# EFFEKTIV PROGRAMMERING I NUMPY.

Eks: Gitt $A, B \in \mathbb{R}^{n \times n}$, beregn $C = A \cdot B$.

Algoritmen: for $i = 1, \cdots, n$
for $j = 1, \cdots, n$
$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

## Kompleksiteten av en algoritme:

Sier noe om hvor arbeidskrevende (CPU-tid, ...) en algoritme er.
Måles ofte i flop (flyttallsoperasjoner).

1 flop = 1 addisjon og 1 multiplikasjon
eller en divisjon.

| Eks: | | krever | | |
|---|---|---|---|---|
| | $C = AB$ | | $n^3$ flop. | $A, B \in \mathbb{R}^{n \times n}$ |
| | $x = Ay$ | | $n^2$ flop | $x, y \in \mathbb{R}^n$ |
| | $c = x^T y$ | | $n$ flop. | |

Implementer og test algoritmen over (Øving 1, oppg. 4)

La $a^{(j)}$, $b^{(j)}$ kolonne $j$ av $A$ og $B$ resp.
$a_{(i)}$, $b_{(i)}$ rad $i$ —"— —"—

Alt 1: Som over

Alt 2: $c_{ij} = a_{(i)} \cdot b^{(j)}$, $i, j = 1, \cdots, n$

Alt 3: $c^{(j)} = A b^{(j)}$, $j = 1, \cdots, n$

Alt. 4: $C = AB$

Eks 2: Diskretisering av varmeledningsligningen.

$$U_i^{j+1} = r \cdot (U_{i+1}^j - 2U_i^j + U_{i-1}^j), \quad i = 0, \cdots, M, \quad j = 0, \cdots N$$

$U_i^0$ er kjent, $U_0^j = U_M^j = 0, \quad j = 0, 1, \cdots, N$

$r \leq 1/2$

MORAL.

Når det er mulig:

- Unngå løkker
- Gjør operasjoner direkte på arrays.

Tenk også over hvor mye minne som brukes.
Ikke lagre data som ikke trengs.

Se time_linalg.py

men prøv gjerne andre ting selv!