

2D Heat Equation

Problem: Solve 2D heat equation for a specified number of iterations within a 2D rectangular object with thermal conductivity on each direction as hyper-parameters.

Methodology:

Suppose, the matter is not gaining any heat from outside. Hence, the temperature at a given (x,y) will be it's previous temperature + what it is gaining from it's neighbors[(x-1,y);(x+1,y);(x,y-1);(x,y+1)] - what it will be distributing to it's neighborhoods.

Sequential Method:

- cx, cy is thermal conductivity on x and y direction
- u[i][j] have the value for current iteration
- w[i][j] is used for next iteration

```
for ( i = 1; i < M - 1; i++ )
{
    for ( j = 1; j < N - 1; j++ )
    {
        w[i][j] = cx * u[i-1][j] + cx * u[i+1][j] + cy * u[i][j-1] + cy * u[i][j+1] - 2 * cx * u[i][j] - 2 * cy * u[i][j];
    }
}
```

OpenMP Method:

As there is no synchronization required the openMP version is very trivial.

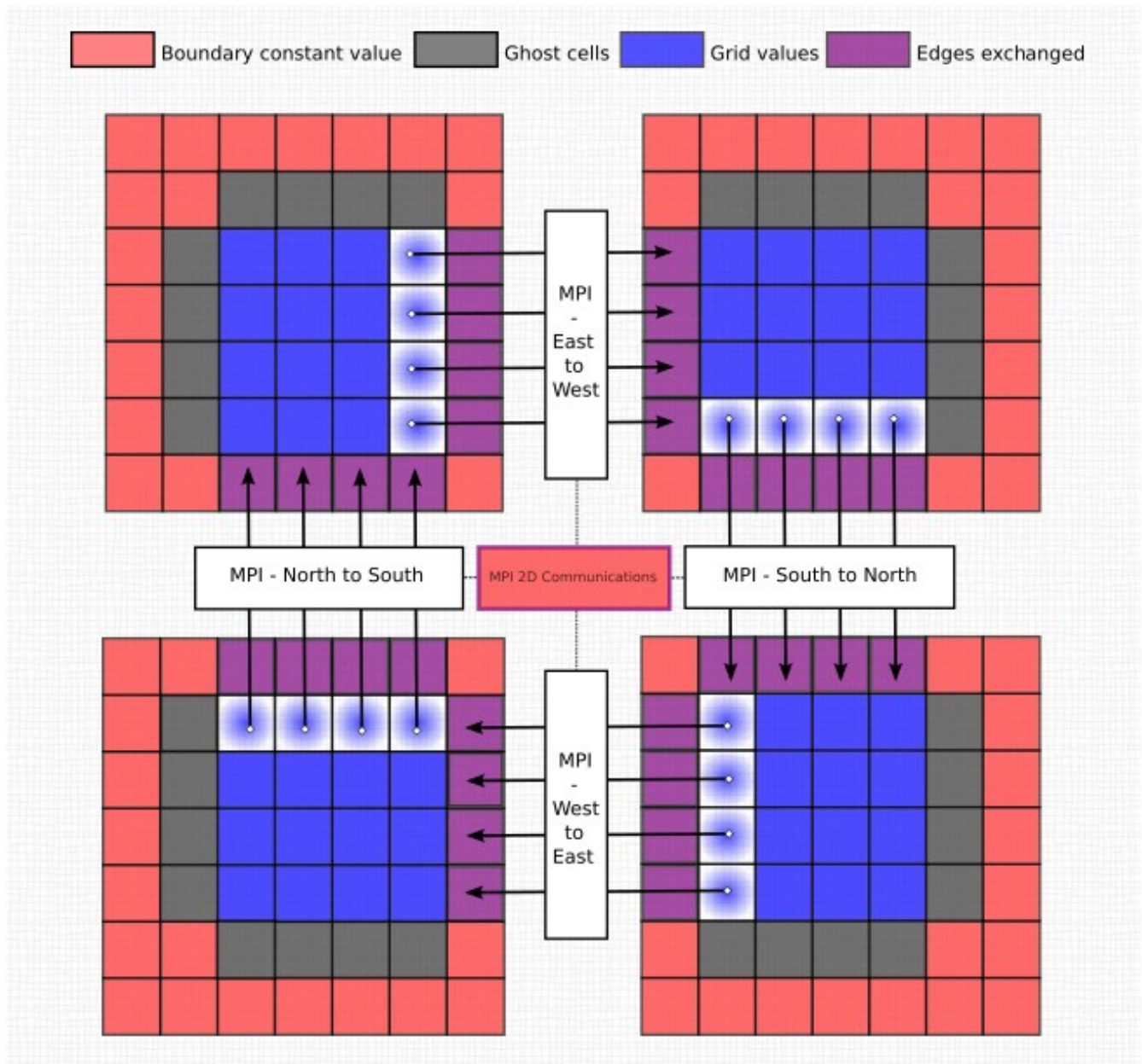
```
# pragma omp parallel shared(cx, cy) private ( ix, iy )
```

```
# pragma omp for
```

```
for ( i = 1; i < M - 1; i++ )
{
    for ( j = 1; j < N - 1; j++ )
    {
        w[i][j] = cx * u[i-1][j] + cx * u[i+1][j] + cy * u[i][j-1] + cy * u[i][j+1] - 2 * cx * u[i][j] - 2 * cy * u[i][j];
    }
}
```

MPI Version:

As any (x,y) require the value of its neighborhoods. The neighbors can belong to different processors. Hence, for border nodes we need to send the information to respective processors.



And it will wait for information from its neighboring nodes.