

- A HiG-Tree (*Hierarchical Grid Tree*) é uma estrutura de dados em árvore usada em malhas adaptativas, que permite refinamento, dividindo o espaço recursivamente, economizando memória e tempo de processamento.
- Nos casos 2D e 3D, chamamos de *quadtree* e *octree*, respectivamente.

Exemplo de HiG-Tree

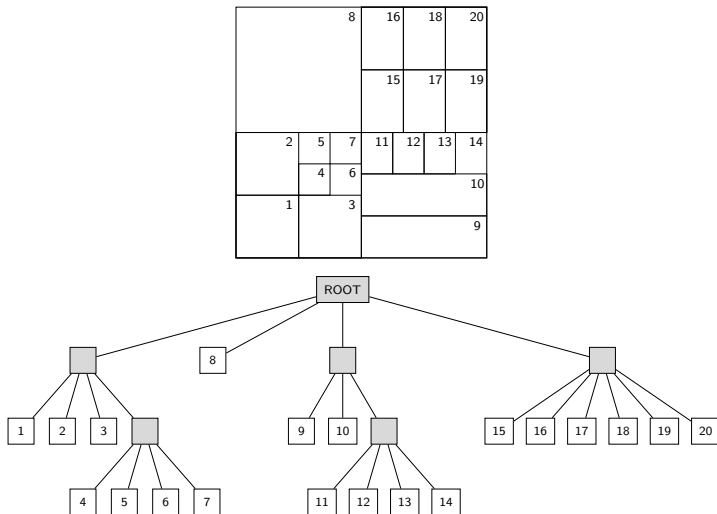


Figure 1.1: Exemplo de malha 2D e sua *quadtree* associada.

```
typedef struct hig_cell {  
    Point lowpoint; ----- ( $x_{1min}, x_{2min}, \dots, x_{mmin}$ )  
    Point highpoint; ----- ( $x_{1max}, x_{2max}, \dots, x_{mmax}$ )  
    int numcells[DIM]; ----- ( $n_1, n_2, \dots, n_m$ )  
    struct hig_cell * parent; -- célula-pai  
    int posinparent; ----- posição em relação ao pai  
    struct hig_cell **children; - células-filho  
    uniqueid ids[NUMIDS]; ----- id de cada célula/faceta  
} hig_cell;
```

Facetas da HiG-Tree

```
typedef struct hig_facet {  
    hig_cell *c; ----- célula associada  
    int dim; ----- m-1  
    int dir; ----- {0,1}  
} hig_facet;
```

dir=0: Facetas geradas pelas arestas que contém o lowpoint.

dir=1: Facetas geradas pelas arestas que contém o highpoint.

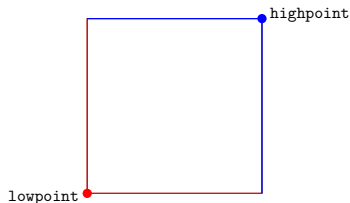


Figure 1.2: Facetas de uma célula 2D.

Visualização das malhas

Escrevemos a malha em formato AMR (*Adaptive Mesh Refinement*) como entrada, por exemplo:

```
0.0 8.0 -1.0 1.0 ----- lowpoint highpoint = xmin ymin xmax ymax
2 ----- numlevels
0.05 0.05 1 ----- (level 1) delta numpatches = delta_x delta_y numpatches
1 1 160 40 ----- initcell patchsize
0.025 0.025 4 ----- (level 2) delta numpatches = delta_x delta_y numpatches
1 1 320 16 ----- initcell patchsize
1 65 320 16 ----- initcell patchsize
1 17 16 48 ----- initcell patchsize
305 17 16 48 ----- initcell patchsize
```

então, geramos um arquivo de saída em formato VTK (*Visualization Toolkit*) com `higtree/src/higtree-io.c` e visualizamos com o ParaView.

Visualização das malhas

```
0.0 8.0 -1.0 1.0
1
0.05 0.05 1
1 1 160 40
```

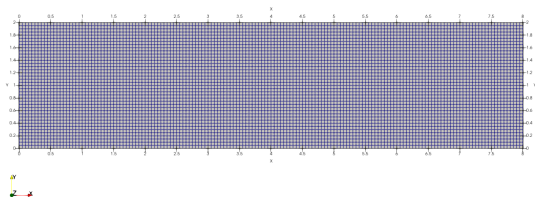


Figure 1.3: Exemplo de malha 2D com 1 nível de refinamento.

Visualização das malhas

0.0 8.0 -1.0 1.0

2

0.05 0.05 1

1 1 160 40

0.025 0.025 4

1 1 320 16

1 65 320 16

1 17 16 48

305 17 16 48

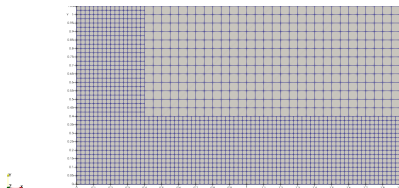
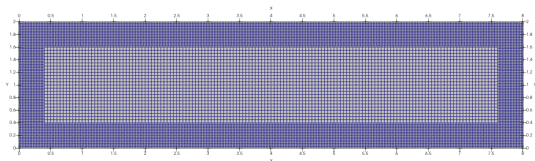


Figure 1.4: Exemplo de malha 2D com 2 níveis de refinamento.

Visualização das malhas

0.0 8.0 -1.0 1.0

3

0.05 0.05 1

1 1 160 40

0.025 0.025 4

1 1 320 16

1 65 320 16

1 17 16 48

305 17 16 48

0.0125 0.0125 4

1 1 640 16

1 145 640 16

1 17 16 128

625 17 16 128

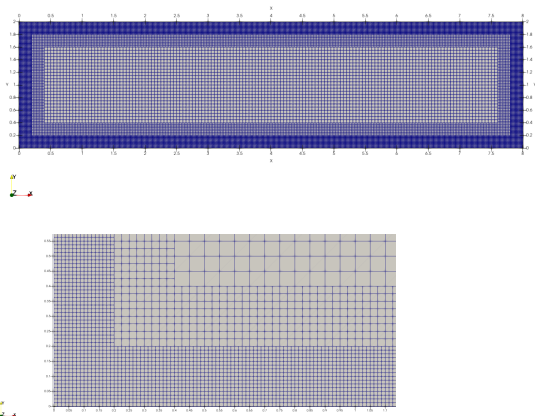


Figure 1.5: Exemplo de malha 2D com 3 níveis de refinamento.

Visualização das malhas

```
0.0 8.0 -1.0 1.0
3
0.05 0.05 1
1 1 160 40
0.025 0.025 4
1 1 320 16
1 65 320 16
1 17 16 48
305 17 16 48
0.0125 0.0125 4
1 1 640 16
1 145 640 16
1 17 16 128
625 17 16 128
```

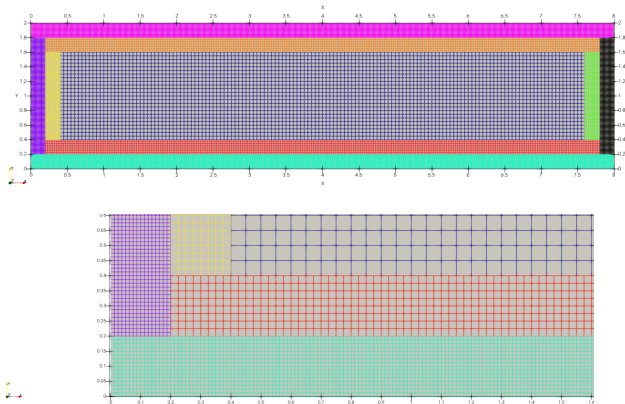


Figure 1.6: Exemplo de malha 2D com 3 níveis de refinamento.