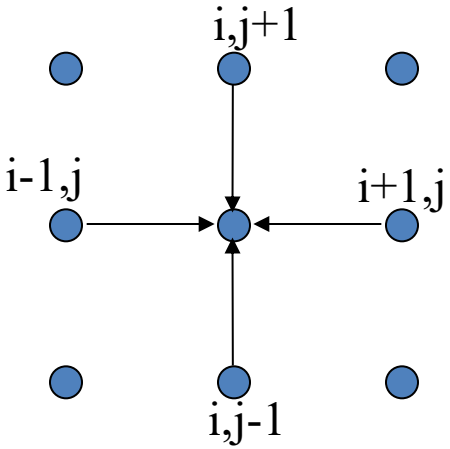
Homework 5

An implementation of Laplace's equation using OpenSHMEM

Deadline: February 26th 2020

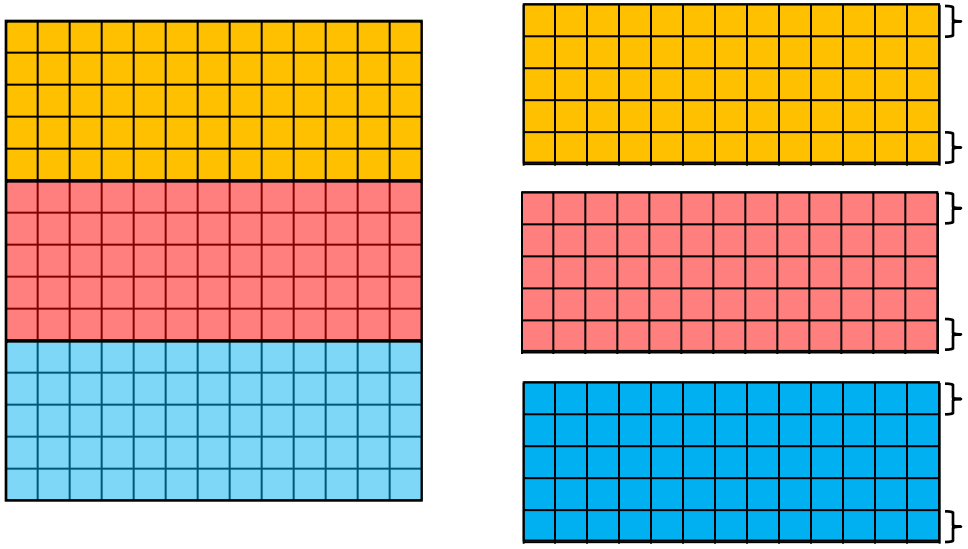
$$U_{i,j}^{n+1} = \frac{1}{4}\left(U_{i-1,j}^{n} + U_{i+1}^{n} + U_{i,j-1}^{n} + U_{i,j+1}^{n}\right)$$

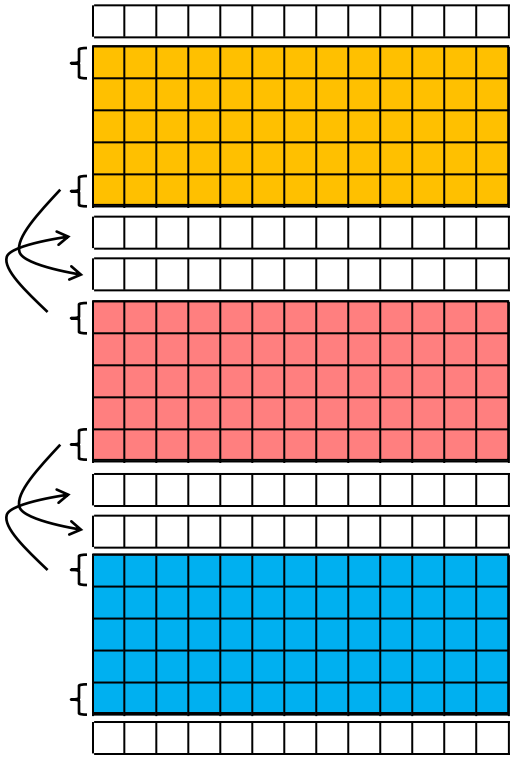
Laplace's equation – Open SHMEM




```
for j = 1 to jmax
  for i = 1 to imax
    Unew(i,j) = 0.25 * ( U(i-1,j) + U(i+1,j)
                       +   U(i,j-1) + U(i,j+1))

  end for
end for
```





- In order to compare the different programming model, namely MPI and Open SHMEM, we will take the same Laplace equation from Homework #4 and will provide a parallel implementation using Open SHMEM.
- Therefore we are working under the same assumptions as before: a 2-dimensional matrix stored in row-major format and will have to compute a well-defined number of iterations of the computation of the Laplace equation using multiple Open SHMEM processes.
  - Minimize memory requirements
  - Use 1D block or 2D block/block data distributions. Justify your choice.
  - Implement support for ghost region
- Originally the matrix is initialized with 0 everywhere except the boundaries (first and last row and first and last column) which are initialized differently.
- Highlight the impact of using multiple Open SHMEM processes to execute this algorithm by doing an analysis of the algorithm's performance. Vary the number of processes and the problem size (weak and strong scaling) and comment on the results.
  - Present the average over multiple runs to account for any measuring error or outside effects

- Benchmarking of the Laplace algorithm should be measured excluding the initialization of parallel programming environment.
  - Keep everything related to setting up the problem outside of the timed section
- A skeleton code is available on the class website (or can be obtained by email from the TA). Add your Open SHMEM code directly there.
- You are strongly encouraged to explore different programming technique in order to improve the scalability of your solution.
  - Detail your approach on the final document.
  - Compare with your MPI implementation.