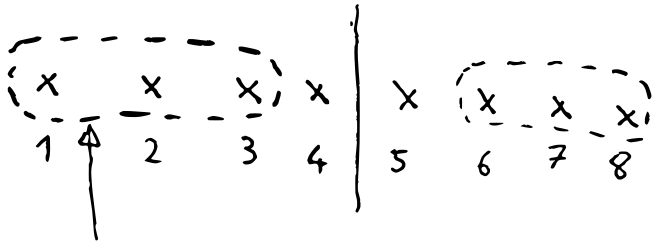
Parallelization:

Wave equation 1D:

$$\partial_t \Pi(x,t) = \Delta_x^2 \varphi(x,t) \equiv K(\varphi)$$
$$\partial_t \varphi(x,t) = \Pi(x,t)$$

$$\Delta_x^2 \varphi(x,t) = \frac{\varphi(n+1,t) + \varphi(n,t) - 2\varphi(n,t)}{\Delta t^2}$$

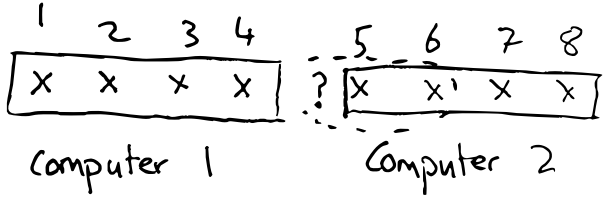


Kernel for $\Pi(n=2, t+1)$ depends only on $\varphi(1,t)$, $\varphi(2,t)$, $\varphi(3,t)$

$\Pi(n=7, t+1)$ —

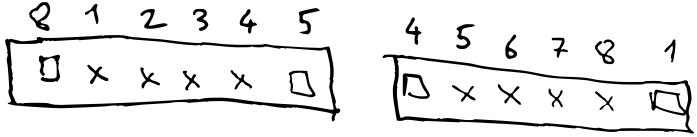
– $\varphi(6,t)$, $\varphi(7,t)$, $\varphi(8,t)$

⇒ Can compute them completely independently! Even on different computers.



Q: How do we compute kernel @ 5?

A: Ghosts cells!



□ are copies of corresponding x.

Need to be updated after x is.

⇒ Communication between computers!

Standard protocol:

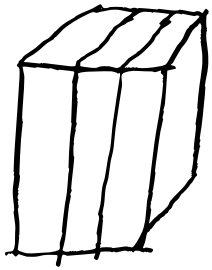
Message Passing Interface (MPI)

⇒ **CosmoLattice** fully hides the parallelization under the hood.

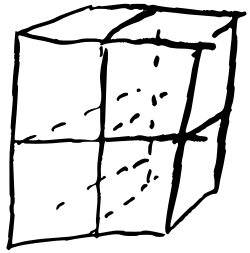
Also use parallelized FFT.

Back to 3D: Two parallelization mode with CL:

- 1D, default FFTW3
- 2D, need to install PFFT



1D



2D

Rules for how many process n_p can use:

- 1D //: $N = m \cdot n_p$, $m \in \mathbb{N}$
- 2D //: $N = m_1 \cdot n_p^1$
  $= m_2 \cdot n_p^2$

with $m_1, m_2, n_p^1, n_p^2 \in \mathbb{N}$

and $n_p = n_p^1 \cdot n_p^2$

Example: $N = 50$

1D //: $n_p = 2$

$n_p = 5$

$n_p = 10$

$n_p = 25$

2D //: $n_p = 2$ (1,2) $n_p = 625$ (25,25)

$n_p = 4$ (2,2)

$n_p = 5$ (1,5)

$n_p = 10$ (2,5)

$n_p = 20$ (2,10)

$n_p = 25$ (5,5)

$n_p = 50$ (5,10)

$n_p = 100$ (10,10)

$n_p = 125$ (5,25)

$n_p = 250$ (10,25)

To use 2D, need to install PFFT (see install script).

See live demo