



HAL
open science

Simulation of soft tissue deformation in real-time using domain decomposition

Ryadh Haferssas, Pierre-Henri Tournier, Frédéric Nataf, Stéphane Cotin

► **To cite this version:**

Ryadh Haferssas, Pierre-Henri Tournier, Frédéric Nataf, Stéphane Cotin. Simulation of soft tissue deformation in real-time using domain decomposition. 2019. hal-02400982v1

HAL Id: hal-02400982

<https://hal.science/hal-02400982v1>

Preprint submitted on 9 Dec 2019 (v1), last revised 11 Dec 2019 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simulation of soft tissue deformation in real-time using domain decomposition

Ryadh Hafessas¹, Pierre-Henri Tournier², and Frédéric Nataf² Stéphane Cotin¹

¹ INRIA France

{ryadh.haferssas,stephane.cotin}@inria.fr

² Sorbonne Université

{frederic.nataf,tournier}@sorbonne-universite.fr

Abstract. Non-linear models used in dynamic simulations usually require the solution of multiple large and sparse linear systems in a successive manner. In this paper, we conduct a study of numerical solvers in the framework of real-time soft tissue deformation. Domain Decomposition paradigm has the potential of providing parallelism at both levels of equation assembling and linear system solving. In our case domain decomposition is employed to solve a non-linear model in a dynamic simulation in order to meet real-time computation by using parallel architecture. Numerical test on liver deformations using a non-linear deformation model is presented to evaluate the acceleration impact of the domain decomposition paradigm. Performances tests clearly show the efficiency of using a domain decomposition approach for real-time feedback.

Keywords: soft tissue deformation · non-linear model · linear systems · domain decomposition

1 Introduction

Image-guided therapy has revolutionized medicine, in its ability to provide care that is both efficient and effective. However, images acquired during an intervention are either incomplete, under-exploited, or can induce adverse outcomes. This can be due, for instance, to the lack of dimensionality of X-ray images and the associated radiation exposure for the patient. In the same time, the scientific computing community developed a particular interest in medical models which attempt to provide numerical simulation to reproduce living anatomy or physiology of the specific patient. The main challenge is to combine numerical models with data extracted from intra-operative images and deliver efficient per-operative guidance to clinicians.

In this paper, we are mainly interested in the simulation of soft tissue, in the context described above. Whether we consider augmented reality or simply real-time elastic registration, the constraints are similar. Models that aim to mimic the mechanical behavior of complex anatomical structures must be accurate enough to predict the location of internal structures invisible in the intra-operative image

while providing visual feedback in real-time. This makes the rise of simulation constraints at several levels: mechanical modeling, equation discretization, and linear solvers. As a result, any simulation attempt would be necessarily looking for the best compromise between accuracy and computation time. At the mechanical level, there are several instances in the literature of real-time simulation of deformable object which rely on linear elasticity like in [3]. The first limitation is inherent to the small strain assumption: to keep a linear model, the magnitude of deformation must be very small. Non-linear models do not suffer from this constraint but they introduce an extra computational cost. At the time discretization level, one of the most know strategies to deal with real-time computation of the liver tissue deformations is based on an explicit integration scheme, as proposed in [14, 13]. Explicit integration methods are particularly well suited for some applications like the real-time non-rigid registration of the brain shift during surgery [9]. However, liver tissues are often very soft, but in most of the pathologic cases, these tissues are much stiffer. That makes the critical time step very small (about $10^{-6}sec$) which would not correspond to the real dynamics. Implicit integration allows the use of larger time steps (about $2 \cdot 10^{-2}sec$) without any stability issues. The counterpart is a heavier computation at each time step.

In practice, implicit integration schemes require to solve large systmes of linear or non-linear equations. In both cases, we need to use a solver of a linear system in a successive manner to compute a solution that represents the state of the simulated model. The computation time of this part is the most important fold of the overall simulation computation time. The main interest which has driven our work in this paper is the efficiency in solving a large algebraic linear system in parallel computation. During these last few years, the computational power available within the hardware of computers is evolving in a different way. Due to frequency and heat-dissipation limits, the current trend is focused on increasing the number of computation units rather than their individual speed. Nowadays, one or two quadcore processors can be found in standard desktop computers. Two families of methods are traditionally used to solve a large linear system arising from discretization of mechanical models on a parallel machine: direct and iterative solvers. Direct solvers are known to be very robust. However, the memory requirement becomes significant with larger systems. On the other hand, iterative solvers [15] e.g., GMRES, CG, are less memory consuming and naturally parallel but they suffer from a lack of robustness. Domain Decomposition method [19, 4] as well as multigrid [8] method are hybrid methods that can take advantage of direct solvers and iterative solvers in the same algorithm. These two groups of approaches are described as hybrid methods because they are ultimately used as a preconditioner for the linear system during an iterative method, but direct methods are also used within the definition of the global preconditioner on some smaller subsystems or auxiliary problems. Such a hybridization provides highly concurrent methods that are robust enough to solve complex real-life problems [12, 6, 18].

In this paper, we aim to assess and achieve performance gain in a typical case of liver deformation simulation using a non-linear model with implicit integration scheme. To this end, we propose to adopt a domain decomposition paradigm that introduces parallelism at two different levels, first at the model assembling then at equation solves. Such a strategy opens new opportunities to deal with accurate real-time simulations of soft organs, in particular in the context of complex interactions.

2 Method

In this section, we summarize the continuum framework, introduce a constitutive model along with the boundary value problem, and its numerical discretization.

2.1 Biomechanical model

To describe the mechanical behavior of the liver, we use a total Lagrangian formalism. In general, we consider a body whose reference configuration is Ω_0 at time t_0 , subjected to a force per unit mass \mathbf{f} , its boundary surface $\partial\Omega$ is divided into a Dirichlet part Γ_0^D constrained by a displacement $\bar{\mathbf{y}}$ and a Neumann part Γ_0^N subjected to a traction force $\bar{\mathbf{T}}$, the continuum equations stated in the strong form are

$$\begin{aligned} \rho_0 \ddot{\mathbf{y}} - \nabla \cdot \boldsymbol{\Sigma} &= \rho_0 \mathbf{f} && \text{in } \Omega_0, \\ \boldsymbol{\Sigma} \cdot \mathbf{n} &= \bar{\mathbf{T}} && \text{on } \Gamma_0^N, \\ \mathbf{y} &= \bar{\mathbf{y}} && \text{on } \Gamma_0^D. \end{aligned} \quad (1)$$

In these relations ρ_0 is the density, $\boldsymbol{\Sigma}$ is the second Piola-Kirchhoff stress tensor, and \mathbf{n} is the unit surface normal in the reference configuration.

Space integration In order to discretize problem (1) by a finite element method, consider a tetrahedral mesh $\{\mathcal{T}_h\}_{h>0}$ of the computational domain Ω_0 . The discretized finite element formulation results in a nonlinear system of algebraic equations

$$M\ddot{Y} + F^{int}(Y) = F^{ext}, \quad \forall t \in [0, T], \quad (2)$$

where initial, internal and external forces are respectively given by

$$\begin{aligned} M\ddot{Y} &= \int_{\mathcal{T}_h} \rho_0 N_i N_j dV \ddot{Y}, \\ F_i^{int} &= \int_{\mathcal{T}_h} \boldsymbol{\Sigma} : \nabla N_i dV, \\ F_i^{ext} &= \int_{\mathcal{T}_h} \rho_0 \mathbf{f} \cdot N_i dV + \int_{\Gamma_{0h}^N} \bar{\mathbf{T}} \cdot N_i^S dS, \end{aligned} \quad (3)$$

where M is the mass matrix, N_i is the conventional shape function corresponding to node $i \in [1, N]$ with N the number of nodes. $Y \in \mathbb{R}^N$ is the vector of the current nodal positions.

Time integration : We chose a conventional implicit integration scheme provided by Newmark [1]. To this end, we consider a positive integer N and define $\Delta t = T/N$, $t_n = n\Delta t$, for $n = 0, 1, \dots, N$. We compute the approximation Y^n by using the following second-order Newmark scheme

$$\begin{aligned} Y^{n+1} &= Y^n + \Delta t \dot{Y}^n + (1/4)\Delta t^2 \ddot{Y}^n, \\ \dot{Y}^{n+1} &= \dot{Y}^n + (1/2)\Delta t \ddot{Y}^n, \\ M\ddot{Y}^{n+1} + F^{int}(Y^{n+1}) &= F_{n+1}^{ext}. \end{aligned} \tag{4}$$

In hyperelastic models, the internal forces are provided by a non-linear function F^{int} . Here, we use the Newton-Raphson method to address the non-linearity at each time step.

The fully discretized problem (4) gives rise to a linear system of the form $Au = b$ which needs to be solved for each simulation time step and more than ones in case of non-linear models. Solving such a linear system could become extremely expensive from the computational point of view. Medical simulations are constrained by the need for real-time computation to enable interactivity of the simulation, this requirement translates into solving concurrently multiple linear systems under a very challenging time constraint.

Traditionally, to solve these linear systems, two types of approaches are used: direct and iterative solvers. Direct solvers provide the solution in a fixed number of steps. It mainly involves two phases: first, the factorization phase, e.g., LU, LDL^T , then, the solving phase. The factorization phase is independent of the right hand side and is computationally more expensive than the solving phase. Iterative solvers, e.g., GMRES, on the other hand, do not modify the matrix and rely solely on matrix-vector products and other basic algebra operations. However, for an iterative solver to be efficient, choosing a good preconditioner [15] is imperative, but in some cases finding a good preconditioner is a difficult task.

To overcome the disadvantages of iterative solvers and to take advantage of the desirable features of direct solvers in the framework of parallel computing, there has been an increasing focus on the so-called hybrid methods such as domain decomposition and multigrid methods. For this paper, we adopt in the numerical implementation a parallel strategy based on domain decomposition method.

2.2 Domain decomposition solver

Domain decomposition methods are known to be a divide & conquer paradigm to accelerate numerical simulations. In our simulation context, we choose to use an overlapping Schwarz method. To describe it, we first divide the mesh $\{\mathcal{T}_h\}_{h>0}$ in N non-overlapping meshes (the sub-domains) $\{\mathcal{T}_i\}_{1 \leq i \leq N}$ using standard graph partitioners, e.g., METIS [11]. If δ is a positive integer, the overlapping decomposition $\{\mathcal{T}_i^\delta\}_{1 \leq i \leq N}$ is defined recursively as follows: \mathcal{T}_i^δ is obtained by including

all elements of $\mathcal{T}_i^{\delta-1}$ plus all adjacent elements of $\mathcal{T}_i^{\delta-1}$. For $\delta = 0$, $\mathcal{T}_i^{\delta} = \mathcal{T}_i$. Let $\{\mathcal{V}_i^{\delta}\}_{1 \leq i \leq N}$ be the local deformation FE spaces defined on $\{\mathcal{T}_i^{\delta}\}_{1 \leq i \leq N}$.

Now, consider the restrictions $\{R_i\}_{1 \leq i \leq N}$ from \mathcal{V}_h to $\{\mathcal{V}_i^{\delta}\}_{1 \leq i \leq N}$ and a local partitions of unity $\{D_i\}_{1 \leq i \leq N}$ such that:

$$\sum_{j=1}^N R_j^T D_j R_j = Id_{n \times n},$$

where Id denotes the identity matrix and n is the global number of unknowns in the deformation space. Algebraically speaking, if n is the global number of deformation unknowns and $\{n_i\}_{1 \leq i \leq N}$ are the numbers of degrees of freedom in each local deformation FE space, then R_i is a Boolean matrix of size $n_i \times n$, and D_i is a diagonal matrix of size $n_i \times n_i$, for all $1 \leq i \leq N$.

Using the partition of unity, one can use the one-level preconditioner, Restricted Additive Schwarz (RAS) method, proposed by Cai and Sarkis [2]:

$$\mathcal{M}_{\text{RAS}}^{-1} = \sum_{i=1}^N R_i^T D_i A_i^{-1} R_i, \quad (5)$$

where the $\{A_i\}_{1 \leq i \leq N}$ are local operators defined by the submatrices $\{R_i A R_i^T\}_{1 \leq i \leq N}$. In this case, we thus chose to use a more sophisticated multilevel domain decomposition method using the GenEO approach [17, 6]. This preconditioner, $\mathcal{M}_{\text{GenEO}}^{-1}$, uses a spectral coarse grid to better couple all sub-domains.

3 Results

This section aims to assess the efficiency of linear solvers described in the previous section in the presence of non-linear deformation model. To this end, we consider Saint-Venant Kirchhoff as a constitutive law to model the liver mechanical response. The Saint-Venant Kirchhoff law is given by the following potential:

$$W(e) = \frac{\lambda}{2} (Tr(e))^2 + \mu (Tr(e^2)), \quad (6)$$

where e is the Green Lagrange tensor $e = \frac{1}{2} (\nabla \mathbf{y} + (\nabla \mathbf{y})^T + (\nabla \mathbf{y})^T \cdot \nabla \mathbf{y})$. Then the second Piola stress tensor is given by $\Sigma = \frac{\partial W}{\partial e}$. λ and μ are the Lamé coefficients that can be determined from the Youngs modulus E and Poissons ratio ν .

The geometry of the model is segmented from a patient pre-operative Computed Tomography (CT) image. The domain is then meshed in a set of linear tetrahedral elements using GMSH [5]. In all the following simulations, we use two different mesh discretizations. An initial tetrahedral mesh with 3316 elements which yields a linear system A with 2000 unknowns. Then, we refine the same mesh by splitting each element into multiple smaller elements to get a

finer mesh with 22208 elements, which yields a linear system A with 12321 unknowns. For domain decomposition purpose, the mesh is decomposed using the graph partitioner METIS [11] (figure 1). The resulting finite element linear system is preconditioned with $\mathcal{M}_{\text{GenEO}}^{-1}$, and the GMRES solver is stopped when the relative preconditioned residual is lower than 10^{-6} .

To implement the physical model, we have employed the open source simulation software FreeFem [7]. The linear solvers and the preconditioners are implemented in HPDDM [10]. We used the PARDISO [16] library for direct solver. Results were obtained on a standard desktop machine equipped with Intel with 6 Intel cores clocked at 3.2 GHz.

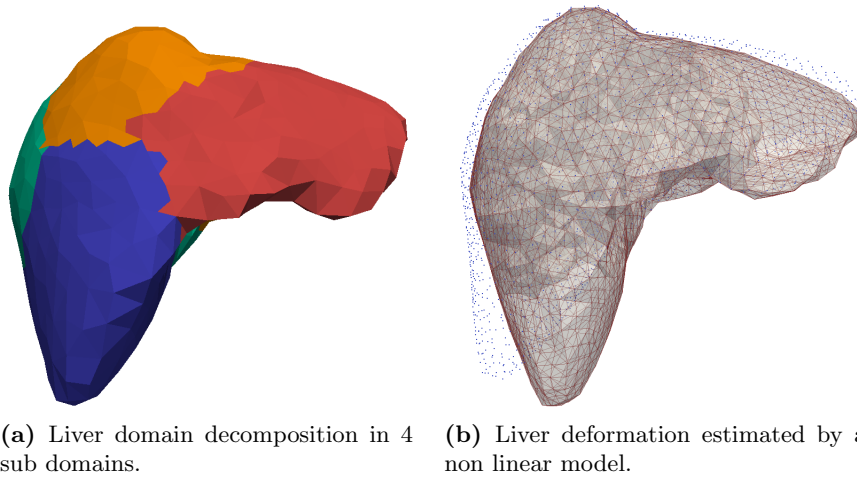


Fig 1: Left: liver computational domain decomposed in 4 sub-domains using the graph partitioner METIS [11] - Right: Liver deformation estimated by the non-linear Saint-Venant Kirchhoff law (the initial configuration is represented by blue points).

3.1 Static non-linear deformation

In this paragraph we evaluate the performances of pure direct solver versus domain decomposition solver for static deformation using a Newton-Raphson algorithm. We simulated an entire liver deformation. A volumic force of $100Pa$ in the $(x + y)$ direction is uniformly applied to the liver while several selected vertices of a plane are fixed (representing the ligament and veins). We considered Young modulus $E = 3 \cdot 10^3$ and Poisson ratio $\nu = 0.35$. The simulated deformation is shown in Fig. 1b, where the deformed mesh is plotted as well as the initial liver configuration (represented by blue points).

In Table 1, we report the time spent in all subroutines included during the multiples Newton iterations solve with respect to the number of subdomains N . The

N	# d.o.f per sub.	Prec(ms).	Solves(ms).	Total(ms).	# Newton iterations	# Iterations per Newton it	Speedup	# of d.o.f.
1	1998	0.00	47.96	47.96	3	1	-	
2	1263	16.50	11.05	27.55	3	8	1.7	$2.00 \cdot 10^3$
4	897	12.88	7.89	20.77	3	9	2.3	
1	12321	0.00	421.97	421.97	4	1	-	
2	7056	142.68	105.86	248.54	4	10	1.7	$12.32 \cdot 10^3$
4	4494	99.00	94.92	193.91	4	13	2.1	

Table 1: Breakdown of the time spent in solver steps for 3D non-linear solver with respect to the number of subdomains, the second column corresponds to the maximum number of unknowns per subdomain, the third column is the time spent in building the DDM preconditioner, and the fourth column corresponds to the time spent in solving the multiple Newton inner linear systems.

case $N = 1$ corresponds direct solver case, where the system is first factorized with and LDL^T than solved. For $N \geq 2$, the system is solved using GMRES with a domain decomposition preconditioner $\mathcal{M}_{\text{GenEO}}^{-1}$. Very few Newton iterations (column number 6) are needed for the solver to converge, independently of the number of subdomains (first column). The scalability of the solving approach is reported in the table using the run of the direct solver (a.k.a $N = 1$) as a reference. For each mesh, we ensure that we are calculating the same solution regardless of the linear solver type (direct or domain decomposition solver). To do that, we make sure that L_2 norm of the final deformation is the same during each scalability test.

We notice that domain decomposition approach is already providing a reasonable speedup with respect to the number of subdomains. Moreover, we observe that using a simple direct solver approach requires to build the factorization again for each inner Newton iteration with the same high cost. Whereas with domain decomposition approach the same preconditioner is reused for all the Newton iterations with no significant impact on the GMRES solver, thanks to the robustness of the DD preconditioner, the number of Krylov iterations remains stable. This suggests that we can take more benefits in a scenario of dynamic deformations, where we need to solve more linear systems successively through both Newton iterations and time integration algorithm.

The scalability is impacted by the number of unknowns per subdomain, which is not scaling linearly with the number subdomains. This fact is first due to the load balancing provided by the graph partitioner and also due to the overlapping regions between subdomains, which seems to be more critical in case of small meshes. We also notice that the increase in the number of unknowns from 2000 to 12321 leads to the increase of the computation time. This happens because the convergence of the iterative solvers is influenced by the condition number of the stiffness matrix, and the condition number will increase with a decreased element size for a given object.

3.2 Dynamic non-linear deformation

The main objective of this paragraph is to show that the cost of the domain decomposition preconditioner is quickly amortized in the scenario of successive solutions of linear systems. Typically in case of stiff elastic deformation where the implicit integration method is more appropriate. To this end, we solve the entire discretized problem (4), where, a Newmark, implicit time integration scheme is used with a time step of 0.01s, in this scenario, the same volume force as the one used in the static case is again uniformly applied to the liver (Fig. 1b) for 0.03s than released to let the system reaches the equilibrium state. We simulate the liver deformation for both discretizations with coarse mesh yielding a system of 2000 unknowns and refined mesh yielding a system of 12321 unknowns. In Fig 2, we show the behavior of computational time spent in solving the successive linear system. The red and blue curves represent, respectively, the coarse and fine mesh discretization. The global domain has been decomposed in 4 subdomains allowing the simulation to run on 4 processors. For each time step, the Newton-Raphson

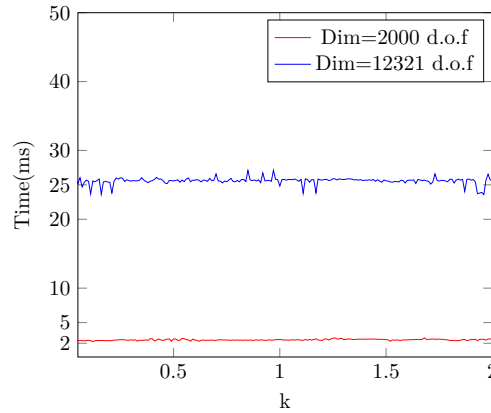


Fig 2: Performances of domain decomposition solver during the simulation of a dynamic non-linear deformation liver response. Using the Saint-Venant Kirchhoff model. Red and blue curves represent the computation time per Newton iteration over time steps with resp to coarse mesh (2000 unknowns) and refined mesh (12321 unknowns).

algorithm is performed to update the deformation state. We use the domain decomposition approach as a solver, where preconditioner $\mathcal{M}_{\text{GenEO}}^{-1}$ is built at the first time step than used for preconditioning all the following Newton inner linear systems

3.3 Contact & deformation

While keeping the context of simulating the behavior of soft tissues, we can point out the importance of providing methods and models to simulate the mechanical

interactions on the organ model: interactions with the surrounding anatomical structures (i.e. contacts), interactions with different surgical instruments leading to contacts or other types of complex interactions. Simulating these interactions necessitate to detect them, to model them, and then to solve them with adapted numerical methods. In practice the numerical system to solve is an augmented version of linear systems that we have solved in previous paragraphs. If we consider the case of a two-body interaction, the augmented system takes the following form

$$AU = b + J^T \lambda \quad (7)$$

which can be formulated as

$$\begin{bmatrix} A_1 & 0 & J_1^T \\ 0 & A_2 & J_1^T \\ J_1 & J_2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ -\delta \end{bmatrix} \quad (8)$$

A_1 and A_2 correspond to the linear operators of each body J_1 , J_1 model the interaction, then the unknowns are the \mathbf{y}_1 , \mathbf{y}_2 are respectively the displacement of the two bodies and λ is the vector of contact forces. We believe that in such case if interaction, domain decomposition approach has a tremendous potential for solving problem (7). And more than accelerating the linear solves, domain decomposition can be specifically designed to consider the interaction area as a single sub-domain domain. This possibility would allow a partial updating of the global operator, and is likely to lead to a substantial gain in simulation time.

4 Conclusion

In this work, we have investigated the computational expense of solving linear systems resulting from a combination of non-linear model and dynamic integration. We showed that employing hybrid solver like a domain decomposition method has a real potential to harness the capability of small parallel machines since it takes full advantage in making the solving procedure fully parallel. On the other hand, the robustness of domain decomposition preconditioners makes it possible to reuse the preconditioner for successive solves. These two strategies combined open up the possibility to significantly accelerate the computation for complex simulation and meet the real-time feedback, which is a hard constraint in surgical training or intra-operative guidance.

The next step will be to integrate the domain decomposition paradigm with fast hyperelastic FEM models and implicit contact schemes. We will also investigate further the limited scalability of the current approach when dealing with real-time applications, which is likely due to load balancing. This can be improved by a better tuning of the graph partitioner .

Bibliography

- [1] Bathe, K.J.: Finite element procedures. Klaus-Jurgen Bathe (2006)
- [2] Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing* **21**, 239–247 (1999)
- [3] Cotin, S., Delingette, H., Ayache, N.: Real-time elastic deformations of soft tissues for surgery simulation. *IEEE transactions on Visualization and Computer Graphics* **5**(1), 62–73 (1999)
- [4] Dolean, V., Jolivet, P., Nataf, F.: An introduction to domain decomposition methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2015), algorithms, theory, and parallel implementation
- [5] Geuzaine, C., Remacle, J.F.: Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering* **79**(11), 1309–1331 (2009)
- [6] Haferssas, R., Jolivet, P., Nataf, F.: An additive schwarz method type theory for lions’s algorithm and a symmetrized optimized restricted additive schwarz method. *SIAM Journal on Scientific Computing* **39**(4), A1345–A1365 (2017)
- [7] Hecht, F.: New development in FreeFem++. *J. Numer. Math.* **20**(3-4), 251–265 (2012)
- [8] Henson, V.E., Yang, U.M.: *BoomerAMG*: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics* **41**(1), 155–177 (2002)
- [9] Joldes, G.R., Wittek, A., Miller, K.: Suite of finite element algorithms for accurate computation of soft tissue deformation for surgical simulation. *Medical Image Analysis* **13**(6), 912–919 (2009)
- [10] Jolivet, P., Hecht, F., Nataf, F., Prud’homme, C.: Scalable domain decomposition preconditioners for heterogeneous elliptic problems. In: *Proceedings of the 2013 ACM/IEEE conference on Supercomputing*. pp. 80:1–80:11. SC13, ACM (2013)
- [11] Karypis, G., Kumar, V.: METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Tech. rep., Department of Computer Science, University of Minnesota (1998)
- [12] Klawonn, A., Lanser, M., Rheinbach, O.: Nonlinear feti-dp and bddc methods. *SIAM Journal on Scientific Computing* **36**(2), A737–A765 (2014)
- [13] Miller, K., Joldes, G., Lance, D., Wittek, A.: Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in numerical methods in engineering* **23**(2), 121–134 (2007)
- [14] Picinbono, G., Delingette, H., Ayache, N.: Real-time large displacement elasticity for surgery simulation: Non-linear tensor-mass model. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 643–652. Springer (2000)

- [15] Saad, Y.: Iterative Methods for Sparse Linear Systems. PWS Publishing Company (1996)
- [16] Schenk, O., Gärtner, K., Fichtner, W.: Efficient sparse LU factorization with left-right looking strategy on shared memory multiprocessors. BIT Numerical Mathematics **40**(1), 158–176 (2000)
- [17] Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. Numer. Math. **126**(4), 741–770 (2014)
- [18] Sundar, H., Biros, G., Burstedde, C., Rudi, J., Ghattas, O., Stadler, G.: Parallel geometric-algebraic multigrid on unstructured forests of octrees. In: Proceedings of the 2012 ACM/IEEE conference on Supercomputing. SC12, IEEE Computer Society (2012)
- [19] Toselli, A., Widlund, O.: Domain Decomposition Methods - Algorithms and Theory, Springer Series in Computational Mathematics, vol. 34. Springer (2005)