

# Instruction of BCT parallelism

Jinyong Feng

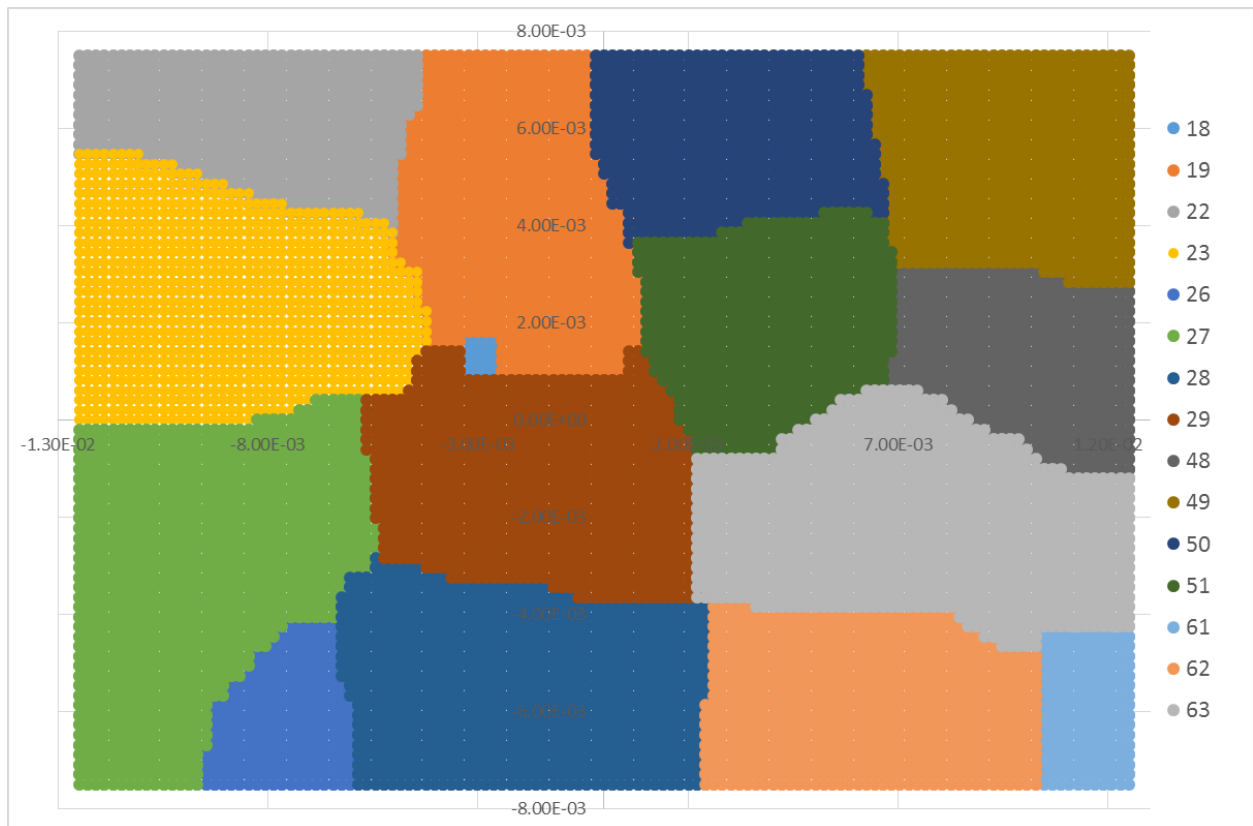
03/13/2015

Step1: modify genbc.f to generate the coordinate positions from different processors on inflow surface

Make sure the “Reading BCT” in solver.inp is 0

Input: one-step simulation from PHASTA (you can change the numstart.dat to be 0 after you are ready to run the case on multi processors); bct.dat (come from bct.\*.dat generated from Varts\_BCT, with \* is the ID of time window)

Output: bct.dat.\*, with \* is the ID of the processors



Step2: modify Coin.f to remove the repeated coordinates

Locations of Coin.f: Process\_bctinflow/Coin.f

Input: myrank.txt (containing the ID of those processors that have boundary points); bct.dat.\*, with \* is the ID of the processors

Output: bctin.dat.\*, with \* is the ID of the cores

Step3: modify Vart\_BCT.f to generate velocity information of all nodes for different time windows

Locations of Vart\_BCT.f: /Varts\_BCT

Input: vbct.inp, xyzts.dat, varts\_run01.dat (come from BCTMerge code, locations: \BCTMerge)

Output: bct.\*.dat, with \* is the ID of the time window

Screenshot of vbct.inp:

Deltat is the simulation time for each bct.dat.\*

Phdt is the timestep size for stable simulation (read from phasta output)

```
Start Simulation Time, End Simulation Time:
      0
      5.58
Deltat
      0.565
phdt
      6.670E-4
Start Time Step, End Time Step
      0
      8400
ioverlap to expand timestep range for slight overlap
      20
```

Step4: create bct\_partition.f to generate the selected nodes information for different time window and cores

Location: /Varts\_BCT

Input: myrank.txt, bctin.dat.\*, bct.\*.dat, "bct\_input" folder (need to create this folder in \varts)

Output: bct.dat.\*1.\*2, with \*1 is the ID of time window and \*2 is the ID of the processors

Step 5: add a step function “Reading BCT” in solver.inp; modify bctint.f to read bct.dat.\* to run the case on different processor; add “iextsbct” in some subroutines, like itrdrv.f and genini.f which will distinguish between those nodes having boundary points and the nodes not having boundary points

Reading BCT: 0 means the PHASTA code will read the bct.dat and generate multiple bct.dat.\* for further steps

Reading BCT: 1 means the PHASTA code will read the bct.dat.\* and run the bct files on different processors

Input: bct.dat.1.\*, with \* is the ID of processor