Parallel unstructured mesh adaptation based on iterative remeshing and repartitioning

L. Cirrottola, A. Froehly

Inria Bordeaux - Sud-Ouest

luca.cirrottola@inria.fr

Context

Sequential remeshing ?

```
## Error: unable to allocate hash table.
## Check the mesh size or increase maximal authorized memory with the -m option.
```

Motivation 1

Exploiting *distributed memory* computer architectures.

Motivation 2

Exploiting *multicore* computer architectures.

Parallel remeshing algorithm

Design choice 1

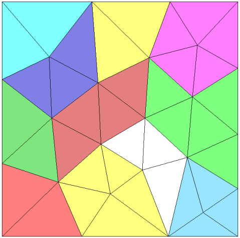
Modularity : Re-use sequential remesher with fixed parallel interfaces.

- Iterative approach :
  1. Input mesh.
  2. *Remesh* with fixed parallel interfaces.
  3. *Repartition* (to move parallel interfaces).
  4. Iterate.
- Implementation : Parallel library (`ParMmg`) on top of the sequential `Mmg3d` remesher (`https://www.mmgtools.org/`).

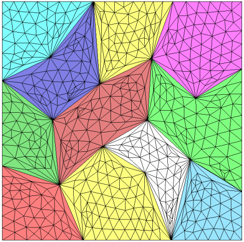
Pierre BENARD et al. "Mesh adaptation for large-eddy simulations in complex geometries". In : *International Journal for Numerical Methods in Fluids* 81.12 (2016), p. 719-740. URL : `https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4204`

Parallel iterative remeshing and repartitioning

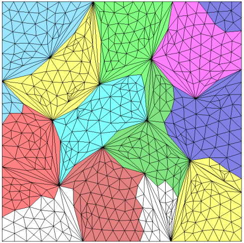
First iteration



Input mesh



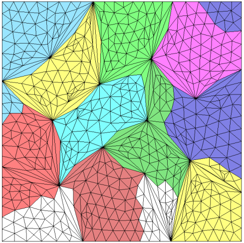
Remeshing



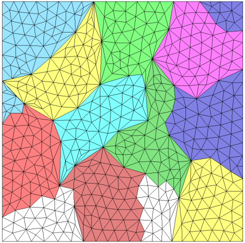
Repartioning

Parallel iterative remeshing and repartitioning

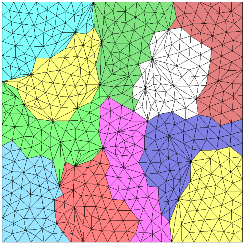
Second (. . .) iteration



Previous mesh



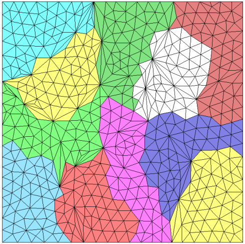
Remeshing



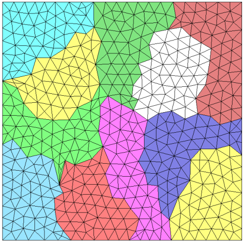
Repartitioning

Parallel iterative remeshing and repartitioning

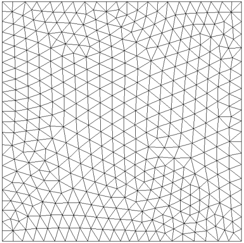
Last iteration



Previous mesh



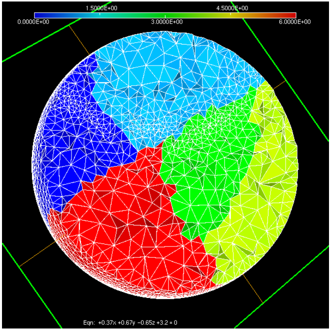
Remeshing



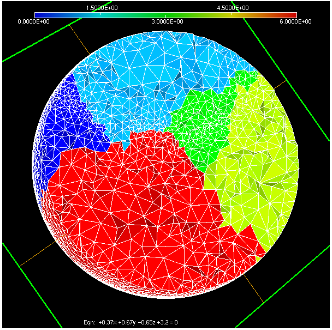
Output mesh

Parallel repartitioning

Interface displacement



Before



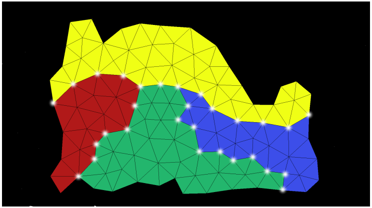
After

Design choice 2

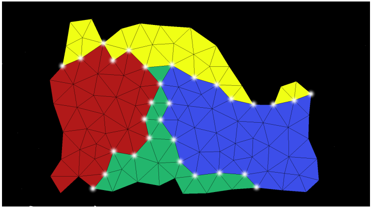
Prioritize interface displacement.

Parallel repartitioning

Interface displacement (2D sketch)



Before

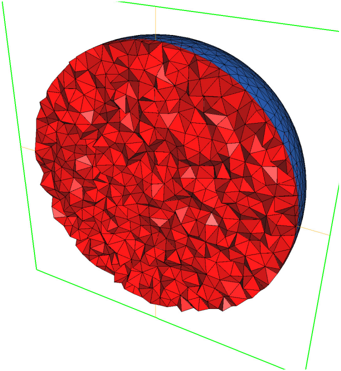


After (without correction)

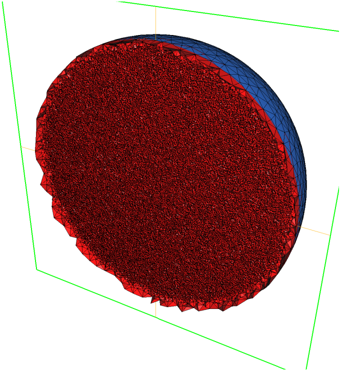
A correction is performed against partitions fragmentation.

Weak scaling performances

Isotropic uniform refinement case



Full input



Full output

On 2 processes.

Weak scaling performances

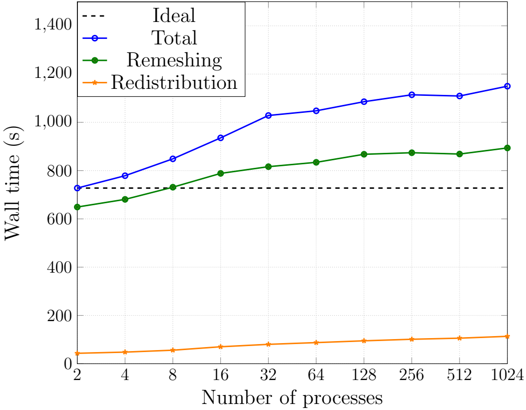
Setting a uniform processors workload

p	n_v^{in}/p	n_v^{out}/p	n_v^{out}/n_v^{in}	n_v^{out}	n_e^{in}/p	n_e^{out}/p	n_e^{out}/n_e^{in}	n_e^{out}
2	3625	1293637	356.81	2587274	18876	7780974	412.21	15561948
4	3467	1341637	386.88	5366549	18798	8072081	429.39	32288325
8	3346	1380055	412.38	11040444	18666	8306084	444.98	66448675
16	3264	1412516	432.66	22600269	18599	8503695	457.2	136059129
32	3190	1437267	450.46	45992552	18557	8654569	466.36	276946210
64	3214	1431186	445.2	91595935	18625	8619098	462.75	551622317
128	3215	1444674	449.31	184918370	18878	8701524	460.91	1113795077
256	3345	1468905	439.01	376039759	19705	8848464	449.03	2265206835
512	3375	1446532	428.52	740624790	19998	8714450	435.74	4461798709
1024	3335	1449215	434.54	1483996788	19821	8731162	440.49	8940710661

Number of vertices n_v and tetrahedra n_e in input and output using p processors.

Weak scaling performances

On the `bora` nodes of `PlaFRIM` (https ://www.plafrim.fr/)



Strong scaling performances

Isotropic test case metrics

Double Archimede's spiral

$$h(x, y) = \min(h_1, h_2)$$

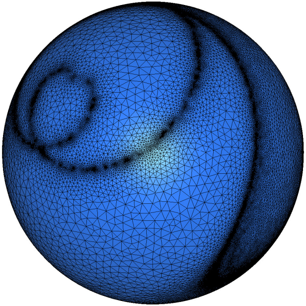
$$h_1 = 1.6 + |\rho - a\theta_1| + 0.005 \qquad h_2 = 1.6 + |\rho + a\theta_2| + 0.0125$$
$$\theta_1 = \phi + \pi \left(1 + \text{floor}\left(\frac{\rho}{2\pi a}\right)\right) \qquad \theta_2 = \phi - \pi \left(1 + \text{floor}\left(\frac{\rho}{2\pi a}\right)\right)$$

$$\phi = \text{atan2}(y, x),\ \ \rho = s\sqrt{x^2 + y^2},\ \ a = 0.6,\ \ s = 0.5$$

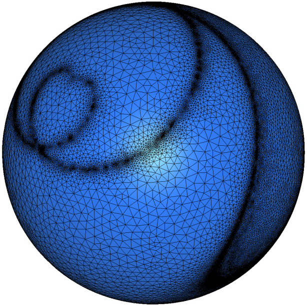
In a sphere of radius 10 with initial edge length 1.

Strong scaling performances

Adapted meshes



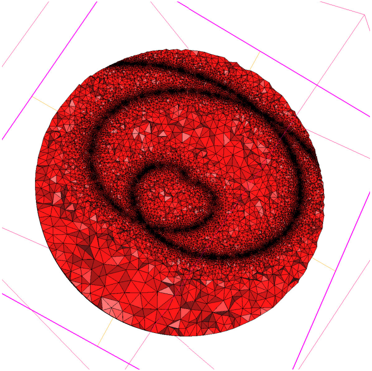
On 1 process.



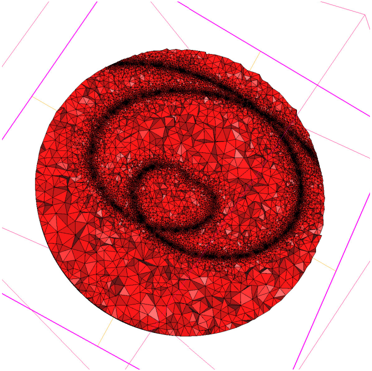
On 1024 processes.

Strong scaling performances

Adapted meshes



On 1 process.



On 1024 processes.

Strong scaling performances

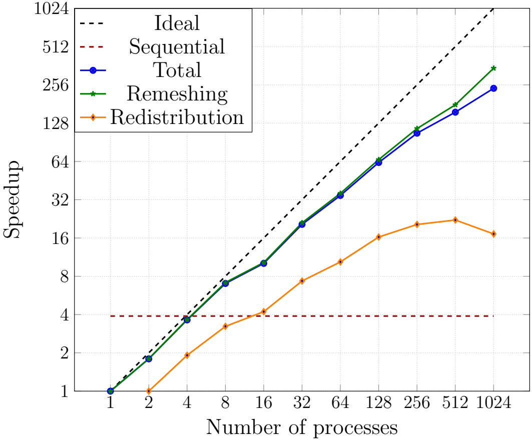
Edge length statistics

p	$N^{(0,0.3]}$	$N^{(0.3,0.6]}$	$N^{(0.6,0.7]}$	$N^{(0.71,0.9]}$	$N^{(0.9,1.3]}$	$N^{(1.3,1.41]}$	$N^{(1.41,2]}$	$N^{(2,5]}$	$N^{>5}$
1	1.34 %	35.34 %	16.89 %	25.49 %	20.15 %	0.63 %	0.16 %	0	0
2	1.14 %	34.87 %	16.97 %	25.79 %	20.45 %	0.63 %	0.15 %	0	0
4	1.04 %	34.20 %	17.03 %	26.08 %	20.85 %	0.64 %	0.16 %	0	0
8	0.98 %	33.66 %	17.06 %	26.30 %	21.18 %	0.66 %	0.16 %	0	0
16	1.07 %	32.41 %	17.06 %	26.78 %	21.84 %	0.68 %	0.16 %	0	0
32	0.91 %	30.78 %	17.29 %	27.59 %	22.57 %	0.70 %	0.16 %	0	0
64	0.93 %	30.22 %	17.20 %	27.76 %	23.01 %	0.72 %	0.17 %	0	0
128	0.88 %	29.48 %	17.20 %	28.06 %	23.48 %	0.73 %	0.17 %	$<$ 0.01 %	0
256	0.70 %	27.63 %	17.20 %	28.84 %	24.67 %	0.77 %	0.18 %	$<$ 0.01 %	$<$ 0.01 %
512	0.66 %	27.19 %	17.14 %	28.96 %	25.04 %	0.80 %	0.20 %	$<$ 0.01 %	$<$ 0.01 %
1024	0.69 %	26.14 %	16.91 %	29.03 %	25.99 %	0.92 %	0.30 %	0.02 %	$<$ 0.01 %

Percentage of edges $N^{(a,b]}$ whose length in the assigned metrics falls in the interval $l_{\mathcal{M}} \in (a, b]$, for each simulation on p processors.

Strong scaling performances

On the bora nodes of PlaFRIM (https ://www.plafrim.fr/)



Conclusions

Motivations

1. Exploiting *distributed memory* computer architectures.
2. Exploiting *multicore* computer architectures.

Strategy

1. Modularity : Re-use sequential remesher with fixed parallel interfaces.
2. Prioritize interface displacement.

Open source implementation into the `ParMmg` library on top of the `Mmg3d` remesher.

Conclusions

Results : Performance scalability

1. Larger meshes than those achievable on a single node memory, but...
   ... Scalability of mesh redistribution can be improved.
2. Faster execution on multicore architectures...
   ... Once the processes overcomes the iterations overhead.

Next work

1. Assess/reduce the influence of mesh gradation from fixed interfaces in anisotropic cases.
2. Extend support for boundary adaptation on non-manifold surfaces.

A few references

Iterative parallel remeshing

- Pierre BENARD et al. "Mesh adaptation for large-eddy simulations in complex geometries". In : *International Journal for Numerical Methods in Fluids* 81.12 (2016), p. 719-740. URL : https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4204
- Hugues DIGONNET et al. "Massively parallel anisotropic mesh adaptation". In : *International Journal of High Performance Computing Applications* (mar. 2017). URL : https://hal.archives-ouvertes.fr/hal-01487424
- Peter A. CAVALLO, Neeraj SINHA et Gregory M. FELDMAN. "Parallel Unstructured Mesh Adaptation Method for Moving Body Applications". In : *AIAA Journal* 43.9 (2005), p. 1937-1945. URL : https://doi.org/10.2514/1.7818

A few references

ParMmg

- Luca Cirrottola et Algiane Froehly. *Parallel unstructured mesh adaptation using iterative remeshing and repartitioning*. Research Report RR-9307. INRIA Bordeaux, équipe CARDAMOM, nov. 2019. URL : https://hal.inria.fr/hal-02386837

- **ExaQUte** European project : http ://exaqute.eu/
- Open source code : https ://github.com/MmgTools/ParMmg/releases/tag/v1.3.0

... Thank you for your attention !