$

Raíz cuadrada matricial: $\mathbf{A} = \sqrt{\mathbf{M}} \quad \text{si} \quad \mathbf{A}^2 = \mathbf{M}$

Raíz cuadrada elemental: $[\sqrt{M_{ij}}]$

7 Inversa de una matriz cuadrada

Inversa de $\mathbf{A} \in \mathbb{R}^{n\times n}$: matriz $\mathbf{A}^{-1}$ tal que $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} = \mathbf{A}\mathbf{A}^{-1}$

Existencia: $\mathbf{A}^{-1}$ existe sii $\det \mathbf{A} \neq 0$; esto es, si $\mathbf{A}$ es no singular

Propiedades: $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$ $\quad (\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ $\quad (\mathbf{A}^{-1})^t = (\mathbf{A}^t)^{-1} = \mathbf{A}^{-t}$

Inversa de $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$: $\quad \mathbf{A}^{-1} = \dfrac{1}{|\mathbf{A}|}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Inversa de una diagonal por bloques: $\begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{-1} \end{pmatrix}$

8 Conceptos básicos de cálculo matricial

8.1 Derivadas

Derivada: de una función $f : \mathbb{R} \to \mathbb{R}$ en un punto

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}$$

Interpretación geométrica: $f'(x)$ puede verse como la pendiente de la línea tangente en $f(x)$

$$f(x+h) \approx f(x) + f'(x)\,h$$

Aproximaciones a la derivada de diferencia finita: dado un $h > 0$ pequeño

$$\begin{aligned}
f'(x) &\approx \frac{f(x+h) - f(x)}{h} && \text{(diferencia forward)} \\
f'(x) &\approx \frac{f(x+h/2) - f(x-h/2)}{h} && \text{(diferencia central)} \\
f'(x) &\approx \frac{f(x) - f(x-h)}{h} && \text{(diferencia backward)}
\end{aligned}$$

La diferenciación es un operador funcional: $D(f) = f'$

Notación de Lagrange: f' denota la derivada; f'' la segunda; $f^{(n)}$ la n-ésima

Notación de Leibniz: si $y = f(x)$ es una función, $\frac{dy}{dx}$ o $\frac{d}{dx}f(x)$ es su derivada y $\left.\frac{df}{dx}\right|_{x=a}$ su derivada en un punto a

8.2 Gradientes

Derivada parcial: de $f : \mathbb{R}^n \to \mathbb{R}$ respecto a x_i

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(\boldsymbol{x} + h\boldsymbol{e}_i) - f(\boldsymbol{x})}{h} \qquad (\boldsymbol{e}_i \text{ es el i-ésimo vector unitario})$$

Gradiente: de f en un punto $\boldsymbol{x}$

$$\boldsymbol{g} = \frac{\partial f}{\partial \boldsymbol{x}} = \nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \qquad \text{o} \qquad \boldsymbol{g}(\boldsymbol{x}^*) = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}^*} \quad (\text{enfatiza que se evalúa en } \boldsymbol{x}^*)$$

8.3 La Jacobiana

Matriz Jacobiana: de $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$

$$\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{x}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^t} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \nabla f_1(\boldsymbol{x})^t \\ \vdots \\ \nabla f_m(\boldsymbol{x})^t \end{pmatrix}$$

8.3.1 Multiplicación de Jacobianas y vectores

Producto Jacobiana-vector: $\mathbf{J} \in \mathbb{R}^{m \times n},\ \boldsymbol{v} \in \mathbb{R}^n$

$$\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{x})\boldsymbol{v} = \begin{pmatrix} \nabla f_1(\boldsymbol{x})^t \\ \vdots \\ \nabla f_m(\boldsymbol{x})^t \end{pmatrix} \boldsymbol{v} = \begin{pmatrix} \nabla f_1(\boldsymbol{x})^t \boldsymbol{v} \\ \vdots \\ \nabla f_m(\boldsymbol{x})^t \boldsymbol{v} \end{pmatrix}$$

Producto vector-Jacobiana: $\mathbf{J} \in \mathbb{R}^{m \times n},\ \boldsymbol{u} \in \mathbb{R}^m$

$$\boldsymbol{u}^t \mathbf{J}_{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{u}^t \left(\frac{\partial \boldsymbol{f}}{\partial x_1}, \ldots, \frac{\partial \boldsymbol{f}}{\partial x_n} \right) = \left(\boldsymbol{u} \frac{\partial \boldsymbol{f}}{\partial x_1}, \ldots, \boldsymbol{u} \frac{\partial \boldsymbol{f}}{\partial x_n} \right)$$

8.3.2 Jacobiana de una composición

Jacobiana de una composición: $h(\boldsymbol{x}) = g(f(\boldsymbol{x}))$

$$\mathbf{J}_h(\boldsymbol{x}) = \mathbf{J}_g(f(\boldsymbol{x}))\mathbf{J}_f(\boldsymbol{x})$$

Ejemplo: $f : \mathbb{R} \to \mathbb{R}^2,\ g : \mathbb{R}^2 \to \mathbb{R}^2$

$$\frac{\partial \boldsymbol{g}}{\partial x} = \begin{pmatrix} \frac{\partial}{\partial x} g_1(f_1(x), f_2(x)) \\ \frac{\partial}{\partial x} g_2(f_1(x), f_2(x)) \end{pmatrix} = \begin{pmatrix} \frac{\partial g_1}{\partial f_1}\frac{\partial f_1}{\partial x} + \frac{\partial g_1}{\partial f_2}\frac{\partial f_2}{\partial x} \\ \frac{\partial g_2}{\partial f_1}\frac{\partial f_1}{\partial x} + \frac{\partial g_2}{\partial f_2}\frac{\partial f_2}{\partial x} \end{pmatrix} = \begin{pmatrix} \frac{\partial g_1}{\partial f_1} & \frac{\partial g_1}{\partial f_2} \\ \frac{\partial g_2}{\partial f_1} & \frac{\partial g_2}{\partial f_2} \end{pmatrix} \begin{pmatrix} \frac{\partial f_1}{\partial x} \\ \frac{\partial f_2}{\partial x} \end{pmatrix} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{f}} \frac{\partial \boldsymbol{f}}{\partial x}$$

8.4 La Hessiana

Matriz Hessiana: de $f : \mathbb{R}^n \to \mathbb{R}$ es la Jacobiana del gradiente

$$\mathbf{H}_f = \frac{\partial^2 f}{\partial \boldsymbol{x}^2} = \nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

9 Gradientes de funciones comunes

9.1 Gradientes de funciones de escalar a escalar

Función diferenciable: $f : \mathbb{R} \to \mathbb{R}$

$$\begin{aligned} \frac{d}{dx} cx^n &= cnx^{n-1} \\ \frac{d}{dx} \log(x) &= 1/x \\ \frac{d}{dx} e^x &= e^x \\ \frac{d}{dx}[f(x) + g(x)] &= \frac{df(x)}{dx} + \frac{dg(x)}{dx} \\ \frac{d}{dx}[f(x)g(x)] &= f(x)\frac{dg(x)}{dx} + g(x)\frac{df(x)}{dx} \\ \frac{d}{dx} f(u(x)) &= \frac{df(u)}{du} \, \frac{du}{dx} \end{aligned}$$

9.2 Gradientes de funciones de vector a escalar

Función diferenciable: $f : \mathbb{R}^n \to \mathbb{R}$

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{a}^t\boldsymbol{x}) = \boldsymbol{a}$$
$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{b}^t\mathbf{A}\boldsymbol{x}) = \mathbf{A}^t\boldsymbol{b}$$
$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^t\mathbf{A}\boldsymbol{x}) = (\mathbf{A}+\mathbf{A}^t)\boldsymbol{x}$$

9.3 Gradientes de funciones de matriz a escalar

Función diferenciable: $f : \mathbb{R}^{m\times n} \to \mathbb{R}$

$$\frac{\partial f}{\partial \mathbf{X}} = \begin{pmatrix} \frac{\partial f}{\partial x_{11}} & \cdots & \frac{\partial f}{\partial x_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_{m1}} & \cdots & \frac{\partial f}{\partial x_{mn}} \end{pmatrix}$$

Identidades para formas cuadráticas:

$$\frac{\partial}{\partial \mathbf{X}}(\boldsymbol{a}^t\mathbf{X}\boldsymbol{b}) = \boldsymbol{a}\boldsymbol{b}^t$$
$$\frac{\partial}{\partial \mathbf{X}}(\boldsymbol{a}^t\mathbf{X}^t\boldsymbol{b}) = \boldsymbol{b}\boldsymbol{a}^t$$

Identidades con la traza:

$$\frac{\partial}{\partial \mathbf{X}}\operatorname{tr}(\mathbf{A}\mathbf{X}\mathbf{B}) = \mathbf{A}^t\mathbf{B}^t$$
$$\frac{\partial}{\partial \mathbf{X}}\operatorname{tr}(\mathbf{X}^t\mathbf{A}) = \mathbf{A}$$
$$\frac{\partial}{\partial \mathbf{X}}\operatorname{tr}(\mathbf{X}^{-1}\mathbf{A}) = -\mathbf{X}^{-t}\mathbf{A}^t\mathbf{X}^{-t}$$
$$\frac{\partial}{\partial \mathbf{X}}\operatorname{tr}(\mathbf{X}^t\mathbf{A}\mathbf{X}) = (\mathbf{A}+\mathbf{A}^t)\mathbf{X}$$

Identidades con el determinante:

$$\frac{\partial}{\partial \mathbf{X}} |\mathbf{AXB}| = |\mathbf{AXB}| \mathbf{X}^{-t}$$
$$\frac{\partial}{\partial \mathbf{X}} \ln(|\mathbf{X}|) = \mathbf{X}^{-t}$$