DISK SELF-GRAVITY CALCULATIONS ON A LEVEL 1 GRID WITH LEVEL 0 REFINEMENT

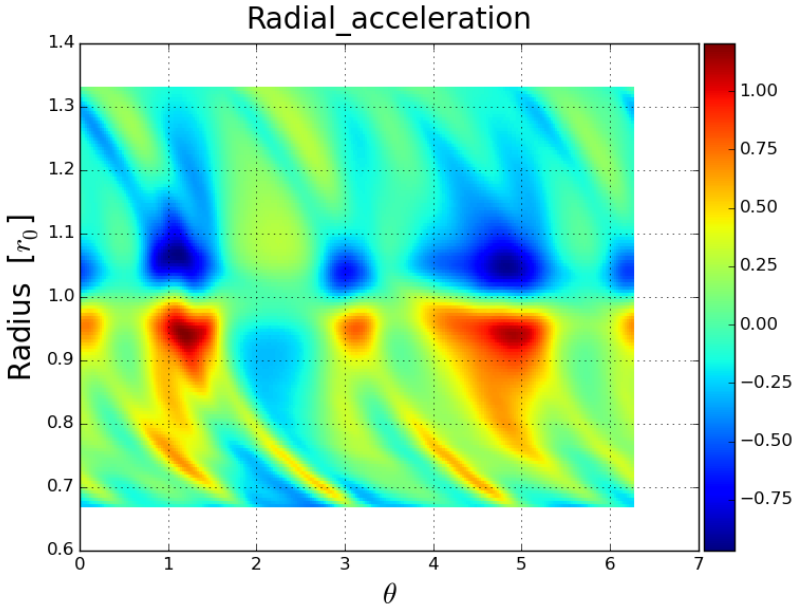
Computational Science II - University of Zurich

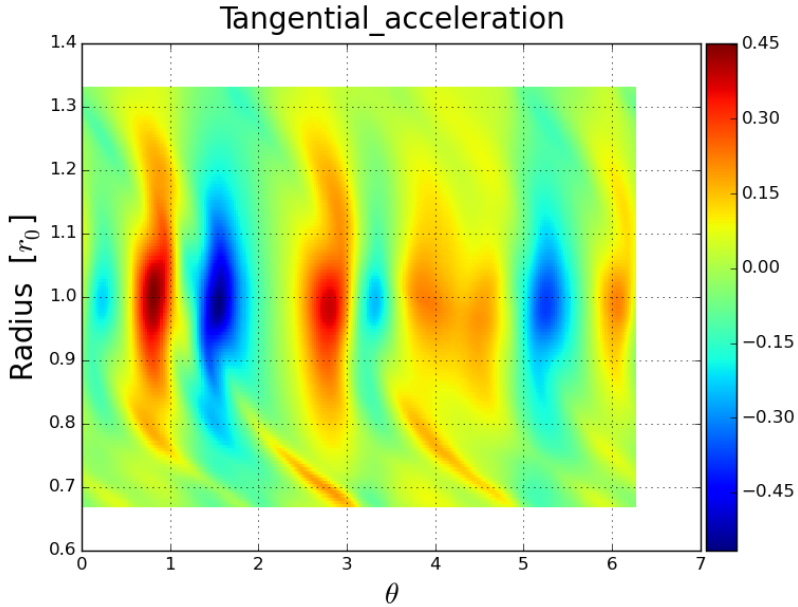
Mihai Tomozeiu

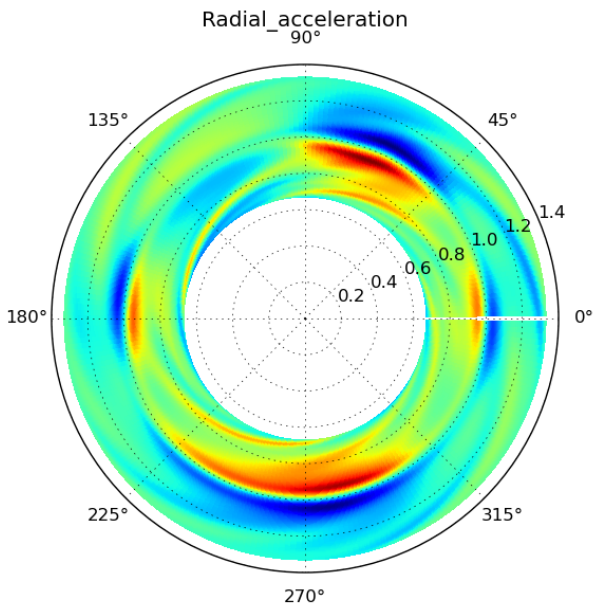
4th June 2015

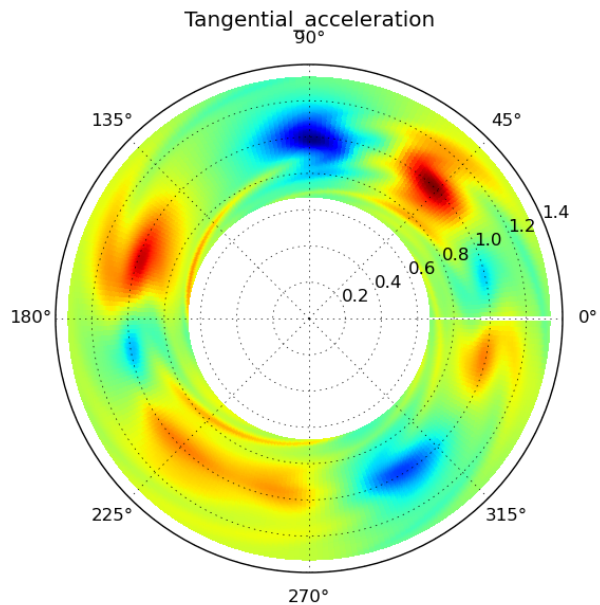
Contents

March starting point

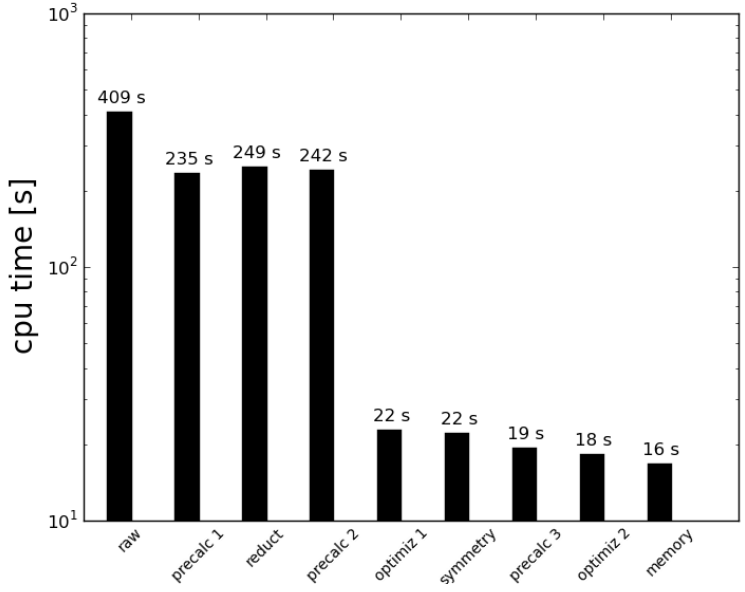








March starting point



- reduction of the number of calculations;
- a priori calculation of 1D and 2D arrays;
- use of trigonometric relations;

New approach

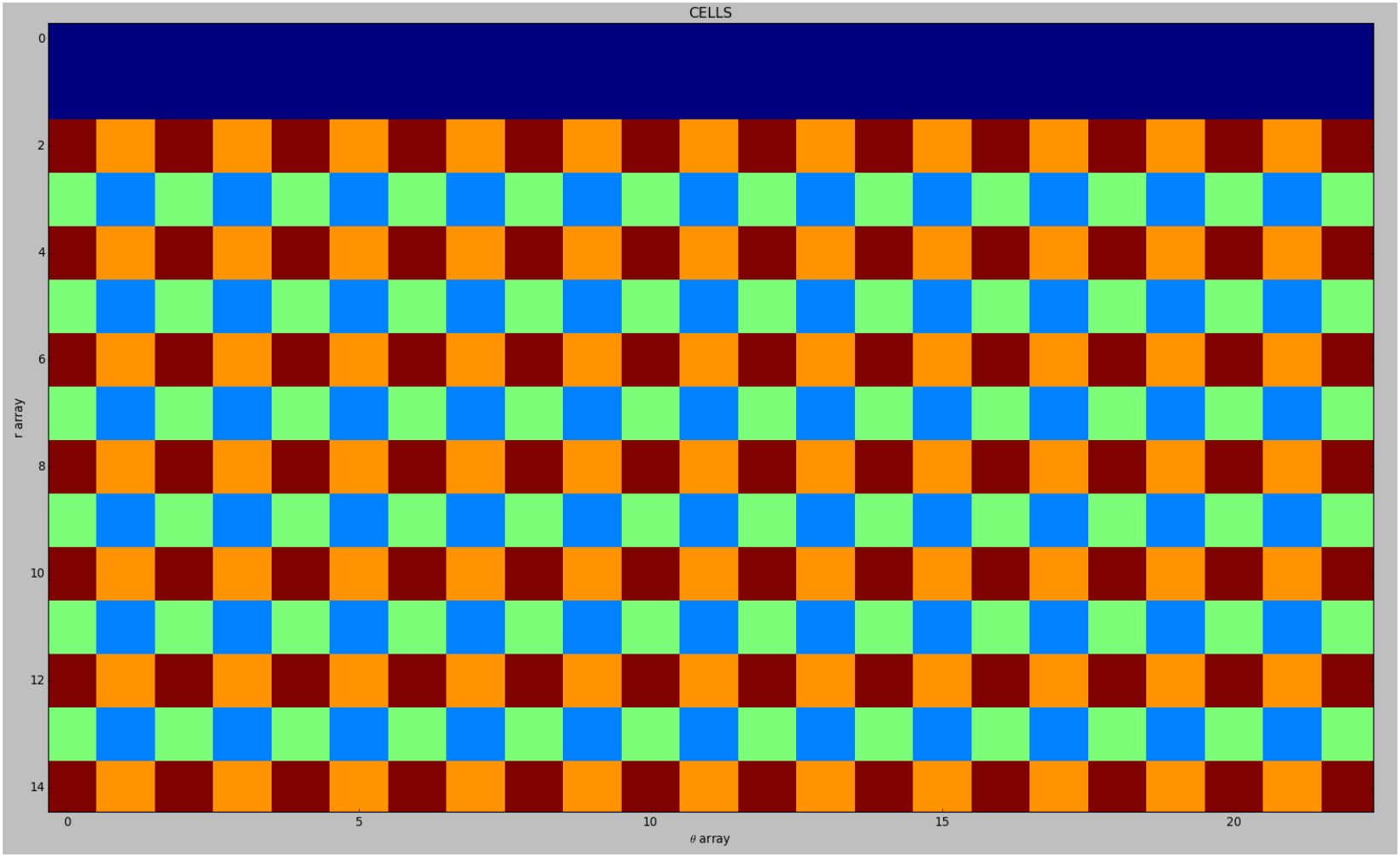


Figure : Cells coloured according to the parity of their indices.

New approach

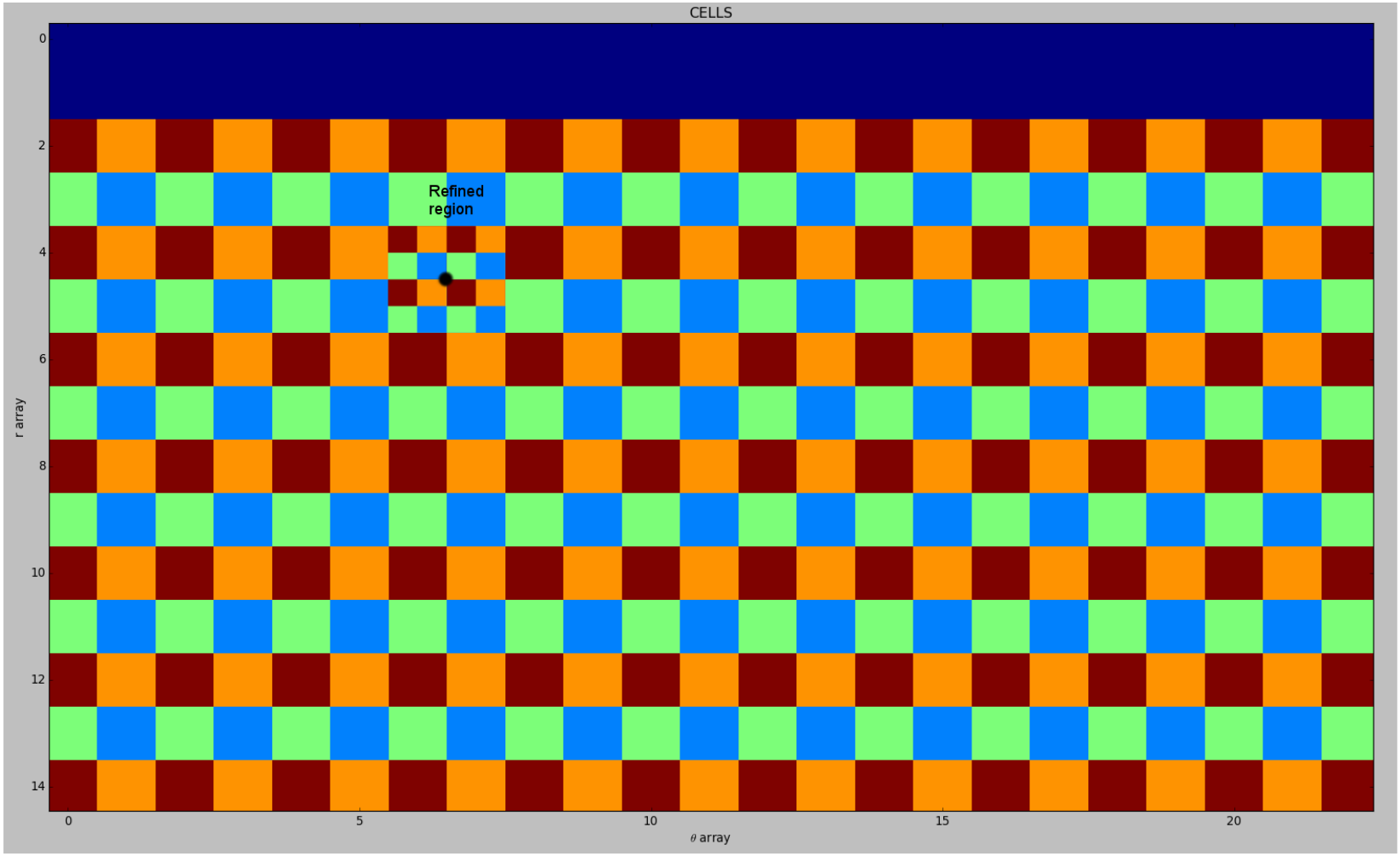
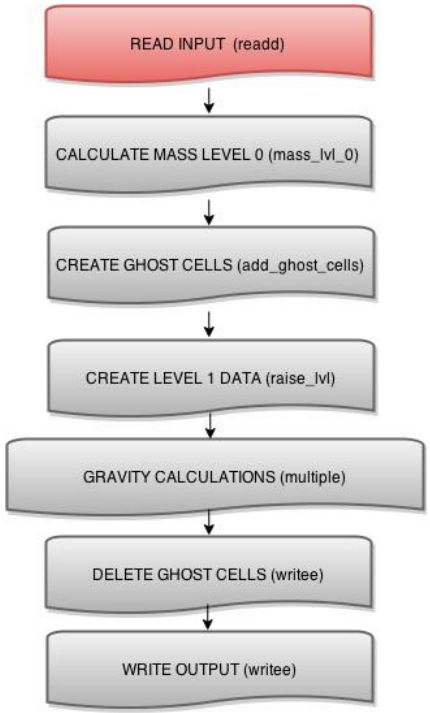
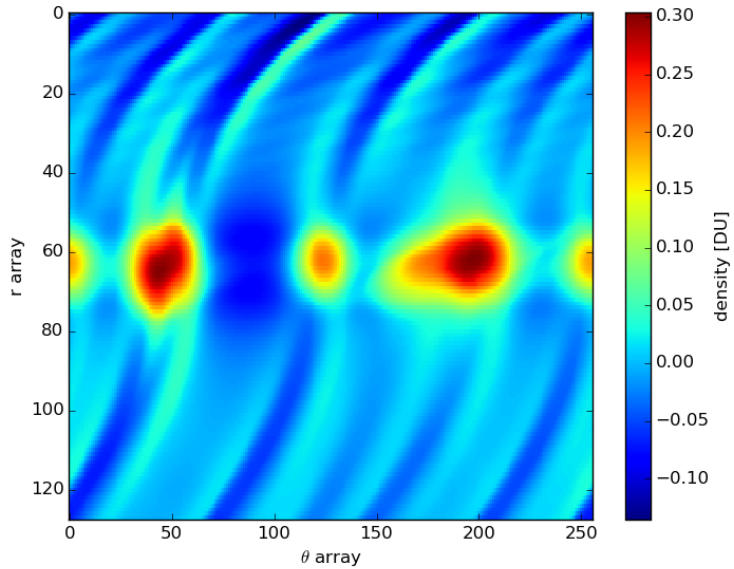


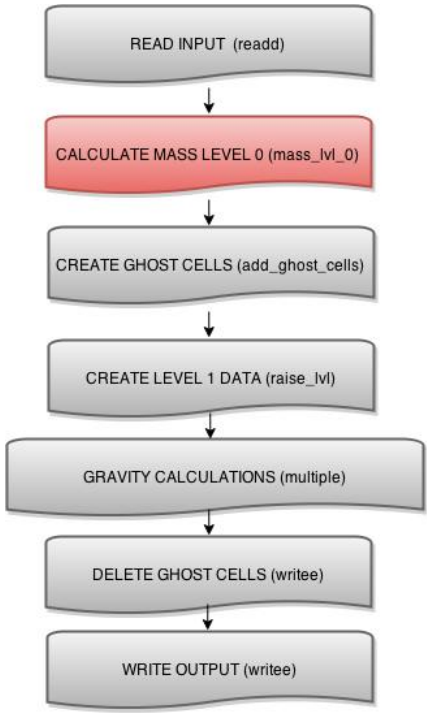
Figure : Refinement to the next lower level

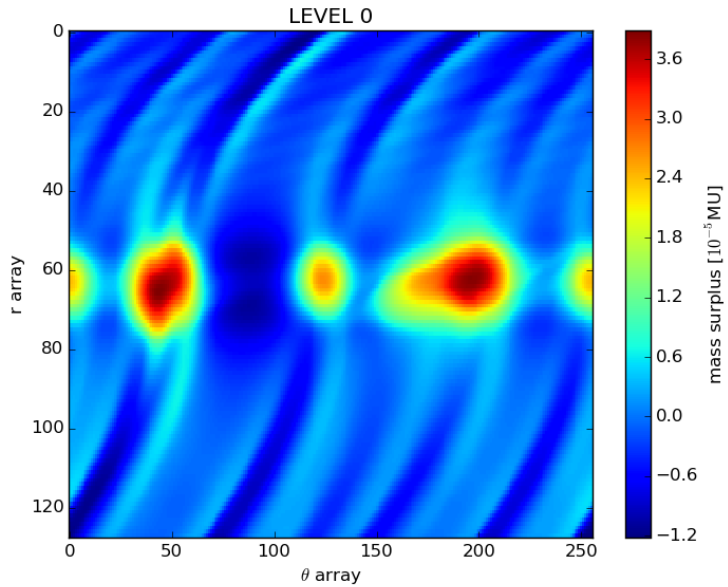
Code schema


READ INPUT (readd)
CALCULATE MASS LEVEL 0 (mass_lvl_0)
CREATE GHOST CELLS (add_ghost_cells)
CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS (multiple)
DELETE GHOST CELLS (writee)
WRITE OUTPUT (writee)

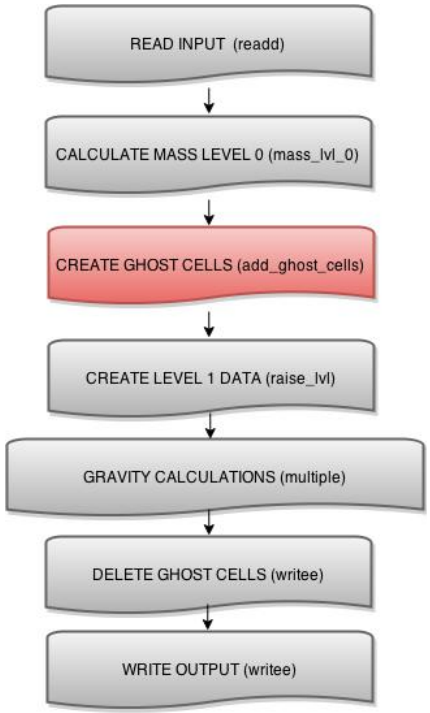

r array
θ array
density [DU]
0
20
40
60
80
100
120
0
50
100
150
200
250
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10

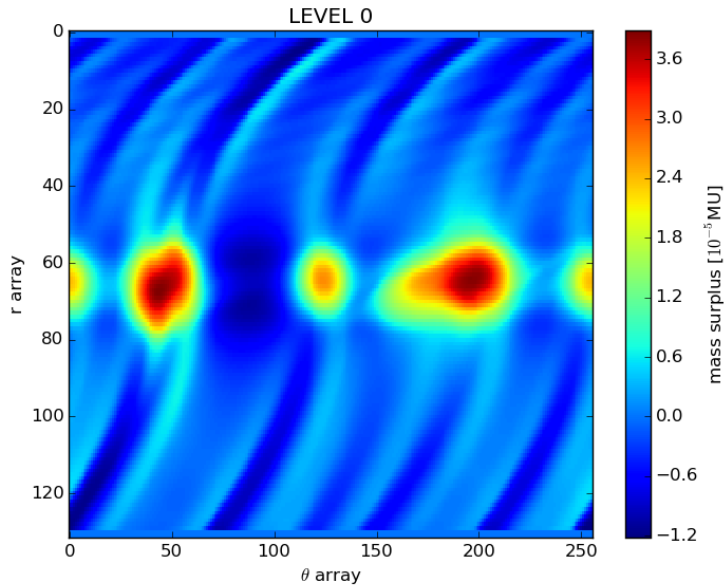
Code schema


READ INPUT (readd)
CALCULATE MASS LEVEL 0 (mass_lvl_0)
CREATE GHOST CELLS (add_ghost_cells)
CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS (multiple)
DELETE GHOST CELLS (writee)
WRITE OUTPUT (writee)

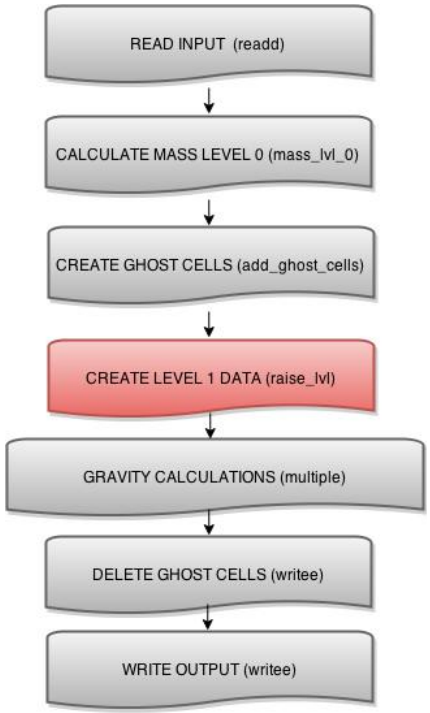

LEVEL 0
r array
θ array
mass surplus [10⁻⁵MU]
0
20
40
60
80
100
120
0
50
100
150
200
250
3.6
3.0
2.4
1.8
1.2
0.6
0.0
−0.6
−1.2

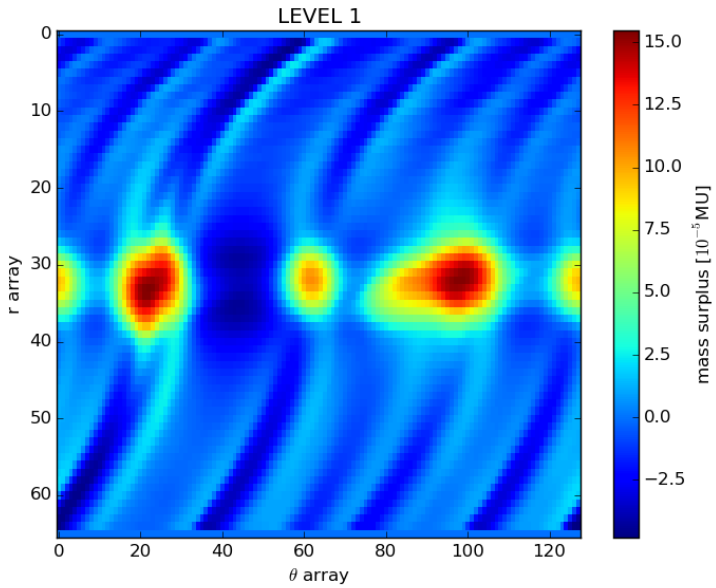
Code schema


READ INPUT (readd)
CALCULATE MASS LEVEL 0 (mass_lvl_0)
CREATE GHOST CELLS (add_ghost_cells)
CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS (multiple)
DELETE GHOST CELLS (writee)
WRITE OUTPUT (writee)

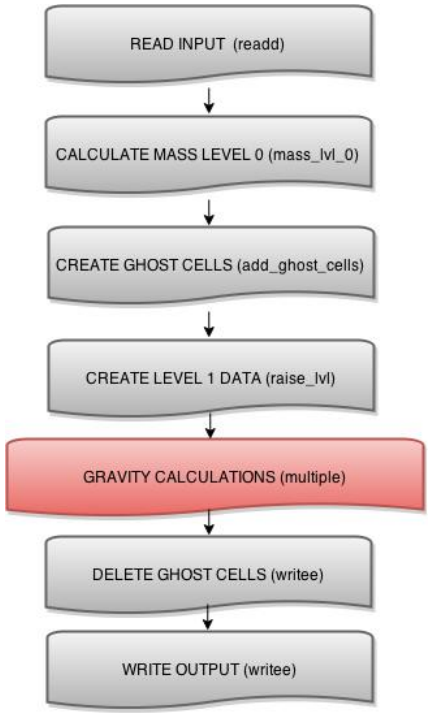

LEVEL 0
r array
θ array
mass surplus [10⁻⁵MU]
0
20
40
60
80
100
120
0
50
100
150
200
250
3.6
3.0
2.4
1.8
1.2
0.6
0.0
−0.6
−1.2

Code schema

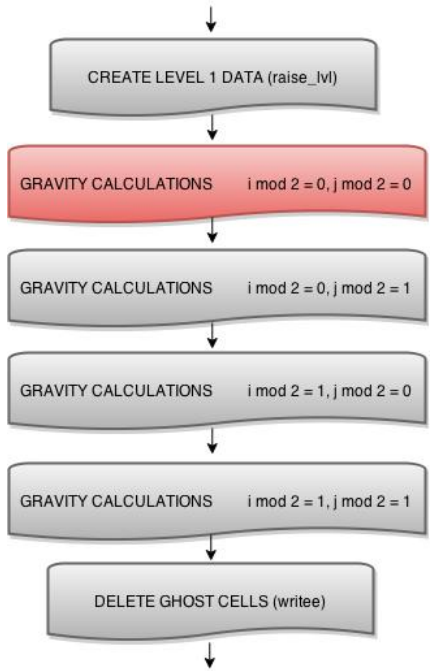

READ INPUT (readd)
CALCULATE MASS LEVEL 0 (mass_lvl_0)
CREATE GHOST CELLS (add_ghost_cells)
CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS (multiple)
DELETE GHOST CELLS (writee)
WRITE OUTPUT (writee)


LEVEL 1
r array
θ array
mass surplus [10⁻⁵MU]
0
10
20
30
40
50
60
0
20
40
60
80
100
120
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5

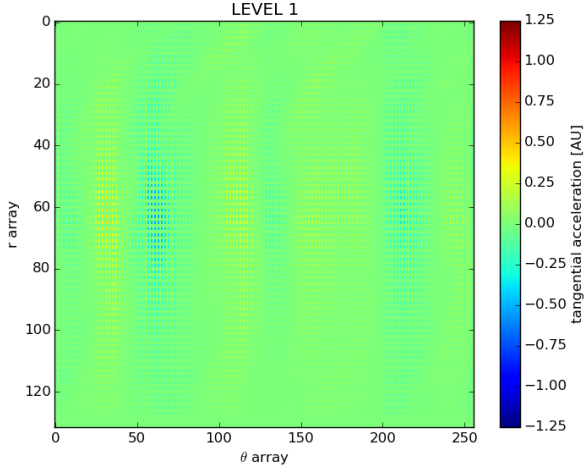
Code schema

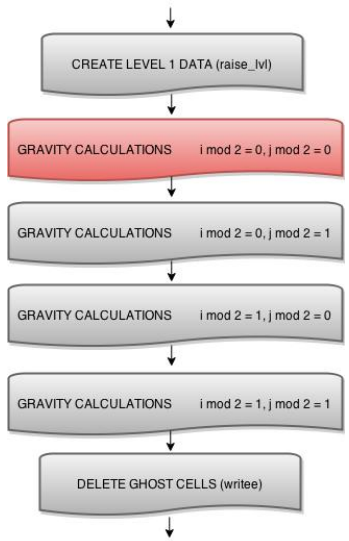

READ INPUT (readd)
CALCULATE MASS LEVEL 0 (mass_lvl_0)
CREATE GHOST CELLS (add_ghost_cells)
CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS (multiple)
DELETE GHOST CELLS (writee)
WRITE OUTPUT (writee)

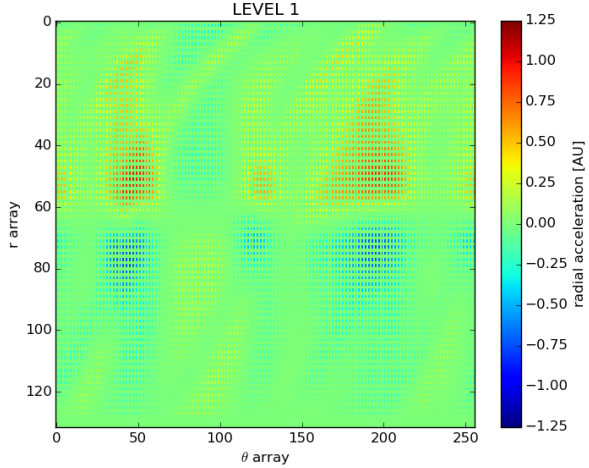
Code schema


CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 1
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 1
DELETE GHOST CELLS (writee)

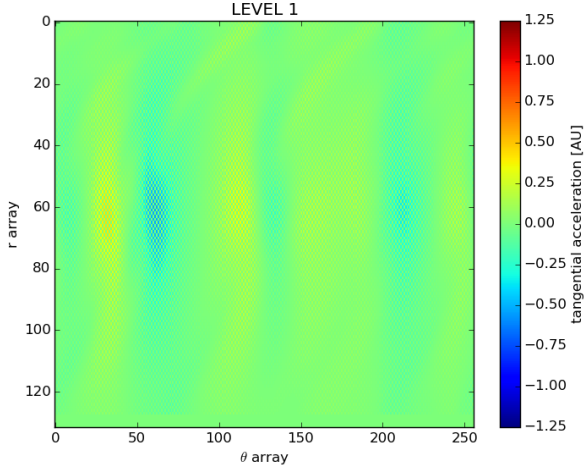
Code schema

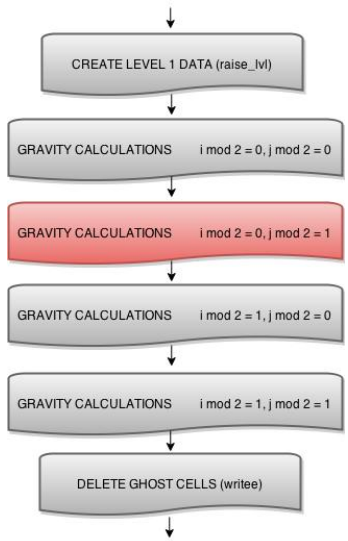

LEVEL 1
r array
θ array
tangential acceleration [AU]

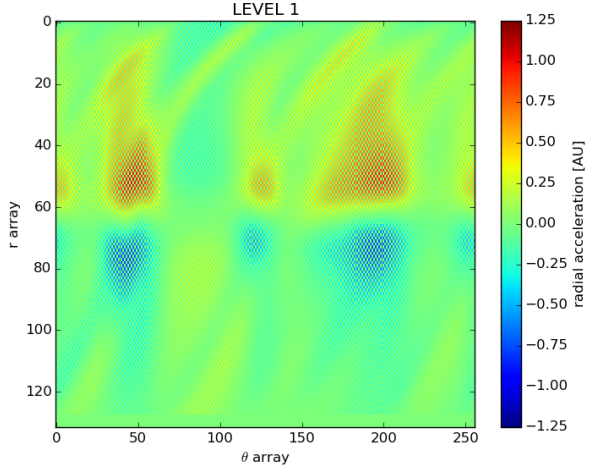

CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 1
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 1
DELETE GHOST CELLS (writee)


LEVEL 1
r array
θ array
radial acceleration [AU]

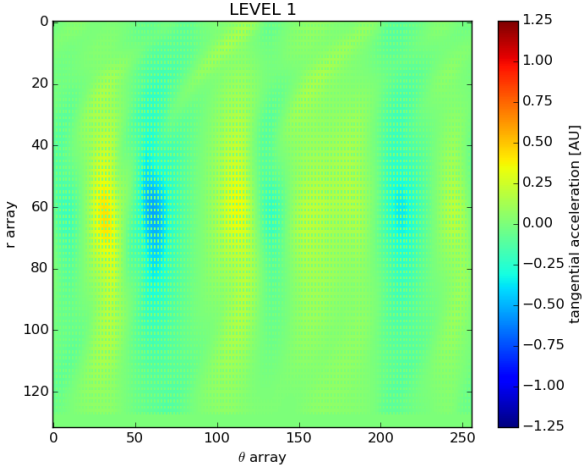
Code schema

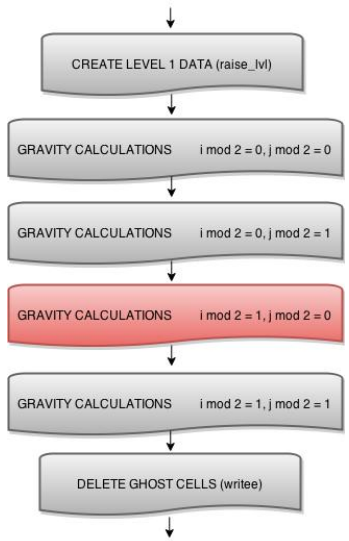

LEVEL 1
r array
θ array
tangential acceleration [AU]

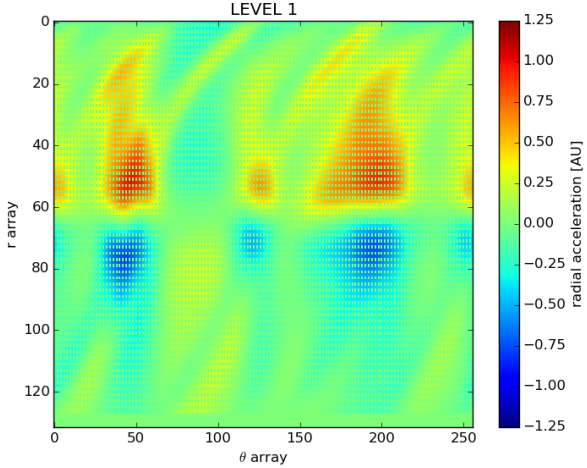

CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 1
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 1
DELETE GHOST CELLS (writee)


LEVEL 1
r array
θ array
radial acceleration [AU]

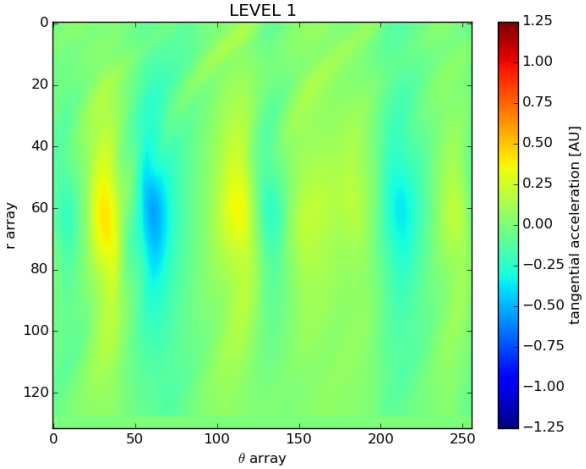
Code schema

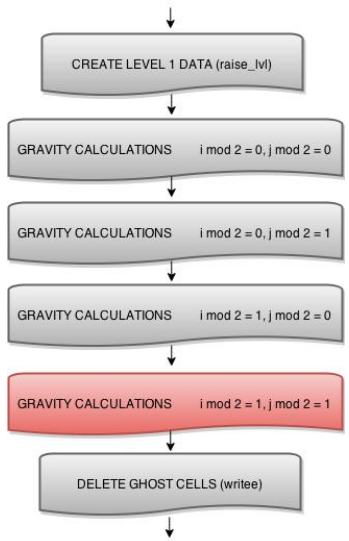

LEVEL 1
r array
θ array
tangential acceleration [AU]

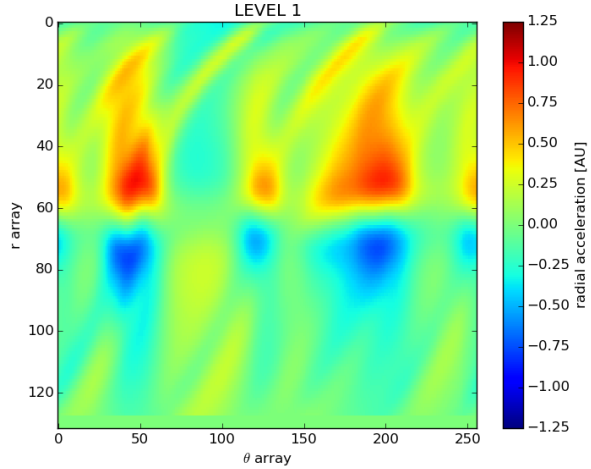

CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 1
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 1
DELETE GHOST CELLS (writee)


LEVEL 1
r array
θ array
radial acceleration [AU]

Code schema

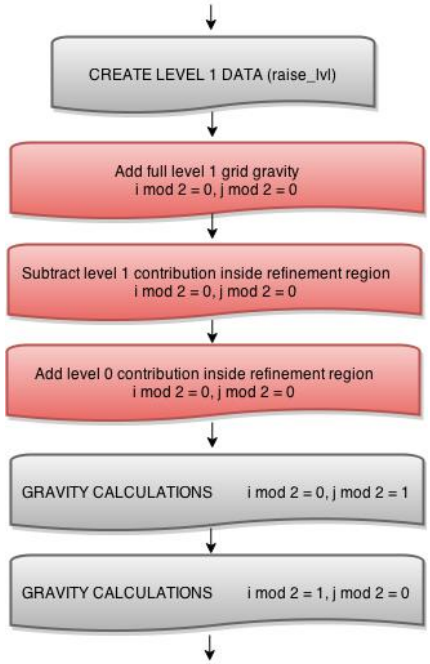

LEVEL 1
r array
θ array
tangential acceleration [AU]


CREATE LEVEL 1 DATA (raise_lvl)
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 0, j mod 2 = 1
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 0
GRAVITY CALCULATIONS i mod 2 = 1, j mod 2 = 1
DELETE GHOST CELLS (writee)


LEVEL 1
r array
θ array
radial acceleration [AU]

Code schema

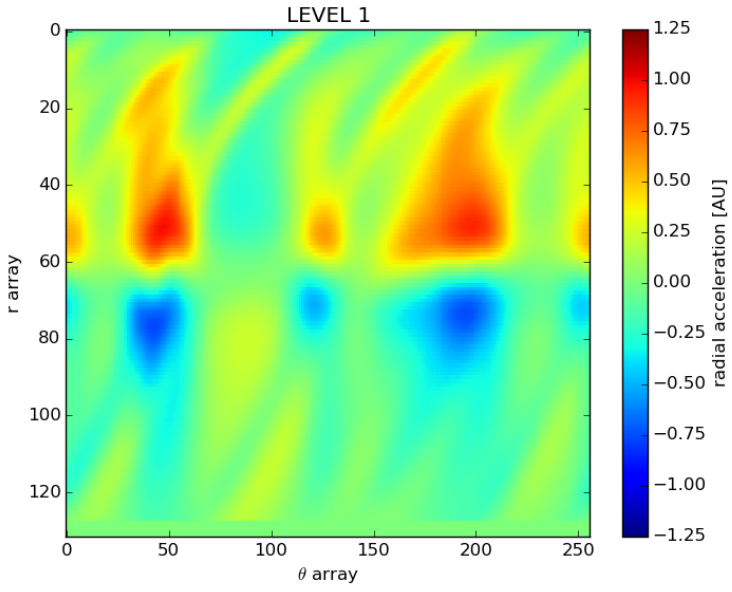
Version with refinement

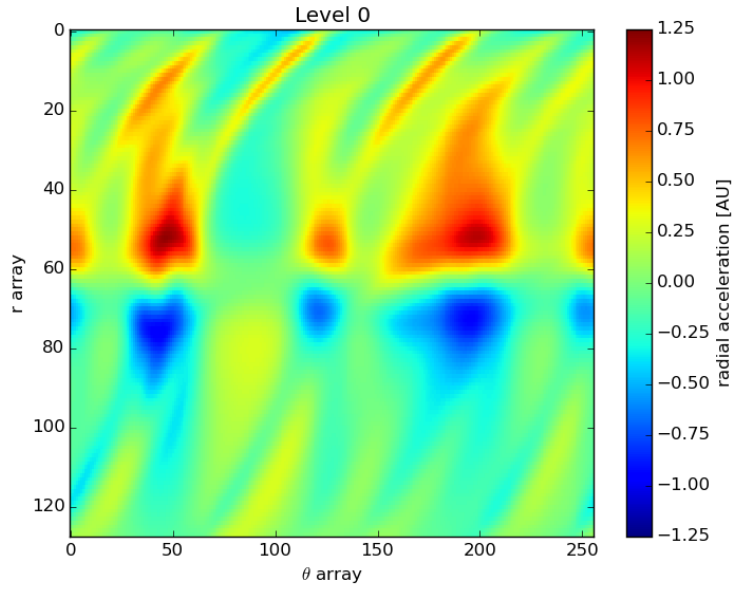


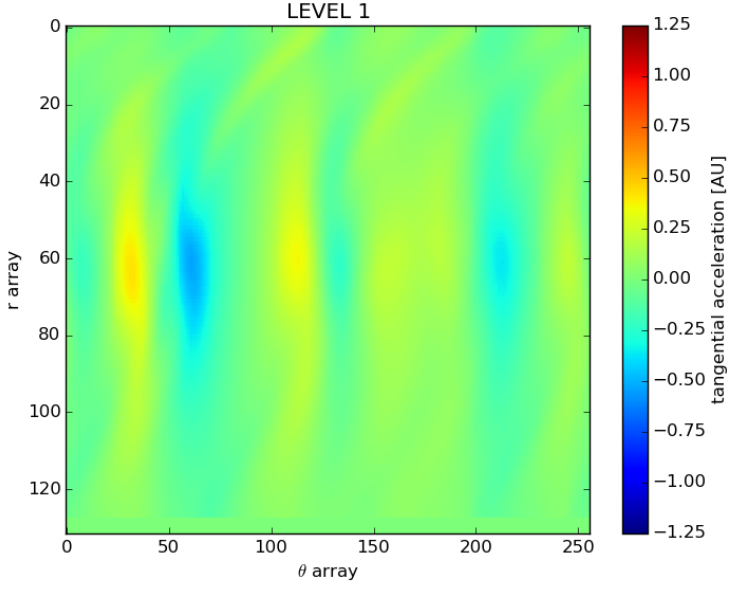
- avoids a series of "if" statements
- main cpu-time usage due to the level1 full calculations
- the larger the number of cells - the smaller the share of the other two

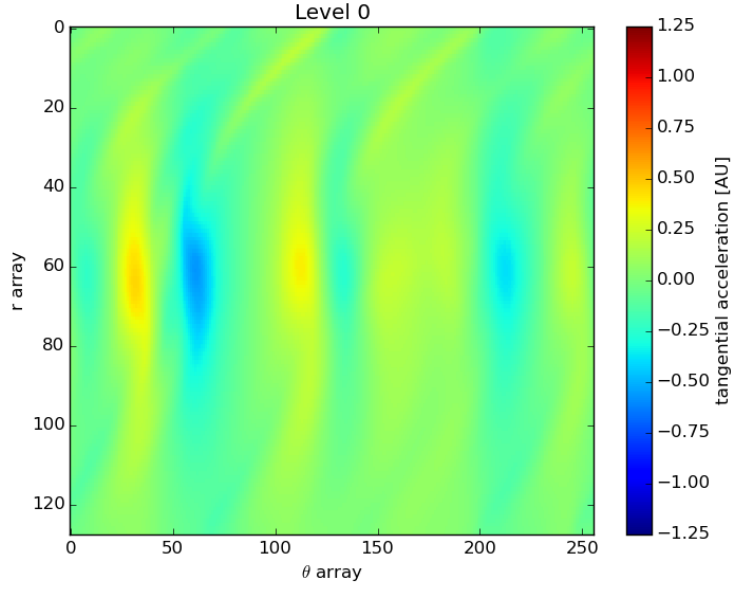
Results

LEVEL 0 vs LEVEL 1



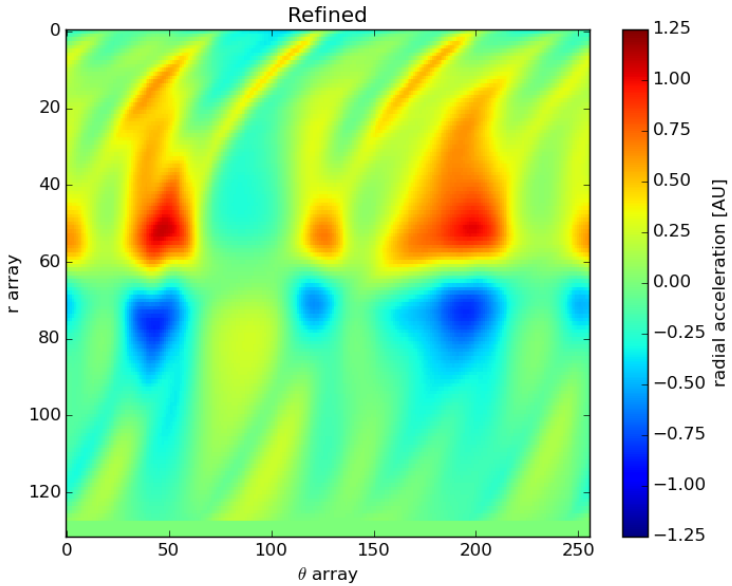


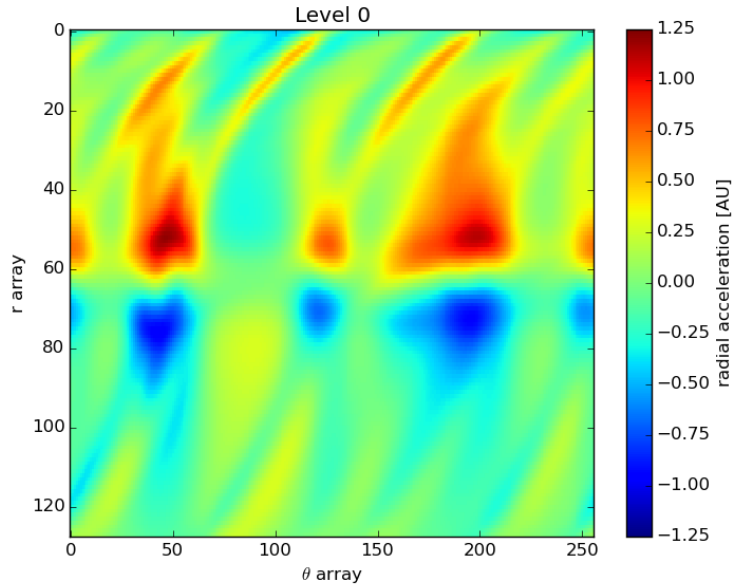


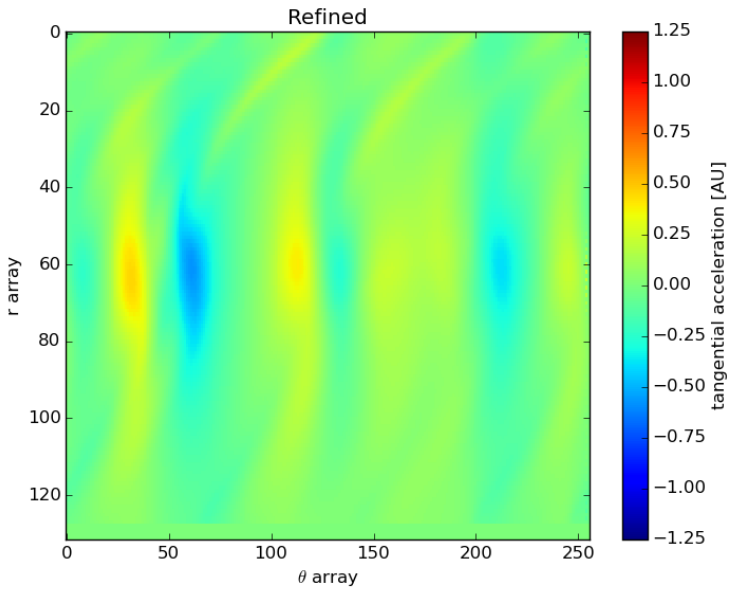


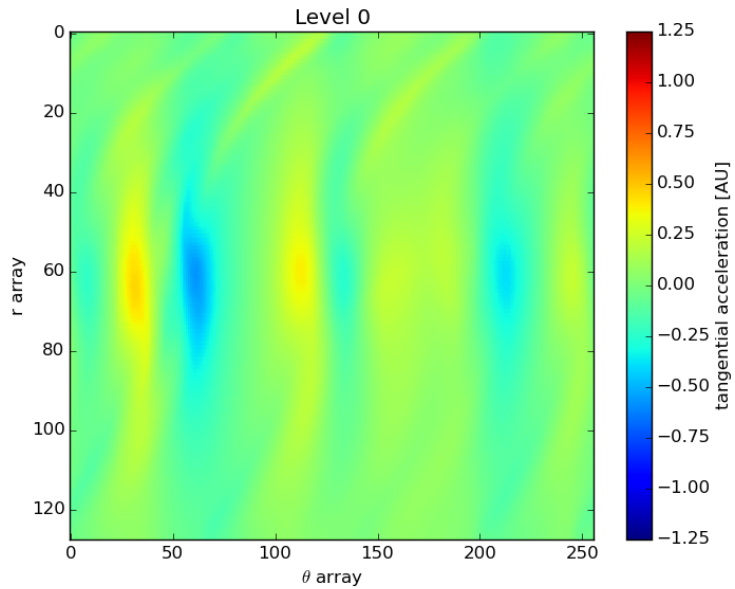
Results

LEVEL 0 vs Refined



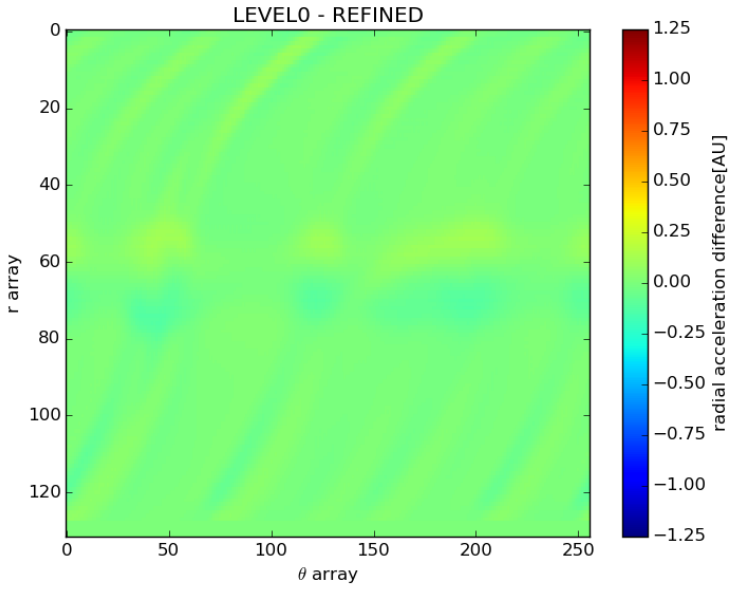


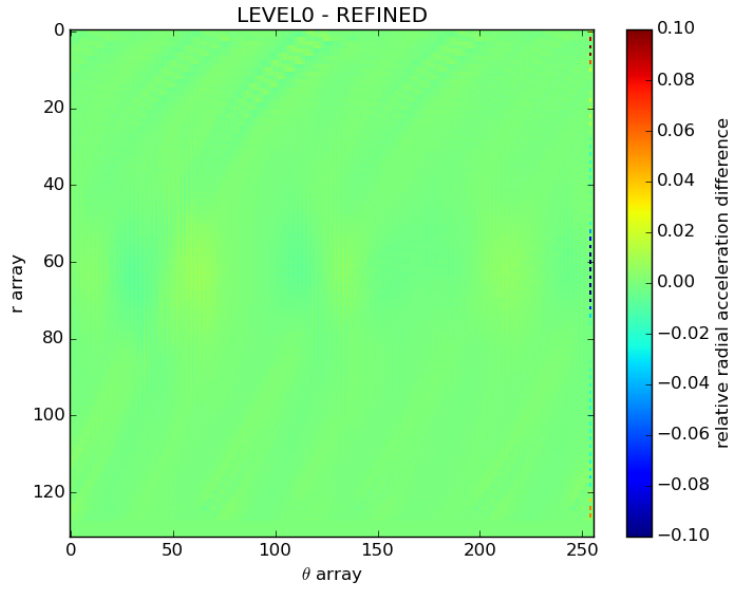


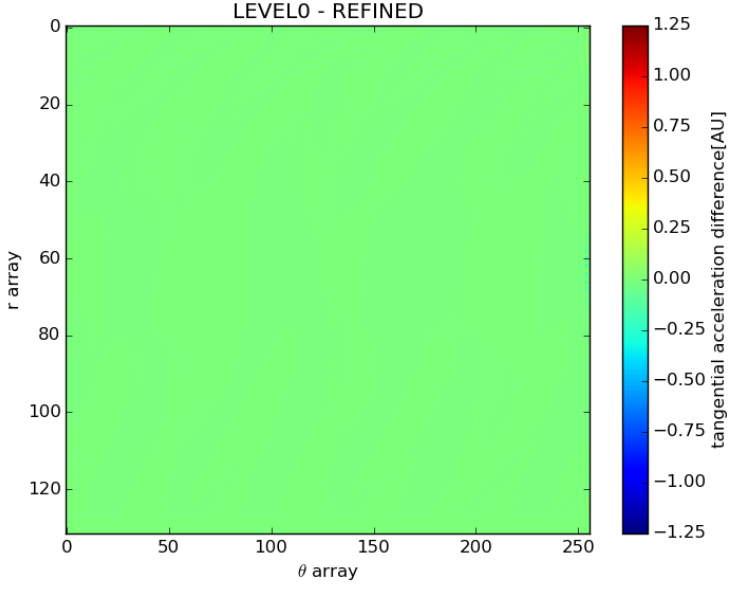


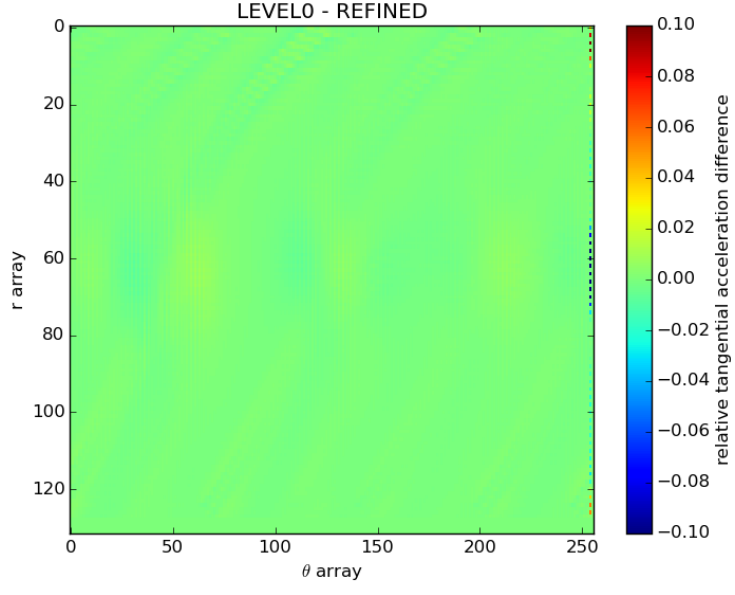
Results

LEVEL 0 vs Refined : Differences



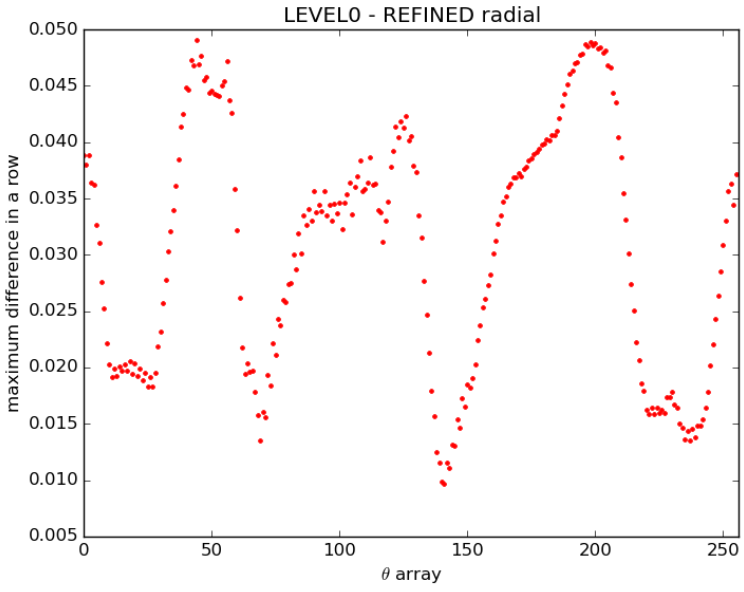


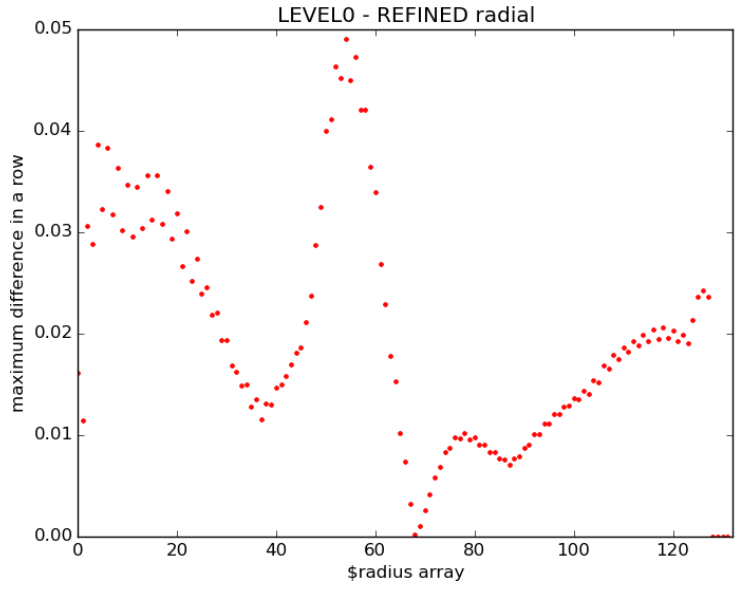


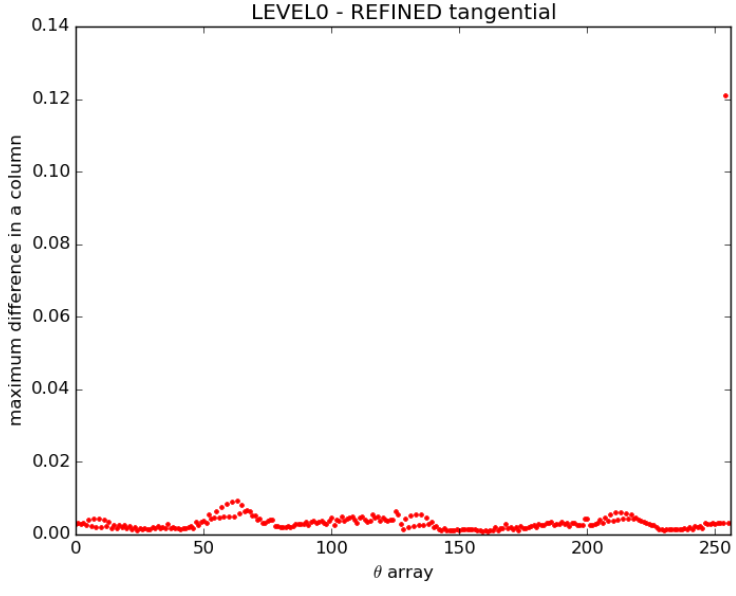


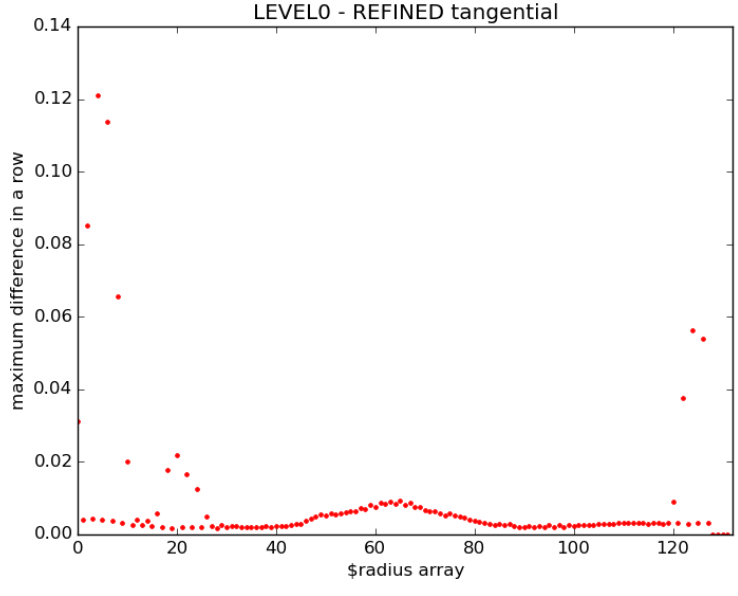
Results

LEVEL 0 vs Refined : Maximum differences







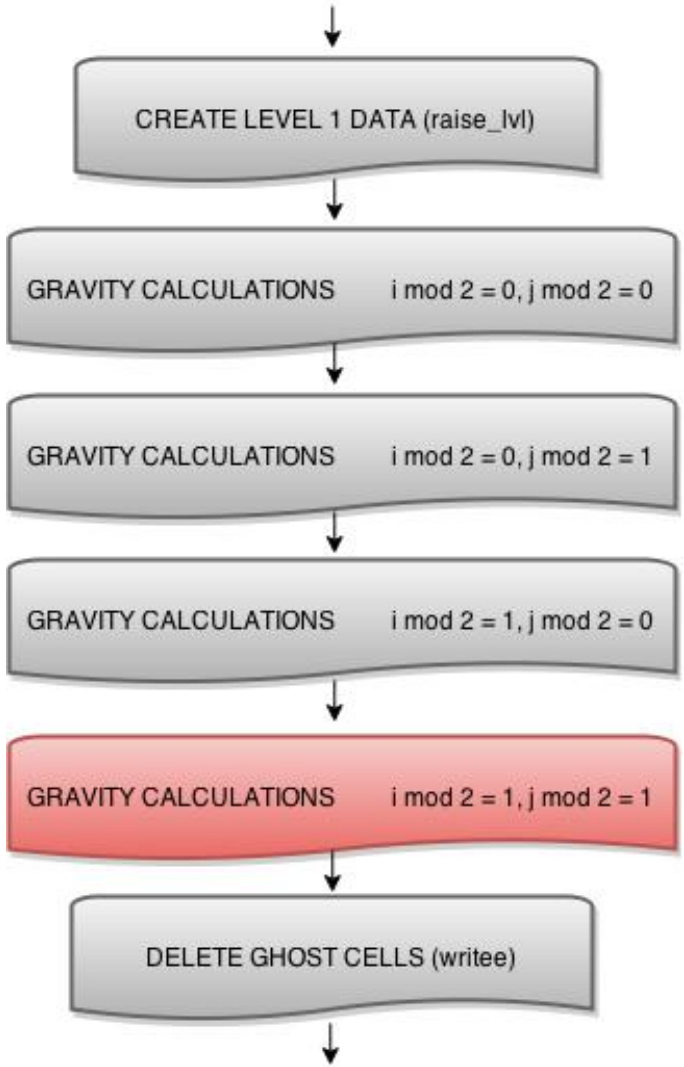


Time measurements

- 16 s level 0 optimization (March results) ;
- 61 s level 0 constrained for refinement method;
- 14 s level 1 with no refinement;
- 15 s level 1 with level 0 refinement;

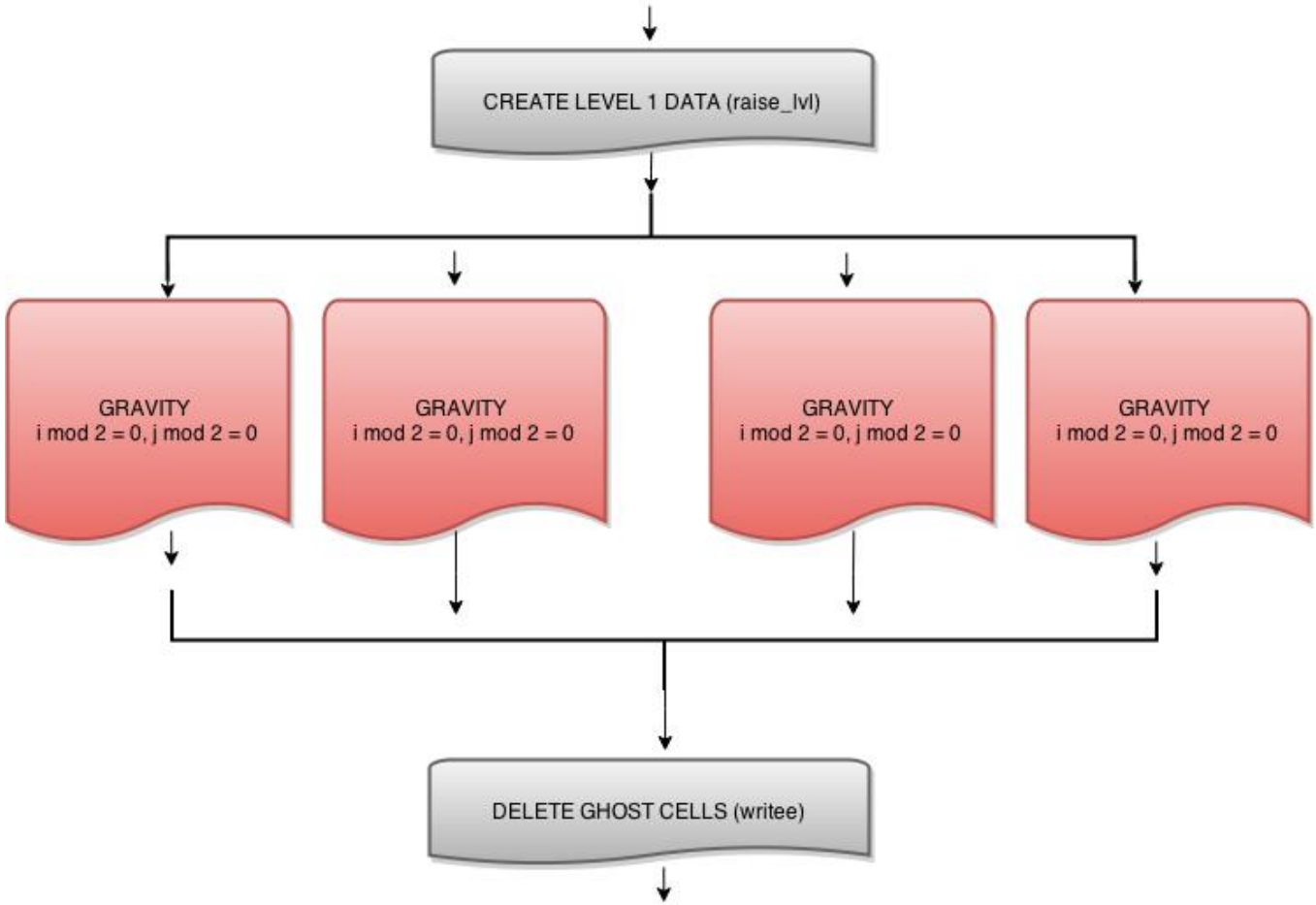
Towards a faster code

Using intra-node parallelism



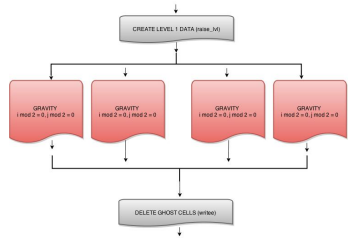
Towards a faster code

Using intra-node parallelism



Towards a faster code

Using intra-node parallelism



- run 4 independent tasks on shared memory ;
- could use i.e. OpenMP;
- simpler distribution of tasks;
- the inter-node MPI structure would not change;
- no MPI communication during one step calculations.
- reduce needed time by 4 to 3-4 s

Towards a faster code

How about using GPUs?

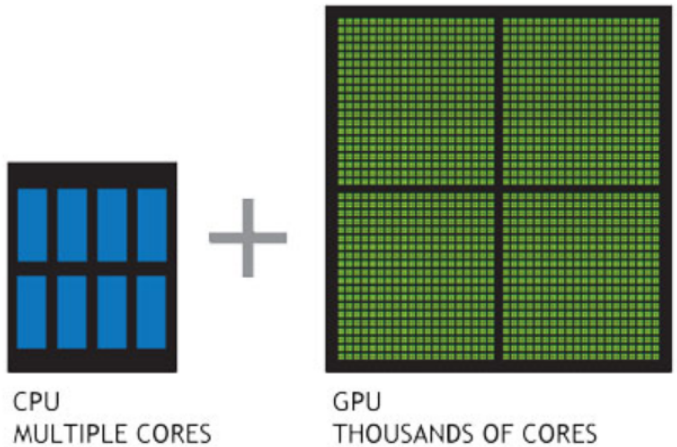


Figure : CPU vs GPU. (Source: NVIDIA)

- rewrite code in GPU languages (i.e. CUDA);
- the gravity calculation part would stay the same;
- the outer loops would need to be replaced to take advantage of the GPU;
- the inter-node MPI would not change;
- intra-node communication problems...

Towards a faster code;

Movement of the industry and scientific community

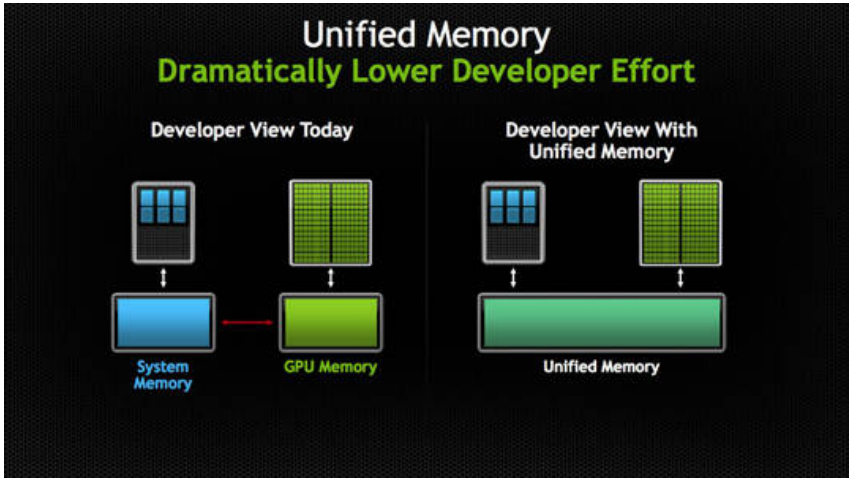


Figure : Future direction for hardware design. (Source: NVIDIA)

- avoid intra-node communication time-loss
- could reduce the needed time to under 1 s

The fortran code

```
program gravity
!Variable section
implicit none
INTEGER i, j, k, l

INTEGER, parameter :: nr = 128, nt = 256, lvl = 1, ng = 2

REAL ag, a, G, dr, dt, rad, den ,den_inv, comm, prod, nothing, c00, c01, c10, c11, massa
double precision t_init, t_end, timed
REAL, dimension(nr) :: radius
REAL, dimension(nt) :: theta, cos_center, cos_corner, sin_center, sin_corner
REAL, dimension(nr,nt) :: density, acc_r, acc_t
REAL, dimension(:,:), allocatable :: acc_rg, acc_tg
REAL, dimension(nr,nt) :: mass
REAL :: cos_diff, sin_diff
REAL:: ratio, ratio_f
REAL, dimension(1:2) :: acc

INTEGER :: nr_l, nt_l, nrg, ii, jj, jjj, lll
REAL, dimension(:) , allocatable:: radius_l, radius_g
REAL, dimension(:) , allocatable:: theta_l
REAL, dimension(:,:), allocatable :: mass_l, mass_g

REAL theta_ad, radius_ad, valu, radius_corn, radius_corn_2_inv, tl1, tl2
REAL c1, c2, c3, c4, unity, contr1, contr2, thetap, thetam
logical don

call readd(nr, nt, radius, theta, density)

call mass_lvl_0(nr, nt, radius, theta, density, mass)

call add_ghost_cells(nr, nt, radius, mass, ng, nrg, radius_g, mass_g)

call raise_lvl(0,0,lvl, nrg, nt, radius_g, theta, mass_g, radius_l, theta_l, mass_l, nr_
```