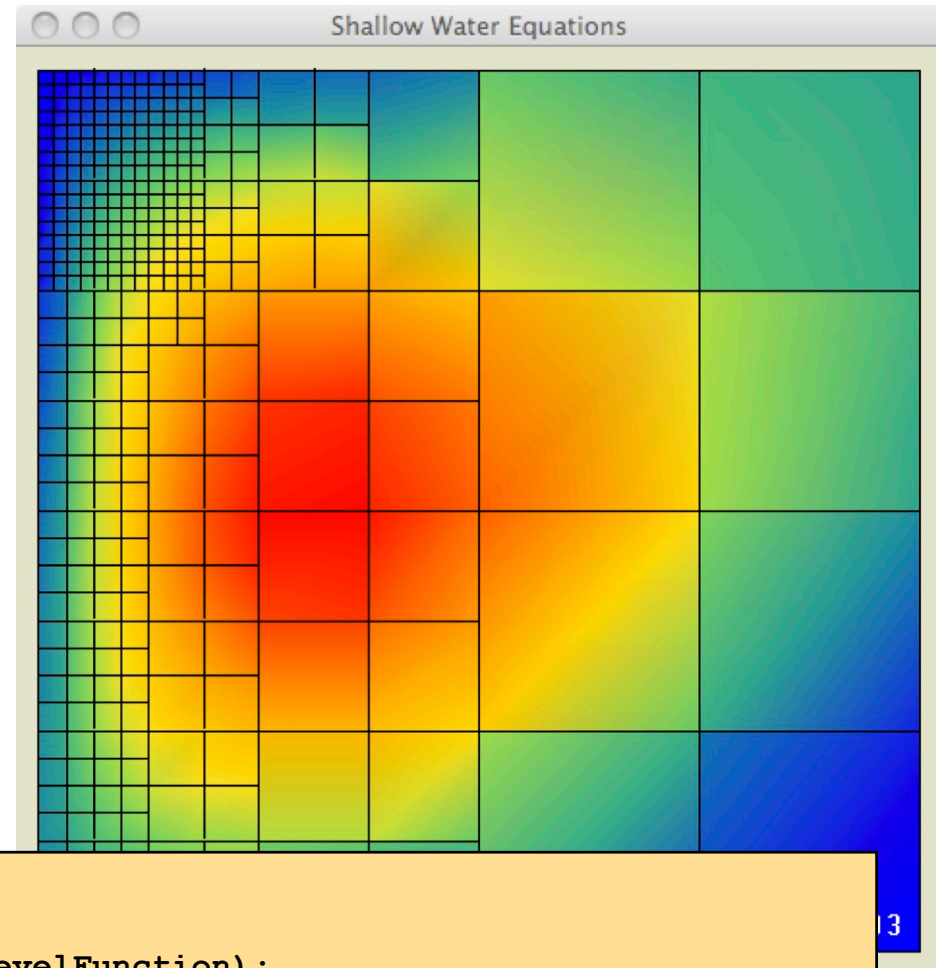
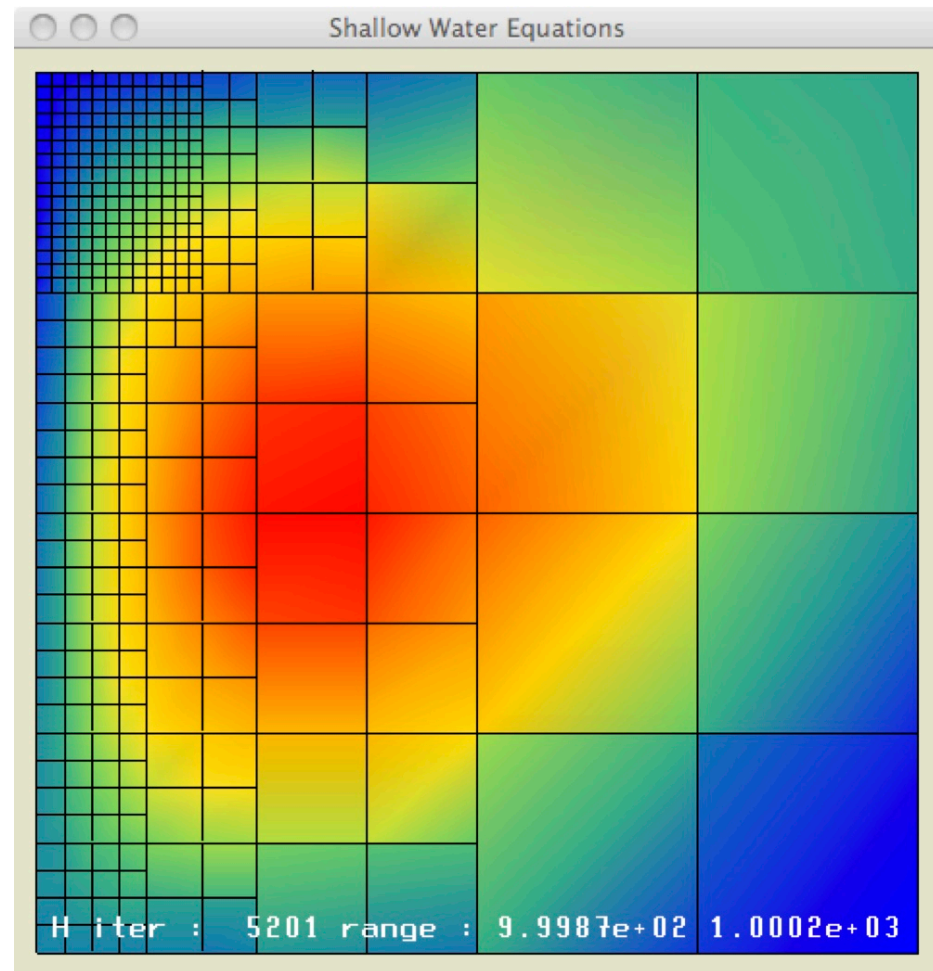


Using adaptive grids !



```
int main(int argc, char **argv)
{
    Grid *grid = gridNew(-5e5, -5e5, 1e6, levelFunction);
    gridPrint(grid);
    glQuadtreeMain(grid->root);
    Erk *erk = erkNew22();
    Swe *swe = sweNew(grid);
    glSweRun(grid, swe, erk, 50);
    gridFree(grid);
    erkFree(erk);
    sweFree(swe);
    return 0;
}
```

Homework 4



- **Computing Stommel problem on this mesh !**
- **Generating intermediate lists**
- **Updating the ghost cells**
- **Adapting the right hand side of the integration scheme**

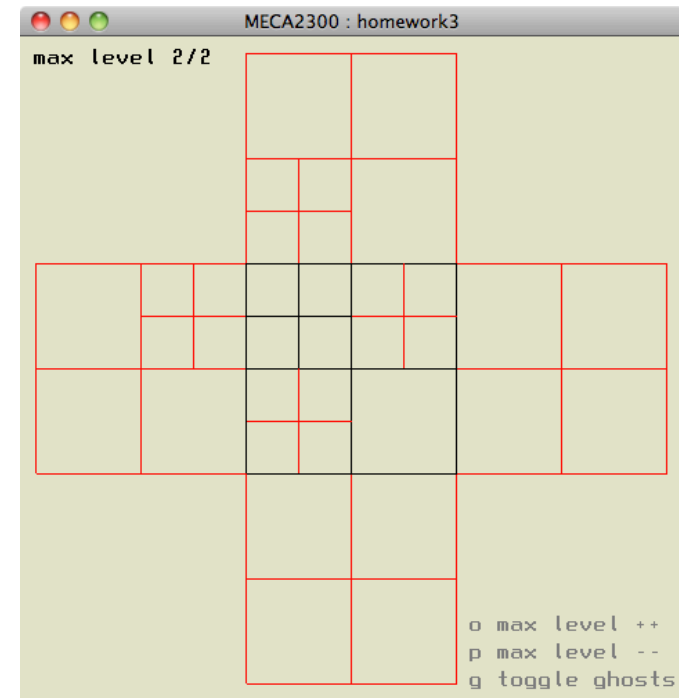
Defining the grid !

```
typedef struct _Cell{
    int i, j, level;
    struct _Cell *child[4];
    struct _Cell *neighbour[4];
    struct _Cell *parent;
    CellFlag flag;
    int id;
} Cell;
```

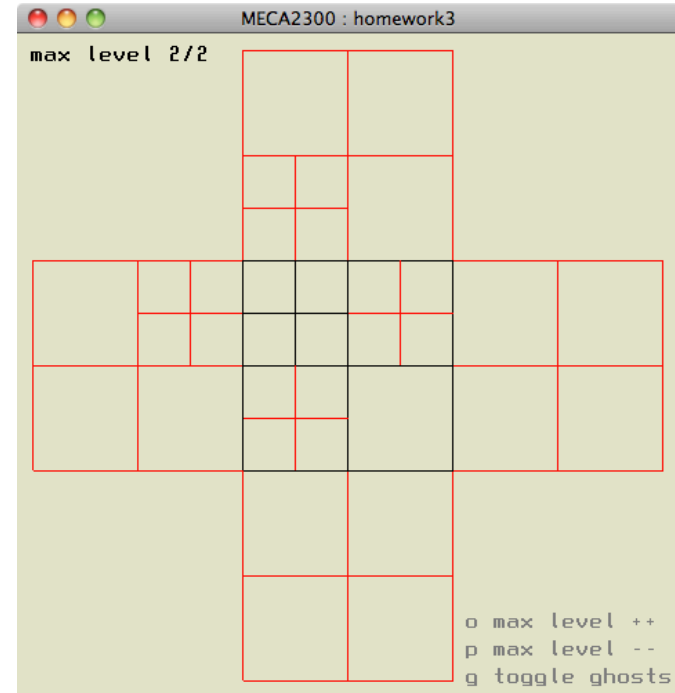
```
typedef enum {INACTIVE_INTERIOR=0, INACTIVE_BOUNDARY,
              ACTIVE_LEAF, ACTIVE_PARENT} CellFlag;
typedef enum {HORIZONTAL= 0, VERTICAL} FaceOrientation;
typedef enum {BOTTOM = 0, RIGHT = 1, TOP = 2, LEFT = 3} Side;
```

```
typedef struct _Face {
    Cell *cell[2];
    FaceOrientation orientation;
} Face;
```

```
typedef struct {
    Cell *cell[2];
    Side side;
} Boundary;
```



The grid :-)



```
typedef struct {  
    Cell *root;  
    int fieldSize;  
    Cell **active;  
    Cell **activeParentOfGhost;  
    Cell **ghostParent;  
    Cell **ghostChild;  
    Boundary *boundary;  
    Face *face;  
    double x0, y0, l;  
    double *dx;  
} Grid;
```

The cells :-)

=== Active-leaf cells

```
0 ( 0 0) (level=1) (active-leaf)
7 ( 1 0) (level=1) (active-leaf)
11 ( 0 2) (level=2) (active-leaf)
13 ( 1 2) (level=2) (active-leaf)
14 ( 0 3) (level=2) (active-leaf)
17 ( 1 3) (level=2) (active-leaf)
19 ( 1 1) (level=1) (active-leaf)
```

=== Active-leaf parent of ghosts cells

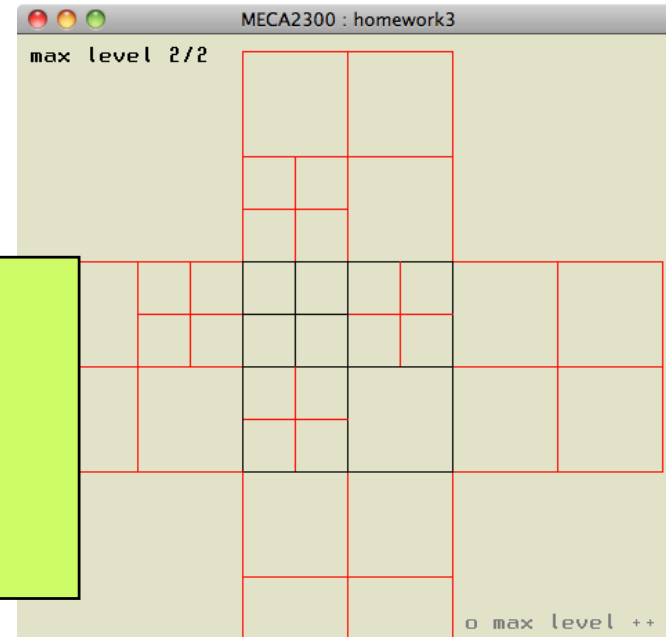
```
0 ( 0 0) (level=1) (active-leaf)
19 ( 1 1) (level=1) (active-leaf)
```

=== Cells with active-leaf childs

```
10 ( 0 1) (level=1) (active-parent)
```

=== Cells with active-leaf parents

```
3 ( 0 1) (level=2) (inactive-interior)
4 ( 1 1) (level=2) (inactive-interior)
20 ( 2 2) (level=2) (inactive-interior)
22 ( 2 3) (level=2) (inactive-interior)
```



```
typedef struct {
    Cell *root;
    int fieldSize;
    Cell **active;
    Cell **activeParentOfGhost;
    Cell **ghostParent;
    Cell **ghostChild;
    Boundary *boundary;
    Face *face;
    double x0, y0, l;
    double *dx;
} Grid;
```

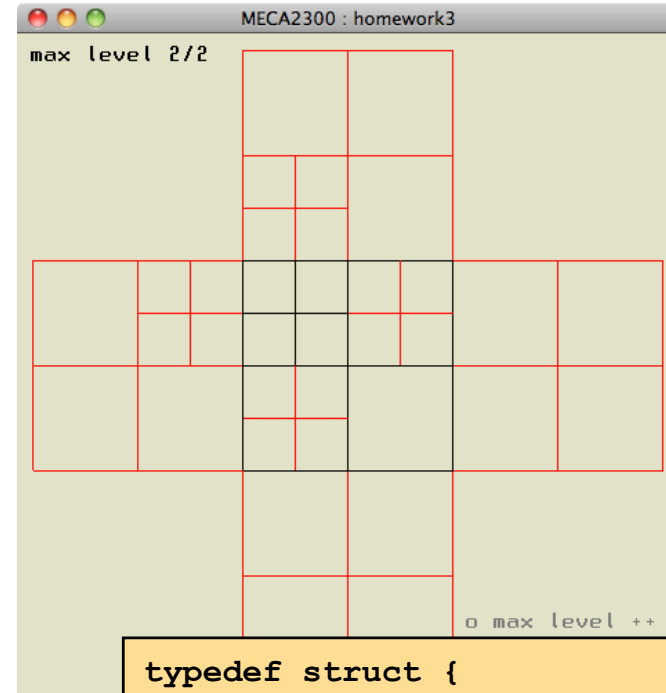
The faces :-)

=== Face list

```
( 5 0) (horizontal)
( 0 7) (vertical)
( 6 0) (vertical)
( 8 7) (horizontal)
( 7 9) (vertical)
( 3 11) (horizontal)
(11 13) (vertical)
(12 11) (vertical)
( 4 13) (horizontal)
(13 20) (vertical)
(11 14) (horizontal)
(14 17) (vertical)
(14 15) (horizontal)
(16 14) (vertical)
(13 17) (horizontal)
(17 22) (vertical)
(17 18) (horizontal)
( 7 19) (horizontal)
(19 24) (vertical)
(19 25) (horizontal)
```

=== Boundary list

```
( 0 5) (bottom)
( 0 6) (left)
( 7 8) (bottom)
( 7 9) (right)
(11 12) (left)
(14 15) (top)
(14 16) (left)
(17 18) (top)
(19 24) (right)
(19 25) (top)
```

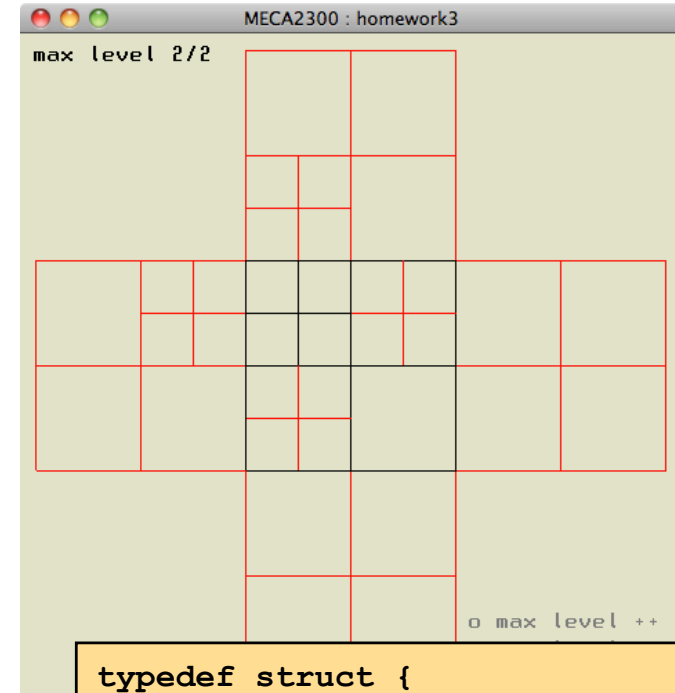


```
typedef struct {
    Cell *root;
    int fieldSize;
    Cell **active;
    Cell **activeParentOfGhost;
    Cell **ghostParent;
    Cell **ghostChild;
    Boundary *boundary;
    Face *face;
    double x0, y0, l;
    double *dx;
} Grid;
```

All cells :-)

```
=== Number of unknowns : 26
=== All cells
-1 ( 0 0) (level=0) (active-parent)
 0 ( 0 0) (level=1) (active-leaf)
 1 ( 0 0) (level=2) (inactive-interior)
 2 ( 1 0) (level=2) (inactive-interior)
 3 ( 0 1) (level=2) (inactive-interior)
 4 ( 1 1) (level=2) (inactive-interior)
 7 ( 1 0) (level=1) (active-leaf)
10 ( 0 1) (level=1) (active-parent)
11 ( 0 2) (level=2) (active-leaf)
13 ( 1 2) (level=2) (active-leaf)
14 ( 0 3) (level=2) (active-leaf)
17 ( 1 3) (level=2) (active-leaf)
19 ( 1 1) (level=1) (active-leaf)
20 ( 2 2) (level=2) (inactive-interior)
21 ( 3 2) (level=2) (inactive-interior)
22 ( 2 3) (level=2) (inactive-interior)
23 ( 3 3) (level=2) (inactive-interior)
-1 ( 0 -1) (level=0) (inactive-boundary)
-1 ( 0 -2) (level=1) (inactive-boundary)
-1 ( 1 -2) (level=1) (inactive-boundary)
 5 ( 0 -1) (level=1) (inactive-boundary)
```

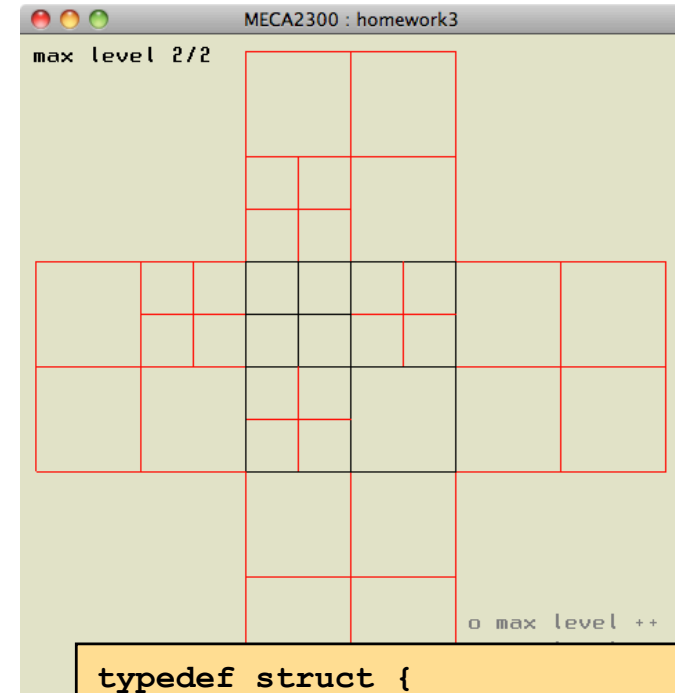
...



```
typedef struct {
    Cell *root;
    int fieldSize;
    Cell **active;
    Cell **activeParentOfGhost;
    Cell **ghostParent;
    Cell **ghostChild;
    Boundary *boundary;
    Face *face;
    double x0, y0, l;
    double *dx;
} Grid;
```

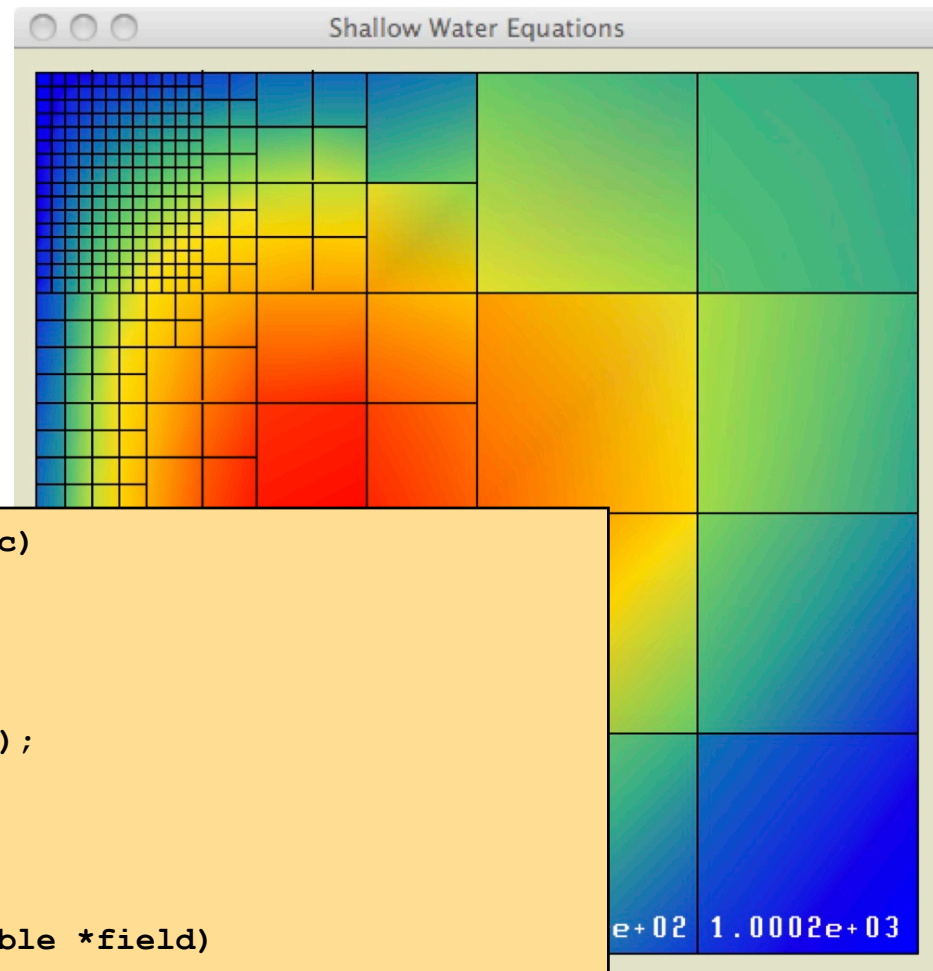
All cells :-)

```
=== Number of unknowns : 26
=== All cells
...
  5 ( 0 -1) (level=1) (inactive-boundary)
  8 ( 1 -1) (level=1) (inactive-boundary)
 -1 ( 1  0) (level=0) (inactive-boundary)
  9 ( 2  0) (level=1) (inactive-boundary)
 -1 ( 3  0) (level=1) (inactive-boundary)
 24 ( 2  1) (level=1) (inactive-boundary)
 -1 ( 3  1) (level=1) (inactive-boundary)
 -1 ( 0  1) (level=0) (inactive-boundary)
 -1 ( 0  2) (level=1) (inactive-boundary)
 15 ( 0  4) (level=2) (inactive-boundary)
 18 ( 1  4) (level=2) (inactive-boundary)
 -1 ( 0  5) (level=2) (inactive-boundary)
 -1 ( 1  5) (level=2) (inactive-boundary)
 25 ( 1  2) (level=1) (inactive-boundary)
 -1 ( 0  3) (level=1) (inactive-boundary)
 -1 ( 1  3) (level=1) (inactive-boundary)
 -1 (-1  0) (level=0) (inactive-boundary)
 -1 (-2  0) (level=1) (inactive-boundary)
  6 (-1  0) (level=1) (inactive-boundary)
 -1 (-2  1) (level=1) (inactive-boundary)
 -1 (-1  1) (level=1) (inactive-boundary)
 -1 (-2  2) (level=2) (inactive-boundary)
 12 (-1  2) (level=2) (inactive-boundary)
 -1 (-2  3) (level=2) (inactive-boundary)
 16 (-1  3) (level=2) (inactive-boundary)
```



```
typedef struct {
    Cell *root;
    int fieldSize;
    Cell **active;
    Cell **activeParentOfGhost;
    Cell **ghostParent;
    Cell **ghostChild;
    Boundary *boundary;
    Face *face;
    double x0, y0, l;
    double *dx;
} Grid;
```


Homework 4



```
void gridFillVectors(Grid *q, Cell *c)
{
    int i;
    for (i = 0; i < 4; ++i) {
        if (c->child[i]) {
            gridFillVectors(q, c->child[i]);
        }
    }
}

void gridUpdateGhost(Grid *grid, double *field)
{
}

void sweCompute(Swe *swe, double *sol, double t, double *f)
{
}
```