High Performance Computing for Science and Engineering

Exercise 9: Sparse Linear Algebra with MPI

Sparse linear systems

$$Au = b$$

- appear from grid-based discretization of PDEs
- less memory compared to dense representation
- less operations for matrix-vector product:

sparse: $O(nnz)$

dense: $O(n^2)$

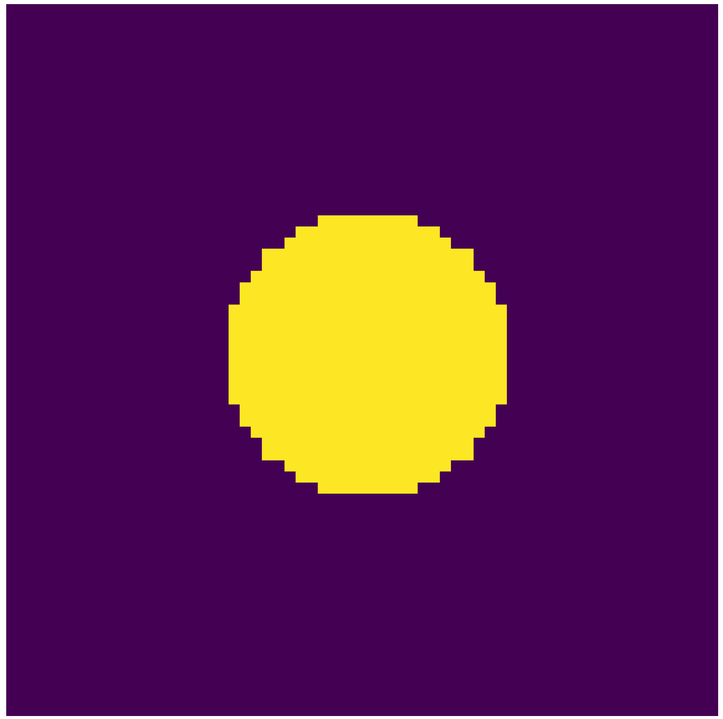
nnz - nonzero elements

n - rows

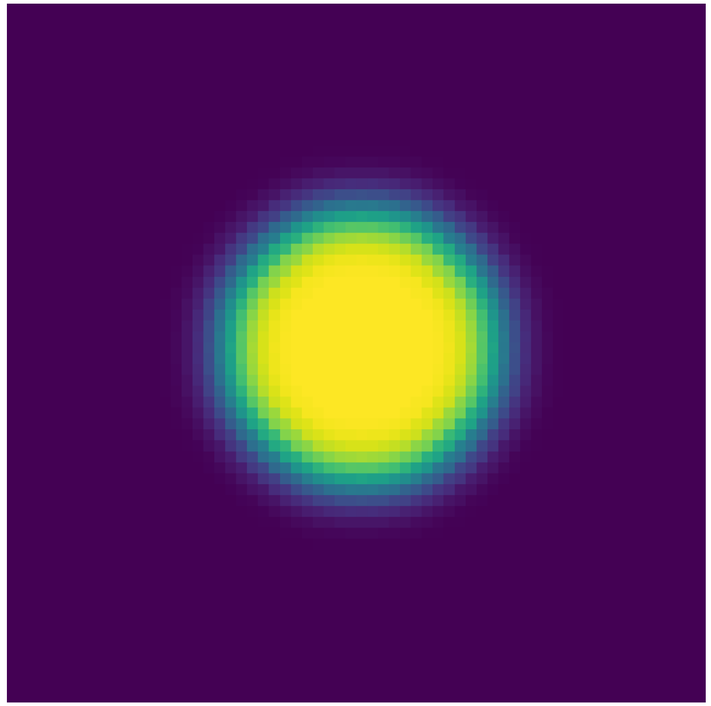
- libraries: MKL Pardiso, Hypre, PETSc

Diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

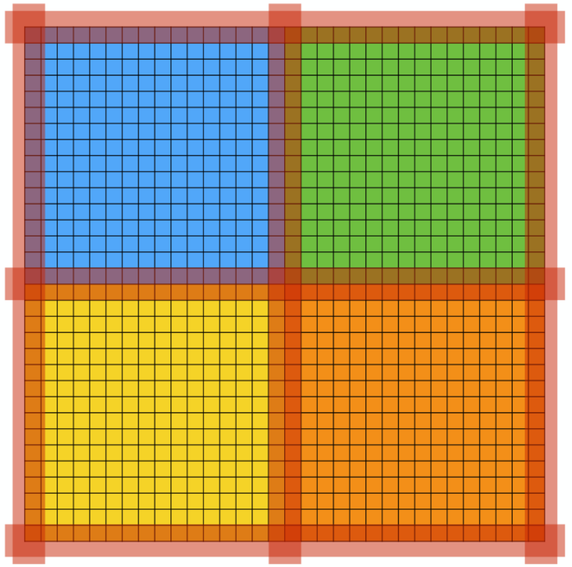


initial



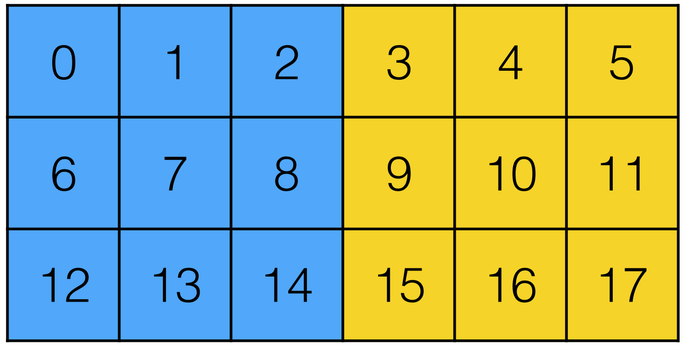
advanced

grid decomposition



Mapping grid to vector

2D grid



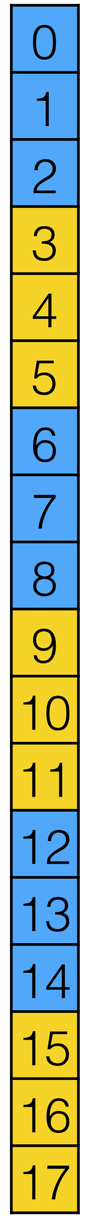
global indices

rank 0

rank 1

```
grid size:

NX = 6

NY = 3


number of blocks:

NBX = 2

NBY = 1
```

vector

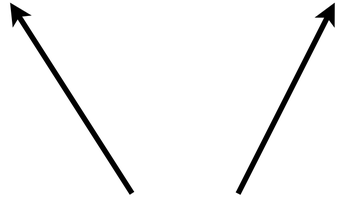


Example: shift operator

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

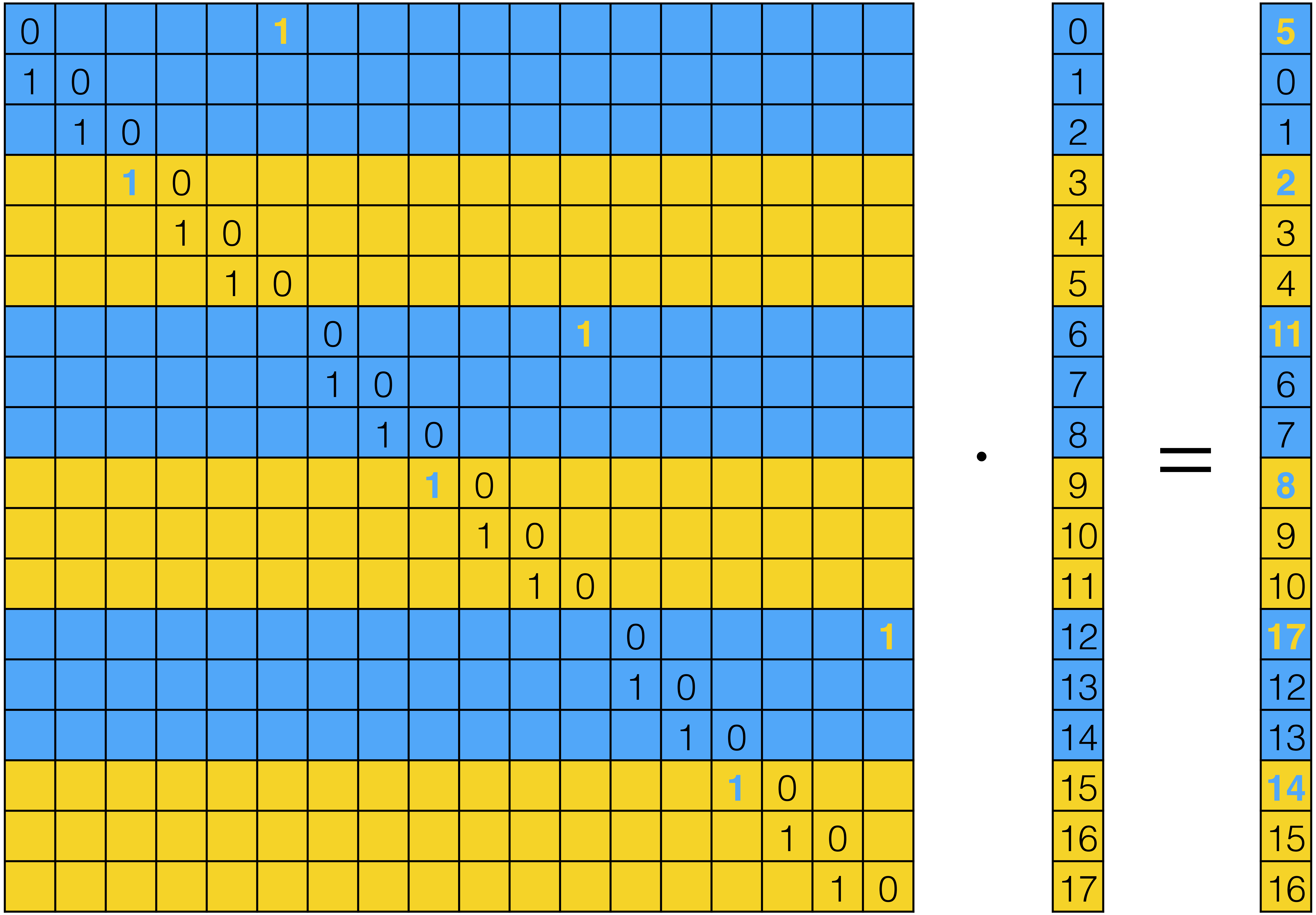
$$Au[x,y] = u[x-1,y]$$

5	0	1	**2**	3	4
11	6	7	**8**	9	10
17	12	13	**14**	15	16



communication needed

Example: shift operator



Hints

- use MPI_ANY_SOURCE to receive from unknown sources
- use MPI_Probe to get the message size
- for weak scaling, keep the number of cells per rank constant:

number of processors	1	4	9	16	36	49
NX=NY=N	64	128	192	256	384	448