AN OBJECT ORIENTED FLOW SOLVER FOR THE CRS4 VIRTUAL VASCULAR PROJECT

G. ABDOULAEV, A. VARONE AND G. ZANETTI

Center for Advanced Studies, Research and Development in Sardinia, C.P. 94, Uta (CA) 09010, Italy
E-mail: gassan@crs4.it

The CRS4 Virtual Vascular Project (*ViVa*) is aimed to the development of software tools for hemodynamic specialists and cardiovascular surgeons in order to study and interpret the information produced by non-invasive imaging equipment. The computational kernel of the *ViVa* system is a solver of the Navier-Stokes equations for viscous incompressible fluid, which govern the blood flow in large vessels (*e.g.* arteries). The computational strategy is based on the domain decomposition with the mortar element method. The mortar element method provides high encapsulation on the level of subdomain computations, *i.e.* subdomain meshes, matrices, preconditioners can be treated completely independently. This feature allows to implement the mortar method efficiently within the frames of the object oriented approach and *C++* programming language. Thanks to the modularity of the code, in different subdomains we can use different meshes (hexagonal, tetrahedral), different matrix storage schemes (band, sparse), different preconditioners. The flexibility in the choice of the subdomain numerical technique makes it possible to construct computationally "optimal" applications, for instance, to use multigrid subdomain preconditioners, or to exploit coarser meshes, where the solution is smooth, etc. Thus a library of *C++ classes* has been developed, which can be used to build a required numerical model.

1 Introduction

In this paper we present a program, *Nast++*, specifically developed for the simulation of blood flow and passive scalars transport in large arteries. *Nast++* is designed as a part of a larger project (*ViVa*), [2] being carried out at the Center for Advanced Studies, Research and Development in Sardinia (CRS4). The aim of *ViVa* is to develop tools for the modern hemodynamicist and cardiovascular surgeon to study and interpret the constantly increasing amount of information being produced by noninvasive imaging equipment. In particular, the system should be able to process and visualize three-dimensional medical data, reconstruct the geometry of arteries of specific patients, and simulate blood flow in them.

Nast++ is structured as a toolbox containing blocks (mapped to *C++ classes*) which can be composed to address a wide scope of applications requiring the solution of series of elliptic (Poisson, Helmholtz) equations. *Nast++* computational approach is based on the division of the physical space in subdomains connected together by mortar elements. The latter computational method takes advantage of the weak continuity condition on interfaces between subdomains to allow for a flexible description of the computational procedure within each subdomain. In particular, it makes possible to have non conforming meshes in different subdomains.

The clinical relevance of blood flow simulation in large arteries is crucially based on the capability of running the simulation on realistic geometrical models of the vessels of the patient treated. In general, and even more so in the pathological cases, there is a large variability in vessels geometries and this, in turn, results in a wide variation of possible blood flow behaviors. The mortar elements approach provides a powerful framework in this applicative context because it allows to increase grid resolution where it is needed, typically close to bifurcation regions, and to substitute parts of the geometry, for instance to simulate the effect of a angioplastic procedure, without the need of rebuilding the grids for the whole computational domain. The mortar domain decomposition method can be implemented very effectively using OO techniques and, moreover, is particularly well suited for parallelization. [1]

In recent years there has been a general effort to use object oriented programming techniques for the numerical solution of partial differential equations (PDEs) using the finite element method. [9,16] Although the object oriented approach makes it easier to develop, modify and adapt a program to a particular application, we found, as already reported by other authors, that it is a non trivial task to write a computationally efficient C++ code while maintaining a object oriented design.

The content of this paper is the following. In the section 2 we present the problem, give the weak formulation of the Navier-Stokes equations for the incompressible flow, and formulate the methods to solve the problem numerically, such as the Lagrangian-Galerkin/projection method for the time-dependent Navier-Stokes equations, and the mortar finite element method. We also give here a brief sketch of the methods used to solve the resulting algebraic system. Section 3 is devoted to the data structure implemented in the code and demonstrates some numerical results of blood flow simulation in prototypical geometries.

2 Problem and Methods

Blood is a complicated fluid with a complex rheology. [17] However, in the specific case of blood flow in large arteries, the length scales and time scale characterizing the flow are such that it is appropriate to describe it as an incompressible Newtonian fluid, [20] governed by the following equations

$$\begin{aligned}
&\frac{\partial \mathbf{u}}{\partial t} - \nu\nabla(\nabla\mathbf{u}) - (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \nabla p = \mathbf{f} \\
&\nabla\cdot\mathbf{u} = 0 \\
&\mathbf{x}\in\Omega, \quad t>0
\end{aligned} \tag{1}$$

where $\mathbf{u} = (u_1, u_2, u_3)$ is a velocity field, p is a pressure, ν is an average value for blood kinematic viscosity. a viscosity coefficient, $\mathbf{f}$ is a volumic external force.

In principle, this model is incomplete because it neglects the interaction between blood flow and the elastic behavior of the arterial walls. In typical applications, however, this interaction is considered a secondary effect and thus ignored. [23] *Nast++* is designed to provide support for fluid–structure interaction modeling, but this feature will be reported elsewhere.

We consider boundary conditions of two types. Assume that the boundary of the domain Ω is composed of two parts:

$$\Omega = \Gamma_0 \cup \Gamma_1, \quad \text{where} \;\; \Gamma_1 = \bigcup_k S_k$$

On Γ_0 the velocity is prescribed: $\mathbf{u} = \mathbf{U}(\mathbf{x}, t), \;\; \mathbf{x}\in\Gamma_0, \;\; t>0$, whereas on Γ_1 the mean pressure boundary condition is applied, [14] that is:

$$\frac{1}{|S_k|}\int_{S_k} p\,ds = P_k(t), \tag{2}$$

where $P_k(t)$ is a prescribed scalar function. In the case of the blood flow problem Γ_0 represents a vessel wall boundary, and S_k are artificial upstream and downstream section boundaries. The system (1) along with the boundary conditions (2) does not result in a well-posed problem, and thus it is necessary to prescribe additional boundary condition in order to close the system. [14] Besides the boundary conditions (2), it is possible to prescribe also a total pressure or a net flux on the upstream and downstream artificial boundaries S_k. [6,14]

A weak formulation of the system (1) reads as follows

$$\begin{aligned}
\left(\frac{D\mathbf{u}}{Dt}, \mathbf{v}\right) + (\nu\nabla\mathbf{u}, \nabla\mathbf{v}) - (p, \nabla\mathbf{v}) \;&=\; (\mathbf{f}, \mathbf{v}) - \sum_k P_k(t)\int\limits_{S_k} \mathbf{v}\cdot\mathbf{n}\,dS \\
(\nabla\cdot\mathbf{u}, q) = 0& \\
\forall(\mathbf{v}, q) \in \mathbf{V}\cup Q&
\end{aligned} \tag{3}$$

where $\dfrac{D\mathbf{u}}{Dt} = \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u}$ is the total (material) time derivative, and $\mathbf{V}$ and Q are suitable functional spaces.

To integrate in time the Navier-Stokes equations we have implemented a combination of the fractional step projection method [11,13,19,22] and the Lagrange-Galerkin (characteristic) method, [18] although the package provides the tools to implement other time advancing schemes. Therefore the problem is reduced to a series of linear elliptic equations with symmetric operators: three Helmholtz equations for the velocity components, and the Poisson equation for the pressure. This time scheme is conditionally stable, although the condition on the time step to guarantee convergence is not very restrictive. [4]

The spatial approximation implemented in the *Nast++* package, is based on the mortar finite element domain decomposition method. [8,21] We assume that the physical domain Ω is decomposed into several non-overlapping subdomains, $\Omega = \bigcup_i \Omega_i$, in such a way that a non-empty intersection between two subdomains can be either a point, or a curve, or a plane. In the latter case we call the intersection a mortar interface.

Thanks to the weak continuity condition on the interfaces between subdomains, we have more freedom to choose mesh and approximation in each subdomain, *i.e.,* the meshes can be nonmatching on the interfaces, and we can use different finite elements in different subdomains. The *Nast++* package provides the choice between tetrahedral and hexahedral meshes, and in the latter case the mesh can be unstructured or structured, *i.e.* "topologically equivalent" to a Cartesian grid. In any case it is assumed that each mesh is derived using several levels of uniform refinement, from a given coarse mesh.

A formulation of the mortar element method with Lagrange multipliers [7] as applied to an elliptic linear equation gives rise to an algebraic saddle point problem with the matrix

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}, \qquad A = \begin{pmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_m \end{pmatrix} \tag{4}$$

where each matrix A_i corresponds to a subdomain problem. To solve a linear algebraic problem with the matrix (4), an iterative method is implemented, with a block diagonal preconditioner, containing subdomain preconditioners, spectrally equivalent to the matrices A_i. [1,15] To make full use of the multilevel structure of subdomain meshes, we have implemented the multigrid preconditioner [12] for *Nast++*.

3 Program Structure and Numerical Results

A typical *Nast++* main routine has the following structure

```
void main(int argc, char **argv) {
   NS_Equations problem(argv[1]);
   problem.makeBoundaryConditions();
   problem.makeTimeScheme();
   problem.runTimeScheme();
}
```

where `NS_Equations` is a *C++* class, implementing the numerical methods and the corresponding data structure. On the "highest" level of the package we tried to take advantage of this approach as much as possible. For instance, the class `NS_Equations` is derived (inherited) from the template class `Domain<Operator_type>`, which is a base for development of applications. It contains basically the information on the geometrical properties of the problem and is defined approximately as follows:

```
template<Operator_type> class Domain {
protected:
   VECTOR<Geom_subdomain> geom_subdomains_;
   VECTOR<Variable>       variables_;
};
```

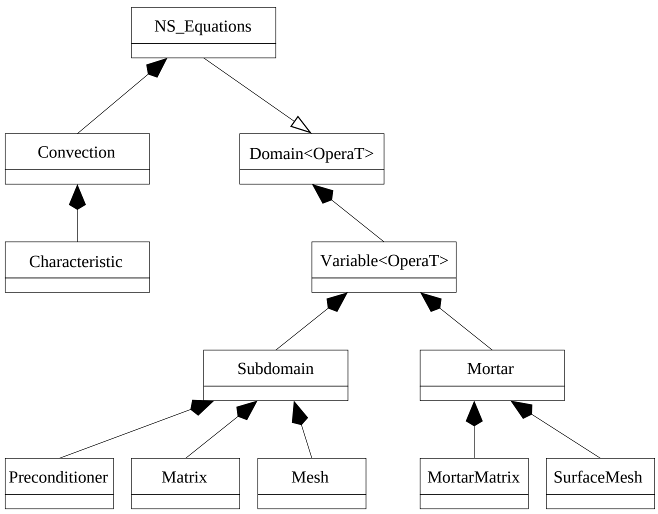


Figure 1. Structure of the *Nast++* package.

The data member `VECTOR<Variable> variables_` indeed represents actual variables of the problem in a "mathematical" sense. For example, in the case of `NS_Equations` class `variables_[0]` represents a velocity component, and `variables_[1]` - a pressure.

The general structure of the *Nast++* package and its basic component are shown on Figure 1. In this section we present some simulation results obtained on prototypical geometries. The main purpose of these simulation is to show that *Nast++* is actually able to drive flows using the average pressure boundary conditions described above and that the subdomain decomposition used does not introduce unphysical discontinuities in the resulting flow. Figure 2 is a model [5] for an aneurism, a local, pathological enlargement of a vessel. The computational domain is split in 3 regions, with one characterized by a high resolution grid in the aneurism region. The adjacent figure shows the flow resulting by an assigned pressure difference between the inlet and the outlet surfaces. The geometry and the in flow, out flow conditions are the same specified in [5], and the resulting flow is, within numerical errors, consistent with what reported there. As can be glimpsed from the image, and proved by more accurate checks, the flow discontinuities are within the numerical approximation errors. The geometry of the second example is described in Figure 3. Here we consider a simplified bifurcation model. Again the geometry is partitioned in several subdomains each with different resolution grids, and, as before, we are using an higher resolution grid in the bifurcation region.

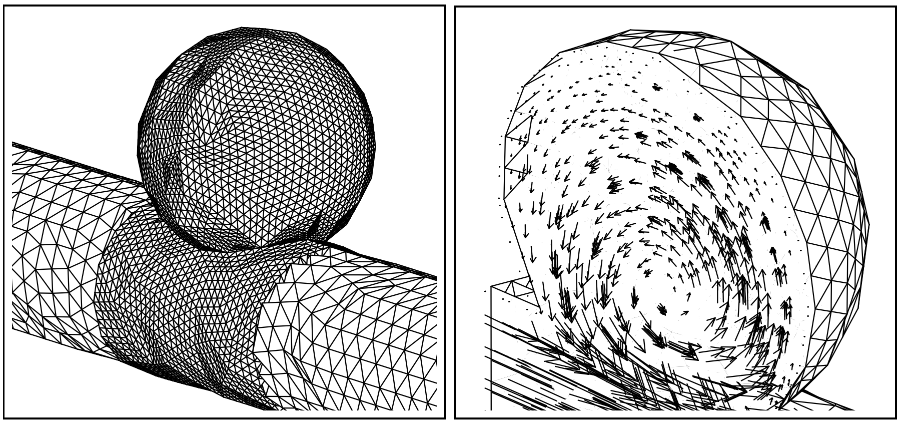


Figure 2. Finite element mesh and velocity profile for the aneurism problem: 3 tetrahedral meshes.

The resulting flow, driven by assigned pressure differences, is consistent with what found in the literature for similar geometries and it does not appear to be affected by the domain decomposition.

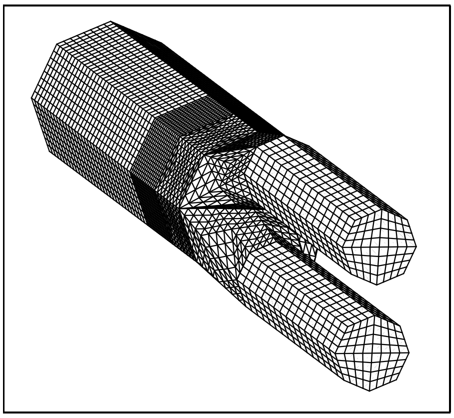
4 Conclusions

We have presented *Nast++* , a package specifically developed for the efficient numerical simulation of blood flow in large arteries. *Nast++* is based on a marriage between an algorithmic framework – the mortar element domain decomposition – and a software technology – object oriented programming – that are well suited to each other. As we hinted in this article, the result is a computationally efficient and flexible tool that is well adapted to the simulation of blood flow in large arteries.

In this paper, we have mainly discussed some of the technological aspects of *Nast++*. In a forthcoming article we will report on its application to the simulation of blood flow in realistic, and clinically relevant, arterial geometries.

5 Acknowledgment

We would like to thank Cristina Manzi for help in grid generation. This research was partially supported by the Sardinian Regional Authorities and by EUREKA project EU 1063.



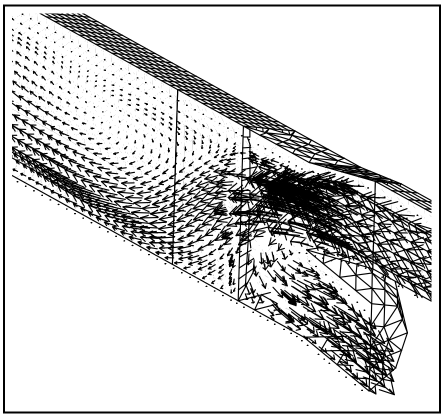


Figure 3. Finite element mesh velocity profile for the bifurcation problem: 4 hexahedral and 1 tetrahedral meshes.

References

1. G. Abdoulaev, Y. Achdou, Y.Kuznetsov and C. Prudhomme, On the parallel implementation of the mortar element method, *Modél. Math. Anal. Numér. (M^2AN)* (to appear).
2. G. Abdoulaev *et al*, *IEEE Transactions on Information Technology in Biomedicine* **2**, 4 (1998).
3. Y. Achdou, G. Abdoulaev, J.-C.Hontand, Y.Kuznetsov, O. Pironneau and C. Prudhomme, Nonmatching grid for fluids, in *Proc. 10th int. Conf. Domain Decompositions Methods, Contemporary Mathematics*, **218** (AMS, 1998).
4. Y. Achdou and J.L. Guermond, *Convergence analysis of a finite element projection/Lagrange-Galerkin method for the incompressible Navier-Stokes equations*, Tech. Report 96-19 (LIMSI CNRS, France, 1996).
5. M. Aenis, A.P. Stancampiano, A.K.Wakhloo and B.B. Lieber, *J. Biomed. Eng.* **119**, 206-212 (1997).
6. C. Begue, C. Conca, F. Murat and O. Pironneau, Les équations de Stokes et de Navier-Stokes avec des conditions aux limites sur la pression, in *Nonlinear Partial Differential Equations and Their Applications, Collége de France Seminar*, eds. H. Brezis and J.Lions, **IX** (Pitman, 1988).
7. F. Ben Belgacem, *The mortar element method with Lagrange multipliers*, Tech. Report (Université Paul Sabatier, Toulouse, 1994).
8. C. Bernardi, Y. Maday and A.T. Patera, A new non conforming approach to domain decomposition:the mortar element method, in *Nonlinear Partial Differential Equations and Their Applications, Collége de France Seminar*, eds. H. Brezis and J.Lions (Pitman, 1993).
9. J. Besson and R. Foerch, *Computer Methods in Applied Mechanics and Engineering* **142**, 1-2 (1997)
10. G. Cokelet, The Rheology and Tube Flow of Blood, in *Handbook of Bioengineering*, eds. R. Skalak and S. Chen (McGraw-Hill, 1987).
11. A.J. Chorin, *Math. Comp.* **23**, 341-353 (1969).
12. M. Griebel, *SIAM J. Sci. Comput.* **15**, 604-620 (1994).
13. J.L. Guermond, *Modél. Math. Anal. Numér. (M^2AN)* **30**, 637-667 (1996).
14. J. Heywood, R. Rannacher and S. Turek, *Int. J. Numer. Meth. Fluids* **22**, 325-352 (1996)
15. Y. Kuznetsov, *Russ. J. Numer. Anal. Math. Modelling* **10**, 187-211 (1995).
16. H.P. Langtangen, *Computational Partial Differential Equations, Numerical Methods and Diffpack Programming*, Lecture Notes in Computational

Science and Engineering, **2** (Springer-Verlag, 1999).
17. D. McDonald, *Blood Flow in Arteries* (Edward Arnold Ltd, London, 1990).
18. O. Pironneau, *Numer. Math.* **38**, 309-332 (1982).
19. R. Rannacher, On Chorin's projection method for the incompressible Navier-Stokes equations, in *Lecture Notes in Mathematics*, **1530** (Springer, Berlin, 1999).
20. M.K. Sharp, G.B. Thurston and J.E. Moore, Jr., *Biorheology* **33**, 3 (1996).
21. P. Le Tallec, T. Sassi and M. Vidrascu, Domains decomposition methods with nonmatching grids, in *Proc. 9th int. Conf. Domain Decompositions Methods* (AMS, 1994).
22. R. Temam, *Bull. Soc. Math. France* **98**, 115-152 (1968).
23. A. Veneziani, *Mathematical and Numerical Modeling of Blood Flow Problems*, Ph. D. Thesis (Universitá degli Studi di Milano, 1998).