

## プログラムリスト 2.1 NumPyで2行2列の行列の行列式を計算する

```
In [2]: import numpy as np                                # (A1) NumPyライブラリをimport
A = np.array([[4, -3], [3, -1]])                        # (B1) 行列の定義
det_A = np.linalg.det(A)                               # (C1) 行列式を計算する関数
print("det(A)=", det_A)                                # (D1) 行列式をそのまま表示
print(f"det(A)= {det_A:.3f}")                          # (D2) 小数点以下3桁に丸めて表示
```

det(A)= 4.999999999999999

det(A)= 5.000

## プログラムリスト 2.2 NumPyで3行3列の行列の転置行列と逆行列を計算する

```
In [3]: import numpy as np                                # (A1) NumPyライブラリをimport
A = np.array([[1, 2, 0], [-1, 2, 1], [5, 0, -3]])      # (B1) 行列の定義
trans_A = A.T                                           # (C1) 転置行列を計算する関数
inv_A = np.linalg.inv(A)                               # (C2) 逆行列を計算する関数
print("trans(A)=\n", trans_A)                          # (D1) 転置行列を表示
print("inv(A)=\n", inv_A)                              # (D2) 逆行列を表示
```

trans(A)=

[[ 1 -1 5]

[ 2 2 0]

[ 0 1 -3]]

inv(A)=

[[ 3. -3. -1. ]

[-1. 1.5 0.5]

[ 5. -5. -2. ]]

## プログラムリスト 2.3 NumPyによる連立方程式の解法

```
In [3]: import numpy as np                                # (A1) NumPyライブラリをimport
A = np.array([[1, 3, -1], [4, -2, 2], [1, -1, -2]])    # (B1) 係数行列の定義
B = np.array([2, 6, -4])                                # (B2) 定数ベクトルの定義
inv_A = np.linalg.inv(A)                               # (C1) 逆行列を計算
inv_A_B = inv_A @ B                                     # (C2) 逆行列を用いて解を計算
SOLVED_X = np.linalg.solve(A, B)                       # (C3) solve関数で解を計算
print("INV:", inv_A_B)                                  # (D1) 逆行列を用いた計算結果を表示
print("SOLVE:", SOLVED_X)                              # (D2) solve関数を用いた計算結果を表示
```

INV: [[1. 1. 2.]

[1. 1. 2.]