MPI

Additional lab notes on

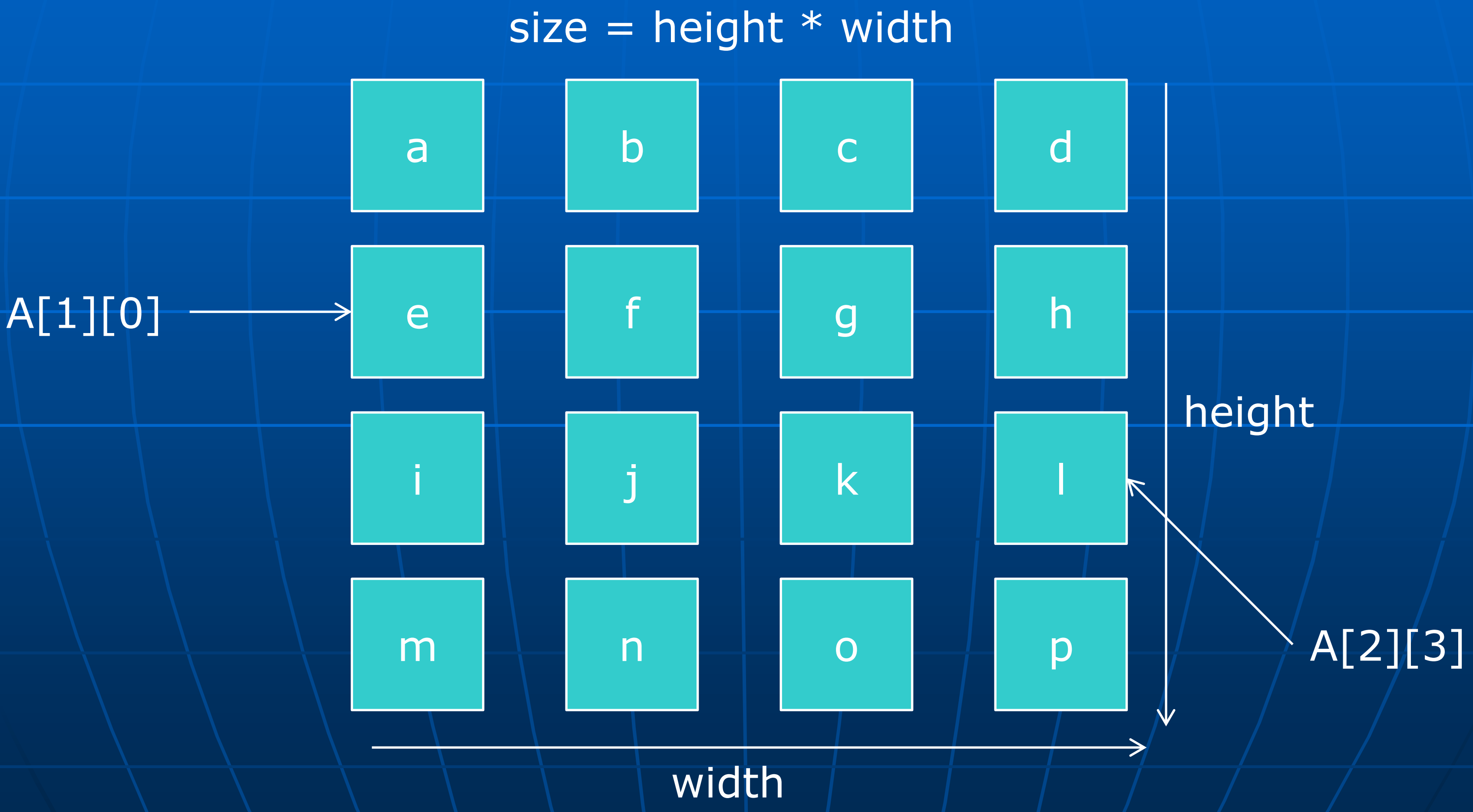
Working with multi-dimensional arrays

MPI n-Dimensional arrays

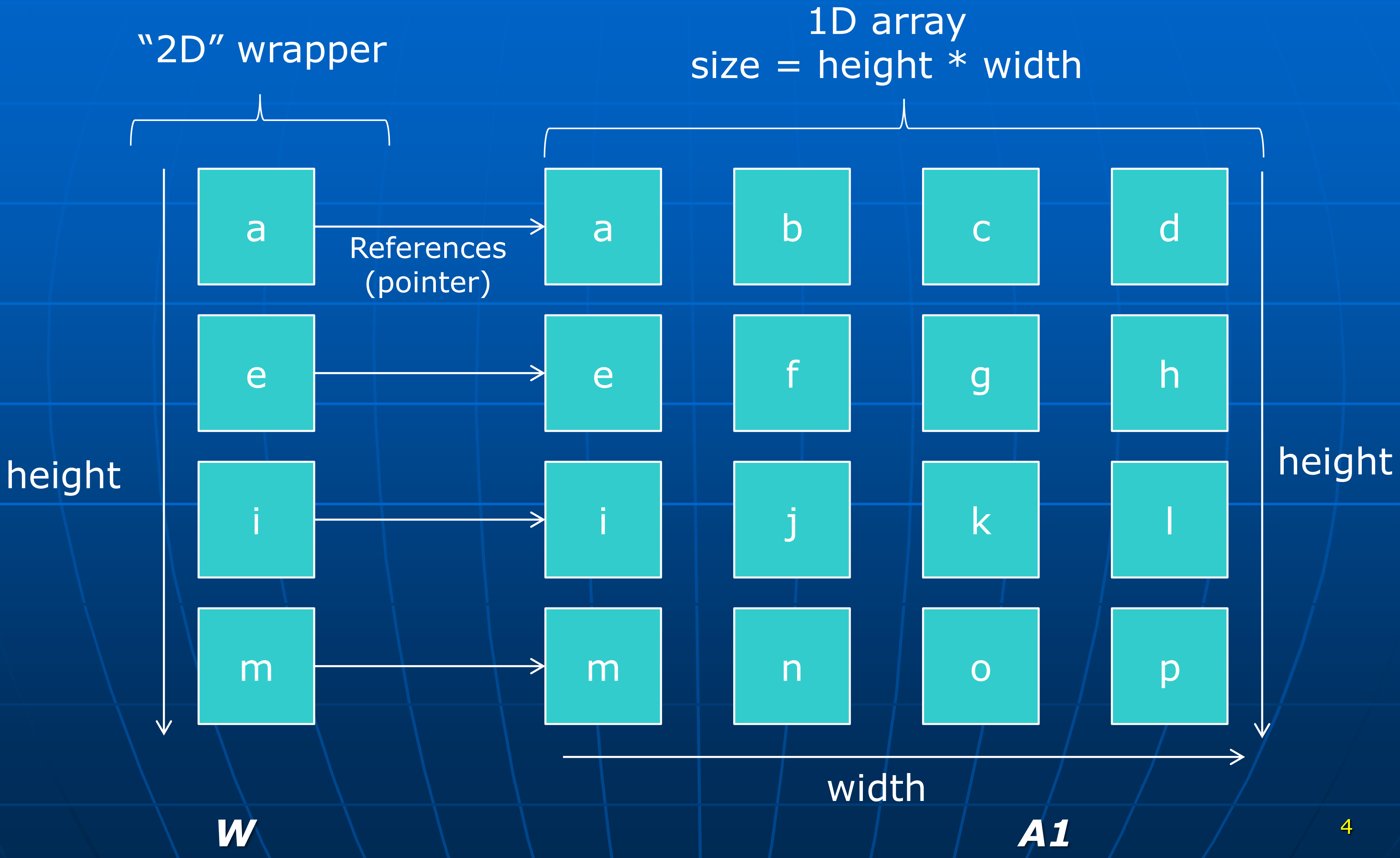
- MPI is inefficient in communicating multidimensional arrays (e.g. A[x][y])
- Solution:
  - Put data in a 1D array.
  - Use 1D array for MPI functions.
  - Use 2D wrapper array for normal programming.

MPI, standard 2D array

Array "***A***", normal 2D array:



MPI, wrapper


"2D" wrapper
1D array
size = height * width
a
e
i
m
References
(pointer)
a b c d
e f g h
i j k l
m n o p
height
height
width
W
A1

MPI, wrapper usage

Previous: ***A***[2][3] = 5

Now: ***W***[2][3] = 5

Trick:

Intermediate step

- ***W***[2]

  = 1D array start + 2*width*itemSize

  = **pointer**
- ***W***[2][3]

  = 1D array start + 2*width*itemSize + 3*itemSize

  = **value**

Use this!

MPI, operation on wrapper

Previous:

Trying to send bits of 2D array

```
MPI_Scatter(
    send_A, chunkSize, MPI_INT,
    recv_A, chunkSize, MPI_INT,
    0, MPI_COMM_WORLD);
```

FAILS: because A is 2D array

Now:

Send bits of 1D array

```
MPI_Scatter(
    send_W[0], chunkSize, MPI_INT,
    recv_W [0], chunkSize, MPI_INT,
    0, MPI_COMM_WORLD);
```

MPI, wrapper creation

Ptr to start of
1D array
(Actual data)

```
W= malloc(height * sizeof(int*))

W [0] = malloc(widht * height * sizeof(int))

For i=1, i<height, ++i
    W[row] = W[row-1] + width;
```

- Free in a simmilar manner!

Lab assignment

- Wrapper is already in parts of current mpi assignment.
  - Naming is different:
    - ***W*** == img->imdata
    - ***A1*** == img->imdata[0]

Actual data