Computers & Fluids 43 (2011) 2–13

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

A 2D block-structured mesh partitioner for accurate flow simulations on non-rectangular geometries

E. Ahusborde *, S. Glockner

TREFLE (UMR CNRS 8508), University of Bordeaux, 33607 Pessac, France

ARTICLE INFO

Article history:
Received 3 May 2010
Received in revised form 15 July 2010
Accepted 21 July 2010
Available online 29 July 2010

Keywords:
Partitioning
Block-structured meshes
Navier–Stokes
Lid driven cavity
Non-rectangular geometries
Parallel computing

ABSTRACT

The motivation of this work is to carry out parallel simulations of incompressible flows on block-structured meshes. A new partitioning method is proposed. The quality of rectangular partitions is checked and compared with other methods, as regards load balance, edge-cut and block numbers. The partitioner is coupled with the massively parallel Hypre solver library and efficiency of the coupling is measured. Finally, the code is applied to study laminar flows (steady and unsteady) on three non-rectangular geometries. Very fine grids are used to compute reference solutions of a Z-shaped channel flow and the L-shaped and double lid driven cavities.

1. Introduction

Flow simulations on complex geometries require either block-structured or unstructured grids. The latter allow very complex geometries to be meshed leading to complex discretization schemes and solvers that require a table of connectivity between nodes and indirect addressing. If the geometry is not too complicated, it can be divided into a reasonable number of structured and conforming blocks. The volume control aspect and lexical numbering facilitate the discretization of the equations (specially if the grid remains orthogonal) and the use of the fastest parallel solvers dedicated to the structured grids.

Solver performances are closely linked to the mesh partitioning or the matrix graph. Partitioning methods can be divided into two classes: geometric and combinatorial [1]. Geometric techniques are based on the coordinates of the mesh nodes whereas combinatorial partitioning uses the graph or the hypergraph of the mesh. Geometric techniques produce lower quality partitions than combinatorial methods but are extremely fast. For unstructured meshes, partitioner libraries such as CHACO [2], METIS [3], SCOTCH [4] are available. Unfortunately, they are not well suited to the block-structured framework since they produce unstructured partitions, as shown in Fig. 1. For block-structured meshes, few works have been carried out. The two main strategies used for the partitioning of such meshes are the recursive edge bisection [5] and the so-called greedy algorithm [6]. These geometric techniques are used in elsA software [7] which is devoted to compressible flows around complex geometries. We can also cite the works of Rantakokko [8] who proposes a framework for partitioning composite grids. In our opinion, his more interesting approach consists in a graph strategy applied at the block level instead of the node level (block refinement is also proposed).

Our goal consists in providing a partitioning strategy for block-structured geometries which produces rectangular partitions. It can be classified as a geometric method even if the coordinates of the nodes are not used. The partitioner is coupled with the massively parallel solver and preconditioner Hypre library [9], more precisely with the semicoarsening geometric multigrid solver [10,11]. Firstly, we are going to present the different steps upon which the proposed method relies. Then, we will compare the quality of the partitions with other approaches and analyse the performance of the coupling with Hypre solvers. Lastly, we will apply our code to compute incompressible flows on non-rectangular geometries that have been scarcely studied so far.

2. Partitioning strategy

Firstly, let us recall the two main qualities of a partitioner:

- It must respect load balancing between processors: each processor should have nearly the same amount of work to do to minimize idle processors. In our context, each processor should have around the same number of nodes, close to the ideal load which is equal to the number of nodes divided by the number of processors.

* Corresponding author.
E-mail address: ahusborde@enscbp.fr (E. Ahusborde).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.07.009

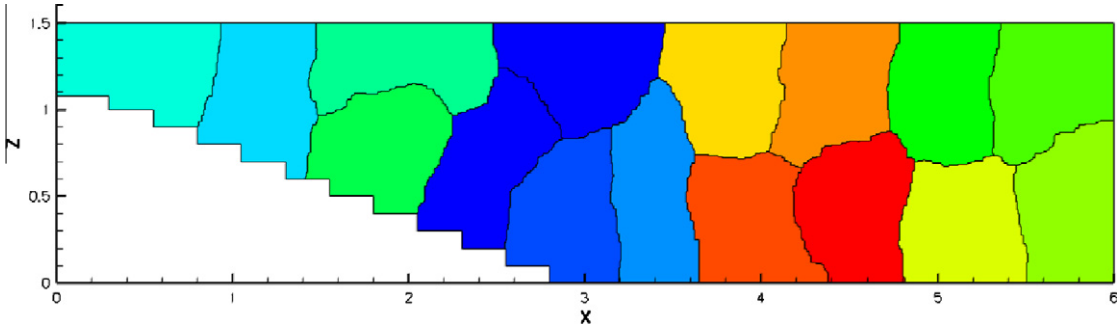

Fig. 1. Partitioning provided by METIS.

- It must minimize explicit communication between processors, *i.e.* the surface-to-volume ratio or edge-cuts. The goal is to delay as far as possible the moment when communications between processors increase such that efficiency collapses as the number of processors rises.

The conceptual interface [12] of Hypre is quite complete and supports four options: structured grid, block-structured grid, finite element interface, and linear algebraic interface. The fastest solvers such as geometric multigrid ones are available for structured grids, and block-structured grids, which is our framework. The interface requires global indexing of the nodes and rectangular boxes that can be non-contiguous. In the next sections, we will present the main steps of the partitioner and finally a complete algorithm that can be used in another solver framework.

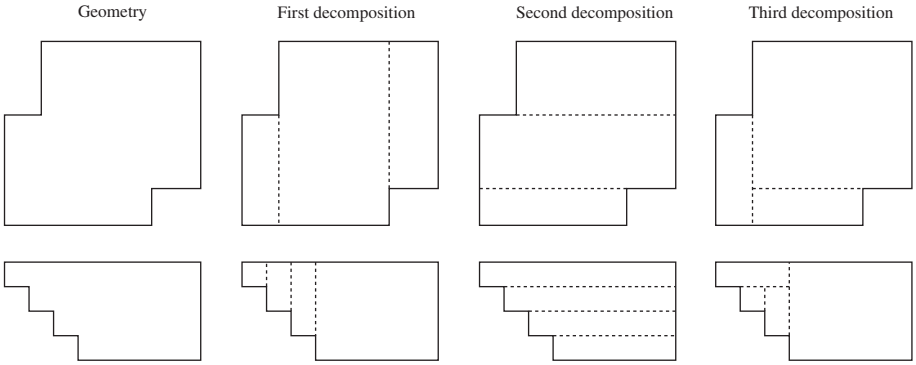

Fig. 2. Example of geometries.

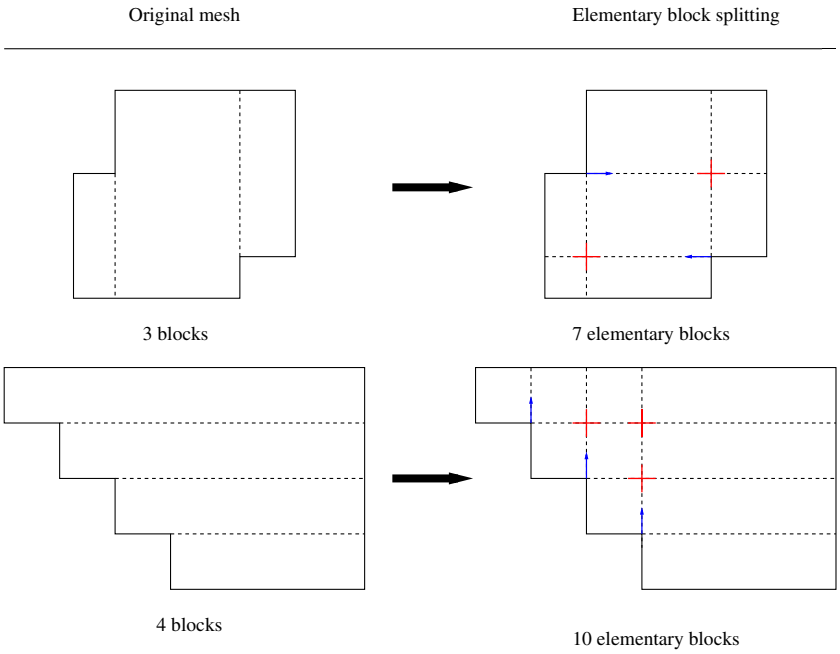

Fig. 3. Elementary block splitting.

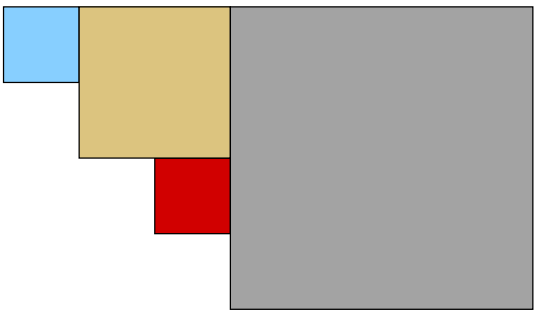

Fig. 4. Merging of elementary block into rectangular macro-blocks.

2.1. An elementary block decomposition

In our opinion, the partitioner should be independent of the initial block construction. For instance, two geometries are defined on the left part of Fig. 2. On the right part, different ways to decompose them, into three or four blocks are presented (that can be later meshed). Meshes are not shown but they are continuous through the interfaces between blocks. In the proposed method, the same partition will be produced for any geometry decomposition. This approach avoids having to consider parallelism during the construction phase of the mesh.

Consequently, the first step consists of splitting the main blocks into elementary ones. This is done by lengthening each boundary line. Intersections between lines define corners of new elementary blocks. For instance in Fig. 3, three blocks of the first geometry are split into seven elementary ones, whereas the four blocks of the second geometry are divided into 10 elementary ones. At this point, we can make three remarks:

- These elementary blocks are now the starting point of our partitioner.
- There is no reason for these blocks to be balanced.
- These blocks could have been created during the construction phase of the mesh, but it can become really fastidious for a large number of blocks.

2.2. Block merging

The second step of the partitioner consists in merging the elementary blocks into macro-blocks. In Fig. 4, we have merged 10 elementary blocks into four macro-blocks. The first macro-block is the largest of all possible macro-blocks. Then with the remaining elementary blocks, we choose the largest remaining macro-block and so on until there is no more elementary blocks. The idea of generating macro-blocks as big as possible is to minimize the number of blocks (and consequently to maximize their size) for which we are able to construct simple and optimal partitioning. Each macro-block is split into three zones (see Fig. 5). The main zone is zone 1 while zones 2 and 3 are residual zones.

2.2.1. Main zone

The size of the main zone is chosen such that it is a multiple of the ideal load. Then, straightforward partitioning that minimizes edge-cuts and respects load balancing is applied.

The number of cells in each direction of space is taken as an input by the partitioning (N_x = number of cells in the x direction, N_y = number of cells in the y direction). It can produce square or rectangular partitions. We also consider a special case if the number of processor associated to the zone 1 is a prime number (see Fig. 6).

2.2.2. Residual zones

Residual zones 2 and 3 of two different macro-blocks are associated to one processor so that the sum of their size is equal to the ideal load. Thus, load balancing is ensured (see Fig. 5).

We can note in Fig. 4 that very small macro-blocks (the darker one) can be composed just of one zone 2 or the sum of zones 2 and

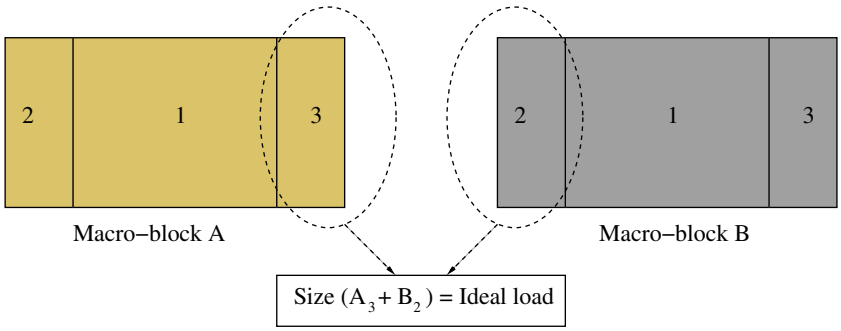

Fig. 5. Residual zones.

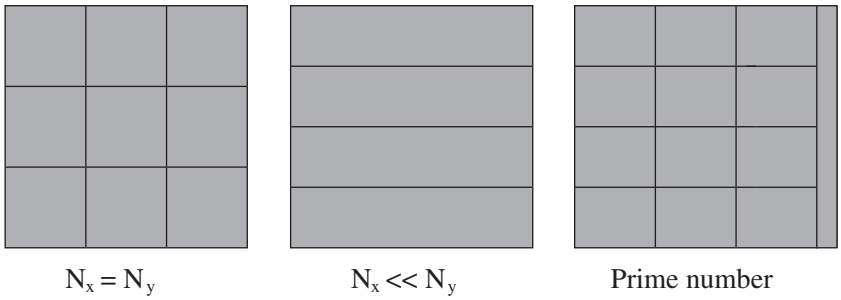

Fig. 6. Partitioning for the main zone.

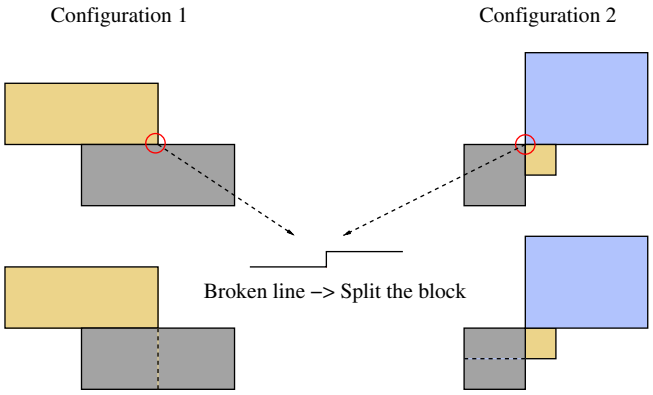

Fig. 7. Block merging rules.

3. These zones can create some non-contiguous regions on the same processor.

2.3. Node partitioning

If we had used cell partitioning, further caution would not have been necessary. With node partitioning, however, care has to be taken to the boundaries between blocks that can lead to non-rectangular macro-blocks. Indeed, as shown in Fig. 7, there exist two configurations that produce a broken boundary line. The only solution is to split the blocks in two parts. These configurations are quite rare in the cases we have studied.

From this set of geometric considerations, an Algorithm 1 has been extracted that can be applied to any 2D block-structured grid.

Algorithm 1. Partitioning strategy

1: Elementary block splitting
2: **for** Each elementary block **do**
3: Create the list of all possible rectangular macro-blocks associated to the elementary block
4: **end for**
5: **while** Macro-block partitioning not finished **do**
6: Choose the biggest remaining macro-block
7: **if** Configuration 1 **then**
8: Reject the macro-block from the list
9: **else if** Configuration 2 **then**
10: Cut the block and update macro-block list
11: **else**
12: Accept the macro-block and update macro-block list
13: **end if**
14: **end while**
15: **for** Each macro-block **do**
16: Construction of the main zone and residual zones
17: Partitioning of the main zone
18: **end for**
19: **return**

3. Block-structured partitioner quality and performance

3.1. Load balancing and edge-cuts

In this section, we test the quality of the partitioner concerning the load balancing, the number of edge-cuts, the number of blocks and the time consumed by the partitioning in comparison with those using METIS, the recursive edge bisection (REB) and the greedy algorithm (GA). For the two latters, results have been obtained with elsA software [7]. Several examples are considered.

The first example is the double cavity geometry, composed of three blocks, four elementary blocks and 4×10^5 nodes. Fig. 8 represents the partitions for 16 and 64 processors (one colour per processor). Table 1 compares load imbalance, edge-cut and block numbers between the different approaches.

The second example concerns a Z-shaped canal with three blocks, five elementary blocks and 5×10^5 nodes. Again, Fig. 9 represents the partition for 64 processors whereas Table 2 compares performances.

Table 1
Partitioner performance for the double cavity mesh.

Number of processors	8	16	32	64
Load imbalance present (%)	0.54	0.44	1.40	2.87
Load imbalance METIS (%)	1.54	2.28	3.17	3.28
Load imbalance REB (%)	17.07	17.07	17.08	17.08
Load imbalance GA 0.001 (%)	0.08	0.08	0.09	0.09
Load imbalance GA 0.05 (%)	0.97	4.49	5	4.85
Edge-cuts present	2829	4584	7513	11,360
Edge-cuts METIS	3230	4853	7507	11,176
Edge-cuts REB	2300	3800	6000	9000
Edge-cuts GA 0.001	3183	5042	7706	13,259
Edge-cuts GA 0.05	3077	4636	7303	12,961
Number of blocks present	11	19	35	66
Number of blocks METIS	8	16	32	64
Number of blocks REB	8	16	32	64
Number of blocks GA 0.001	21	49	87	171
Number of blocks GA 0.05	14	22	41	94

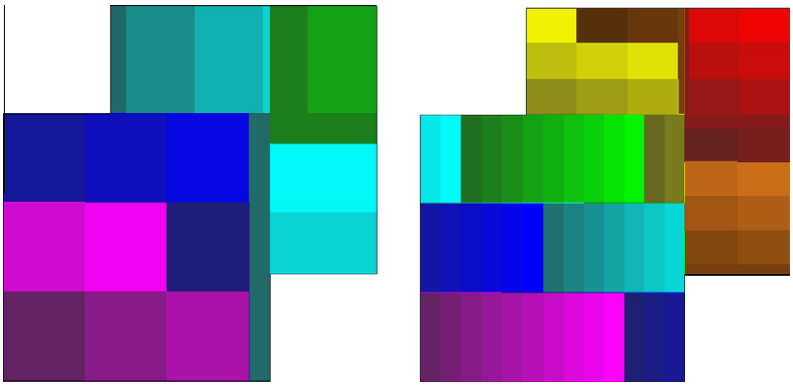

Fig. 8. Partitioning of a double cavity mesh.

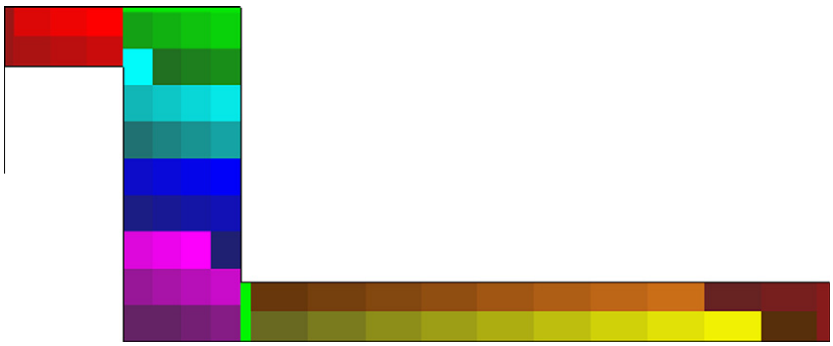

Fig. 9. Partitioning of a Z-shaped canal mesh.

Table 2
Partitioner performance for the Z-shaped canal mesh.

Number of processors	8	16	32	64
Load imbalance present (%)	0.42	0.40	0.91	1.37
Load imbalance METIS (%)	1.66	3.07	3.24	3.15
Load imbalance REB (%)	47.63	47.63	47.63	47.64
Load imbalance GA 0.001 (%)	0.08	0.09	0.08	0.1
Load imbalance GA 0.05 (%)	1.45	4.14	4.77	4.93
Edge-cuts present	2016	3692	6351	10,949
Edge-cuts METIS	2247	4055	7096	11,700
Edge-cuts REB	1773	3140	5640	8843
Edge-cuts GA 0.001	2521	4345	7625	14,317
Edge-cuts GA 0.05	2335	3864	7219	13,756
Number of blocks present	10	18	34	66
Number of blocks METIS	8	16	32	64
Number of blocks REB	8	16	32	64
Number of blocks GA 0.001	21	50	80	164
Number of blocks GA 0.05	12	20	49	91

The third and last example is a ring with 10 blocks, 27 elementary blocks and 1.5×10^5 nodes. Again, Fig. 10 represents the partition for 64 processors while Table 3 compares performances.

These test cases underline very good load imbalance lower than 1% if the number of processors is not too high. It confirms that splitting macro-blocks into three zones is efficient, residual zones being associated to verify ideal load. Lower load imbalance cannot be reached because the precision of the partitioner is equal to the length of a mesh line (necessary to keep rectangular partitions). If the number of processors increases, load imbalance increases but remains lower than 3%. Here, the partitioning effect of zone 1 is more visible: the size of a line of the mesh is relatively high in comparison with the size of the partition. The number of edge-cuts is overall very good, better than with METIS. REB method shows optimal results as regards edge-cut and block numbers but very high load imbalance (up to 47%) which is a crippling default. Moreover, this method did not provide acceptable results for the ring mesh probably because of the circular aspect of the geometry. Load imbalance produced by the GA is controlled by an epsilon parameter which has a consequence on the number of blocks generated: the lower is ϵ the lower is the load imbalance, but the greater is the number of blocks. Two values for ϵ have been used: 0.05 and 0.001. For $\epsilon = 0.001$ load imbalance is very low, edge-cut number is acceptable but the number of blocks is more than the double of the number of processors. That leads to a high number of non-contiguous subdomains associated to a processor: it can reduce solver efficiency and it increases the memory requirement due to the multiplication of ghost cells necessary to the communications between processors [7]. For $\epsilon = 0.05$, load imbalance is higher than our method, as well as edge-cut and block numbers. Finally for the ring example, CPU time shows that the proposed block-approach is much faster (nearly 30 times) than the other methods. This point could be even more relevant for 3D partitioning where CPU time is much longer. We can conclude that the proposed

Table 3
Partitioner performance for the ring mesh.

Number of processors	8	16	32	64
Load imbalance present (%)	0.10	0.15	0.35	0.86
Load imbalance METIS (%)	0.43	1.17	2.39	2.92
Load imbalance REB (%)	–	–	–	–
Load imbalance GA 0.001 (%)	0.08	0.05	0.08	0.05
Load imbalance GA 0.05 (%)	0.56	1.19	2.46	4.99
Edge-cuts present	2254	4346	7884	14,439
Edge-cuts METIS	2240	4746	10,113	16,482
Edge-cuts REB	–	–	–	–
Edge-cuts GA 0.001	12,730	14,052	15,805	19,524
Edge-cuts GA 0.05	12,504	13,409	14,800	18,342
Number of blocks present	17	25	41	73
Number of blocks METIS	8	16	32	64
Number of blocks REB	–	–	–	–
Number of blocks GA 0.001	50	68	109	180
Number of blocks GA 0.05	42	48	60	92
CPU present time (s)	0.04	0.048	0.048	0.048
CPU time METIS (s)	1.24	1.23	1.29	1.32
CPU time REB (s)	–	–	–	–
CPU time GA 0.001 (s)	1.63	1.64	1.65	1.67
CPU time GA 0.05 (s)	1.63	1.64	1.63	1.65

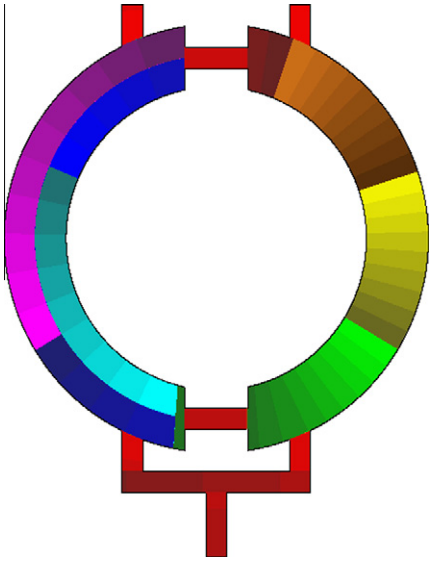

Fig. 10. Partitioning of a ring mesh.

method is efficient for the studied geometries and shows a good compromise between all the partitioning requirements.

3.2. Scalability

The efficiency of coupling the partitioner and the Hypre library is illustrated solving the Poisson equation obtained from one velocity correction step [13] which is very CPU time consuming in a Navier–Stokes solver. The studied problem is the double lid driven cavity flow.

The solver is a Generalized Minimal Residual Method (GMRES) associated to the geometric semicoarsening multigrid preconditioner. The relative residual is set to 10^{-10}. The code runs on an SGI ICE cluster. Two types of processors have been used: Harpertown nodes linked to a DDR Infiniband network and Nehalem nodes linked to a QDR Infiniband network.

In parallel computing, two types of scalability are defined. The first is the strong scaling, which represents the relation between the computation time and the number of processors for a fixed total problem size. The second is the weak scaling, for which the load per processor is constant.

3.2.1. Weak scaling

Fig. 11 displays for each type of processor (Harpertown and Nehalem) CPU time as a function of the number of processors, with 32,500 and 65,000 degrees of freedom (dof) per processor. We can see that processors Nehalem are much faster than Harpertown ones, particularly if the number of processors is low. A time ratio from 1.3 to 2.1 can be observed.

Weak efficiency is given by:

$$\mathrm{WE}\ (N) = \frac{\text{CPU time on } p \text{ processors}}{\text{CPU time on } N \text{ processors}}$$

where p denotes the number of processors used for the reference time (not always equal to one for heavy computations). Efficiency equal to one indicates an optimal behaviour for the algorithm and the computer architecture. Indeed, CPU times remains constant, equal to the reference time, while the total size of the problem increases with the number of processors. Usually, this property is hardly verified and curves with plateaus can be observed. Values of the plateaus rise toward one with the load of each processor. This phenomenon is illustrated in Fig. 12. Weak efficiency is better for the Harpertown cluster than the Nehalem one, besides a longer computation time.

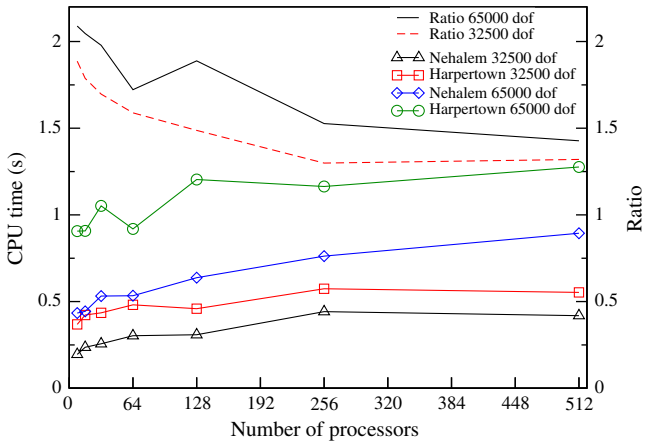

Fig. 11. CPU time versus the number of processors.

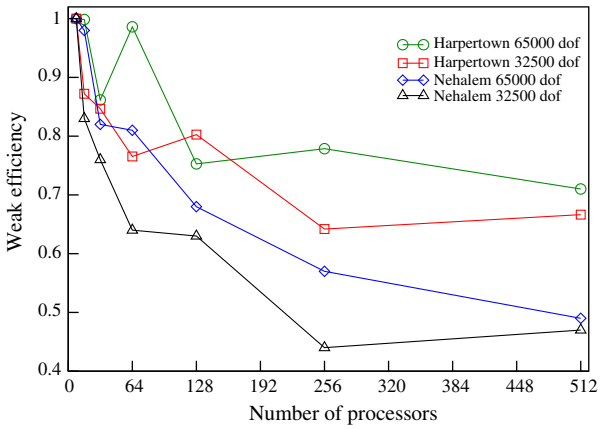

Fig. 12. Weak scaling versus the number of processors.

3.2.2. Strong scaling

Fig. 13 displays for each type of processor (Harpertown and Nehalem) on a logarithmic scale, CPU time as a function of the number of processors for two fixed size problems of 1 and 16 million degrees of freedom. Again, processors Nehalem are much faster than Harpertown ones.

Strong efficiency is given by:

$$\mathrm{SE}\ (N) = \frac{\text{CPU time on } p \text{ processors} \times p}{\text{CPU time on } N \text{ processors} \times N}$$

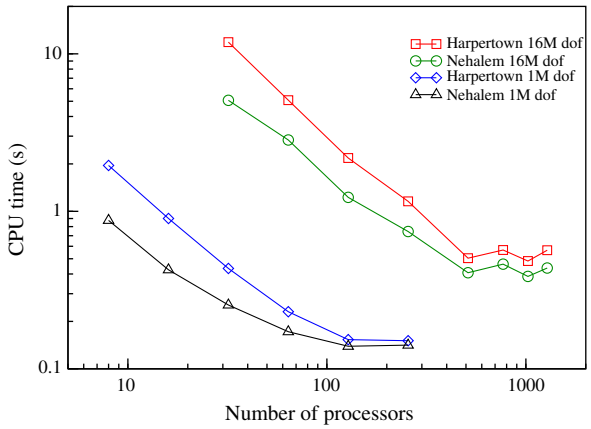

Fig. 13. CPU time versus the number of processors.

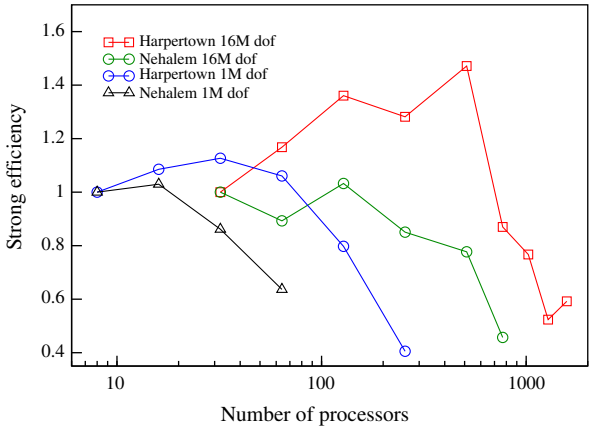

Fig. 14. Strong scaling versus the number of processors.

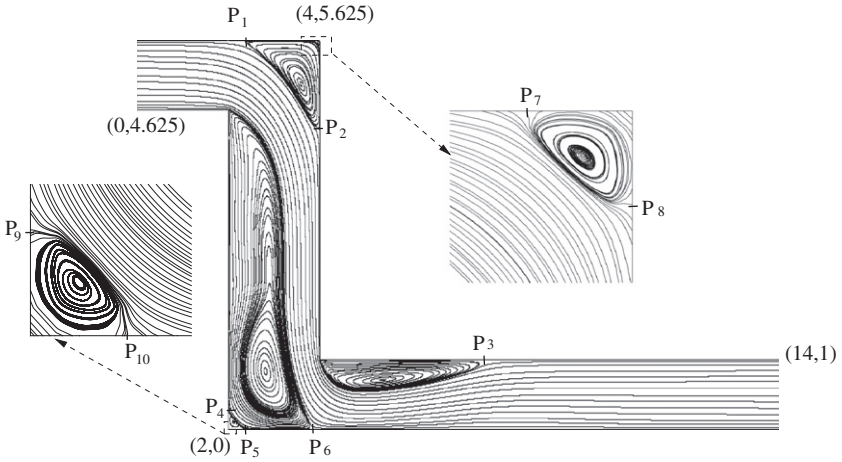

Fig. 15. Streamlines for the Z-shaped flow.

where p denotes the number of processors used for the reference time (not always equal to one for heavy computations). It points out an optimal use of the parallel resources. Efficiency equal to one indicates that communications and synchronizations between processors are negligible.

Fig. 14 represents the strong scaling versus the number of processors on a semi-logarithmic scale. With the Harpertown architecture and 16 million dof, a very high efficiency greater than 0.8 for up to 1024 processors can be observed (16,000 dof per processor). The first part of the graph being over the expected efficiency is due to memory bandwidth saturation when the number of processors is low that leads to a long reference time in the strong efficiency formula. Using more processors leads to smaller tasks that lead to a performance increase when more and more data can be kept in cache. With 1 million dof, this effect is less visible ans scaling is very good up to 128 processors (8000 dof per processor). With the Nehalem architecture which has a much higher memory bandwidth (more than three times), efficiency curves have the expected behaviour. Consequently, optimal efficiencies are obtained for 512 and 32 processors respectively for the 16 and 1 million dof problems. The saturation of the efficiency due to the increase of the communications between processors appears earlier with this architecture.

The next part of the article is devoted to the study of incompressible flows in non-rectangular geometries using the approach proposed here.

4. Computations of incompressible flows on non-rectangular geometries

Laminar flows in rectangular geometries, such as the lid driven cavity [15,16], have been extensively studied in the literature. Several numerical methods have been compared and reference solutions are available for a wide range of Reynolds numbers (leading to stationary or unsteady flows). In the present study, we propose a precise solution of flows for three non-rectangular geometries, scarcely studied so far.

Time discretization of the Navier–Stokes equations is implicit thanks to Gear's second order backward differentiation formula [14]. A pressure correction method (see Goda [13]) is used to solve the velocity–pressure coupling. Spatial discretization (second order centered scheme) is based on the finite volume method on a staggered grid of the Marker and Cells type. Solvers of the Hypre library are used.

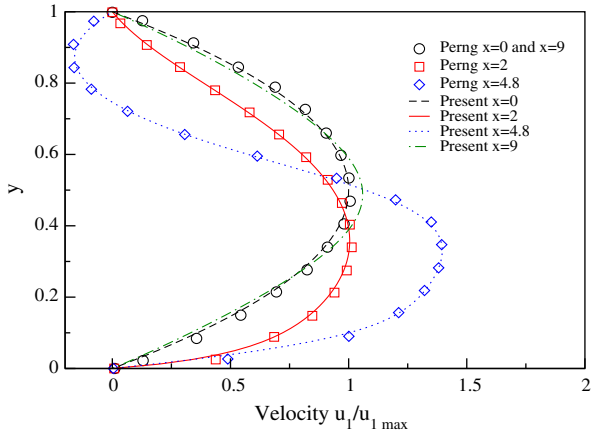

Fig. 16. Velocity u_1/u_{1max} profiles at different sections versus y.

4.1. Z-shaped channel

Flow in a Z-shaped channel has been studied in [17,18]. The channel has an abrupt expansion and contraction of it section (with a ratio of two), associated with a change of direction (horizontal contracted sections and a vertical expanded one). In our study, the Reynolds number Re is based on the width of the channel and the mean value of the inlet velocity. We can note that in [17], the Reynolds number is defined from the mean velocity u_{mean} while in [18], it is defined from the maximal velocity $u_{max} = 1.5u_{mean}$. A Poiseuille velocity profile is imposed at the inlet while Neumann and wall conditions are respectively imposed at the outlet, upper and lower sides. We have studied the flow for $Re = 200$. The streamlines and the points $P_i(x_i, y_i)$ of detachment and reattachment of the flow are presented in Fig. 15. Three main recirculations are created. The first and biggest one occupies almost one-half of the expanded region. The second eddy is developed in the upper right-hand corner while the third one is attached to the upper boundary at the beginning of the last contracted horizontal section. An infinite series of Moffat corner vortices [19] of increasingly smaller amplitude appears in the lower left and the upper right corners (see zooms in Fig. 15). Four increasingly fine grids with 3.75×10^3, 1.5×10^5, 6×10^5, 2.4×10^6 nodes were used.

4.1.1. Velocity profiles

In Fig. 16, we compare our velocity profiles at different sections with those obtained in [17]. We can note good accordance in the

Table 4
Positions (x, y) and intensities of the primary vortices.

Reference	Main primary vortex (x, y)	Vorticity
Mesh 1	(2.8097, 0.83996)	−1.29348
Mesh 2	(2.8117, 0.83881)	−1.28848
Mesh 3	(2.8127, 0.83797)	−1.28474
Mesh 4	(2.8133, 0.83748)	−1.28253
Reference	Upper primary vortex (x, y)	Vorticity
Mesh 1	(3.5884, 4.9967)	0.75678
Mesh 2	(3.5861, 4.9945)	0.75393
Mesh 3	(3.5849, 4.9925)	0.75299
Mesh 4	(3.5845, 4.9914)	0.75279
Reference	Lower primary vortex (x, y)	Vorticity
Mesh 1	(5.3985, 0.70982)	4.4441
Mesh 2	(5.3893, 0.71240)	4.3645
Mesh 3	(5.3716, 0.71392)	4.3239
Mesh 4	(5.3684, 0.71469)	4.3054

results except for the section $x = 9$ where the Poiseuille flow is restored more quickly for [17]. The very coarse grid (7000 nodes) used in [17] could explain this difference. There are no other data in the literature to compare with. We thus present our values of positions and intensities of the vortices.

4.1.2. Positions and intensities of the vortices

Table 4 represents the positions and intensities of the three primary vortices versus the mesh size for $Re = 200$. Positions and intensities of the secondary and ternary vortices are presented in Table 5. Only mesh 4 is fine enough to capture the ternary vortex. Finally, Table 6 reports the positions of the detachment and reattachment points $P_i (i = 1, 10)$ defined in Fig. 15. Convergence is achieved of two to four significant digits.

4.2. L-shaped driven cavity

Flow in an L-shaped driven cavity has been studied in [17,18,20]. The fluid is driven by the upper side while the other boundaries are walls. The configuration and the streamlines are presented in Fig. 17 for a Reynolds number $Re = 1000$ (based on the cavity width and the lid velocity). The presence of a step inside the cavity induces three recirculation zones, two in the upper part of the cavity and one in the lower. The main recirculation is located in the upper right part. Secondary vortices, whose size increases with increasing Re, appear in the corners of the cavity. If the grid is fine enough, ternary vortices can be computed. The study is carried out for $Re = 1000$. Four increasingly fine grids with 1.25×10^5, 5×10^5, 2×10^6 and 8×10^6 nodes were used.

Table 5
Positions (x, y) and intensities of the secondary vortices.

Reference	Upper secondary vortex (x, y)	Vorticity
Mesh 1	–	–
Mesh 2	(3.9676, 5.5924)	−0.00429
Mesh 3	(3.9659, 5.5907)	−0.00512
Mesh 4	(3.9655, 5.5902)	−0.00533
Reference	Lower secondary vortex (x, y)	Vorticity
Mesh 1	(2.1176, 0.10110)	0.10198
Mesh 2	(2.1188, 0.10185)	0.10283
Mesh 3	(2.1191, 0.10215)	0.10272
Mesh 4	(2.1193, 0.10219)	0.10241
Reference	Lower ternary vortex (x, y)	Vorticity
Mesh 4	(2.00632, 0.00632)	-6.22×10^{-4}

Table 6
Positions (x_i, y_i) of the detachment and reattachment points $P_i (i = 1, 10)$.

Reference	$P_1(x_1, y_1)$	$P_2(x_2, y_2)$
Mesh 1	(2.4249, 5.625)	(4, 4.3519)
Mesh 2	(2.3750, 5.625)	(4, 4.3519)
Mesh 3	(2.3500, 5.625)	(4, 4.3394)
Mesh 4	(2.3406, 5.625)	(4, 4.3332)
Reference	$P_3(x_3, y_3)$	$P_4(x_4, y_4)$
Mesh 1	(7.6250, 1)	(2, 0.20008)
Mesh 2	(7.6250, 1)	(2, 0.22499)
Mesh 3	(7.6313, 1)	(2, 0.23124)
Mesh 4	(7.6281, 1)	(2, 0.23126)
Reference	$P_5(x_5, y_5)$	$P_6(x_6, y_6)$
Mesh 1	(2.2749, 0)	(3.7750, 0)
Mesh 2	(2.2875, 0)	(3.7749, 0)
Mesh 3	(2.3063, 0)	(3.7812, 0)
Mesh 4	(2.3094, 0)	(3.7812, 0)
Reference	$P_7(x_7, y_7)$	$P_8(x_8, y_8)$
Mesh 1	–	–
Mesh 2	(3.9374, 5.625)	(4, 5.5624)
Mesh 3	(3.9249, 5.625)	(4, 5.5496)
Mesh 4	(3.9187, 5.625)	(4, 5.5437)
Reference	$P_9(x_9, y_9)$	$P_{10}(x_{10}, y_{10})$
Mesh 4	(2, 0.0125)	(2.0125,0)

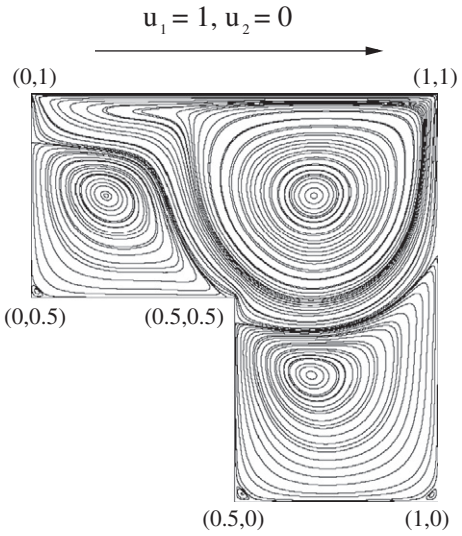

Fig. 17. Streamlines for the L-shaped driven cavity.

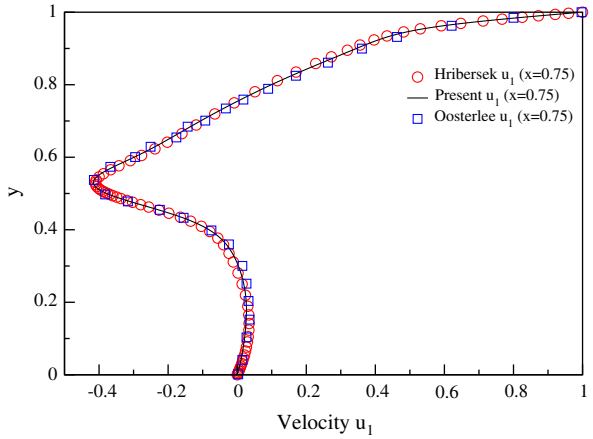

Fig. 18. u_1 profile at $x = 0.75$ versus y.

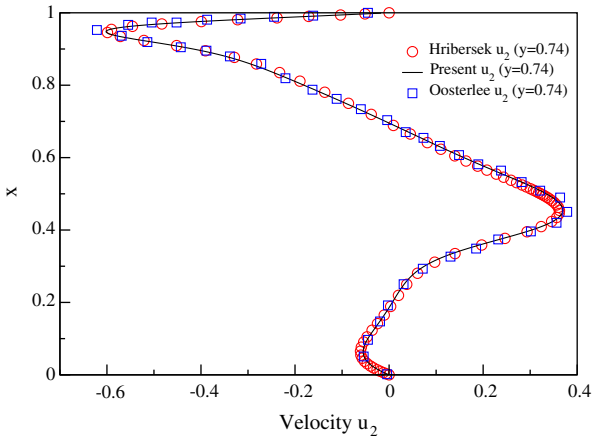

Fig. 19. u_2 profile at $y = 0.74$ versus x.

4.2.1. Velocity profiles

In Figs. 18 and 19, the velocity profiles at $x = 0.75$ and $y = 0.74$ present good accordance with those obtained in [20,18].

4.2.2. Positions and intensities of the vortices

Tables 7 and 8 represent the positions and the intensities of the primary and secondary vortices. Table 9 displays the positions and the intensities of the ternary vortices that can be only computed for meshes 3 and 4. Convergence of up to 2–4 digits is reached.

4.3. Double lid driven cavity

Flow in a double lid driven cavity has been recently studied in [21,22]. This configuration is presented in the left upper part of Fig. 20. The fluid is driven by the lower and upper boundaries in opposite directions. The other boundaries are walls. The Reynolds number Re is based on the cavity length L and the lid velocity. We focused on two flow ranges according to the value of the Reynolds number in relation to its critical value Re_c:

- $Re < Re_c$: the flow is stationary. We can note three different regimes represented in Fig. 20. On the upper right part for $Re = 100$, twin primary eddies are created between the two driving lids while secondary vortices appear in the left and right corners. In the lower left part for $Re = 1000$, the two primary vortices coalesce and two secondary vortices appear. Finally, in the lower right part for $Re = 3000$, the primary eddy becomes horizontal. The size of the two latter secondary vortices increases and two new secondary eddies appear (vertically on the upper right and lower left parts of the domain). However, it is now well-known [23] that cavity flows experience 3-dimensional global instability well below Re_c. Consequently, for $Re \geqslant Re_c^{3D}$ 2-dimensional studies are not physical anymore even if they can present numerical interests. Re_c^{3D} has been recently identified for the double and cross-sectional cavity flows [24]. For the former, it characterizes the transition between the two first flow regimes.
- $Re \geqslant Re_c$: the flow becomes 2D unsteady, from periodic to chaotic. In [15,16], a study was carried out to identify the transition from stable to periodic flow in the case of the 2D lid driven square cavity flow. The flow becomes unstable via a Hopf bifurcation. In [16], the first Lyapunov exponent was used to compute a critical Reynolds number Re_c close to 8000. In [15], thanks to a different approach (unsteady simulations with small time step), the first Hopf bifurcation occurs for $Re_c = 7402$. Several subcritical and supercritical flow regimes were identified.

Table 7
Positions (x, y) and intensities of the primary vortices.

Reference	Upper right primary vortex (x, y)	Vorticity
Oosterlee [20]	(0.6938,0.7509)	–
Mesh 1	(0.69440, 0.75113)	−3.87659
Mesh 2	(0.69408, 0.75101)	−3.87413
Mesh 3	(0.69394, 0.75097)	−3.87279
Mesh 4	(0.69388, 0.75094)	−3.87212
Reference	Upper left primary vortex (x, y)	Vorticity
Oosterlee [20]	(0.1822,0.7515)	–
Mesh 1	(0.18417, 0.75078)	1.21124
Mesh 2	(0.18386, 0.75091)	1.20968
Mesh 3	(0.18417, 0.75095)	1.20901
Mesh 4	(0.18417, 0.75097)	1.20870
Reference	Lower right primary vortex (x, y)	Vorticity
Oosterlee [20]	(0.6866,0.3089)	–
Mesh 1	(0.68741, 0.30940)	0.96079
Mesh 2	(0.68731, 0.30923)	0.96074
Mesh 3	(0.68729, 0.30918)	0.96078
Mesh 4	(0.68728, 0.30911)	0.96081

Table 8
Positions (x, y) and intensities of the secondary vortices.

Reference	Upper left secondary vortex (x, y)	Vorticity
Mesh 1	(0.01430, 0.51450)	−0.01112
Mesh 2	(0.01441, 0.51463)	−0.01140
Mesh 3	(0.01443, 0.51466)	−0.01147
Mesh 4	(0.01444, 0.51467)	−0.01148
Reference	Lower left secondary vortex (x, y)	Vorticity
Mesh 1	(0.51804, 0.01850)	−0.00780
Mesh 2	(0.51814, 0.01859)	−0.00796
Mesh 3	(0.51815, 0.01861)	−0.00801
Mesh 4	(0.51815, 0.01861)	−0.00802
Reference	Lower right secondary vortex (x, y)	Vorticity
Mesh 1	(0.98356, 0.01641)	−0.004610
Mesh 2	(0.98343, 0.01653)	−0.004739
Mesh 3	(0.98341, 0.01656)	−0.004768
Mesh 4	(0.98340, 0.01657)	−0.004778

Table 9
Positions (x, y) and intensities of the ternary vortices.

Reference	Upper left ternary vortex (x, y)	Vorticity
Mesh 3	$(6.864 \times 10^{-4}, 0.50068)$	3.17×10^{-5}
Mesh 4	$(8.279 \times 10^{-4}, 0.50082)$	7.00×10^{-5}
Reference	Lower left ternary vortex (x, y)	Vorticity
Mesh 3	$(0.99922, 7.772 \times 10^{-4})$	1.40×10^{-5}
Mesh 4	$(0.99902, 9.705 \times 10^{-4})$	3.21×10^{-5}
Reference	Lower right ternary vortex (x, y)	Vorticity
Mesh 3	$(0.50094, 9.416 \times 10^{-5})$	3.06×10^{-5}
Mesh 4	$(0.50107, 1.070 \times 10^{-4})$	5.31×10^{-5}

4.3.1. Steady flow

For the study of the steady flow, we focus on $Re = 1000$. The lower left part of Fig. 20 displays the streamlines. As described above, a primary vortex and four secondary eddies appear. With very fine meshes, ternary vortices appear between the secondary ones and the corner of the domain (see right part of Fig. 21). These vortices were not shown in previous studies [21,22]. Fig. 21 represents two zooms in which we define the points $P_i(x_i, y_i)$ of detachment and reattachment of the flow. The results found in the literature are given on the intensities and positions of the vortices.

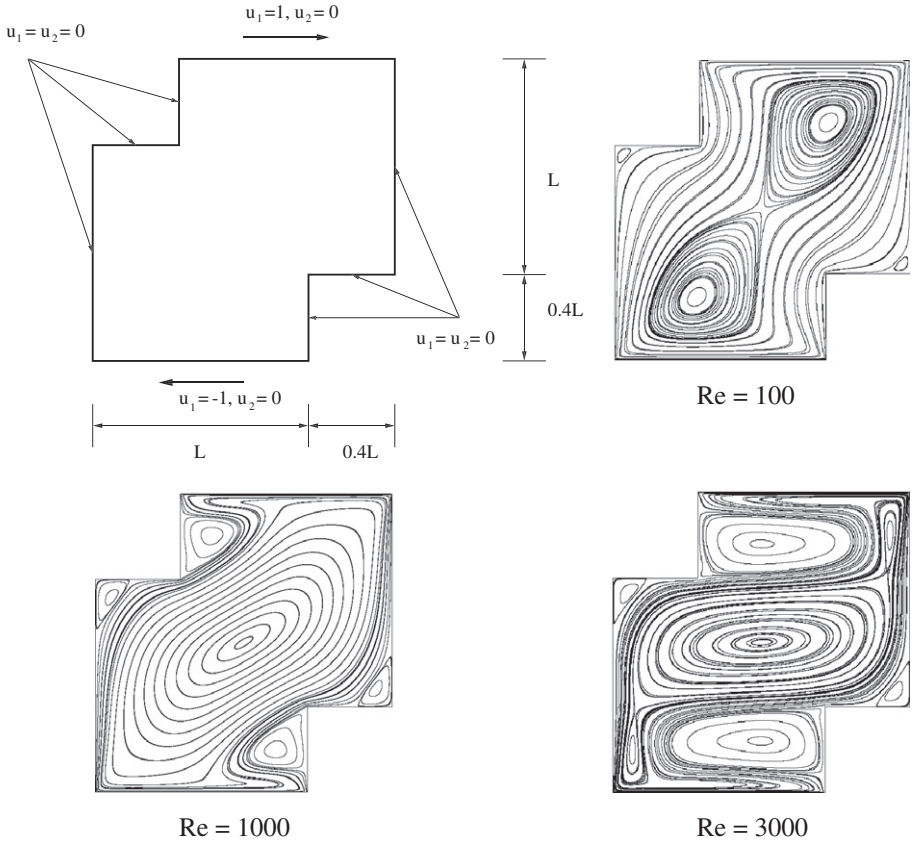

Fig. 20. Configuration of the double lid driven cavity and streamlines for different Reynolds numbers.

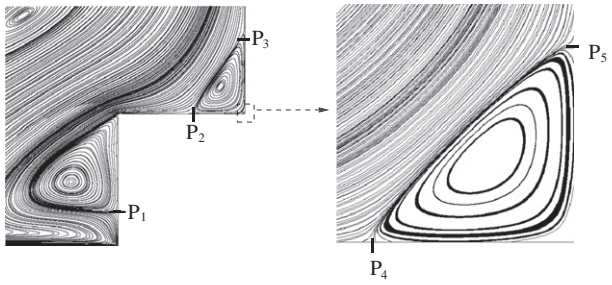

Fig. 21. Streamlines for the double lid driven cavity for *Re* = 1000.

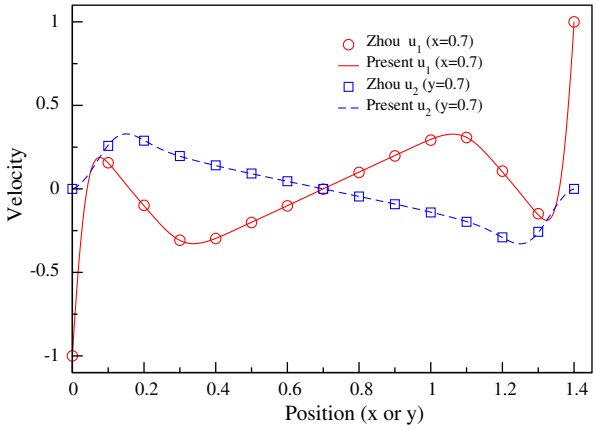

Fig. 22. Velocity profiles at $x = 0.7$ and $y = 0.7$.

In the present study, the results were obtained with convergence criteria on stationarity below 10^{-12} between two consecutive iterations. Four increasingly fine grids with 6.25×10^4, 2.5×10^5, 10^6 and 4×10^6 nodes have been used.

Table 10
Positions (x, y) and intensities of the main vortex.

Reference	Main vortex (x, y)	Vorticity
Zhou [21]	(0.70000, 0.70000)	−1.41562
Nithiarasu [22]	(0.68950, 0.69690)	−1.52363
Present (mesh 1)	(0.70099, 0.70070)	−1.49753
Present (mesh 2)	(0.69996, 0.69996)	−1.49974
Present (mesh 3)	(0.70000, 0.70000)	−1.49858
Present (mesh 4)	(0.69999, 0.69999)	−1.49838

Table 11
Positions (x, y) and intensities of the secondary vortices.

Reference	Lower secondary vortex (x, y)	Vorticity
Zhou [21]	(0.72560, 0.20000)	2.38559
Nithiarasu [22]	(0.85230, 0.20150)	2.60588
Present (mesh 1)	(0.84952, 0.19744)	2.60184
Present (mesh 2)	(0.85055, 0.19637)	2.59861
Present (mesh 3)	(0.85112, 0.19587)	2.59770
Present (mesh 4)	(0.85139, 0.19563)	2.59745
Reference	Right secondary vortex (x, y)	Vorticity
Zhou [21]	(1.32500, 0.48440)	0.53846
Nithiarasu [22]	(1.32210, 0.48360)	0.65005
Present (Mesh 1)	(1.32226, 0.48356)	0.63099
Present (Mesh 2)	(1.32243, 0.48353)	0.62956
Present (Mesh 3)	(1.32249, 0.48349)	0.62971
Present (Mesh 4)	(1.32253, 0.48346)	0.62996

Table 12
Positions (x, y) and intensities of the right ternary vortex.

Reference	(x, y)	Vorticity
Mesh 1	–	–
Mesh 2	(1.39579, 0.40420)	−0.002911
Mesh 3	(1.39531, 0.40468)	−0.004380
Mesh 4	(1.39522, 0.40478)	−0.004718

Table 13
Positions (x_i, y_i) of the detachment and reattachment points $P_i (i = 1,5)$.

Reference	$P_1(x_1, y_1)$	$P_2(x_2, y_2)$	$P_3(x_3, y_3)$
Mesh 1	(1, 0.09999)	(1.23999, 0.4)	(1, 0.63999)
Mesh 2	(1, 0.09999)	(1.23748, 0.4)	(1, 0.64499)
Mesh 3	(1, 0.09874)	(1.23624, 0.4)	(1, 0.64874)
Mesh 4	(1, 0.09937)	(1.23563, 0.4)	(1, 0.64999)
Reference	$P_4(x_4, y_4)$		$P_5(x_5, y_5)$
Mesh 1	–		–
Mesh 2	(1.3925, 0.4)		(1, 0.4075)
Mesh 3	(1.3900, 0.4)		(1, 0.4100)
Mesh 4	(1.3887, 0.4)		(1, 0.4112)

4.3.1.1. Velocity profiles. In Fig. 22, the velocity profiles at $x = 0.7$ and $y = 0.7$ present good accordance with those obtained by Zhou et al. [21].

4.3.1.2. Positions and intensities of the vortices. Tables 10 and 11 represent the positions and intensities of the primary and secondary vortices as a function of the mesh size for $Re = 1000$. Positions and intensities of the ternary vortex are shown in Table 12. For the sake of conciseness, we only focus on the positions and the intensities of one ternary and two secondary vortices. The others can be obtained symmetrically in relation to the center of the cavity. We observed symmetrical values to up to four or five significant digits. Table 13 displays the positions of the detachment and reattachment points $P_i (i = 1,5)$ defined in Fig. 21. In [21,22], the authors use coarser grids which could explain the difference between their values and ours for the primary vortices.

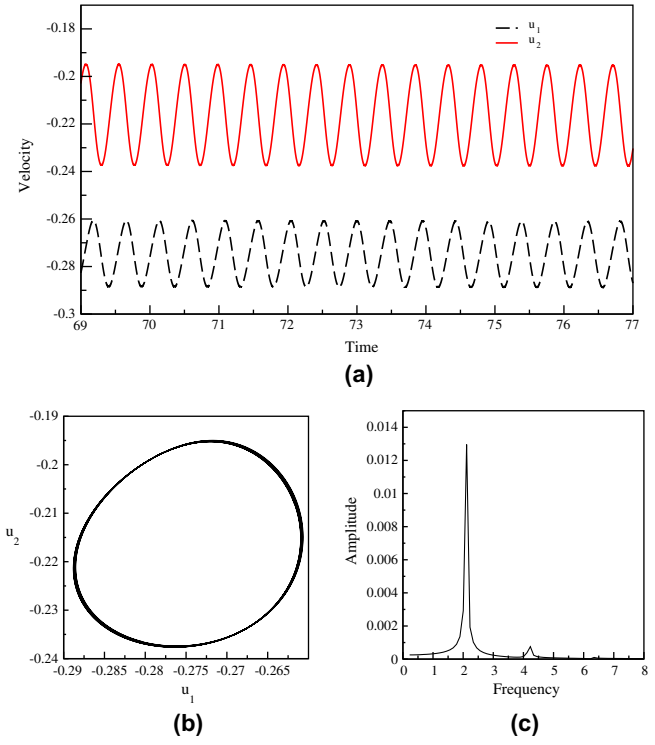

Fig. 23. Flow for $Re = 5000$. (a) u_1 and u_2 versus time t. (b) Phase trajectory on $u_1 - u_2$ plane. (c) Fourier power spectrum of the u_1 velocity.

4.3.2. Unsteady flow

In this section, we propose a first insight of an unsteady double lid driven cavity. The transition process is illustrated for $Re = 5000$ and $Re = 10{,}000$ using time velocity histories, Fourier power spectra and phase-space trajectories.

In the present case, we used mesh 3 (10^6 nodes). Flow becomes unsteady and periodic for $3480 < Re < 3500$ whereas Zhou [21] presents an unsteady flow at $Re = 3200$. This difference can be due to his coarser grids.

For $Re = 5000$, the time evolution of u_1 and u_2 variables, the phase trajectory on $u_1 - u_2$ plane and the power spectrum at the point $(x = 1, y = 1)$ between $t = 69$ s and $t = 77$ s are represented in Fig. 23. The flow is periodic with a fundamental frequency equal to 2.1151 Hz. Fig. 24 represents the evolution of the streamlines during a period giving prominence to the periodicity of the flow for $Re = 5000$. The mean flow is not symmetrical. The right lower vortex is very unstable while the others remain quite stable. This phenomenon has already been noticed in the 2D lid driven square cavity [16].

For $Re = 10{,}000$, time evolution of u_1 and u_2 variables, the phase trajectory on u_1–u_2 plane and the power spectrum at the point

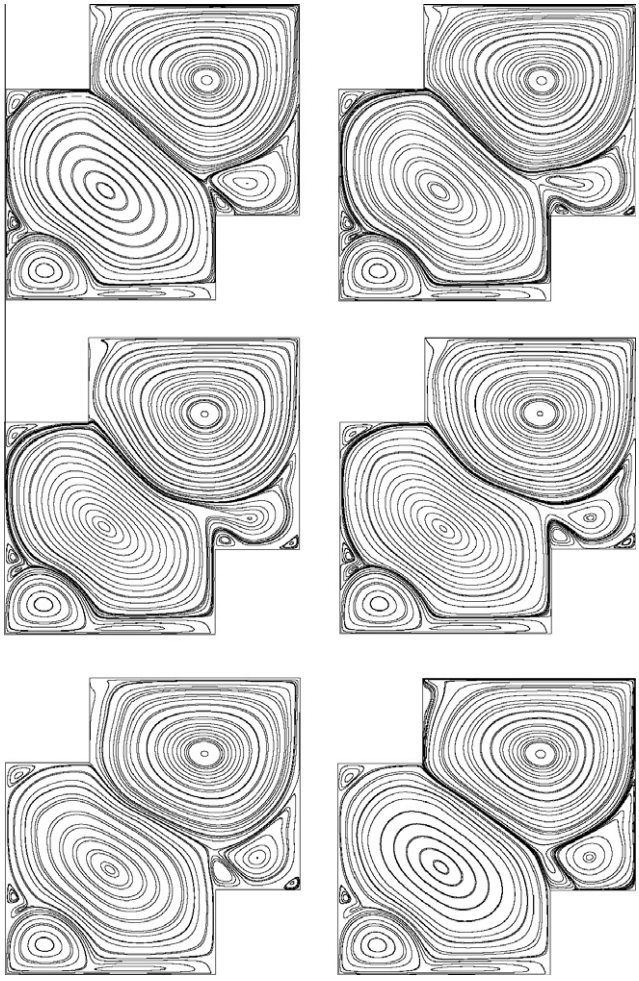

Fig. 24. Evolution of the streamlines during a period at $Re = 5000$ (it reads from left to right and vertically).

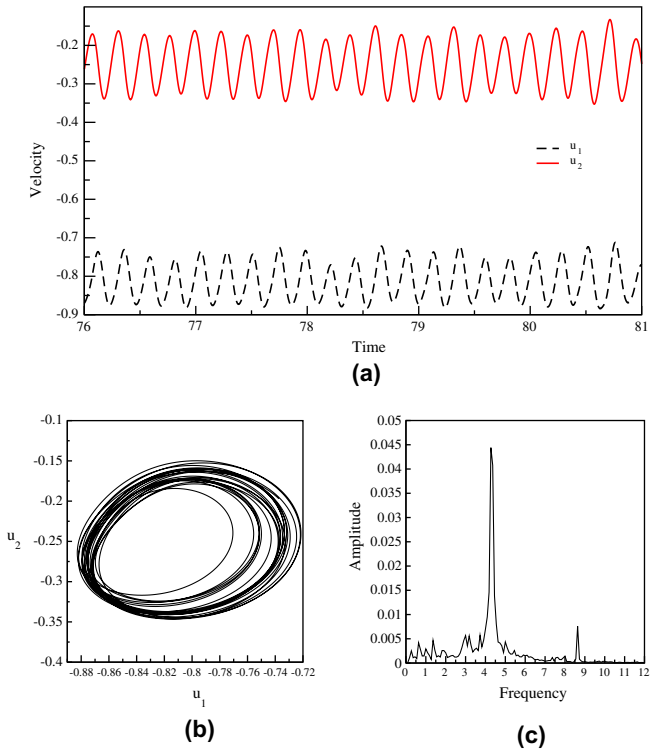

Fig. 25. Flow for $Re = 10000$. (a) u_1 and u_2 versus time t. (b) Phase trajectory on $u_1 - u_2$ plane. (c) Fourier power spectrum of the u_1 velocity.

$(x = 1, y = 1)$ between $t = 76$ s and $t = 81$ s are represented in Fig. 25. The flow becomes quasiperiodic with non-negligible variations in amplitude. The main frequency equals 4.2729 Hz.

5. Conclusion

In this paper, we proposed a method for partitioning 2D block-structured meshes. The goal was to compute flow simulations on non-rectangular geometries. Our geometrical partitioner was coupled with the massively parallel solver and preconditioner Hypre library. Several examples of partitioning are presented to check, both the efficiency and the performance of our strategy in comparison with other partitioners. Finally, we computed flow on non-rectangular geometries with very fine grids to propose reference solutions.

Acknowledgements

The authors thank Professor Michel Deville for his invitation to contribute to this special issue and Michel Gazaix, Engineer at ONERA who provided the partitioning results for the REB and GA methods. We acknowledge the calculation facilities financially supported by the Conseil Régional d'Aquitaine and the French Ministry of Research and Technology.

References

[1] Dongarra J, Foster I, Fox G, Gropp W, Kennedy K, Torczon L, et al. Sourcebook of parallel computing. Morgan Kaufmann Publishers Inc.; 2003.
[2] Hendrickson B, Leland R. The Chaco users guide, version 2.0. Technical report, Sandia National Laboratories, Albuquerque, NM; July 1995.
[3] Karypis G, Kumar V. METIS: unstructured graph partitioning and sparse matrix ordering system, Technical report, University of Minnesota, Computer Science Department, Minneapolis; 1995.
[4] Pellegrini F. SCOTCH 3.1 users guide. Technical report 1137-96, LaBRI, University of Bordeaux, France; 1996.
[5] Berger MJ, Bokhari SH. A partitioning strategy for nonuniform problems on multiprocessors. IEEE Trans Comput 1987;C-36:571–80.
[6] Ytterström A. A tool for partitioning structured multiblock meshes for parallel computational mechanics. Int J Supercomput Appl High Perform Comput 1997;11:336–43.
[7] Gourdain N, Gicquel L, Montagnac M, Vermorel O, Gazaix M, Staffelbach G, et al. High performance parallel computing of flows in complex geometries: I. Methods. Comput Sci Discovery 2009;2 [art. no. 015003].
[8] Rantakokko J. Partitioning strategies for structured multiblock grids. Parallel Comput 2000;26(12):1661–80.
[9] Falgout RD. Hypre high performance preconditioners user's manual. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory; 2008.
[10] Schaffer S. A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients. SIAM J Sci Comput 1998;20:228–42.
[11] Brown PN, Falgout RD, Jones JE. Semicoarsening multigrid on distributed memory machines. SIAM J Sci Comput 2000;21:1823–34.
[12] Falgout RD, Jones JE, Yang UM. Conceptual interfaces in hypre. Future Generation Comput Sys 2006;22:239–51 [Special issue on PDF software].
[13] Goda K. A multistep technique with implicit difference schemes for calculating two and three dimensional cavity flows. J Comput Phys 1979;30:76–95.
[14] Gear C. Numerical initial value problems in ordinary differential equation. Prentice-Hall; 1971.
[15] Peng YF, Shiau YH, Hwang RR. Transition in a 2-D lid driven cavity flow. Comput Fluids 2003;32:337–52.
[16] Bruneau CH, Saad M. The 2D lid-driven cavity problem revisited. Comput Fluids 2006;35:326–48.
[17] Perng CY, Street L. A coupled multigrid-domain-splitting technique for simulating incompressible flows in geometrically complex domains. Int J Numer Methods Fluids 1991;13:269–86.
[18] Hribersek L, Skerget L. Boundary domain integral method for high Reynolds viscous fluid flows in complex planar geometries. Comput Methods Appl Mech Eng 2005;194:4196–220.
[19] Moffatt HK. Viscous and resistive eddies near a sharp corner. J Fluid Mech 1964;18:1–18.
[20] Oosterlee CW, Wesseling P, Segal A, Brakkee E. Benchmark solutions for the incompressible Navier–Stokes equations in general co-ordinates on staggered grids. Int J Numer Methods Fluids 1993;17:301–21.
[21] Zhou YC, Patnaik BSV, Wan DC, Wei GW. DSC solution for flow in a staggered double lid driven cavity. Int J Numer Methods Fluids 2003;57:211–34.
[22] Nithiarasu P, Liu CB. Steady and unsteady incompressible flow in a double driven cavity using the artificial compressibility (AC)-based characteristic-based split (CBS) scheme. Int J Numer Methods Fluids 2005;63:380–97.
[23] Theofilis V. Advances in global linear instability analysis of nonparallel and three-dimensional flows. Prog Aerospace Sci 2003;39:249–315.
[24] De Vicente J, Rodríguez D, Theofilis V, Valero E. On high-Re numerical solutions in spanwise-periodic lid-driven cavity flows with complex cross-sectional profiles. Comput Fluids, submitted for publication.