Archive of Numerical Software
vol. 1, no. S1, 2011, pages 1–28
DOI: ???

The LifeV library: engineering mathematics beyond the proof of concept

Luca Bertagna[1], Simone Deparis*[2], Luca Formaggia[3], Davide Forti[2], and Alessandro Veneziani[1]

[1]Department of Mathematics and Computer Science, Emory University, Atlanta (GA) 30322 USA
[2]CMCS–MATHICSE–SB, École Polytechnique Fédérale de Lausanne, Station 8, Lausanne, CH–1015, Switzerland
[3]MOX, Dipartimento di Matematica, Politecnico di Milano, Italy

Received: Jan 1, 1677; **final revision:** Jun 1, 1677; **published:** Jun 13, 1677.

Abstract:

LifeV is a library for the finite element (FE) solution of partial differential equations in one, two, and three dimensions. It is written in C++ and designed to run on diverse parallel architectures, including cloud and high performance computing facilities. In spite of its academic research nature, meaning a library for the development and testing of new methods, one distinguishing feature of LifeV is its use on real world problems and it is intended to provide a tool for many engineering applications. It has been actually used in computational hemodynamics, including cardiac mechanics and fluid-structure interaction problems, in porous media, ice sheets dynamics for both forward and inverse problems. In this paper we give a short overview of the features of LifeV and its coding paradigms on simple problems. The main focus is on the parallel environment which is mainly driven by domain decomposition methods and based on external libraries such as MPI, the Trilinos project, HDF5 and ParMetis.

Dedicated to the memory of Fausto Saleri.

1 Introduction

LifeV[1] is a parallel library written in C++ for the approximation of Partial Differential Equations (PDEs) by the finite element method in one, two and three dimensions. The project started in 1999/2000 as a collaboration between the modeling and scientific computing group (CMCS) at EPFL Lausanne and the MOX laboratory at Politecnico di Milano. Later the REO and ESTIME groups at INRIA joined the project. In 2006 the library has been progressively parallelized using MPI with the Trilinos library suite as back-end interface. In 2008 the Scientific Computing group

*Corresponding author, simone.deparis@epfl.ch

[1]Pronounced “life five”, the name stands for Library for Finite Elements, 5th edition as V is the Roman notation for the number 5

at Emory University joined the LifeV consortium. Since then, the number of active developers has fluctuated between 20 and 40 people, mainly PhD students and researchers from the laboratories within the LifeV consortium.

LifeV is open source and currently distributed under the LGPL license on github[2], and migration to BSD License is currently under consideration. The developers page is hosted by a Redmine system at `http://www.lifev.org`. LifeV has two specific aims: (i) it provides tools for developing and testing novel numerical methods for single and multi-physics problems, and (ii) it provides a platform for simulations of engineering and, more generally, real world problems. In addition to "basic" Finite Elements tools, LifeV also provides data structures and algorithms tailored for specific applications in a variety of fields, including fluid and structure dynamics, heat transfer, and transport in porous media, to mention a few. It has already been used in medical and industrial contexts, particularly for cardiovascular simulations, including fluid mechanics, geometrical multiscale modeling of the vascular system, cardiac electro-mechanics and its coupling with the blood flow. When in 2006 we decided to introduce parallelism, the choice has turned towards available open-source tools: MPI (mpich or openmpi implementations), ParMETIS, and the Epetra, AztecOO, IFPACK, ML, Belos, and Zoltan packages distributed within Trilinos Heroux et al. [2005].

LifeV has benefited from the contribution of many PhD students, almost all of them working on project financed by public funds. We provide a list of supporting agencies hereafter.

In this review article we explain the parallel design of the library and provide two examples of how to solve PDEs using LifeV. Section 2 is devoted to a description of how parallelism is handled in the library while in Section 3 we discuss the distinguishing features and coding paradigms of the library. In Section 4 we illustrate how to use LifeV to approximate PDEs by the finite element method, using a simple Poisson problem as an example. In Section 5 we show how to approximate unsteady Navier–Stokes equations and provide convergence, scalability, and timings. We conclude by pointing to some applications of LifeV.

Before we detail technical issues, let us briefly address the natural question when approaching this software, namely *yet another finite element library*?

No question that the research and the commercial arenas offer a huge variety of finite element libraries (or, in general, numerical solvers for partial differential equations) to meet diverse expectations in different fields of engineering sciences. LifeV is - strictly speaking - no exception.

Since the beginning, LifeV was intended to address two needs: (i) *a permanent playground for new methodologies in computational mechanics*; (ii) *a translational tool to shorten the time-to-market of new successful methodologies to real engineering problems*. Also, over the years, we organized portions of the library to be used for teaching purposes. It was used as a sort of *gray box* tool for instance in Continuing Education initiatives at the Politecnico di Milano, or in undergraduate courses at Emory University (MATH352: Partial Differential Equations in Action) - see Formaggia et al. [2012].

As such, LifeV incorporated since the beginning the most advanced methodological developments on topics of interest for the different groups involved. In fact, state-of-the-art methodologies have been rapidly implemented, particularly in incompressible computational fluid dynamics, to be tested on problems of real interest, so to quickly assess the real performances of new ideas and their practical impact. On the other hand, advanced implementation paradigms and efficient parallelization were prioritized, as we will describe in the paper.

When a code is developed with a strong research orientation, the working force is mainly provided by young and junior scholars on specific projects. This has required a huge coordination effort, in each group and overall. Stratification of different ideas evolving has sometimes made the crystallization of portions of the code quite troublesome, also for the diverse background

[2]`https://github.com/lifev`, install instructions at `https://bitbucket.org/lifev-dev/lifev-release/wiki/lifev-ubuntu`

of the developers. Notwithstanding this, the stimulus of the applications has promoted the development of truly advanced methods for the solution of specific problems. In particular, the vast majority of the stimuli were provided by computational hemodynamics, as all the groups involved worked in this field with a strict connection to medical and healthcare institutions. The result is a library extremely advanced regarding performances and with a sort of unique deep treatment of a specific class of problems. Beyond the overall approach to the numerical solution of the incompressible Navier-Stokes equations (with either monolithic or algebraic partitioning schemes), fluid-structure interaction problems, patient-specific image based modeling, defective boundary conditions and geometrical multiscale modeling have been implemented and tested in LifeV in an extremely competitive and unique way. The validation and benchmarking on real applications, as witnessed by several publications out of the traditional field of computational mechanics, makes it a reliable and efficient tool for modern engineering. So, it is another finite element library yet with peculiarities that make it a significant - somehow unique - example of modern scientific computing tools

1.1 Financial support

LifeV has been supported by the 6th European Framework Programme (Haemodel Project, 2000-2005, PI: A. Quarteroni), the 7th European Framework Programme (VPH2 Project, 2008-2011, ERC Advanced Grant MATHCARD 2009-2014, PI: A. Quarteroni), the Italian MIUR (PRIN2007, PRIN2009 and PRIN12 projects, PI: A. Quarteroni and L. Formaggia), the Swiss National Science Foundation (several projects, from 1999 to 2017, 59230, 61862, 65110, 100637, 117587, 125444, 122136, 122166, 141034, PI: A. Quarteroni), the Swiss supercomputing initiatives HP2C and PASC (PI: A. Quarteroni), the Fondazione Politecnico di Milano with Siemens Italy (Aneurisk Project, 2005-2008, PI: A. Veneziani), the Brain Aneurysm Foundation (Excellence in Brain Aneurysm Research, 2010, PI: A. Veneziani), Emory University Research Committee Projects (2007, 2010, 2015, PI: A. Veneziani), ABBOTT Resorb Project (2014-2017, PI: H. Samady, D. Giddens, A. Veneziani), the National Science Foundation (NSF DMS 1419060, 2014-2017, PI: A. Veneziani, NSF DMS 1412963, 2014-2017, PI: A. Veneziani, NSF DMS 1620406, 2016-2018, PI: A. Veneziani), iCardioCloud Project (2013-2016, PI: F. Auricchio, A. Reali, A. Veneziani), the Lead beneficiary programm (Swiss SNF and German DFG, 140184, 2012-2015, PI: A. Klawonn, A. Quarteroni, J. Schröder), and the French National Research Agency. Some private companies also collaborated to both the support and the development, in particular MOXOFF SpA (2012-2104) exploited the geometric multiscale paradigm for the simulation of an industrial packaging system and and Eni SpA (2011-2014) contributed to the development of the Darcy solver and extended FE capabilities, as well as the use of the library for the simulation the evolution of sedimentary basins Cervone et al. [2012].

1.2 Main Contributors

The initial core of developers was the group of A. Quarteroni at MOX, Politecnico di Milano, Italy and at the Department of Mathematics, EPFL, Lausanne, Switzerland from an initiative of L. Formaggia, J.F. Gerbeau, F. Saleri and A. Veneziani. The group of J.F. Gerbau at the INRIA, Rocquencourt, France gave significant contributions from 2000 through 2009 (in particular with M. Fernandez and, later, M. Kern). The important contribution of C. Prud'homme and G. Fourestey during their stays at EPFL and of D. Di Pietro at University of Bergamo are acknowledged too. S. Deparis has been the coordinator of the LifeV consortium since 2007.

Here, we limit to summarize the list of main contributors who actively developed the library in the last five years. We group the names by affiliation. As some of the authors moved over the years to different institutions, they may be listed with multiple affiliations hereafter.

[D. Baroli, A. Cervone, M. Del Pra, N. Fadel, E. Faggiano, L. Formaggia, A. Fumagalli, G. Iori, R.M. Lancellotti, A. Melani, S. Palamara, S. Pezzuto, S. Zonca][3]. [L. Barbarotta, C. Colciago, P.

[3]MOX, Politecnico di Milano IT

Crosetto, S. Deparis, D. Forti, G. Fourestey, G. Grandperrin, T. Lassila, C. Malossi, R. Popescu, S. Quinodoz, S. Rossi, R. Ruiz Baier, P. Tricerri][4], M. Kern[5], [S. Guzzetti, L. Mirabella, T. Passerini, A. Veneziani][3,6], U. Villa[6], L. Bertagna[6, 7,8], M. Perego[3,6,7,8], A. Quaini[9,6], H. Yang[6,7], A. Lefieux[10,6].

Abbreviations : Algebraic Additive Schwarz (AAS), FE (Finite Elements), FEM (Finite Element Method), DoF (Degree(s) of Freedom), OO (Object Oriented), PCG (Preconditioned Conjugate Gradient), PGMRes (Preconditioned Generalized Minimal Residual), ML (Multi Level), DD (Domain Decomposition), MPI (Message Passing Interface), ET (Expression Template), HDF (Hierarchical Data Format), HPC (High Perfromance Computing), CSR (Compact Sparse Row)

2 The Parallel Framework

The library can be used for the approximation of PDEs in one dimension, two dimensions, and three dimensions. Although it can be used in serial mode (i.e., with one processor), parallelism is crucial when solving three dimensional problems. To better underline the ability of LifeV to tackle large problems, in this review we focus on PDEs discretized on unstructured linear tetrahedral meshes, although we point out that LifeV also supports hexahedral meshes as well as quadratic meshes.

Parallelism in LifeV is achieved by domain decomposition (DD) strategies, although it is not mandatory to use DD preconditioners for the solution of sparse linear systems. In a typical simulation, the main steps involved in the parallel solution of the finite element problem using LifeV are the following:

1. All the MPI processes load the same (not partitioned) mesh.
2. The mesh is partitioned in parallel using ParMETIS or Zoltan. At the end each process keeps only its own local partition.
3. The DoFs are distributed according to the mesh partitions. By looping on the local partition, a list of local DoF in global numbering is built.
4. The FE matrices and vectors are distributed according to the DoFs list. In particular, the matrices are stored in row format, for which whole rows are assigned to the process owning the associated DoF.
5. Each process assembles its local contribution to the matrices and vectors. Successively, global communication consolidates contributions on shared nodes (at the interface of two subdomains).
6. The linear system is solved using an iterative solver, typically either a Preconditioned Conjugate Gradient (PCG) when possible or a Preconditioned GMRes (PGMRes). The preconditioner runs in parallel. Ideally, the number of preconditioned iterations should be independent of the number of processes used.
7. The solution is downloaded to mass storage in parallel using HDF5 for post-processing purposes (see Sect. 2.4).

The aforementioned steps are explained in detail in the next subsections.

[4]CMCS, EPFL Lausanne, CH
[5]INRIA, Rocquencourt, FR
[6]Dept. Math & CS, Emory Univ, Atlanta GA USA
[7] Department of Scientific Computing, Florida State University, Tallahassee, FL USA
[8]Sandia National Lab, Albuquerque, NM USA
[9]Department of Mathematics, University of Houston, TX USA
[10]Department of Civil Engineering and Structures, University of Pavia, IT

2.1 Mesh partitioning: ParMETIS and Zoltan

As mentioned above, LifeV achieves parallelism by partitioning the mesh among the available processes. Typically, this is done "online": the entire mesh is loaded by all the processes but it is deleted after the partitioning phase, so that each process keeps only the part required for the solution of the local problem and to define inter-process communications. As the mesh size increases, the "online" procedure may become problematic. Therefore for large meshes it is possible, and sometimes necessary, to partition the mesh offline on a workstation with sufficient memory Popescu [2013]. It is also possible to include an halo of ghost elements such that the partitions overlap by one or more layers of elements (see e.g. Guzzetti et al. [2015]). This may be relevant for schemes that require a large stencil. To perform the partition, LifeV can interface with two third party libraries: ParMetis and Zoltan Karypis and Kumar [1998], Devine et al. [2002].

2.2 Distributed arrays: Epetra

The sparse matrix class used in LifeV is a wrapper to the Epetra matrix container Epetra_FECrsMatrix and, similarly, the vector type is a wrapper to Epetra_FEVector, both provided by the Epetra Heroux [2009] package of Trilinos. The distribution of the unknowns is determined automatically by the partitioned mesh: with a loop over each element of the local mesh we create the list of DoF managed by the current processor. This procedure in fact creates a *repeated* map, i.e., an instance of an Epetra_Map with some entries referring to the DoF associated with geometric entities lying on the interface between two (or more) subdomains. Then, a *unique* map is created, in such a way that, among all the owners of a *repeated* DoF, only one will also own it in the *unique* map. The unique map is used for the vectors and matrices to be used in the linear algebra routines as well as for the solution vector. The repeated map is used to access information stored on other processors, which is usually necessary only in the assembly and post-processing phases.

The assembly of the FE matrices is typically performed by looping on the local elements Ern and Guermond [2006], Formaggia et al. [2012]. To reduce latency time, the loops on each subdomain are performed in parallel, without need of any communication during the loop. Just a single communication phase takes place once all processes have assembled their local contribution, to complete the assembly for interface dofs.

Efficiency and stability may be improved by two further available operations, (i) precomputing the matrix graph; (ii) using overlapping meshes. The former demands for the creation at the beginning of the simulation of an Epetra_Graph, associated to the matrix. Since it depends on the problem at hand and the chosen finite elements, its computation needs a loop on all the elements. This is coded by *Expression Templates* (ET), see Sect. 3.2, using the same call sequence as for the matrix assembly. The latter further reduces communications by allowing all processes to compute the local finite element matrix also on all elements sharing a DoF on the interface. As a result, each process can independently compute all the entries of matrices and vectors pertaining to the DoF it owns at the price of some extra computation. Yet, the little overhead is justified by the complete elimination of the post-assembly communication costs.

2.3 Parallel preconditioners

The solution of linear systems in LifeV relies on the Trilinos Heroux et al. [2005] packages AztecOO and Belos Bavier et al. [2012], which provide an extensive choice of iterative or direct solvers. LifeV provides a common interface to both of them.

The proper use of ParMETIS and Zoltan for the partitioning, and of Epetra matrices and vectors for the linear algebra, ensures that matrix-vector multiplication and vector operations are properly parallelized, i.e., they scale well with the number of processes used, and communications are optimized. In this situation the parallel scalability of iterative solvers like PCG or PGMRES depends essentially on the properties of the preconditioner.

The choice of preconditioner is thus critical. In our experience it may follow two directions: (i) parallel preconditioners for the generic linear systems, like single or multilevel overlapping Schwarz preconditioners, or multigrid preconditioners, that are generally well suited for highly coercive elliptic problems, or incomplete factorization (ILU), which are generally well suited for advection-dominated elliptic problems; (ii) problem specific preconditioners, typically required for multifield or multiphysics problem. These preconditioners exploits specific features of the problem at hand to recast the solution to standard problems that can be eventually solved with the generic strategies in (i). Preconditioners of this class are, e.g., the SIMPLE, the Least Square Commutator, the Caouet-Chabard and the Yosida preconditioners for the incompressible Navier-Stokes equations Patankar and Spalding [1972], Elman et al. [2014], Cahouet and Chabard [1988], Veneziani [2003] or the Monodomain preconditioner for the Bidomain problem in electrocardiology Gerardo-Giorda et al. [2009].

In LifeV preconditioners for elliptic problems are indeed an interface to the Trilinos package IFPACK Sala and Heroux [2005], which is a suite of algebraic preconditioners based on incomplete factorization, and to ML Gee et al. [2006] or MueLu Prokopenko et al. [2014], which are two Trilinos packages for multi-level preconditioning based on algebraic multigrid. Typically we use IFPACK to define algebraic overlapping Schwarz preconditioners with exact or inexact LU factorization of the restricted matrix. The preconditioner P in this case can be formally written as

$$P^{-1} = \sum_{i=1}^{n} R_i^T A_i^{-1} R_i, \text{ where } A_i = R_i A^{-1} R_i^T, \tag{1}$$

where A is the finite element matrix related to the PDE approximation, n is the number of partitions (or subdomains) Ω_i, R_i is the restriction operator to Ω_i and R_i^T the extension operator from Ω_i to the whole domain Ω. A_i is inverted many times during the iterations of PCG or PGMRES, which is why it is factorised by LU or ILU. In LifeV, the choice of the factorization is left to the user.

Similarly, it is possible to use multilevel preconditioners via the ML Gee et al. [2006] or MueLu Prokopenko et al. [2014], Hu et al. [2014] packages. They work at the algebraic level too and the coarsening and extension are done either automatically, or by user defined strategies, based on the parallel distribution of the matrix and its graph. The current distribution of LifeV does not offer the last option, but the interested developer could add this extra functionality with relatively little effort.

2.4 Parallel I/O with HDF5

When dealing with large meshes and large number of processes, input/output access to files on disk deserves particular care. A prerequisite is that the filesystem of the supercomputing architecture being used provides the necessary access speed in parallel. However this is not enough. MPI itself offers parallel I/O capabilities, for which HDF5 is one of the existing front-ends The HDF Group [1997]. Although other formats are also supported (see section 3.1), LifeV strongly encourages the use HDF5 for I/O processing essentially for three reasons.

- The number of files produced is independent of the number of running processes: each process accesses the same file in parallel and writes out his own chunks of data. As a result, LifeV generally produces one single large output binary file, along with a xmf text file describing its contents.
- HDF5 is compatible with open source post-processing visualization tools like Paraview Ahrens et al. [2005] and VisIt Childs et al. [2012]).
- Having one single binary file makes it very easy to use for restarting a simulation.

The interface to HDF5 in LifeV exploits the facilities of the EpetraExt package in Trilinos, and since the LifeV vectors are compatible with Epetra format the calls are simple Heroux [2009].

3 Features and Paradigms

3.1 I/O data formats

In a typical simulation, the user provides a text file containing input data (including physical and discretization parameters, and options for the linear/nonlinear solvers), and the code will generate results, which need to be stored for post-processing. Although it is up to the user to write the program main file where the data file is parsed, LifeV makes use of two particular classes in order to forward the problem data to all the objects involved in the simulation: the GetPot class Schaefer [2007] (for which LifeV also provides an *ad hoc* re-implementation inside the core module), and the ParameterList class from the Teuchos package in Trilinos. The former has been the preferred way since the early development of LifeV, and is therefore supported by virtually all classes that require a setup. The latter is used mostly for the linear and nonlinear solvers, since it is the standard way to pass configuration parameters to the Trilinos solvers. Both classes map strings representing the names of properties to their actual values (be it a number, a string, or other), and they both allow the user to organize the data in a tree structure. Details on the syntax of the two formats are available online.

When it comes to mesh handling, LifeV has the built-in capability of generating structured meshes on domains of the form $\Omega = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$. If a more general mesh is required, the user needs to create it beforehand. Currently, LifeV supports the FreeFem Hecht [2012] and Gmsh Geuzaine and Remacle [2009] formats (both usually with extension `.msh`) in 2D, while in 3D it supports the formats of Gmsh, Netgen Schöberl [1997] (usually with extension `.vol`) and Medit Frey [2001] (usually with extension `.mesh`). Additionally, as mentioned in Section 2.1, LifeV offers the capability of offline partitioning. The partitions of a mesh are stored and subsequently loaded using HDF5 for fast and parallel input.

Finally, LifeV offers three different formats for storing the simulation results for post-processing: Ensight Computational Engineering International, Inc. [2011], HDF5 The HDF Group [1997], which is the preferred format when running in parallel, and VTK Schöberl et al. [2006]. All these formats are supported by the most common scientific visualization software packages, like Paraview Ahrens et al. [2005] and VisIt Childs et al. [2012]. The details on these formats can be found on their respective webpages.

3.2 Expression Templates for Finite Elements

One of the aims of LifeV is to be used in multiple contexts, ranging from industrial and social applications to teaching purposes. For this reason, it is important to find the best trade-off between computational efficiency and code readability. The high level of abstraction proper of C++ is in principle perfectly matched by the abstraction of mathematics. However, versatility and efficiency may conflict. Quoting Furnish [1997], a "natural union has historically been more of a stormy relationship". This aspect is crucial in High Performance Computing, where efficiency is a priority. Operator overloading has a major impact on efficiency, readability, maintainability and versatility, however it may adversely affect the run time. *Expression Templates* (ET) have been originally devised to minimize this drawback by T. Veldhuizen Veldhuizen [1995], and later further developed in the context of linear algebra Härdtlein et al. [2009], Iglberger et al. [2012] and solution of partial differential equations Pflaum [2001], Di Pietro and Veneziani [2009a].

In the context of linear algebra, the technique was developed to allow high level vector syntax without compromising on code speed, due to function overloading. The goal of ET is to write high level *Expressions*, and use *Template* meta-programming to parse the expression at compile-time, generating highly efficient code. Put it simply, in the context of PDE's, ET aims to allow a syntax of the form

```
auto weak_formulation = alpha*dot(grad(u), grad(v)) + sigma*u*v;
```


which is an *expression* very close to the abstract mathematical formulation of the problem. However, during the assembly phase, the resolution of the overloaded operators and functions would yield a performance hit, compared to a corresponding *ad-hoc* for loop. To overcome this issue, the above *expression* is implemented making massive use of *template* meta-programming, which allows to expand the *expression* at compile time, resulting in highly efficient code. Upon expansion, the *expression* takes the form of a combination of polynomial basis functions and their derivative at a quadrature point. At run time, during the assembly phase, such combination is then evaluated *at once* at a given quadrature point, as opposed to the more *classical* implementation, where all contributions are evaluated separately and then summed up together.

For instance, for the classical linear advection-diffusion-reaction equation in the unknown u, $-\mu\Delta u+\beta\cdot\nabla u+\sigma u$, we need to combine three different differential operators weighted by coefficients μ, β and σ. These may be numbers, prescribed functions or pointwise functions (inherited for instance by another FE computation). In nonlinear problems — after proper linearization — expressions may involve finite element functions too. Breaking down the assembly part to each differential operator individually with its own coefficient is possible but leads to duplicating the loops over the quadrature nodes, as opposed to the assembly of their sum. In LifeV the possible differential operators for the construction of linear and nonlinear advection-diffusion-reaction problems are enucleated into specific *Expression*s, following the idea originally proposed in Di Pietro and Veneziani [2009b]. The ET technique provides readable code with no efficiency loss for operator parsing. As a matter of fact, the final gathering of all the assembly operations in a single loop, as opposed to standard approaches with a separate assembly for each elemental operator, introduces computational advantages. Indeed, numerical tests have pointed out a significantly improved performance for probelms with non-constant coefficients when using the ET technique.

A detailed description of ET definition and implementation in LifeV can be found in Quinodoz [2012] and in the code snapshots presented later on.

4 Basics: Life = Library of Finite Elements

The library supports different type of finite elements. The use of ET makes the set-up of simple problems easy, as we illustrate hereafter.

4.1 The Poisson problem

As a first example, we present the setup of a finite element solver for a Poisson problem. We assume a polygonal Ω with boundary $\partial\Omega$ split into two subsets Γ^D and Γ^N of positive measure such that $\Gamma^D \cup \Gamma^N = \partial\Omega$ and $\Gamma^D \cap \Gamma^N = \emptyset$. Let $V_h^D \subset H^1_{\Gamma^D}$ be a discrete finite element space relative to a mesh of Ω, for example continuous piecewise linear functions vanishing on Γ_D. The Galerkin formulation of the problem reads: find $u_h \in V_h^D$ such that

$$\int_\Omega \kappa \nabla u_h : \nabla \varphi_h = -\int_\Omega \kappa \nabla u^D : \nabla \varphi_h + \int_\Omega f \varphi_h + \int_{\Gamma^N} g^N \varphi_h \qquad \forall \varphi_h \in V_h^D, \tag{2}$$

where κ is the diffusion coefficient, possibly dependent on the space coordinate, g^N is the Neumann boundary condition, $\frac{\partial u}{\partial n} = g^N$ on Γ^N, and f are the volumetric forces. The lifting u_D can be any finite element function such that $u^D|_{\Gamma^D}$ is a suitable approximation of g^D. As usually done, u^D is such that $u^D|_{\Gamma^D}$ is the Lagrange interpolation of g^D, extended to zero inside Ω.

In LifeV, DoF associated with Dirichlet boundary conditions are not physically eliminated from the FE unknown vectors and matrices. Even though this elimination would certainly affect positively the performances of the linear algebra solver, it introduces a practical burden in the implementation and memory particularly in 3D unstructured problems, that makes it less appealing. The enforcement of these conditions can be done alternatively in different ways as illustrated e.g. in Formaggia et al. [2012], after the matrix assembly. We illustrate the strategy adopted in the following sections.

4.2 Matrix form and expression templates

Firstly, we introduce the matrix assembled for homogeneous Natural conditions associated with the differential operator at hand (aka "do-nothing" boundary conditions, as they do not require any extra work to the pure discretization of the differential operator), i.e.

$$A = (a_{ij})_{i,j=1,\ldots,n} \text{ and } a_{ij} = \int_\Omega \kappa \nabla \varphi_j : \nabla \varphi_i, i, j = 1, \ldots, n,$$

where φ_j, $j = 1, \ldots, n$ are the basis functions of the finite element space V_h. Thanks to the ET framework Di Pietro and Veneziani [2009b], Quinodoz [2012], once the mesh, the solution FE space and the quadrature rule have been created, the assembly of the stiffness matrix A is as simple as the following instruction

```
integrate ( elements (localMeshPtr), quadratureRule, uFESpace, uFESpace,
            value(kappa) * dot ( grad ( phi_i ) , grad ( phi_j ) )     )  >> systemMatrixPtr;
```

We emphasize how the ET syntax clearly highlights the differential operator being assembled, making the code easy to read and maintain. In a similar way, we define the right hand side of the linear problem as the vector

$$\mathbf{b} = (b_i)_{i=1,\ldots,n} \text{ where } b_i = \int_\Omega f \varphi_i + \int_{\Gamma^N} g^N \varphi_i, i = 1, \ldots, n.$$

In this case, possible non-homogeneous Neumann or natural conditions are included. Finally, the DoF related to the Dirichlet boundary conditions are enforced by setting the associated rows of A equal to zero except for the diagonal entries. In this way, the equation associated with the i-th Dirichlet DoF is replaced by $cu_i = cg_i$, where c is a scaling factor, depending in general on the mesh size, to be used to control the condition number of the matrix. A general strategy is to pick up values of the same order of magnitude of the entries of the row of the do-nothing matrix being modified.

Without further modification, the system matrix is not symmetric anymore. Many of the problem faced in application are not symmetric, therefore we describe here only how to deal with non-symmetric matrices.

It is worth noting that the symmetry break does not prevent using specific methods for symmetric systems like CG when appropriate as pointed out in Ern and Guermond [2006]. A symmetrization of the matrix can be also achieved by enforcing the condition $cu_i = cg_i$ column-wise, i.e. by setting to 0 also the off-diagonal entries of the columns of Dirichlet DoF. Some sparse-matrix formats oriented to row-wise access of the matrix, like the popular CSR, need in this case to be equipped with specific storage information that could make the column-wise access convenient Formaggia et al. [2012].

4.3 Linear algebra

The linear system $A\mathbf{x} = \mathbf{b}$ can be solved by a preconditioned iterative method like PCG, PGMres, BiCGStab, etc, available in the packages AztecOO or Belos of Trilinos. The following snippet highlights the simplicity of the usage of LifeV's linear solver interface.

```
// Load the linear solver parameters
Teuchos::RCP<Teuchos::ParameterList> params = Teuchos::getParametersFromXmlFile("params.xml");
// Create and setup the preconditioner
std::shared_ptr<Preconditioner> precPtr (new PreconditionerIfpack);
precPtr->setParametersList (params->sublist("Preconditioner").sublist("Ifpack"));
// Create and setup the linear solver
LinearSolver linearSolver (Comm);          // Comm is the communicator
linearSolver.setOperator (systemMatrix);   // Set the matrix of the system
linearSolver.setParameters (*params);      // Set the linear solver parameters
linearSolver.setPreconditioner (precPtr);  // Set the preconditioner
linearSolver.setRightHandSide (rhs);       // Set the right hand side
// Solve the linear system, and store the solution in the given vector
linearSolver.solve (solution);
```

The choice of the method and its settings are to be set via an input xml file. For the above example (which used AztecOO as solver and Ifpack as preconditioner), a minimal input file `params.xml` would have the following form

```
<ParameterList>
    <Parameter name="Solver Type"  type="string"  value="AztecOO"/>
    <Parameter name="Prec Type"    type="string"  value="Ifpack"/>
    <ParameterList name="Solver: Operator List">
        <ParameterList name="Trilinos: AztecOO List">
            <Parameter name="solver"    type="string"  value="gmres"/>
            <Parameter name="conv"      type="string"  value="rhs"/>
            <Parameter name="tol"       type="double"  value="1.e-9"/>
            <Parameter name="max_iter"  type="int"     value="50"/>
            <Parameter name="kspace"    type="int"     value="20"/>
        </ParameterList>
    </ParameterList>
    <ParameterList name="Preconditioner">
        <ParameterList name="Ifpack">
            <Parameter name="prectype"               type="string"  value="ILU"/>
            <Parameter name="overlap level"          type="int"     value="2"/>
            <Parameter name="fact: level-of-fill"    type="int"     value="2"/>
        </ParameterList>
    </ParameterList>
</ParameterList>
```

The main difficulty is to set up a scalable preconditioner. As pointed out , in LifeV there are several options based on Algebraic Additive Schwarz (AAS) or Multigrid preconditioners. In the first case, the local problem related to A_i in (1) has to be solved. It is possible to use an LU factorization, using the interface with Amesos Sala et al. [2006a,b] or incomplete factorizations (ILU). LU factorizations are more robust than incomplete ones in the sense that they do not need any parameter tuning, which is delicate in particular within AAS, while incomplete ones are much faster and require less storage. In a parallel context though, for a given problem, the size of the local problem is inversely proportional to the number of subdomains. The LU factorization, whose cost depends only on the number of unknowns, is perfectly scalable, but more memory demanding. An example of the scalability in this settings is given in Figure 1. LifeV leaves the choice to the user depending on the type of problem and computer architecture at hand.

The previous example is just an immediate demonstration of LifeV coding. For other examples we refer the reader to Formaggia et al. [2012].

5 The CFD Portfolio

A core application developed since the beginning, consistently with the tradition of the group where the library has been originally conceived, is incompressible fluid dynamics, which is particularly relevant for hemodynamics. It is well-known that the problem has a *saddle point* nature that stems from the incompressiblity constraint. From the mathematical stand point, this introduces specific challenges, for instance the choice of finite element spaces for velocity and pressure that should satisfy the so called *inf-sup condition*, unless special stabilization techniques are used Elman et al. [2014]. LifeV offers both possibilities. In fact, one can choose among inf-sup stable P2-P1 finite element pairs or equal order P1-P1 or P2-P2 stabilized formulations, either

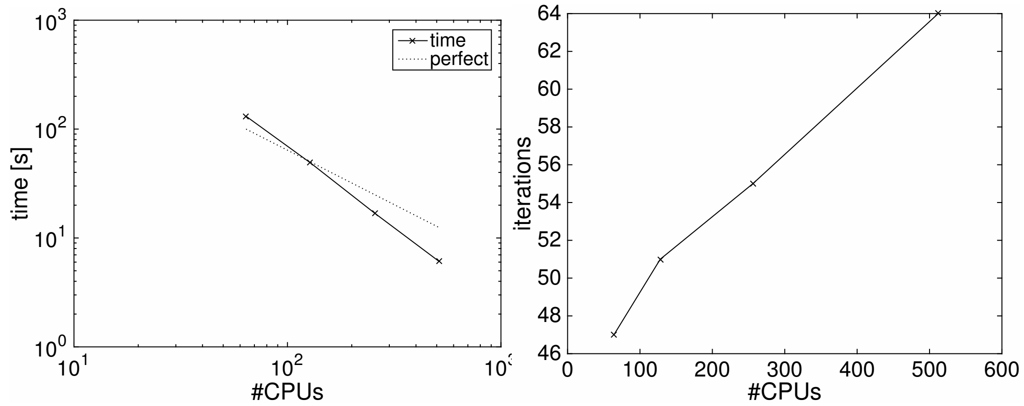

Figure 1: Solving a Poisson problem in a cube with P2 finite elements with 1'367'631 degrees of freedom. The scalability in terms of CPU time (left) is perfect, however the number of iterations (right) linearly increases. The choice of the preconditioner is not optimal, the use of a coarse level or of multigrid in the preconditioner is essential and allows to use more processes with no loss of resources, cf. also Figures 2 and 3.

by interior penalty Burman and Fernández [2007] or SUPG-VMS Bazilevs et al. [2007], Forti and Dedè [2015].

As it is well known, for high Reynolds flow it is important to be able to describe turbulence by modeling it, being impossible to resolve it in practice. Being hemodynamics the main LifeV application, where turbulence is normally less relevant, LifeV has not implemented a full set of turbulence models. However, it includes the possibility of using the Large Eddie Simulation (LES) approach, which relies on the introduction of a suitable filter of the convective field in the Navier-Stokes equations with the role of bringing the unresolved scales of turbulence to the mesh scale. The Van Cittert deconvolution operator considered in Bowers and Rebholz [2012] has been recently introduced in LifeV in Bertagna et al. [2016]. Validation up to a Reynolds number 6500 has been validated with the FDA Critical Path Initiative Test Case.

Another LES procedure based on the variational splitting of resolved and unresolved parts of the solution has been considered in Forti and Dedè [2015], while other LES filtering techniques, in particular the σ-model Nicoud et al. [2011], have been implemented in Lancellotti [2015].

5.1 Preconditioners for Stokes problem

In Section 2.3 we have introduced generic parallel preconditioners based on AAS or multigrid. These algorithms have been originally devised for elliptic problems. In our experience, their use for saddle-point problems like Darcy, Stokes and Navier–Stokes equations, or in fluid structure interaction problems, is not effective. However, their combination with specific preconditioners, like e.g. SIMPLE, Least Square Commutator, Yosida for unsteady Navier–Stokes equations or Dirichlet-Neumann for FSI, leads to efficient and scalable solvers.

In Deparis et al. [2014b] this approach has been applied with success to unsteady Navier-Stokes problems with inf-sup stable finite elements, and then extended to a VMS-LES stabilized formulation with equal order elements for velocity and pressure Forti and Dedè [2015], see also Figure 2.

Applying the same techniques to build a parallel preconditioner for FSI based on a Dirichlet–Neumann splitting Crosetto et al. [2011a] and a SIMPLE Patankar and Spalding [1972], Elman et al. [2008] preconditioner does not lead to a scalable algorithm. To this end, it is necessary to add additional algebraic operations which leads to a FaCSI preconditioner Deparis et al. [2016], see also Figure 3.

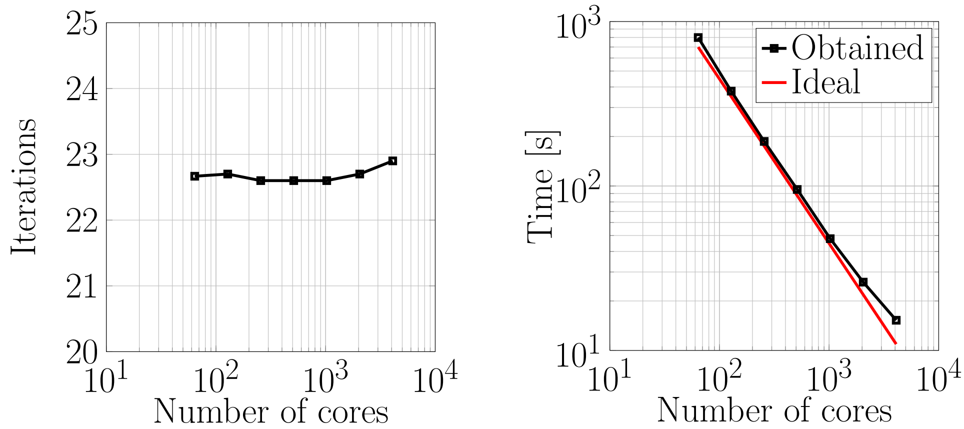

Figure 2: Flow around a cylinder, Reynolds number equal to 22'000. Scalability of hybrid preconditioners for stabilized Navier–Stokes equations Forti and Dedè [2015]. Simulations on PizDora at CSCS. VMS-LES stabilized P2-P2 finite elements, 9'209'040 degrees of freedom and time step 0.0025 s. On the left the number of iterations and on the right the time to solve an entire time step.

Mesh	Fluid DoF	Structure DoF	Coupling DoF	Geometry DoF	Total
Coarse	9'029'128	2'376'720	338'295	8'674'950	20'419'093
Fine	71'480'725	9'481'350	1'352'020	68'711'934	151'026'029

Table 1: Femoropopliteal bypass test case: number of Degrees of Freedom (DoF).

5.2 Algebraic Factorizations

Another issue is to reduce the computational cost by separating pressure and velocity computations. The origin of splitting schemes for Navier-Stokes problems may be dated back to the separate work of A. Chorin and R. Temam that give rise to the well-known scheme that bears their name. The basic scheme operates at a differential level by exploiting the Helmholtz decomposition Theorem (also known as Ladhyzhenskaja Theorem) to separate the differential problem into the sequence of a vector advection-diffusion-reaction problem and a Poisson equation, with a final correction step for the velocity. As opposed to the "split-then-discretized" paradigm, B. Perot in Perot [1993] advocates a "discretize then split" strategy, by pointing out the formal analogy between the Chorin-Temam scheme and an inexact LU factorization of the matrix obtained after discretization of the Navier-Stokes equations. This latter approach, often called "algebraic factorization" is easier to implement, particularly when one has to treat general boundary conditions Quarteroni et al. [2000].

Let us introduce briefly a general framework. Let A be the matrix obtained by the finite element discretization of the (linearized) incompressible Navier-Stokes equations. The discretized problem at each step reads

$$A\begin{bmatrix}\boldsymbol{u}\\ \boldsymbol{p}\end{bmatrix} = \begin{bmatrix}\boldsymbol{f}\\ 0\end{bmatrix} \qquad \text{with} \quad A = \begin{bmatrix}C & D^T\\ D & 0\end{bmatrix}$$

where A collects the contribution of the linearized differential operator acting on the velocity field in the momentum equation, D and D^T are the discretization of the divergence and the gradient operators, respectively. Notice that

$$A = LU = \begin{bmatrix}C & 0\\ D & -DC^{-1}D^T\end{bmatrix}\begin{bmatrix}I & C^{-1}D^T\\ 0 & I\end{bmatrix}.$$

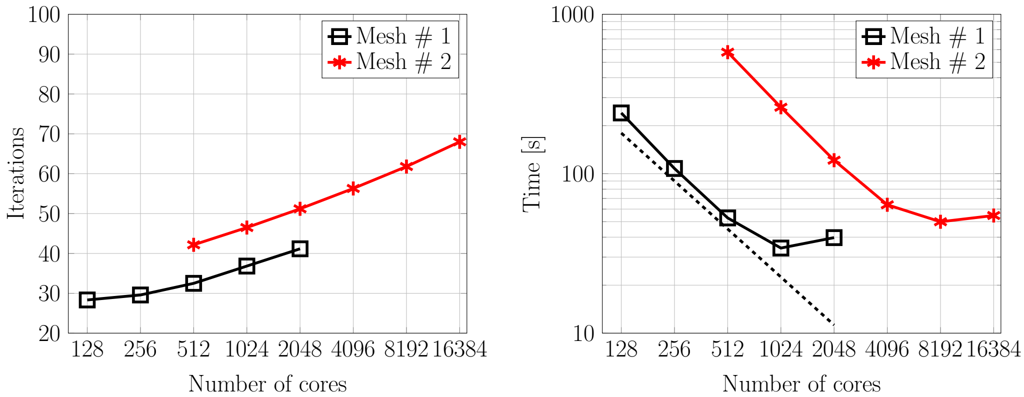

Figure 3: Blood flow in a patient-specific arterial bypass. Scalability of FaCSI preconditioner for FSI simulations Deparis et al. [2016]. Simulations on PizDora at CSCS. Finite elements used: fluid P2-P1, structure P2, ALE mapping P2. See Table 1 for the number of degrees of freedom.

This "exact" LU factorization of the problem formally realizes a velocity-pressure splitting. However there is no computational advantage because of the presence of the matrix C^{-1}, which is not explicitly available, so any matrix-vector product with this matrix requires to solve a linear system. The basic idea of algebraic splitting is to approximate this factorization. A first possibility is to replace C^{-1} with the inverse of the velocity mass matrix scaled by Δt. This is the result of the first term truncation of the Neumann expansion that may be used to represent C^{-1}. The advantage of this approximation is that the mass matrix can be further (and harmless) approximated by a diagonal matrix by the popular "mass-lumping" step. In this way, $DC^{-1}D^T$ is approximated by a s.p.d matrix — sometimes called "discrete Laplacian" for its spectral analogy with the Laplace operator — that can be tackled with many different convenient numerical strategies. In addition, it is possible to see that the splitting error gathers in the first block row, i.e. in the momentum equation. We finally note that replacing the original C^{-1} with the velocity mass matrix in the U factor of the splitting implies that the exact boundary conditions cannot be enforced exactly. The Yosida strategy, on the other hand, follows a similar pathway, except for not approximating C^{-1} in U. Similar properties can be proved as for the Perot scheme, however in this case the splitting error affects only the mass conservation (with a moderate mass loss depending on the time step) and the final step does actually enforce the exact boundary conditions for the velocity. Successively, different splittings have been proposed in Gauthier et al. [2004], Saleri and Veneziani [2005], Gervasio et al. [2006], Gervasio and Saleri [2006], Gervasio [2008], Veneziani [2009] to reduce the impact of the splitting error by successive corrections of the pressure field. In particular, in Veneziani [2003], Gauthier et al. [2004] the role of inexact factorizations as preconditioners for the original problem was investigated.

LifeV incorporates these last developments. In particular, the Yosida scheme has been preferred since the error on the mass conservation has less impact on the interface with the structure in fluid-structure interaction problems. It is worth noting that a special block operator structure reflecting the algebraic factorization concept has been implemented in Villa [2011].

It is worth noting that a robust validation of these methods has been successfully performed not only against classical analytical test cases but also within the framework of one *Critical Path Initiative* promoted by the US Food and Drug Administration (FDA) Passerini et al. [2013], (https://fdacfd.nci.nih.gov). Also, extensions of the inexact algebraic factorization approach to the steady problem have been recently proposed in Viguerie and Veneziani [2017].

5.2.0.1 Time adaptivity An interesting follow up of the pressure corrected Yosida algebraic factorizations is presented in Veneziani and Villa [2013]. This work stems from the fact that the sequence of pressure corrections not only provides an enhancement of the overall splitting error, but also provides an error estimator in time for the pressure field - with no additional computational cost. Based on this idea, a sophisticated time adaptive solver has been introduced Villa [2011], Veneziani and Villa [2013], with the aim of cutting the computational costs by a smart and automatic selection of the time step. The latter must be the trade-off among the desired accuracy, the computational efficiency and the numerical stability constraints introduced by the splitting itself. The final result is a solver that automatically detects the optimal time step, possibly performing an appropriate number of pressure correction to attain stability.

This approach is particularly advantageous for computational hemodynamics problems featuring a periodic alternation of fast and slow transients (the so called systolic and diastolic phases in circulation). As a matter of fact, for the same level of accuracy, the total number of time steps required within a heart beat is reduced to one third of the ones required by the non adaptive scheme.

In fact, in Veneziani and Villa [2013] a smart combination of algebraic factorizations as solvers and preconditioners of the Navier-Stokes equations based on the *a posteriori* error estimation provided by the pressure corrections is proposed as a potential optimal trade-off between numerical stability and efficiency.

6 Beyond the proof of concept

As explained in Sect. 1, LifeV is intended to be a tool to work aggressively on real problems, aiming at a general scope of bringing most advanced methods for computational predictive tools in the engineering practice (in broad sense).

In particular, one of the most important applications — yet non exclusive — is the simulation of cardiovascular problems. Examples of the use of LifeV for real clinical problems are: simulations of Left Ventricular Assist Devices (LVAD) Bonnemain et al. [2013, 2012], the study of the physiological Crosetto et al. [2011b] and abnormal fluid-dynamics in ascending aorta in presence of a bicuspid aortic valve Bonomi et al. [2015], Vergara et al. [2012], Faggiano et al. [2013], Viscardi et al. [2010], of Thoracic EndoVascular Repair (TEVAR) van Bogerijen et al. [2014], Auricchio et al. [2014], of the Total Cavopulmonary Connection Mirabella et al. [2013], Restrepo et al. [2015], Tang et al. [2015], of blood flow in stented coronary arteries Gogas et al. [2013] and in cerebral aneurysms Passerini et al. [2012]. The current trend in this field is the setting up of *in silico* or "Computer Aided" Clinical TrialsVeneziani [2015], i.e. of systematic investigations on large pools of patients to retrieve data of clinical relevance by integrating traditional measures and numerical simulations[11] In this scenario, numerical simulations are part of a complex, integrated pipeline involving: (i) Image/Data retrieval; (ii) Image Processing and Reconstruction (extracting the patient-specific morphology); (iii) Mesh generation and preprocessing (encoding of the boundary conditions); (iv) Numerical simulation (with LifeV); (v) Postprocessing and synthesis. As in the following sections we mainly describe applications relevant to step (iv), it is important to stress the integrated framework in which LifeV developers work, toward a systematic automation of the process, required from the large volume of patients to process. In this respect, we may consider LifeV as a vehicle of *methodological transfer* or *translational mathematics*, as leading edge methods are made available to the engineering community with a short time-to-market. Here we present a series of distinctive applications where we feel that using LifeV actually allowed to to bring rapidly new methods to real problems beyond the proof of concept stage.

[11] The ABSORB Project granted by Abbott Inc. at Emory University and the iCardioCloud Project granted by Fondazione Cariplo to University of Pavia have been developed in this perspective using LifeV.

6.1 FSI

In the Fluid-Structure Interaction (FSI) context, LifeV has offered a very important bench for testing novel algorithms. For example, it has been possible to test Robin-based interface conditions for applications in hemodynamics Nobile and Vergara [2008], Nobile et al. [2013, 2014], or compare segregated algoritms, the monolithic formulation, and the Steklov-Poincaré formulation Deparis et al. [2006].

An efficient solution method for FSI problem considers the physical unknowns and the fluid geometry problem for the Arbitrary Lagrangian-Eulerian (ALE) mapping as a single variable. This monolithic description implies to use ad-hoc parallel preconditioners. Most often they rely on a Dirichlet-Neumann inexact factorization between fluid and structure and then specific preconditioners for the subproblems Crosetto et al. [2011a]. Recently a new preconditioner FaCSI Deparis et al. [2016] has been developed and tested with LifeV with an effective scalability up to 4 thousands processors. A next step in FSI has been the use of non-conforming meshes between fluid and structure using rescaled-localized radial basis functions Deparis et al. [2014a].

A study on different material constitutive models for cerebral arterial tissue — in particular Hypereslatic isotropic laws, Hyperelastic anisotropic laws — have been studied in Tricerri et al. [2015]. A benchmark for the simulation of the flow inside carotids and the computation of shear stresses Balzani et al. [2015] has been tested with LifeV coupled with the FEAP library Taylor [2014], which includes sophisticated anisotropic material models.

6.2 Geometric Multiscale

The cardiovascular system features coupled local and global dynamics. Modeling its integrity by three dimensional geometries including FSI is either unfeasible or very expensive computation-wise and, most of the time, useless. A more efficient model entails for the coupling of multiple dimensions, like lumped zero-dimensional models, hyperbolic one-dimensional ones, and three dimensional FSI, leading to the so called *geometric multiscale* modeling as advocated in Veneziani [1998], Formaggia et al. [2001].

LifeV implements a full set of tools to integrate 0D, 1D and 3D-FSI models of the cardiovascular system Malossi et al. [2011, 2013], Blanco et al. [2013], Passerini et al. [2009], with also a multirate time stepping scheme to improve the computational efficiency Malossi et al. [2012]. It has been used to simulate integrated models of the cardiovascular system Bonnemain et al. [2013].

A critical aspect of this approach is the management of the dimensional mismatch between the different models, as the accurate 3D problems require more information at the interface that the one provided by the other surrogate models, This required the accurate analysis of “defective boundary problems” Formaggia et al. [2002], Veneziani and Vergara [2005, 2007], Formaggia et al. [2008, 2010]. A recent review on these topics can be found in Quarteroni et al. [2016].

6.3 Heart dynamics

Electrocardiology is one of the problems - beyond CFD but still related to cardiovascular mathematics - where LifeV has cumulated extensive experience. An effective preconditioner for the bidomain equations has been proposed and demonstrated in Gerardo-Giorda et al. [2009]. The basic idea is to use the simplified extended monodomain model to precondition the solution of the more realistic bidomain equations. Successively, the idea has been adapted to reduce the computational costs by mixing Monodomain and Bidomain equations in an adaptive procedure. A suitabe *a posteriori* estimator is used to decide when the Monodoimain equations are enough or the bidomain solution is needed Mirabella et al. [2011], Gerardo-Giorda et al. [2011]. Ionic models solved in LifeV ranges from the classical Rogers McCulloch, Fenton Karma, Luo Rudy I and II Clayton et al. [2011] to more involved ones Dupraz et al. [2015]. Specific high order methods (extending the classical Rush Larsen one) have been proposed and implemented in the library Perego and Veneziani [2009].

In addition, one research line has been oriented to the coupling of electrocardiology with cardiac mechanics. Hyperelasticity problems based on non-trivial mixed and primal formulations with applications in cardiac biomechanics have been studied in Rossi et al. [2011, 2012]. LifeV-based simulations of fully coupled electromechanics (using modules for the abstract coupling of solvers) can be found in Nobile et al. [2012], Ruiz-Baier et al. [2013], Rossi et al. [2014], Andreianov et al. [2015] for whole organ models, and in Ruiz-Baier et al. [2014], Gizzi et al. [2015], Ruiz-Baier [2015] for single-cell problems. The coupling with ventricular fluid dynamics and arterial tree FSI description are possible through a multiscale framework Quarteroni et al. [2015]. The coupling the Purkinje network, a network of high electrical conductivity myocardium fibers, has been implemented in Vergara et al. [2016].

Another research line in this field successfully carried out with LifeV is the variational estimation of cardiac conductivities (the tensor coefficients that are needed by the Bidomain equations) from potential measures Yang and Veneziani [2013].

6.4 Inverse problems and data assimilation

One of the most recent challenging topics in computational hemodynamics is the quantification of uncertainty and the improvement of the reliability in patient-specific settings. As a matter of fact, while the inclusion of patient specific geometries is now a well established procedure (as we recalled above), many other aspects of the patient-specific modeling still deserve attention. Parameters like viscosity, vessel wall rigidity, or cardiac conductivity are not routinely measured (or measurable) in the specific patient and however have generally a major impact on the numerical results. These concepts have been summarized in Veneziani and Vergara [2013]. Variational procedures have been implemented in LifeV, where the assimilation with available data or the parameter estimation are obtained by minimizing a mismatch functional. In D'Elia et al. [2012] this approach was introduced to incorporate into the numerical simulation of the incompressible Navier-Stokes equations sparse data available in the region of interest; in Perego et al. [2011] the procedure was introduced for estimating the vascular rigidity by solving an inverse FSI problem, while a similar procedure in Yang and Veneziani [2013, 2017] aims at the estimate of the cardiac conductivity.

6.5 Model reduction

One of the major challenges of modern scientific computing is the controlled reduction of the computational costs. In fact, practical use of HPC demands for extreme efficiency — even real time solutions. Improvement of computing architectures and cloud solutions that make relatively easy the access to HPC facilties is only a partial answer to this need Guzzetti et al. [2017]. From the modeling and methodological side, we need also customized models that can realize the trade off between efficiency and accuracy. These may be found by a smart combination of available High Fidelity solutions, according to the offline/online paradigm; or by the inclusion of specific features of a problem that may bring a significant advantage in comparison with general versatile but expensive methods. In LifeV these strategy have been both considered. For instance, in Colciago et al. [2014], a model for blood flow dynamics in a fixed domain, obtained by transpiration condition and a membrane model for the structure, has been compared to a full three dimesional FSI simulation. The former model is described only in the lumen with the Navier–Stokes equations, the structure is taken into account by surface Laplace–Beltrami operator on the surface representing the fluid-strucutre interface. The reduced model allows for roughly one third of the computational time and, in situations where the displacement of the artery is pretty small, the dynamics, including e.g. wall shear stresses, are very close if not indistinguishable from a full three dimensional simulation

In Bertagna and Veneziani [2014] a solution reduction procedure based on the Proper Orthogonal Decomposition (POD) was used to accelerate the variational estimate of the Young modulus of vascular tissues by solving an Inverse Fluid Structuire Interaction problem. POD wisely combines available offline High Fidelity solutions to obtain a rapid (online) parameter estimation. More challenging is the use of a similar approach for cardiac conductivities Yang [2015], Yang and Veneziani [2017], requiring nonstandard procedures.

A directional model reduction procedure called HiMod (*Hierarchical Model Reduction* - see Perotto et al. [2010], Aletti et al. [2015], Perotto [2014], Perotto and Veneziani [2014], Blanco et al. [2015], Mansilla Alvarez et al. [2016]) to accelerate the computation of advection diffusion reaction problems as well as incompressible fluids in pipe-like domains (or generally domains with a clear dominant direction, like in arteries) has been implemented in LifeV Aletti et al. [2015], Guzzetti [2014]. These modules will be released in the library soon.

6.6 Darcy equations and porous media

Single and multi-phase flow simulators in fractured porous media are of paramount importance in many fields like oil exploration and exploitation, CO_2 sequestration, nuclear waste disposal and geothermal reservoirs. A single-phase flow solver is implemented with the standard finite element spaces Raviart-Thomas for the Darcy velocity and piecewise constant for the pressure. A global pressure-total velocity formulation for the two-phase flow is developed and presented in Fumagalli and Scotti [2011], Fumagalli [2012] where the equations are solved using an IMPES-like technique. To handle fractures and faults in an efficient and accurate way the extended finite element method is adopted to locally enrich the cut elements, see Iori [2011], Del Pra et al. [2015], and a one-codimensional problem for the flow is considered for these objects, see Ferroni et al. [2016].

6.7 Ice sheets

Another application that has used the LifeV library (at Sandia Nat Lab, Albuquerque, NM) is the simulation of ice sheet flow. Ice behaves like a highly viscous shear-thinning incompressible fluid and can be modeled by nonlinear Stokes equations. In order to reduce computational costs several simplifications have been made to the Stokes model, exploiting the fact that ice sheets are very shallow. LifeV has been used to implement some of these models, including the Blatter-Pattyn (also known as First Order) approximation and the L1L2 approximation. The former model is a three-dimensional nonlinear elliptic PDE and the latter a depth-integrated integro-differential equations (see Perego et al. [2012]). In all models, nonlinearity has been solved with Newton method, coupling LifeV with Trilinos NOX package. The LifeV based ice sheet implementation has been mentioned in Evans et al. [2012] as an example of modern solver design for the solution of earth system models. Further, in Tezaur et al. [2015] the authors verified the results of another ice sheet code with those obtained using LifeV.

LifeV ice sheet module has been coupled with the climate library MPAS and used in inter-comparison studies to asses sensitivities to different boundary conditions and forcing terms, see Edwards et al. [2014] and Shannon et al. [2013]. The latter (Shannon et al. [2013]) has been considered in the IPCC (Intergovernmental Panel on Climate Change) report of 2014.

In Perego et al. [2014] the authors perform a large scale PDE-constrained optimization to estimate the basal friction field (70K parameters) in Greenland ice sheet. For this purpose LifeV has been coupled with the Trilinos package ROL to perform a reduced-gradient optimization using BFGS. The assembly of state and adjoint equations and the computation of objective functional and its gradient have been performed in LifeV.

7 Perspectives

LifeV has proved to be a versatile library for the study of numerical techniques for large scale and multiphysics computations with finite elements. The code is in continuous development. The latest introduction of expression templates has increased the easiness of usage at the high level. Unfortunately, due to the fact the way the code has been developed, mainly by PhD students, not all the applications have been ported yet to this framework. Work is ongoing in that direction. The library has been coded using the C++98 yet porting is ongoing to exploit new features of the C++11 and C++14 standard, which can make the code more readable, user friendly and efficient.

As for the parallelization issues, LifeV relies strongly on the tools provided by the Trilinos libraries. We are following their development closely and we will be ready to integrate all the new features the library will offer, in particular with respect to hybrid type parallelism.

A target use of the library the team is working on is running on cloud facilities Slawinski et al. [2012], Guzzetti et al. [2017] and GPL architectures.

Acknowledgments

Besides the funding agencies cited in the introduction of this paper, we have to acknowledge all the developers of LifeV, who have contributed with new numerical methods, provided help to beginners, and struggled for the definition of a common path. Porting and fixing bugs is also a remarkable task to which they have constantly contributed. It is difficult to name people, the list would be anyway incomplete, a good representation of the contributors is given in the text of the paper and, of course, on the developers website. However, the three seniors authors of this paper wish to thank their collaborators who over the years gave passion and dedication to the development. SD and LF from Lausanne and Milan groups wish to thank A. Quarteroni, who has contributed LifeV in many ways, by supporting its foundation, by dedicating human resources to the common project, outreaching, and mentoring. Also, worth of specific acknowledgment is G. Fourestey who has invested a lot of efforts to parallelize the code by introducing Trilinos and many of the concepts described in the paper. SD wishes to thank all his collaborators at CMCS in Lausanne, who have been able to work toghether in a very constructive environment. Particularly P. Crosetto for introducing and testing the parallel framework for monolithic FSI and associated preconditioners; C. Malossi for the design and implementation of the multiscale framework; S. Quinodoz for the design and implementation of the expression template module which is now at the core of the finite element formulation in LifeV; A. Gerbi, S. Rossi, R. Ruiz Baier, and P. Tricerri for their work on the mechanics of tissues, including active electromechainics; G. Grandperrin and R. Popescu for thier studies on parallel algorithms for the approximation of PDEs, in particolar the former and D. Forti for their essential contribution for the Navier–Stokes equations and FSI; the promising and successfull coupling of LifeV with a reduced basis solver for 3D problems has been carried out by C. Colciago and N. Dal Santo.

LF wishes to thank specially Antonio Cervone, for his work on the phase field formulation for stratified fluids and the Stokes solver, Alessio Fumagalli for the Darcy solver, Guido Iori and Marco del Pra for the XFEM implementation for Darcy's flow in fractured media and the porting to C++11, Davide Baroli for his many contributions and the support of several Master students struggling with the code, Stefano Zonca for the DG-XFEM implementation of fluid structure interaction problems, Michele Lancellotti for his work on Large Eddy Simulation. And, finally, a thank to Christian Vergara who has exploited LifeV for several real life applications of simulations of the cardiovascular system, as well as giving an important contribution to the development of numerical techniques for defective boundary conditions and FSI.

AV wishes to thank all the Emory collaborators for their passion, dedication and courage joining him in the new challenging and enthusiastic adventure in Atlanta; specifically (i) T. Passerini for the constant generous development and support not only in the development but also in the dissemination of LifeV among Emory students; (ii) M. Perego for his rigorous contribution and expansion of using LifeV at Sandia on the Ice-Sheet activity; (iii) U. Villa for the seminal and fundamental work in the Navier-Stokes modules that provided the basis of the current CFD in LifeV at Emory; (iv) L. Mirabella for bridging LifeV with the community of computational electrocardioologists and biomedical engineers at GA Tech. The early contribution of Daniele Di Pietro (University of Montpellier) for Expression Templates is gratefully acknowledged too.

Finally, all the authors together join to remember Fausto Saleri. Fausto Saleri introduced the name "LiFE" many years ago, in a 2D, Fortran 77 serial code for advection diffusion problems. Over the years, we made changes and additions, yet we do hope to have followed his enthusiasm, passion and dedication to scientific computing.

References

J. Ahrens, B. Geveci, and C. Law. 36 - paraview: An end-user tool for large-data visualization. In C. D. H. R. Johnson, editor, *Visualization Handbook*, pages 717 – LXXII. Butterworth-Heinemann, Burlington, 2005. ISBN 978-0-12-387582-2. doi: http://dx.doi.org/10.1016/B978-012387582-2/50038-1. URL http://www.sciencedirect.com/science/article/pii/B9780123875822500381.

M. Aletti, S. Perotto, and A. Veneziani. An educated-basis approach for the hierarchical model reduction of elliptic problems. Tech report, Dept. Math and CS, Emory University, 2015.

B. Andreianov, M. Bendahmane, A. Quarteroni, and R. Ruiz-Baier. Solvability analysis and numerical approximation of linearized cardiac electromechanics. *Math. Models Meth. Appl. Sci.*, 25(05):959–993, 2015. doi: 10.1142/S0218202515500244.

F. Auricchio, M. Conti, A. Lefieux, S. Morganti, A. Reali, F. Sardanelli, F. Secchi, S. Trimarchi, and A. Veneziani. Patient-specific analysis of post-operative aortic hemodynamics: a focus on thoracic endovascular repair (tevar). *Comp Mech*, 54(4):943–953, 2014. doi: 10.1007/s00466-014-0976-6.

D. Balzani, S. Deparis, S. Fausten, D. Forti, A. Heinlein, A. Klawonn, A. Quarteroni, O. Rheinbach, and J. SchrÃűder. Numerical modeling of fluidâĂŞstructure interaction in arteries with anisotropic polyconvex hyperelastic and anisotropic viscoelastic material models at finite strains. *International Journal for Numerical Methods in Biomedical Engineering*, (on line):1–41, 2015. ISSN 2040-7947. doi: 10.1002/cnm.2756. URL http://dx.doi.org/10.1002/cnm.2756. cnm.2756.

E. Bavier, M. Hoemmen, S. Rajamanickam, and H. Thornquist. Amesos2 and belos: Direct and iterative solvers for large sparse linear systems. *Scientific Programming*, 20(3):241–255, 2012. doi: 10.3233/SPR-2012-0352.

Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1âĂŞ4):173 – 201, 2007. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/j.cma.2007.07.016. URL http://www.sciencedirect.com/science/article/pii/S0045782507003027.

L. Bertagna and A. Veneziani. A model reduction approach for the variational estimation of vascular compliance by solving an inverse fluid-structure interaction problem. *Inverse Problems*, 30(5):055006, 2014. doi: 10.1088/0266-5611/30/5/055006.

L. Bertagna, A. Quaini, and A. Veneziani. Deconvolution-based nonlinear filtering for incompressible flows at moderately large reynolds numbers. *Int J Num Meth Fluids*, 81:463–488, 2016.

P. Blanco, L. M. Alvarez, and R. Feijóo. Hybrid element-based approximation for the navier–stokes equations in pipe-like domains. *Computer Methods in Applied Mechanics and Engineering*, 283:971–993, 2015.

P. J. Blanco, S. Deparis, and A. C. I. Malossi. On the continuity of mean total normal stress in geometrical multiscale cardiovascular problems. *Journal of Computational Physics*, 251:136–155, 2013. doi: 10.1016/j.jcp.2013.05.037.

J. Bonnemain, S. Deparis, and A. Quarteroni. Connecting Ventricular Assist Devices to the Aorta: a Numerical Model. In M. Emmer, editor, *Imagine Math. Between Culture and Mathematics*, pages 211–224, Venice, Italy, 2012. Springer-Verlag. ISBN 978-88-470-2426-7.

J. Bonnemain, A. C. I. Malossi, M. Lesinigo, S. Deparis, A. Quarteroni, and L. von Segesser. Numerical simulation of left ventricular assist device implantations: comparing the ascending and the descending aorta cannulations. *Medical Engineering and Physics*, 35(10):1465–1475, 2013. doi: 10.1016/j.medengphy.2013.03.022.

D. Bonomi, C. Vergara, E. Faggiano, M. Stevanella, C. Conti, A. Redaelli, G. Puppini, G. Faggian, L. Formaggia, and G. B. Luciani. Influence of the aortic valve leaflets on the fluid-dynamics in aorta in presence of a normally functioning bicuspid valve. *Biomechanics and Modeling in Mechanobiology*, 14(6):1349–1361, 2015. ISSN 1617-7940. doi: 10.1007/s10237-015-0679-8. URL http://dx.doi.org/10.1007/s10237-015-0679-8.

A. L. Bowers and L. G. Rebholz. Increasing accuracy and efficiency in fe computations of the leray-deconvolution model. *Numerical Methods for Partial Differential Equations*, 28(2):720–736, 2012. ISSN 1098-2426. doi: 10.1002/num.20653. URL http://dx.doi.org/10.1002/num.20653.

E. Burman and M. A. Fernández. Continuous interior penalty finite element method for the time-dependent Navier-Stokes equations: space discretization and convergence. *Numer. Math.*, 107(1):39–77, 2007. ISSN 0029-599X. doi: 10.1007/s00211-007-0070-5. URL http://dx.doi.org/10.1007/s00211-007-0070-5.

J. Cahouet and J.-P. Chabard. Some fast 3d finite element solvers for the generalized stokes problem. *International Journal for Numerical Methods in Fluids*, 8(8):869–895, 1988.

A. Cervone, N. A. Fadel, and L. Formaggia. Simulations of large three-dimensional sedimentary basin dynamics through domain decomposition technniques. In J. Eberhardsteiner, H. BÃűhm, and F. Rammerstorfer, editors, *ECCOMAS 2012 e-book*, pages 2652–2663. Vienna University of Technology, Vienna, 2012.

H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, and P. Navrátil. VisIt: An end-user tool for visualizing and analyzing very large data. In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, pages 357–372. Taylor and Francis, Boca Raton, Oct 2012.

R. Clayton, O. Bernus, E. Cherry, H. Dierckx, F. Fenton, L. Mirabella, A. Panfilov, F. B. Sachse, G. Seemann, and H. Zhang. Models of cardiac tissue electrophysiology: progress, challenges and open questions. *Progress in biophysics and molecular biology*, 104(1):22–48, 2011.

C. M. Colciago, S. Deparis, and A. Quarteroni. Comparisons between reduced order models and full 3D models for fluid-structure interaction problems in haemodynamics. *Journal Of Computational And Applied Mathematics*, 265:120–138, 2014. doi: 10.1016/j.cam.2013.09.049.

Computational Engineering International, Inc. EnSight User Manual, 2011. https://www.ensight.com/.

P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM J. Sci. Comput.*, 33(4):1598–1622, 2011a. ISSN 1064-8275. doi: 10.1137/090772836. URL http://dx.doi.org/10.1137/090772836.

P. Crosetto, P. Reymond, S. Deparis, D. Kontaxakis, N. Stergiopulos, and A. Quarteroni. Fluid-structure interaction simulation of aortic blood flow. *Comput. & Fluids*, 43:46–57, 2011b. ISSN 0045-7930. doi: 10.1016/j.compfluid.2010.11.032. URL http://dx.doi.org/10.1016/j.compfluid.2010.11.032.

M. Del Pra, A. Fumagalli, and A. Scotti. Well posedness of fully coupled fracture/bulk darcy flow with xfem. Submitted to: SIAM Journal on Numerical Analysis, May 2015.

M. D'Elia, M. Perego, and A. Veneziani. A variational data assimilation procedure for the incompressible navier-stokes equations in hemodynamics,. *J Sci Comp*, 52(2):340–359, 2012.

S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. *Comput. Methods Appl. Mech. Engrg.*, 195(41-43):5797–5812, 2006. ISSN 0045-7825. doi: 10.1016/j.cma.2005.09.029. URL http://dx.doi.org/10.1016/j.cma.2005.09.029.

S. Deparis, D. Forti, and A. Quarteroni. A Rescaled Localized Radial Basis Function Interpolation on Non-Cartesian and Nonconforming Grids. *SIAM Journal on Scientific Computing*, 36:A2745–A2762, 2014a.

S. Deparis, G. Grandperrin, and A. Quarteroni. Parallel preconditioners for the unsteady Navier–Stokes equations and applications to hemodynamics simulations. *Computers & Fluids*, 92: 253–273, 2014b. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2013.10.034. URL http://www.sciencedirect.com/science/article/pii/S0045793013004179.

S. Deparis, D. Forti, G. Grandperrin, and A. Quarteroni. FaCSI: A block parallel preconditioner for fluid–structure interaction in hemodynamics. *Journal of Computational Physics*, 327:700–718, 2016.

K. Devine, E. Boman, R. Heaphy, B. Hendrickson, and C. Vaughan. Zoltan data management services for parallel dynamic applications. *Computing in Science and Engineering*, 4(2):90–97, 2002.

D. A. Di Pietro and A. Veneziani. Expression templates implementation of continuous and discontinuous Galerkin methods. *Computing and Visualization in Science*, 12(8):421–436, 2009a.

D. A. Di Pietro and A. Veneziani. Expression templates implementation of continuous and discontinuous galerkin methods. *Comput. Vis. Sci.*, 12(8):421–436, Sept. 2009b. ISSN 1432-9360. doi: 10.1007/s00791-008-0117-x. URL http://dx.doi.org/10.1007/s00791-008-0117-x.

M. Dupraz, S. Filippi, A. Gizzi, A. Quarteroni, and R. Ruiz-Baier. Finite element and finite volume-element simulation of pseudo-ecgs and cardiac alternans. *Math. Methods Appl. Sci.*, 38 (6):1046–1058, 2015. doi: 10.1002/mma.3127.

T. L. Edwards, X. Fettweis, O. Gagliardini, F. Gillet-Chaulet, H. Goelzer, J. M. Gregory, M. Hoffman, P. Huybrechts, A. J. Payne, M. Perego, S. Price, A. Quiquet, and C. Ritz. Effect of uncertainty in surface mass balanceâĂŞelevation feedback on projections of the future sea level contribution of the greenland ice sheet. *The Cryosphere*, 8(1):195–208, 2014. doi: 10.5194/tc-8-195-2014. URL http://www.the-cryosphere.net/8/195/2014/.

H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 227(3):1790–1808, 2008. ISSN 0021-9991. doi: 10.1016/j.jcp.2007.09.026. URL http://dx.doi.org/10.1016/j.jcp.2007.09.026.

H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, Oxford, UK, 2014.

A. Ern and J. L. Guermond. *Theory and Practice of Finite Elements*. Springer, New York, 2006.

K. J. Evans, A. G. Salinger, P. H. Worley, S. F. Price, W. H. Lipscomb, J. A. Nichols, J. B. White, M. Perego, M. Vertenstein, J. Edwards, and J.-F. Lemieux. A modern solver interface to manage solution algorithms in the community earth system model. *International Journal of High Performance Computing Applications*, 26(1):54–62, 2012. doi: 10.1177/1094342011435159. URL http://hpc.sagepub.com/content/26/1/54.abstract.

E. Faggiano, L. Antiga, G. Puppini, A. Quarteroni, G. Luciani, and C. Vergara. Helical flows and asymmetry of blood jet in dilated ascending aorta with normally functioning bicuspid valve. *Biomechanics and Modeling in Mechanobiology*, 12(4):801–813, 2013.

A. Ferroni, L. Formaggia, and A. Fumagalli. Numerical analysis of darcy problem on surfaces. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(6):1615–1630, 2016.

L. Formaggia, J.-F. Gerbeau, F. Nobile, and A. Quarteroni. On the coupling of 3d and 1d navier–stokes equations for flow problems in compliant vessels. *Computer Methods in Applied Mechanics and Engineering*, 191(6):561–582, 2001.

L. Formaggia, J. F. Gerbeau, F. Nobile, and A. Quarteroni. Numerical treatment of defective boundary conditions for the navier–stokes equations. *SIAM Journal on Numerical Analysis*, 40(1):376–401, 2002. doi: 10.1137/S003614290038296X. URL http://dx.doi.org/10.1137/S003614290038296X.

L. Formaggia, A. Veneziani, and C. Vergara. A new approach to numerical solution of defective boundary value problems in incompressible fluid dynamics. *SIAM Journal on Numerical Analysis*, 46(6):2769–2794, 2008.

L. Formaggia, A. Veneziani, and C. Vergara. Flow rate boundary problems for an incompressible fluid in deformable domains: formulations and solution methods. *Computer Methods in Applied Mechanics and Engineering*, 199(9):677–688, 2010.

L. Formaggia, F. Saleri, and A. Veneziani. *Solving Numerical PDEs: Problems, Applications, Exercises: Problems, Applications, Exercises*. Springer-Verlag, Milan, Italy, 2012.

D. Forti and L. Dedè. Semi–implicit BDF time discretization of the Navier–Stokes equations with VMS–LES modeling in High Performance Computing framework. *Computer & Fluids*, 117: 168–182, 2015.

P. Frey. Medit: An interactive mesh visualization software. Technical Report RT-0253, Institut National de Recherche en Informatique et en Automatique, 2001.

A. Fumagalli. *Numerical Modelling of Flows in Fractured Porous Media by the XFEM Method*. PhD thesis, Politecnico di Milano, 2012.

A. Fumagalli and S. Scotti. Numerical modelling of multiphase subsurface flow in the presence of fractures. *Communications in Applied and Industrial Mathematics*, 3(1):1–23, 2011. ISSN 2038-0909. doi: 10.1685/journal.caim.380. URL http://caim.simai.eu/index.php/caim/article/view/380.

G. Furnish. Disambiguated glommable expression templates. *Comp Phys*, 11:263–269, 1997.

A. Gauthier, F. Saleri, and A. Veneziani. A fast preconditioner for the incompressible navier stokes equations. *Computing and Visualization in science*, 6(2-3):105–112, 2004.

M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala. ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

L. Gerardo-Giorda, L. Mirabella, F. Nobile, M. Perego, and A. Veneziani. A model-based block-triangular preconditioner for the Bidomain system in electrocardiology. *J. Comput. Phys.*, 228 (10):3625–3639, 2009. doi: http://dx.doi.org/10.1016/j.jcp.2009.01.034.

L. Gerardo-Giorda, M. Perego, and A. Veneziani. Optimized schwarz coupling of bidomain and monodomain models in electrocardiology. *ESAIM: Mathematical Modelling and Numerical Analysis*, 45(02):309–334, 2011.

P. Gervasio. Convergence analysis of high order algebraic fractional step schemes for time-dependent stokes equations. *SIAM Journal on Numerical Analysis*, 46(4):1682–1703, 2008.

P. Gervasio and F. Saleri. Algebraic fractional-step schemes for time-dependent incompressible navier–stokes equations. *Journal of Scientific Computing*, 27(1-3):257–269, 2006.

P. Gervasio, F. Saleri, and A. Veneziani. Algebraic splitting methods for the incompressible navier-stokes equations. *J Comp Phys*, 214(1):347–365, 2006.

C. Geuzaine and J.-F. Remacle. 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. ISSN 1097-0207. doi: 10.1002/nme.2579. URL http://dx.doi.org/10.1002/nme.2579.

A. Gizzi, R. Ruiz-Baier, S. Rossi, A. Laadhari, C. Cherubini, and S. Filippi. A three-dimensional continuum model of active contraction in single cardiomyocytes. In A. Quarteroni, editor, *Modeling the Heart and the Circulatory System*, pages 157–176. Springer-Verlag, Milano, 2015. doi: 10.1007/978-3-319-05230-4_6.

B. Gogas, L. Timmins, T. Passerini, K. Piccinelli M, M. S., G. D. D., S. King, and H. Samady. Biomechanical assessment of bioresorbable devices. *JACC: Cardiovascular Interventions*, 6(7): 760–761, 2013.

S. Guzzetti. *Hierarchical Model Reduction for the Incompressible Navier-Stokes Equations*. Master thesis, 2014.

S. Guzzetti, A. Veneziani, and V. Sunderam. Performances of overlapping domain decomposition methods. Tech rep, Dept. Math and CS, Emory University, 2015.

S. Guzzetti, T. Passerini, J. Slawinski, U. Villa, A. Veneziani, and V. Sunderam. Platform and algorithm effects on computational fluid dynamics applications in life sciences. *Future Generation Computer Systems*, 67:382–396, 2017.

J. Härdtlein, C. Pflaum, A. Linke, and C. H. Wolters. Advanced expression templates programming. *Computing and Visualization in Science*, 13(2):59–68, 2009. ISSN 1433-0369. doi: 10.1007/s00791-009-0128-2. URL http://dx.doi.org/10.1007/s00791-009-0128-2.

F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012. ISSN 1570-2820.

M. A. Heroux. Epetra performance optimization guide. Technical report, Sandia National Laboratories, 2009.

M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005. ISSN 0098-3500. doi: http://doi.acm.org/10.1145/1089014.1089021.

J. J. Hu, A. Prokopenko, C. M. Siefert, R. S. Tuminaro, and T. A. Wiesner. MueLu multigrid framework. http://trilinos.org/packages/muelu, 2014.

K. Iglberger, G. Hager, J. Treibig, and U. Rüde. Expression templates revisited: a performance analysis of current methodologies. *SIAM Journal on Scientific Computing*, 34(2):C42–C69, 2012.

G. Iori. Una metodologia XFEM per problemi ellittici 3D con superfici di discontinuità. Master's thesis, Politecnico di Milano, Dec. 2011. Advisor: Luca Formaggia. Co-advisor: Alessio Fumagalli.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

R. M. Lancellotti. *Large eddy simulations in haemodynamics: models and applications*. PhD thesis, Department of Mathematics, Politecnico di Milano, Italy, 2015.

A. C. I. Malossi, P. J. Blanco, S. Deparis, and A. Quarteroni. Algorithms for the partitioned solution of weakly coupled fluid models for cardiovascular flows. *International Journal for Numerical Methods in Biomedical Engineering*, 27(12):2035–2057, 2011. doi: 10.1002/cnm.1457.

A. C. I. Malossi, P. J. Blanco, and S. Deparis. A two-level time step technique for the partitioned solution of one-dimensional arterial networks. *Computer Methods in Applied Mechanics and Engineering*, 237-240:212–226, 2012. doi: 10.1016/j.cma.2012.05.017.

A. C. I. Malossi, P. J. Blanco, P. Crosetto, S. Deparis, and A. Quarteroni. Implicit coupling of one-dimensional and three-dimensional blood flow models with compliant vessels. *Multiscale Modeling and Simulation*, 11(2):474–506, 2013. doi: 10.1137/120867408.

L. Mansilla Alvarez, P. Blanco, C. Bulant, E. Dari, A. Veneziani, and R. Feijóo. Transversally enriched pipe element method (tepem): An effective numerical approach for blood flow modeling. *International Journal for Numerical Methods in Biomedical Engineering*, (on line):1–24, 2016.

L. Mirabella, F. Nobile, and A. Veneziani. An a posteriori error estimator for model adaptivity in electrocardiology. *Computer Methods in Applied Mechanics and Engineering*, 200(37):2727–2737, 2011.

L. Mirabella, C. Haggerty, P. T., M. Piccinelli, P. J. Del Nido, A. Veneziani, and A. P. Yoganathan. Treatment planning for a tcpc test case: a numerical investigation under rigid and moving wall assumptions. *Int J Num Meth Biomed Eng*, 29(2):197–216, 2013.

F. Nicoud, H. B. Toda, O. Cabrit, S. Bose, and J. Lee. Using singular values to build a subgrid-scale model for large eddy simulations. *Physics of Fluids*, 23(8):085106, 2011.

F. Nobile and C. Vergara. An effective fluid-structure interaction formulation for vascular dynamics by generalized Robin conditions. *SIAM J Sc Comp*, 30(2):731–763, 2008.

F. Nobile, A. Quarteroni, and R. Ruiz-Baier. An active strain electromechanical model for cardiac tissue. *Int. J. Numer. Methods Biomed. Engrg.*, 28(1):52–71, 2012. doi: 10.1002/cnm.1468. URL ./myPapers/nqr_ijnmbe12.pdf.

F. Nobile, M. Pozzoli, and C. Vergara. Time accurate partitioned algorithms for the solution of fluid-structure interaction problems in haemodynamics. *Computer & Fluids*, 86:470–482, 2013.

F. Nobile, M. Pozzoli, and C. Vergara. Inexact accurate partitioned algorithms for fluid-structure interaction problems with finite elasticity in haemodynamics. *Journal of Computational Physics*, 273:598–617, 2014.

T. Passerini, M. de Luca, L. Formaggia, A. Quarteroni, and A. Veneziani. A 3d/1d geometrical multiscale model of cerebral vasculature. *Journal of Engineering Mathematics*, 64(4):319–330, 2009.

T. Passerini, L. M. Sangalli, S. Vantini, M. Piccinelli, S. Bacigaluppi, L. Antiga, E. Boccardi, P. Secchi, and A. Veneziani. An integrated statistical investigation of the internal carotid ar- teries hosting cerebral aneurysms. *Cardiov Eng Techn*, 3(1):26–40, 2012.

T. Passerini, A. Quaini, U. Villa, A. Veneziani, and S. Canic. Validation of an open source framework for the simulation of blood flow in rigid and deformable vessels. *Int J Num Meth Biomed Eng*, 29(11):1192–1213, 2013.

S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows. *International J. on Heat and Mass Transfer*, 15:1787–1806, 1972. doi: 10.1016/0017-9310(72)90054-3.

M. Perego and A. Veneziani. An efficient generalization of the rush-larsen method for solving electro-physiology membrane equations. *Electronic Transactions on Numerical Analysis*, 35:234–256, 2009.

M. Perego, A. Veneziani, and C. Vergara. A variational approach for estimating the compliance of the cardiovascular tissue: an inverse fluid-structure interaction problem. *SIAM J Sc Comp*, 33(3):1181–1211, 2011.

M. Perego, M. Gunzburger, and J. Burkardt. Parallel finite-element implementation for higher-order ice-sheet models. *Journal of Glaciology*, 58(207):76–88, 2012. doi: 10.3189/2012JoG11J063. URL http://www.ingentaconnect.com/content/igsoc/jog/2012/00000058/00000207/art00007.

M. Perego, S. Price, and G. Stadler. Optimal initial conditions for coupling ice sheet models to earth system models. *Journal of Geophysical Research: Earth Surface*, 119(9):1894–1917, 2014. ISSN 2169-9011. doi: 10.1002/2014JF003181. URL http://dx.doi.org/10.1002/2014JF003181.

J. B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108(1): 51–58, 1993.

S. Perotto. A survey of hierarchical model (hi-mod) reduction methods for elliptic problems. In *Numerical simulations of coupled problems in engineering*, pages 217–241. Springer, 2014.

S. Perotto and A. Veneziani. Coupled model and grid adaptivity in hierarchical reduction of elliptic problems. *Journal of Scientific Computing*, 60(3):505–536, 2014.

S. Perotto, A. Ern, and A. Veneziani. Hierarchical local model reduction for elliptic problems i: A domain decomposition approach. *SIAM MMS*, 8(4):1102–1127, 2010.

C. Pflaum. Expression templates for partial differential equations. *Computing and Visualization in Science*, 4(1):1–8, 2001.

R. Popescu. *Parallel algorithms and efficient implementation techniques for finite element approximations*. PhD thesis, SB, Lausanne, 2013.

A. Prokopenko, J. J. Hu, T. A. Wiesner, C. M. Siefert, and R. S. Tuminaro. Muelu userâĂŹs guide 1.0. Technical Report SAND2014-18874, Sandia National Labs, 2014.

A. Quarteroni, F. Saleri, and A. Veneziani. Factorization methods for the numerical approximation of navier–stokes equations. *Computer methods in applied mechanics and engineering*, 188(1):505–526, 2000.

A. Quarteroni, S. Rossi, T. Lassila, and R. Ruiz-Baier. Integrated heart – coupling multiscale and multiphysics models for simulation of the total heart function. *In preparation*, 2015.

A. Quarteroni, A. Veneziani, and C. Vergara. Geometric multiscale modeling of the cardiovascular system, between theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 302:193–252, 2016.

S. Quinodoz. *Numerical Simulation of Orbitally Shaken Reactors*. PhD thesis, SB, 2012.

M. Restrepo, M. Luffel, J. Sebring, K. Kanter, P. Del Nido, A. Veneziani, J. Rossignac, and A. Yoganathan. Surgical planning of the total cavopulmonary connection: robustness analysis. *Annals of biomedical engineering*, 43(6):1321–1334, 2015.

S. Rossi, R. Ruiz-Baier, L. F. Pavarino, and A. Quarteroni. Active strain and activation models in cardiac electromechanics. *Proc. Appl. Math. Mech.*, 11(1):119–120, 2011. doi: 10.1002/pamm.201110051.

S. Rossi, R. Ruiz-Baier, L. F. Pavarino, and A. Quarteroni. Orthotropic active strain models for the numerical simulation of cardiac biomechanics. *Int. J. Numer. Methods Biomed. Engrg.*, 28(6–7):761–788, 2012. doi: 10.1002/cnm.2473. URL ./myPapers/rrpq_ijnmbe12.pdf.

S. Rossi, T. Lassila, R. Ruiz-Baier, A. Sequeira, and A. Quarteroni. Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics. *Eur. J. Mech. A/Solids*, 48:129–142, 2014. doi: 10.1016/j.euromechsol.2013.10.009.

R. Ruiz-Baier. Primal-mixed formulations for reaction-diffusion systems on deforming domains. *J. Comput. Phys.*, in press, 2015.

R. Ruiz-Baier, D. Ambrosi, S. Pezzuto, S. Rossi, and A. Quarteroni. Activation models for the numerical simulation of cardiac electromechanical interactions. In G. Holzapfel and E. Kuhl, editors, *Computer Models in Biomechanics: From Nano to Macro*, pages 189–201. Springer, Netherlands, 2013. doi: 10.1007/978-94-007-5464-5_14.

R. Ruiz-Baier, A. Gizzi, S. Rossi, C. Cherubini, A. Laadhari, S. Filippi, and A. Quarteroni. Mathematical modeling of active contraction in isolated cardiomyocytes. *Math. Medicine Biol.*, 31(3):259–283, 2014. ISSN 1477-8599. doi: 10.1093/imammb/dqt009.

M. Sala and M. Heroux. Robust algebraic preconditioners with IFPACK 3.0. Technical Report SAND-0662, Sandia National Laboratories, 2005.

M. Sala, K. Stanley, and M. Heroux. On the design of interfaces to sparse direct solvers. *submitted*, 2006a.

M. Sala, K. Stanley, and M. Heroux. Amesos: A set of general interfaces to sparse direct solver libraries. In *Proceedings of PARA'06 Conference, Umea, Sweden*, 2006b.

F. Saleri and A. Veneziani. Pressure correction algebraic splitting methods for the incompressible navier–stokes equations. *SIAM journal on numerical analysis*, 43(1):174–194, 2005.

F. R. Schaefer. Getpot, powerful input file and command line parser, 2007. http://getpot.sourceforge.net/.

J. Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, 1997. ISSN 1432-9360. doi: 10.1007/s007910050004. URL http://dx.doi.org/10.1007/s007910050004.

J. Schöberl, K. Martin, and B. Lorensen. *The Visualization Toolkit*. Kitware, 2006. ISBN 978-1-930934-19-1. 4th edition.

S. Shannon, A. Payne, I. Bartholomew, M. Van Den Broeke, T. Edwards, X. Fettweis, O. Gagliardini, F. Gillet-chaulet, H. Goelzer, M. Hoffman, P. Huybrechts, D. Mair, P. Nienow, M. Perego, S. Price, C. Smeets, A. Sole, R. Van De Wal, and T. Zwinger. Enhanced basal lubrication and the contribution of the greenland ice sheet to future sea-level rise. *Proceedings of the National Academy of Sciences of the United States of America - PNAS*, 110(35):14156âĂŞ14161, 2013. ISSN 0027-8424. doi: 10.1073/pnas.1212647110.

J. Slawinski, T. Passerini, U. Villa, A. Veneziani, and V. Sunderam. Experiences with target-platform heterogeneity in clouds, grids, and on-premises resources. In T. INSERT, editor, *Proceedings of HCW 2012 21st International Heterogeneity in Computing Workshop*, 2012.

E. Tang, Z. A. Wei, K. K. Whitehead, A. Veneziani, M. A. Fogel, and A. P. Yoganathan. Respiratory pulsations affect fontan connection power loss: using real time velocity mapping to improve the accuracy of computational simulations. *Journal of Cardiovascular Magnetic Resonance*, 17(1): 1, 2015.

R. L. Taylor. FEAP - finite element analysis program, 2014. URL http://www.ce.berkeley/feap.

I. K. Tezaur, M. Perego, A. G. Salinger, R. S. Tuminaro, and S. F. Price. Albany/felix: a parallel, scalable and robust, finite element, first-order stokes approximation ice sheet solver built for advanced analysis. *Geoscientific Model Development*, 8(4):1197–1220, 2015. doi: 10.5194/gmd-8-1197-2015. URL http://www.geosci-model-dev.net/8/1197/2015/.

The HDF Group. Hierarchical Data Format, version 5, 1997. http://www.hdfgroup.org/HDF5/.

P. Tricerri, L. Dede', S. Deparis, A. Quarteroni, A. M. Robertson, and A. Sequeira. Fluid-structure interaction simulations of cerebral arteries modeled by isotropic and anisotropic constitutive laws. *Computational Mechanics -International Journal then Research Journal-*, 55(3):479–498, 2015. doi: 10.1007/s00466-014-1117-y.

G. H. W. van Bogerijen, F. Auricchio, M. Conti, A. Lefiueux, A. Reali, A. Veneziani, J. Tolenaar, V. R. F.L. Moll, and S. Trimarchi. Aortic hemodynamics after thoracic endovascular aortic repair with the role of bird-beak. *J Endov Therapy*, 21:791–802, 2014.

T. Veldhuizen. Expression templates. *C++ Report Magazine*, 7:26–31, 1995.

A. Veneziani. *Mathematical and Numerical Modeling of Blood Flow Problems*. Phd, 1998.

A. Veneziani. Block factorized preconditioners for high-order accurate in time approximation of the navier-stokes equations. *Numerical Methods for Partial Differential Equations*, 19(4):487–510, 2003.

A. Veneziani. A note on the consistency and stability properties of yosida fractional step schemes for the unsteady stokes equations. *SIAM Journal on Numerical Analysis*, 47(4):2838–2843, 2009.

A. Veneziani. *Stent design and improvement: a matter of Mathematics too*. A. Gruenztig Center, Emory University, 2015.

A. Veneziani and C. Vergara. Flow rate defective boundary conditions in haemodynamics simulations. *International Journal for Numerical Methods in Fluids*, 47(8-9):803–816, 2005.

A. Veneziani and C. Vergara. An approximate method for solving incompressible Navier-Stokes problems with flow rate conditions. *CMAME*, 196(9-12):1685–1700, 2007.

A. Veneziani and C. Vergara. Inverse problems in cardiovascular mathematics: toward patient-specific data assimilation and optimization. *International journal for numerical methods in biomedical engineering*, 29(7):723–725, 2013.

A. Veneziani and U. Villa. ALADINS: An algebraic splitting time adaptive solver for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 238:359–375, 2013.

C. Vergara, F. Viscardi, L. Antiga, and G. Luciani. Influence of bicuspid valve geometry on ascending aortic fluid-dynamics: a parametric study. *Artificial Organs*, 36(4):368–378, 2012.

C. Vergara, M. Lange, S. Palamara, T. Lassila, A. F. Frangi, and A. Quarteroni. A coupled 3dâĂŞ1d numerical monodomain solver for cardiac electrical activation in the myocardium with detailed purkinje network. *Journal of Computational Physics*, 308:218 – 238, 2016. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2015.12.016. URL http://www.sciencedirect.com/science/article/pii/S0021999115008359.

A. Viguerie and A. Veneziani. Inexact algebraic factorization methods for the steady incompressible navier stokes equations at moderate reynolds numbers., 2017. URL www.mathcs.emory.edu.

U. Villa. *Scalable efficient methods for incompressible fluid-dynamics in engineering problems*. Phd, 2011.

F. Viscardi, C. Vergara, L. Antiga, S. Merelli, A. Veneziani, G. Puppini, G. Faggian, A. Mazzucco, and G. Luciani. Comparative finite element model analysis of ascending aortic flow in bicuspid and tricuspid aortic valve. *Artificial organs*, 34(12):1114–20, Dec. 2010. ISSN 1525-1594. doi: 10.1111/j.1525-1594.2009.00989.x. URL http://www.ncbi.nlm.nih.gov/pubmed/20618222.

H. Yang. *Parameter Estimation and Reduced-order Modeling in Electrocardiology*. PhD thesis, Emory University, 2015.

H. Yang and A. Veneziani. Variational estimation of cardiac conductivities by a data assimilation procedure. Tech rep, Dept. of Math and CS, Emory University, 2013.

H. Yang and A. Veneziani. Efficient estimation of cardiac conductivities via pod-deim model order reduction. 2017.

