Technical Report

TR-2015-004

Experimental Optimization of Parallel 3D Overlapping Domain Decomposition Schemes

by

Sofia Guzzetti, Vaidy Sunderam, Alessandro Veneziani

Mathematics and Computer Science

EMORY UNIVERSITY

Experimental Optimization of Parallel 3D Overlapping Domain Decomposition Schemes*

Sofia Guzzetti[1], Alessandro Veneziani[1], and Vaidy Sunderam[1]

Department of Mathematics and Computer Science, Emory University, Atlanta GA, USA

Abstract. Overlapping domain decomposition is a technique for solving complex problems described by Partial Differential Equations in a parallel framework. The performance of this approach strongly depends on the size and the position of the overlap since on the one hand more overlap increases the computational costs for the single subdomain but on the other hand it accelerates the iterative procedure of the global decomposition solving. In this paper we test the overlapping domain decomposition method on the finite element discretization of a diffusion reaction problem in both idealized and real 3D geometries. Results confirm that the detection of the optimal overlapping in real cases is not trivial but has the potential to significantly reduce the computational costs of the entire solution process.

1 Introduction

Advanced applications in several fields of engineering based on partial differential equations (PDEs) challenge high performance and parallel computing to reduce computational costs within timelines of practical relevance. In this context, *domain decomposition techniques* (DD) provide an important framework to associate mathematical formalism to the parallel solution of a complex PDEs system - see e.g. [QV99,TW05]. With this technique, the problem over a region of interest Ω is decomposed in subproblems to be iteratively solved by single processors or clusters up to the fulfillment of a convergence criterion stating that the solution found is equivalent to the one of the unsplit problem. Several types of DD approaches have been investigated from the viewpoint of mathematical and numerical performances. Their relevance is increasing with the adoption of heterogeneous computational resources and the detection of optimal splitting is of paramount importance for the global performance of the parallel solver. In fact, splitting is important not only from the mathematical point of view (for the numerical efficiency of the iterative-by-subdomain solver), but also from the computational viewpoint, since the subdivision drives the communications among the different processors and therefore it has a major impact on the global computational time.

In previous work [Sla14] we have investigated the performance of a non-overlapping DD solver for problems in fluid dynamics of blood in cerebral arteries when using heterogeneous architectures. *Non overlapping* means that the different subdomains do not share portions of Ω. The information iteratively exchanged by subdomains at the interfaces must obey compatibility constraints to guarantee that the decomposed problem is equivalent to the unsplit one. In this work, we focus on a different approach, based on the so called *overlapping* DD. The advantages of overlapping are that the iterative solver requires in general a lower number of iterations to converge and that the conditions to be exchanged at the interface can be selected freely (apart from obvious constraints of physical consistency). On the other hand, with overlapping DD the PDE problem is solved multiple

* Research supported in part by National Science Foundation grant OCI-1124418.

times on the overlapping regions, with potential computational duplication overhead. The interplay of (i) additional numerical costs due to the overlap, (ii) efficiency advantages induced by the specific iterative methods and (iii) versatility of the selection of domain interfaces (and the associated conditions) for the communication time, is not trivial in problems of practical interest. Numerical testing in 2D in a special class of differential problems (Poisson equation) has been carried out in [DEK06]. Here we want to perform a similar extensive analysis of the performances of overlapping DD in more general problems (reaction-diffusion) in both idealized and realistic 3D domains. The specific goal is to verify the performances of these schemes in problems where the geometry of the domain of interest plays a major role. In particular, we have a specific interest in problems related to computational hemodynamics. The realistic geometries here are retrieved from a data base of cerebral arteries [APPV].

Our results show that an appropriate selection of overlap and interfaces introduces significant computational benefits, consistent with the finding of [DEK06]. In fact, for a 3D cylindrical geometry, we found an optimal p of 20 %, that is similar to the values found on rectangular domains in that paper.

The outline of the paper is as follows: we recall basics of Overlapping DD in Section 2. Detailed description of the software used for numerical tests is provided in Section 3. Section 4 describes the test cases solved, while results are commented in Section 5. Conclusions and perspectives are drawn in Section 6.

2 Overlapping DD in a nutshell

Let us consider the following differential *Diffusion-Reaction* (DR) problem

$$-\sum_{i=1}^{3}\frac{\partial}{\partial x_i}\left(\mu\frac{\partial u}{\partial x_i}\right)+\sigma u=f \tag{1}$$

for $(x_1,x_2,x_3)\in\Omega\subset\mathbb{R}^3$ with $\mu>0$ and σ coefficients for simplicity assumed to be constant. The forcing term f is a given function of x_1,x_2,x_3. Hereafter it will be set to 0. To the equation, we associate the boundary conditions $u(\Gamma_D)=g(x_1,x_2,x_3)$, $\dfrac{\partial u}{\partial \mathbf{n}}(\Gamma_N)=0$, where Γ_D and Γ_N are two disjoint portions of the boundary of Ω such that $\Gamma_D\cup\Gamma_N=\partial\Omega$. Here the unknown u may represent the density of a species in a region where it diffuses with diffusivity μ and it undergoes to a chemical reaction with rate σ.

To take advantage of domain decomposition, we first split the domain Ω into two overlapping subdomains Ω_1 and Ω_2, such that $\Omega_1\cap\Omega_2=\Omega_o$ and $\Omega_1\cup\Omega_2=\Omega$. Let us denote by Γ_j the interfaces between the two subdomains ($j=1,2$), that is the portion of the boundary of Ω_j that is not also boundary of Ω, in short $\Gamma_j\equiv\partial\Omega_j\setminus(\partial\Omega_j\cap\partial\Omega)$. The solution of the problem in each subdomain will be denoted by $u_j(x_1,x_2,x_3)$. We reformulate the original problem in an iterative fashion. Given an initial guess $u_j^{(0)}$ (typically $=0$), we solve on each subdomain for $k=1,2,\ldots$

$$-\sum_{i=1}^{3}\frac{\partial}{\partial x_i}\left(\mu\frac{\partial u_j^{(k)}}{\partial x_i}\right)+\sigma u_j^{(k)}=f\qquad\text{in }\Omega_j, j=1,2 \tag{2}$$

with boundary conditions

$$u_j^{(k)}(\Gamma_D\cap\partial\Omega_j)=g(x_1,x_2,x_3),\quad \frac{\partial u}{\partial \mathbf{n}}(\Gamma_N\cap\partial\Omega_j)=0,\quad u_j^{(k)}(\Gamma_j)=u_{\widehat{\jmath}}^{(k-1)}(\Gamma_j), \tag{3}$$

(where $\widehat{j} = 2$ for $j = 1$ and $\widehat{j} = 1$ for $j = 2$) up to the fulfillment of the convergence condition. In our case this condition checks that the solution in the overlapping region is not changing significantly along the iterations (as we will detail later on).

Notice that at each iteration we solve two independent problems in each subdomain, while the communication by subdomain occurs in the latter of boundary conditions (3).

The convergence of the iterative scheme depends on the size of the overlapping region. In fact, if the overlapping is 100 % of Ω, convergence is trivially guaranteed as at the first iteration (23) we are solving (twice) the unsplit problem. On the other hand, if the overlapping reduces to a volume-zero region, convergence is not guaranteed, as in general the juxtaposition of the two problems does not coincide with the original problem (this occurs only if the interface conditions are chosen properly).

The one presented here is the so called *additive* formulation of the overlapping DD method, where the two subdomain problems can be solved simultaneously - as opposed to the *multiplicative* version, where one subdomain can be solved only when the problem on the other subdomain is completed. In the multiplicative formulation a faster convergence is guaranteed in terms of number of iterations (about one half of the additive scheme), but the advantage of the parallel setting is limited by the sequential structure of the algorithm. From now on, we refer only to the additive algorithm.

The numerical solution of the problem by subdomains can be performed for instance by the so called Finite Element Methods. This is done by introducing a tessellation or *mesh* of each subdomain in a (generally large) number of tetrahedra where the solution is locally assumed to be linear. The number of the tetrahedra on each mesh is denoted by N. The more tetrahedra we have, the more accurate the numerical approximation is, but the more computationally expensive it will be since we resort to solving a larger system. For the same reason, the larger the computational domain, the higher the computational costs.

The selection of the interfaces Γ_j - provided that the overlapping volume is not null - is relatively free. Here we want to investigate the optimal selection as the result of the trade-off between the computational cost of each subproblem and the reduction of the communications between processors attained by a smart selection of the interfaces.

3 Software Tools

Once a mathematical representation of Ω is available, numerical solution consists of the following steps.
(a) The region is meshed by a public domain software called `NetGen`. A mesh for complex geometries like the ones we consider in this paper in general cannot rely on particular assumptions on the shape of Ω and it is called *unstructured*. A text file storing the coordinates of the vertexes of the reticulation and their connectivity is the result of this step. The maximal size of elements is denoted by h.
(b) The mesh is split by subdomains using a specific library based on the package ParMETIS. This is a MPI-based parallel library that implements a variety of algorithms for partitioning large unstructured meshes. ParMETIS extends the functionality of the serial package METIS [Lab] for large-scale parallel numerical simulations. In particular it aims at (i) reducing the communication time by minimizing the number of interface elements; (ii) balancing the number of elements for subdomain so to create a nearly uniform load distribution. Graph interpretation of the mesh and successive partitioning successfully drives to the fulfillment of these conditions. Unfortunately, the original package does not manage overlapping. Ad hoc MATLAB scripts were prepared to add to an existing non-overlapping partition an overlapping region symmetrically with respect to the original

splitting. At the end of the partitioning step, we have two mesh files and the map of the nodes of each interface Γ_j in the corresponding domain $\Omega_{\widehat{j}}$.
(c) On each subdomain, after a proper labeling of the vertexes to identify the portions of the boundary/interface nodes associated with different boundary conditions (3), we solve the problem (2) by the Finite Element method. This means that on each element of the mesh (a tetrahedron) we assume the solution of the problem to be linear in the space coordinates x_1, x_2, x_3 so that equations (3) can be conveniently reformulated as a linear algebraic system. The assembly of these matrices has been performed through the MATLAB package `fast_fem_assembly` [RV13]. More precisely, subroutines of this library build the so-called *stiffnes* and *mass* matrices associated with the discrete formulation of the problem. The system is then solved with a built-in MATLAB function that implements the iterative method called GMRES. As it can be proved from the theory of the FEM, the larger N and the larger the linear system to be solved (higher computational costs) and the more accurate the numerical solution. Similarly, the larger the subdomain, the more elements will be needed to provide an accurate solution, so the larger the computational costs will be.

4 Benchmarking

In our tests we include in the computational time the iterative-by-subdomain process. Meshing, partitioning and matrix assembly are not included in this analysis, since they are off-line costs that do not depend on the specific solution procedure. The time $T_{it}^{(k)}$ of each iteration (k) is computed as the maximum of the two parallel subdomain solution times $T_j^{(k)}$,

$$T_{it}^{(k)} = \max_{j=1,2} T_j^{(k)}.$$

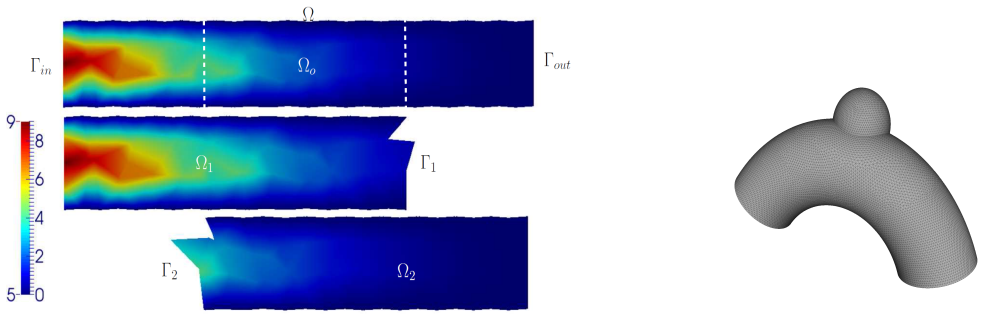
The single processor time is given by the time for solving the linear system added by the communication time to read from the other processor the last of conditions (3), $T_j^{(k)} = T_{j,sol}^{(k)} + T_{j,com}^{(k)}$. For this particular problem, we may speculate that the computational cost per iteration is constant (denoted by $\overline{T_{sol} + T_{com}}$) , so we get

$$T = \sum_{k=1}^{N_{it}} T_{it}^{(k)} \approx N_{it}(\overline{T_{sol} + T_{com}}).$$

If we denote by p the percentage of overlap in the domain splitting (i.e. the ratio of the volume of the intersection of the domains to the total volume of the geometry), theory of overlapping DD proves that N_{it} *decreases* with p, T_{sol} increases with p while T_{com} depends on the position of the interfaces (precisely on the number of vertexes of the mesh on the interface), so it may change with p in an unpredictable way for a complicated geometry. We therefore expect that the value p has a major impact on the solver performances depending on the different geometries. We will test this hypothesis in the next Section. For this reason, we consider both idealized and real geometries from computational fluid dynamics applications and described hereafter. It is worth noting that the total cost is a function of the mesh size N too. In this case, both factors N_{it} and $\overline{T_{sol} + T_{com}}$ get larger with h, as a price to pay to the improvement of the accuracy of the approximated solution achieved in this way. The sensitivity of our results to N will be tested too.

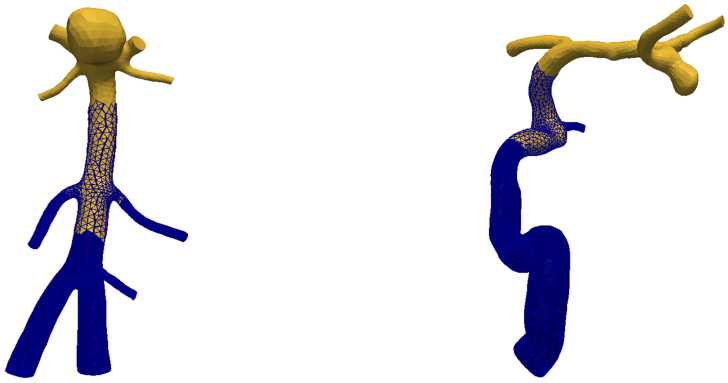
4.1 Idealized geometries

Test 1: Cylinder We consider a cylinder of length $L = 6cm$ and radius $R = 0.5cm$. We use five meshes with different values of N, for each of the sizes of the overlap. The numerical solution shown in Figure 1a, representing a longitudinal cut of the 3D solution, displays the overlapping subdomains (bottom panels) and the associated solution perfectly matching at the interfaces.



(a) Test 1 (b) Test 2

Fig. 1: Overlapping DD solution in a cylinder (longitudinal section): in the aggregated domain Ω (top) and by subdomains (bottom). (b) Idealized geometry of an aneurysm, made of a sphere merging with a portion of a torus.



(a) Test 3 (b) Test 4

Fig. 2: Geometries of patient-specific cerebral aneurysms from the Aneuriskweb Repository, used for Test 3 and 4 respectively.

Test 2: Idealized Aneurysm. In this case (shown in Figure 1b) we consider an idealized representation of a cerebral aneurysm (a pathology of the cerebral circulation) where a curved cylinder (namely a torus with radius 2) representing an artery is merged with a sphere of radius 0.5, representing the sac of the aneurysm. This test intends to emphasize the role of communication time. In fact

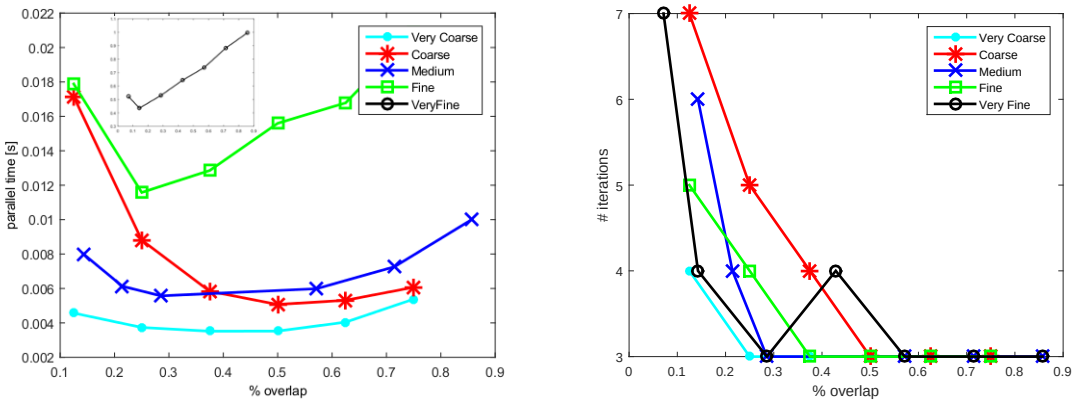
a splitting with an interface intersecting the sac has more vertices than with interfaces involving only the artery. Overlapping DD allows to manage the location of the interfaces so to avoid many vertices on the interface yet preserving workload balance between the subdomains.

4.2 Real geometries

A web repository of cerebral aneurysm geometries is freely available [APPV]. We selected two cases with significant features for the purpose of the paper.

Test 3: Patient 1. This geometry is approximately a cylindrical shape with a terminal spherical aneurysm located at the bifurcation (see Figure 2a). In spite of the simplicity, the collateral ramifications make the *a priori* prediction of performances quite problematic.

Test 4: Patient 2. This geometry is more complicated for the curvature and the torsion of the main vessel (see Figure 2b). The size of the aneurysm is smaller and a major bifurcation occurs along the main vessel (the carotid siphon).



(a) Parallel time as a function of the overlap.

(b) Number of iterations of the parallel solver.

Fig. 3: Parallel time performed by an ADR parallel solver as a function of the entity of overlap for five levels of refinement of the mesh on a cylinder (a). Corresponding number of iterations for very coarse (•), coarse (∗), medium (×), fine (□) and very fine (○) meshes (b).

5 Results

Test 1 Figure 3a shows the parallel running time as a function of p. The varying dependence of number of iterations and cost per iteration on p results in a convex behavior of the computational time. This behavior is expected, since for small p the high number of iterations dominates the cost, while beyond a certain value it does not decrease any longer (Fig. 3a), while the cost per processor increases.

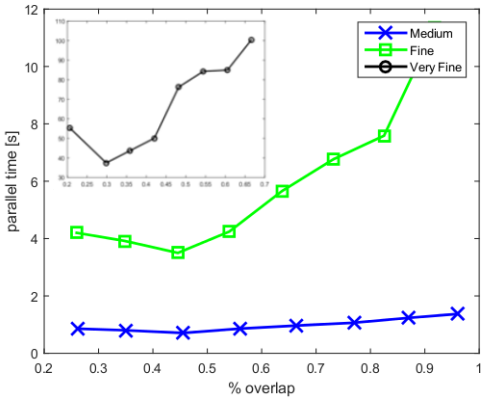
The size of the meshes for all the tests is $N \simeq 3000$ (very coarse), $N \simeq 6000$ (coarse), $N \simeq 12000$ (medium), $N \simeq 24000$ (fine), $N \simeq 300000$ (very fine). In addition, the value of the optimal p changes with N (Fig. 3a). For the very coarse and coarse meshes, the optimal overlap happens between 37.5% and 50%. The medium and fine meshes perform better with 20-37.5% overlap, while the very fine grid reaches the minimum at 14%. The optimal overlap shifts leftward as the mesh size decreases. In fact, for coarse meshes, the cost-per-iteration is small and a large overlap is totally beneficial. For fine meshes, a larger overlap results in a significant burden for the solution on the subdomains.

The number of nodes that lie on the inner interfaces - that is directly proportional to the communication time - is nearly constant with p since we have a simplified geometry with a constant number of vertexes along the transversal sections. However, it gets larger when N increases.

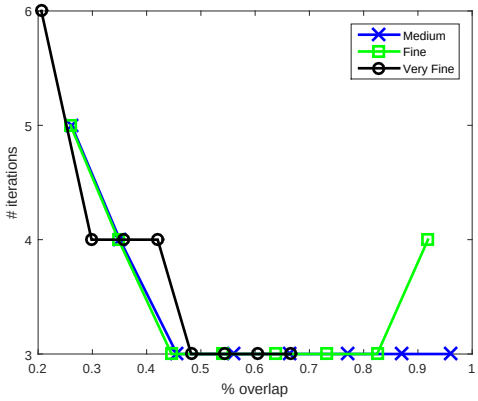
Notice that the optimal range for the medium-refined meshes agrees with the results in [DEK06] on a 2D rectangular domain.

Test 2 The numerical results in Figure 4 show that medium and fine grids perform better with a superposition around 45%. Interestingly, the curve related to the very fine mesh features a minimum at a fraction of overlap of $\sim$ 30%. This corresponds to the minimal overlap including the sphere entirely, so that the interfaces do not cross the bulb. In this way, the number of vertices is contained (as opposed to the case with an interface through the aneurysm) with a benefit for the communication time. This shows that the overlap allows the minimization of communication time thanks to a flexible positioning of the interfaces, yet preserving workload balance by subdomains. For coarser meshes, the extra computational cost associated with an increase of p does not have a great impact on the cost of an iteration and a wider intersection is allowed.

In addition, Figure 4b shows that the minimum fraction of volume at which the minimum number of iterations is attained grows when N increases. In particular, the finest mesh requires the greatest value of p to minimize N_{it}, as expected from the theory of DD.



(a) Parallel time as a function of the overlap.



(b) Number of iterations of the parallel solver.

Fig. 4: Parallel time performed as a function of p for three levels of refinement of the mesh on an idealized aneurysm (a). Corresponding number of iterations for medium ($\times$), fine ($\square$) and very fine ($\bigcirc$) meshes (b).

Test 3 In the real case, the complexity of the geometry makes the analysis of the results more difficult. We comment only the case we ran on the very fine mesh, since similar considerations hold for the other cases, but this is the case of practical interest.

For this geometry the optimal value of overlap is localized around 7%. This happens when the number of subdomain iterations is minimized to the value of 3 and a further increment of the overlap (in the range considered here) does not bring any further reduction of this number. In particular Table 1 points out that the optimal partitioning occurs when a large number of interface vertices is corresponded by a well balanced workload of the subdomains. This partition is located at the branching of collateral vessels of the artery. A shift of the interfaces would include the entire set of those vessels in one of the two subdomains, with an additional burden to the solution in that subdomain. In fact, in this case the complexity of the problem weights the cost per subdomain more than the communication time.

% overlap	DoF1	DoF0	InterNodes	Parallel Time [s]
1.60	25,469	28,017	446	1.30770
3.63	25,906	28,273	549	0.84232
7.74	27,420	28,810	869	0.65015
12.13	29,105	19,194	601	0.70613
16.34	30,320	29,618	510	0.72133
20.51	31,201	30,127	534	0.75693
24.73	32,132	30,620	489	0.75093

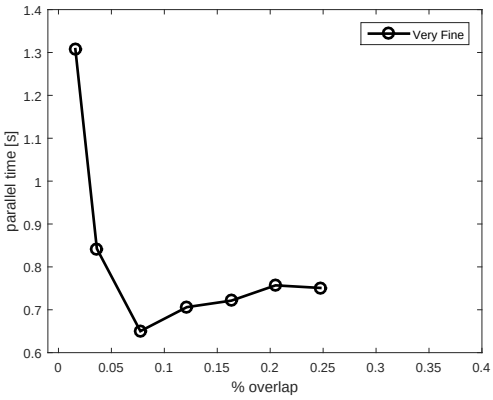
Table 1: Number of nodes of each partition (DoF1, DoF0), total number of nodes on the interfaces (InterNodes) and parallel times (in s) for different levels of overlap on a very fine mesh for Case C0095.

Test 4 For Patient 2 we find similar results to Patient 1. The optimal partition penalizes the size of the local problem more than the communication time induced by the number of vertices at the interface. The minimal time corresponds therefore to interfaces cutting collateral vessels so to balance the workload between subdomains. The tortuosity of the vessel and the size of the aneurysm do not play any significant role in this respect. This is related to the particular problem we are solving, where the species undergoes to "isotropic" dynamics (diffusion and reaction) as opposed to more directional phenomena like the presence of drift. This will be subject of further investigations.

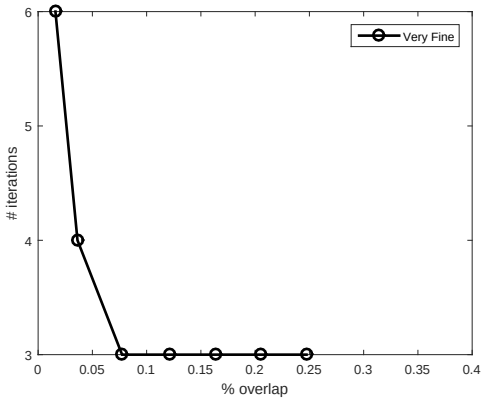
6 Conclusions and Perspectives

This work presents preliminary results on finding the optimal partition in an overlapping DD approach to minimize the computational time for parallel solution. Different aspects have competitive dynamics resulting in a nontrivial optimization. As a matter of fact, the dependence of the number of iterations on the iterative-by-subdomain method decreases with the overlap, while the cost of the solution on each subdomain increases. The communication time depends on the location of the interface.

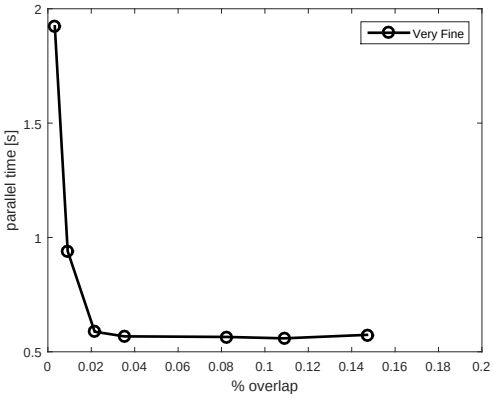
Overlapping DD has some advantages, in particular it allows a more flexible positioning of the interfaces. In addition, the subdomain conditions (3) can be designed for a further optimization



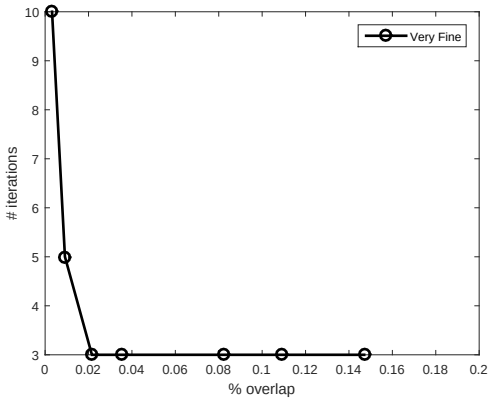
(a) Patient 1 - Parallel time.



(b) Patient 1 - Number of iterations.



(c) Patient 2 - Parallel time.



(d) Patient 2 - Number of iterations.

Fig. 5: Parallel time performed by an ADR parallel solver as a function of the entity of overlap (left) for a very fine mesh fore Patient 1 (top) and Patient 2 (bottom). Corresponding number of iterations (right).

of the procedure, in what is called *Optimized Schwartz approach*. The definition of the optimal partitioning depends on many aspects related to both the mathematical algorithm and the computer architecture. The nature of the PDE problem to be solved has generally an impact too. For this reason it is not easy to draw general conclusions. However, our results in realistic geometries point out the efficacy of an appropriate selection of overlapping to reduce costs in a parallel computing setting. In general, a small amount of overlap results in a good trade-off of all the competitive mechanisms affecting the total computational time. This is just the first of several experiments we intend to pursue. In particular we will investigate the impact of overlapping on

1. different differential problems in particular related to incompressible fluids (as done in [SPV$^+$12]);
2. optimized interface conditions (optimized Schwartz);
3. flexible strategies, when the accuracy pursued to solve each subdomain is reduced to accelerate the procedure, possible performing more iterations (see [DEK06]);
4. heterogeneous architectures (possibly dislocated in space) on splitting featuring many subdomains.

The ultimate goal is the definition of practical criteria for a partitioning software able to perform an optimal splitting of the domain Ω based not only on the work balance by subdomains but also on the communication time and more in general the parallel architectures used.

References

[APPV] Luca Antiga, Tiziano Passerini, Marina Piccinelli, and Alessandro Veneziani. Aneurisk web, ecm2.mathcs.emory.edu/aneuriskweb.

[DEK06] David Darjany, Burkhard Englert, and Eun Heui Kim. Implementing Overlapping Domain Decomposition Methods on a Virtual Parallel Machine. In Geyong Min, Beniamino Di Martino, Laurence T. Yang, Minyi Guo, and Gudula Rünger, editors, *Frontiers of High Performance Computing and Networking ISPA 2006 Workshops*, volume 4331 of *Lecture Notes in Computer Science*, pages 717–727. Springer Berlin Heidelberg, 2006.

[Lab] Karypis Lab. Metis, http://glaros.dtc.umn.edu/gkhome/views/metis.

[PSVS14] Tiziano Passerini, Jaroslaw Slawinski, Umberto Villa, and Vaidy Sunderam. Experiences with cost and utility trade-offs on iaas clouds, grids, and on-premise resources. In *Cloud Engineering (IC2E), 2014 IEEE International Conference on*, pages 391–396. IEEE, 2014.

[QV99] Alfio Quarteroni and Alberto Valli. Domain decomposition methods for partial differential equations. Technical report, Oxford University Press, 1999.

[RV13] Talal Rahman and Jan Valdman. Fast MATLAB assembly of FEM matrices in 2D and 3D: Nodal elements. *Applied Mathematics and Computation*, 219(13):7151–7158, 2013.

[Sla14] Jaroslaw K. Slawinski. *Adaptive Approaches to Utility Computing for Scientific Applications*. PhD thesis, Emory University, 2014.

[SPV$^+$12] J Slawinski, T Passerini, U Villa, A Veneziani, and V Sunderam. Experiences with Target-Platform Heterogeneity in Clouds, Grids, and On-Premises Resources. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, pages 41–52, 2012.

[TW05] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods: Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, 2005.