Introduction to Languages for Scientific Computing

Prof. **Paolo Bientinesi**

`pauldj@aices.rwth-aachen.de`

1) y = clock_MV(n, v)

As input, the function `clock_MV` takes an integer `n` and a vector `v` of size `n`; as output, it returns the vector `y`, also of size `n`, resulting from the multiplication of the "clock" matrix `C` with the input vector `v`. Mathematically, `y = C * v`.
The $n \times n$ clock matrix `C` is defined as follows.

n is even

$$
C = \begin{bmatrix}
4 & & & & & & 1 \\
& \ddots & & & & \cdot^{\cdot^{\cdot}} & \\
& & 4 & 1 & & & \\
& & 3 & 2 & & & \\
& \cdot^{\cdot^{\cdot}} & & & \ddots & & \\
3 & & & & & & 2
\end{bmatrix}
$$

n is odd

$$
C = \begin{bmatrix}
7 & & & 8 & & & 1 \\
& \ddots & & \vdots & & \cdot^{\cdot^{\cdot}} & \\
& & 7 & 8 & 1 & & \\
6 & \dots & 6 & 0 & 2 & \dots & 2 \\
& & 5 & 4 & 3 & & \\
& \cdot^{\cdot^{\cdot}} & & \vdots & & \ddots & \\
5 & & & 4 & & & 3
\end{bmatrix}
$$

- Write two Matlab functions that implement `clock_MV`.
  Goal #1 (`clock_MV_nice`): Elegance, readability.
  Goal #2 (`clock_MV_fast`): Low computational complexity.

2) M = distr(A, p, q, type)

Write a Matlab function that implements `distr`.

As input, the function `distr` takes 4 arguments: an $m \times n$ matrix `A`, two positive integers `p` and `q`, and the string `type`, which can be 'cols', 'rows', and '2D'; as output, `distr` returns a matrix `M` of the same size as `A`. You can assume that both `m` and `n` are multiples of $p \times q$.

Figuratively, the function `distr` distributes the elements of the matrix `A` among $t = p * q$ processors. The distribution is determined by the parameter `type`.

- type == 'cols'

  The columns of `A` are assigned to the `t` processors in round robin fashion. The matrix `M` is obtained by collecting all the columns assigned to processor 1 first, then all those assigned to processor 2, and so on.
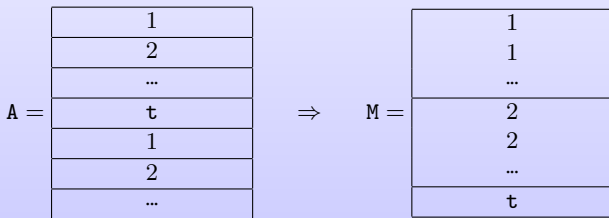
$$A = \boxed{1\,|\,2\,|\cdots|\,t\,|\,1\,|\,2\,|\cdots} \quad \Rightarrow \quad M = \boxed{1\ \ 1\ \cdots\ 2\ \ 2\ \cdots\ t}$$
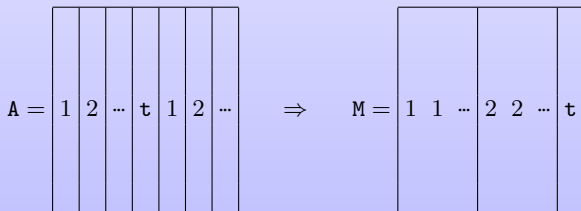
3) M = distr(A, p, q, type)

- type == 'rows'

  The rows of A are assigned to the t processors in round robin fashion.
  The matrix M is obtained by collecting all the rows assigned to processor 1 first,
  then all those assigned to processor 2, and so on.

3) M = distr(A, p, q, type)

- type == '2D'

  The elements of A are assigned according to the imaginary p×q grid of processors.

  Example with p = 2 and q = 3:

Submission

- Individual assignment.

- Prepare 3 files: clock_MV_nice.m, clock_MV_fast.m and distr.m.
  Include your name in each file.

- Archive the files as your_name.zip or your_name.tgz and submit.

- Submission by email to   pauldj@aices.rwth-aachen.de

- Email's subject: ``LSC-15 HW2 <your last name>''

- **Deadline: Wednesday, November 18, 23.59pm.**