

## PARALLELISING THE SOR ALGORITHM

So far I have been using just one logical processor of my CPU however my CPU has 8 of them, so I might obtain a speed boost by utilising more of them

↓  
multiprocessing

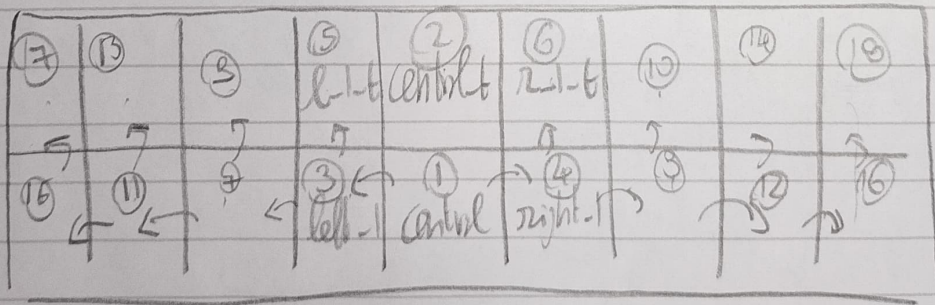
```
import multiprocessing as mp
```

Multiprocessing module in Python allows to optrel, to send a function to multiple cores and evaluate those functions simultaneously

Good for independent processes

Check Parallel Processing in Python for more information.

## Work Flow of Weighting Field Calculation



The numbers indicate the order in which the blocks are SOR updated in a sequential software

However we should notice that the calculation in the left blocks are INDEPENDENT of the calculation in the right

blocks.

Therefore we can think of parallelizing the evaluation of some of the blocks.

A parallelised version would be;

13	8	6	4	control 2	4	6	8	10
5	7	5	3	control 1	3	5	7	9

10 steps vs 18 we have almost halved the computational time (not the number of processors) by a factor of 2!

In this case our pool would need 2 logical processors

After all the domain is divided, we must compute the global residuals

22 global residuals  $\rightarrow$  speeds up the process calling the full pool of 8 processors

Over the domain

22 blocks  $\rightarrow$  evaluates in 12 "steps" } more computationally demanding  
22 global residuals  $\rightarrow$  evaluates in about 2.5 steps