INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING
Int. J. Numer. Meth. Biomed. Engng. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cnm.2519

A black-box decomposition approach for coupling heterogeneous components in hemodynamics simulations

Pablo J. Blanco [1,4,*,†], Jorge S. Leiva [2] and Gustavo C. Buscaglia [3,4]

[1] *Laboratório Nacional de Computação Científica, Av. Getúlio Vargas 333, Quitandinha, 25651-075, Petrópolis, Brazil*
[2] *Instituto Balseiro, Av. Bustillo Km. 9.5, San Carlos de Bariloche, 8400, Río Negro, Argentina*
[3] *Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, Av. do Trabalhador Sãocarlense 400, 13560-970, São Carlos, SP, Brazil*
[4] *Instituto Nacional de Ciência e Tecnologia em Medicina Assistida por Computação Científica, Petrópolis, Brazil*

SUMMARY

This work presents a generic and efficient black-box approach for the strong iterative coupling of dimensionally heterogeneous flow models in computational hemodynamics. A heterogeneous model of the cardiovascular system is formed by several vascular black-box components, which are connected through coupling equations. The associated system of equations is solved using the Broyden algorithm. In addition, a multiple time-stepping strategy is introduced to meet different component requirements. The proposed algorithm is employed to split a 3D–1D–0D closed-loop model of the cardiovascular system into corresponding black-box components standing for the 3D (specific vessels), 1D (systemic arteries/peripheral vessels), and 0D (venous/cardiac/pulmonary circulation) components. Examples of application are presented showing the robustness and suitability of this novel approach. Copyright © 2012 John Wiley & Sons, Ltd.

Received 27 February 2012; Revised 16 August 2012; Accepted 15 September 2012

KEY WORDS: strong coupling; cardiovascular system; closed-loop model; multi-scale; blood flow

1. INTRODUCTION

In recent years, several efforts have been directed at integrating different mathematical models of the cardiovascular system (CVS) [1–4]. Hence, the use of dimensionally heterogeneous models, also known as multi-scale analysis [5, 6], has become customary in blood flow simulations [7–15].

In contrast to stand-alone 3D models, dimensionally heterogeneous models allow to incorporate in a natural manner the complex effects of the vasculature present upstream and downstream of an artificially isolated 3D region, that is, to consider the hemodynamics environment posed by the global state of the CVS. In this sense, nonstationary conditions as those encountered when considering regulation mechanisms [11] or analyses in multiple cardiovascular scenarios [12] are clear examples for which the dimensionally heterogeneous modeling provides a natural and consistent approach in order to account for the dynamically evolving environment of each 3D flow model. In these cases, stand-alone 3D models would require boundary data, which has to be somehow provided (i) for all the artificial interfaces generated by the isolation of the 3D geometry and (ii) for the different varying conditions. Either in cases with several 3D artificial inlets/outlets (consider, for instance, the full aorta artery, or multiple 3D models in the same analysis) or in the nonperiodic scenarios mentioned previously, this is nowadays unfeasible.

*Correspondence to: Pablo J. Blanco, Laboratório Nacional de Computação Científica, Av. Getúlio Vargas 333, Quitandinha, 25651-075, Petrópolis, RJ, Brazil.

[†]E-mail: pjblanco@lncc.br

The literature about mathematical models of vascular components is vast and will be referred to in due course. Specifically, we understand the CVS as formed by the coupling of components, each of which accounts for different levels of circulation, for example, local circulation in arterial vessels, systemic circulation, peripheral circulation, and cardiac circulation. These components are, most of the time, associated with black-box codes specifically developed to deal with their own mathematical characteristics.

A black-box partitioned approach to the modeling of the CVS is desirable in view of the following advantages:

(i) no need for full access to the computational codes;
(ii) based on simple input/output relations of well-validated codes;
(iii) impact of nonlinearities of each component bounded to each specific component; and
(iv) time discretization according to each component requirements.

It is worthwhile to exemplify the last two points listed earlier in order to understand the relevance of the black-box approach. Consider that the most expensive component is nonlinear and takes three to four nonlinear iterations to converge in a standard simulation (e.g., the Navier–Stokes equations in the present context). Consider also that this component is coupled to a highly nonlinear component, but far cheaper to be solved, which takes 10–15 iterations to converge (e.g., the 0D model for the cardiac valves in the present context). The monolithic approach forces the 3D model to be solved jointly with the 0D model more times than needed, making the monolithic computation more expensive than the iterative counterpart. A similar situation occurs when choosing the time step. In a monolithic setting, the time step is necessarily the same for all the components. This fact may signify unnecessary expensive computations if the required time step for a 3D model to advance in time is larger than the time step needed by the rest of the (cheaper) components. In contrast to the monolithic approach, in a black-box setting, these restrictions on the time step, which depend upon the time-advancing schemes chosen for each component, do not affect the time step of the coupled problem.

The iterative solution of heterogeneous coupling in computational hemodynamics has been seldom addressed in the literature. The first approaches made use of explicit numerical schemes in time to address the interaction between the heterogeneous models [1, 2, 6, 16]. In these works, coupling strategies are based on explicit methods as well as on implicit Gauss–Seidel-like methods, and eventually in the latter case, subrelaxation is considered. The common feature of those approaches is that it is necessary to define a hierarchy in the network of components regarding boundary data. For instance, assume that we are coupling a 1D model with a 3D model and that we use a Gauss–Seidel method (the same holds for Jacobi or for explicit methods) to solve the problem in an iterative manner. Further, suppose that we decide to impose the pressure as boundary data to the 1D model. This model will give back a certain flow rate that is to be imposed as boundary data on the 3D model, from which a certain pressure is going to be retrieved to be subsequently applied to the 1D model, and so on. Clearly, this way of handling the problem poses restrictions on the nature of boundary data for the different components. This is known in the domain decomposition literature as subdomain coloring and is not a minor issue if we have to couple several components with multiple coupling interfaces each. Besides, assuming that this issue has been somehow solved, recall that explicit methods have poor convergence properties, and the same happens with the Gauss–Seidel-like methods. That is, it is relatively easy to find practical situations in which these methods fail to converge [17].

Recent developments of more sophisticated iterative strategies for the strong coupling of heterogeneous components have been presented in [18]. This has been applied in [9] to the decomposition of 1D networks for the solution of the arterial blood flow and in [17] to the decomposition of hydraulic networks in the context of incompressible fluids and rigid pipes. In the latter work, extensive comparisons between the Gauss–Seidel, Newton–Generalized Minimal Residual method (GMRES), and Broyden methods are presented, from which it is concluded that the Broyden method turns out to be the best choice in terms of computational cost to solve dimensionally heterogeneous flow models.

DOI: 10.1002/cnm

In this work, we develop and test a methodology to address the solution of a dimensionally heterogeneous closed-loop model of the CVS through the iterative solution of stand-alone components, borrowing the ideas from [17]. Specifically, we propose an algorithm based on the Broyden method with the possibility of setting different time steps for the different components. These components stand for blood flow in specific vessels (3D models), wave propagation phenomena in systemic arteries (1D models), and peripheral, venous, cardiac, and pulmonary circulations (0D models) and form a 3D–1D–0D closed-loop model of the CVS.

The core idea of the strategy presented here is the following: components used in the description of the CVS are connected through coupling (interface) equations; the problem is rewritten in terms of interface unknowns; an input/output reasoning is applied to define the character of the interface unknowns for each black-box component; and an iterative method is used to solve the associated system of nonlinear equations, the Broyden method in the current work. The performance and robustness of the proposed methodology are assessed through several examples of application.

The work is organized as follows. In Section 2, the mathematical models are briefly described, and in Section 3, the algorithm for the iterative solution is presented. The application to practical situations is given in Section 4, and further discussions and the final remarks are outlined in Section 5.

2. HETEROGENEOUS MODELING

In this section, we briefly describe the main features of the components used in the composition of the CVS and the interface equations used to couple such components.

2.1. Components in the vascular modeling

The closed-loop circulation is described in terms of three classes of components:

High-pressure (`HP`) *component*: This component is composed of the arterial network and peripheral beds, for which we use 1D and 0D (lumped) representations, respectively. The systemic circulation is modeled as the flow of an incompressible fluid in compliant vessels [19–21]. At each terminal vessel, a Windkessel element is considered to model the peripheral circulation [20]. The topological configuration and data set (based on [22]) are taken from our previous works [1, 9, 10, 12].

Low-pressure (`LP`) *component*: This component closes the cardiovascular circuit incorporating venules, veins, cavas, atria, ventricles, and valves, modeled through 0D models. The venous and pulmonary circulations are modeled using standard analog electric circuit models as in [23]. The four cardiac chambers are introduced following also [23]. The models for the four valves are the ones presented in [24]. The data for the different vascular regions were taken from [23, 24], correspondingly.

Specific-vessel (`SV`) *component*: This component stands for specific vessels of interest in which blood flow is to be assessed in detail such as bifurcations and aneurysms, among others. The blood flow is modeled using the Navier–Stokes equations in deforming domains [25, 26]. The structural behavior is considered through independent ring models [3]. Standard or patient-specific arterial vessels can be considered in the model.

This three-component closed-loop description of the CVS is schematically shown in Figure 1. Hereafter, for notational simplicity, we assume that we have just one `SV` component in the model.

Although we chose to name the components as `HP` and `LP`, these components are not strictly high-pressure and low-pressure components, respectively. Indeed, in the `HP` component, we include all the systemic arteries (high-pressure part) and the peripheral beds, in which the pressure drops to the level of the venous system. Likewise, in the `LP` component, we have the venous system (low-pressure part) and the cardiac chambers, in which the left ventricle raises the pressure to the level of the arterial system. Then, the nomenclature must be understood by virtue of the main characteristic of the components.

DOI: 10.1002/cnm

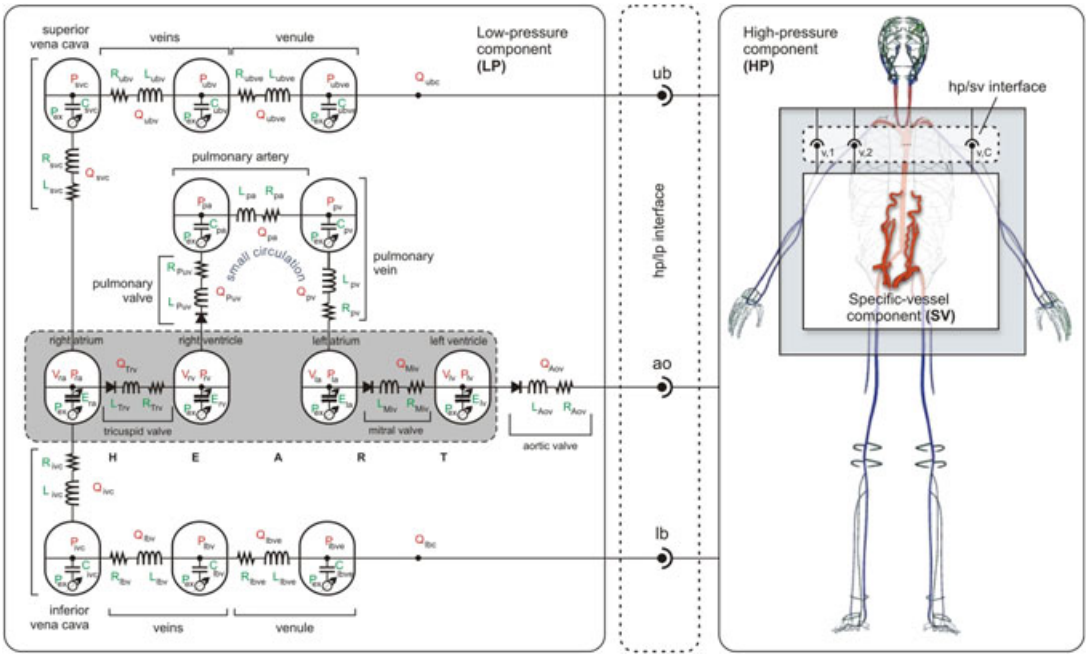

Figure 1. Representation of the CVS through heterogeneous components. The closed-loop is formed by the LP–HP components, whereas in this case, the SV component is embedded in the HP one.

The closed-loop is formed by the LP and HP components through three coupling interfaces. The first is placed at the aortic root right after the aortic valve (ao). The second and third interfaces correspond to the upper (ub) and lower (lb) body parts. The upper and lower body part connections (Figure 1) gather all the corresponding Windkessel terminals from the arterial network (see also [8] for a full description of this model). In the SV component, there are C coupling interfaces shared with the HP component, which correspond to the artificial inlets/outlets caused by the geometrical isolation of the vessel from the rest of the system. Therefore, the total number of coupling interfaces in the network is $N = 3 + C$.

To end this section, recall that the dimensionally heterogeneous coupling entails dealing with defective information for the SV component. In the present work, this defective problem has been tackled using the variational approach presented in [1], which leads to continuity of the normal component of the traction force between the SV and HP components at a given coupling interface, while considering homogeneous the tangent component of the traction force for the SV at those interfaces.

2.2. *Coupling equations and input–output data*

At each coupling interface between two components, we define two degrees of freedom: flow rate Q and pressure P. Mass conservation and force equilibrium hold, then Q and P are continuous quantities across these interfaces. The system is fully described by the vector of interface unknowns $\mathbf{X} = (\mathbf{Q}, \mathbf{P})$, where $\mathbf{Q} = \left(Q_{\mathtt{ao}}, Q_{\mathtt{ub}}, Q_{\mathtt{lb}}, \{Q_{\mathtt{v},i}\}_{i=1}^{C}\right)$ and $\mathbf{P} = \left(P_{\mathtt{ao}}, P_{\mathtt{ub}}, P_{\mathtt{lb}}, \{P_{\mathtt{v},i}\}_{i=1}^{C}\right)$. Component-wise, we identify the following degrees of freedom

$$\begin{aligned} \mathbf{Q}_{\mathtt{HP}} &= \left(Q_{\mathtt{ao}}, Q_{\mathtt{ub}}, Q_{\mathtt{lb}}, \{Q_{\mathtt{v},i}\}_{i=1}^{C}\right), & \mathbf{P}_{\mathtt{HP}} &= \left(P_{\mathtt{ao}}, P_{\mathtt{ub}}, P_{\mathtt{lb}}, \{P_{\mathtt{v},i}\}_{i=1}^{C}\right), \\ \mathbf{Q}_{\mathtt{LP}} &= \left(Q_{\mathtt{ao}}, Q_{\mathtt{ub}}, Q_{\mathtt{lb}}\right) & \mathbf{P}_{\mathtt{LP}} &= \left(P_{\mathtt{ao}}, P_{\mathtt{ub}}, P_{\mathtt{lb}}\right), \\ \mathbf{Q}_{\mathtt{SV}} &= \left(\{Q_{\mathtt{v},i}\}_{i=1}^{C}\right) & \mathbf{P}_{\mathtt{SV}} &= \left(\{P_{\mathtt{v},i}\}_{i=1}^{C}\right). \end{aligned}$$

Our black-box approach relies on the following input/output reasoning for each component: From the component interface unknowns vector $\mathbf{X}_{\mathtt{X}} = (\mathbf{Q}_{\mathtt{X}}, \mathbf{P}_{\mathtt{X}})$, we define those quantities considered as input data $\mathbf{I}_{\mathtt{X}}$ and those considered output data $\mathbf{O}_{\mathtt{X}}$, $\mathtt{X} = \mathtt{HP}, \mathtt{LP}, \mathtt{SV}$. For the SV component,

DOI: 10.1002/cnm

Table I. Input and output data for the different components (see Figure 1 for notation).

Component	
	Input data
HP	$\mathbf{I}_{\mathtt{HP}} = \left(P_{\mathtt{ao}}, P_{\mathtt{ub}}, P_{\mathtt{lb}}, \{P_{\mathrm{v},i}\}_{i=1}^{C}\right)$
LP	$\mathbf{I}_{\mathtt{LP}} = (P_{\mathtt{ao}}, Q_{\mathtt{ub}}, Q_{\mathtt{lb}})$
SV	$\mathbf{I}_{\mathtt{SV}} = (P_{\mathrm{v},1}, P_{\mathrm{v},2}, \ldots, P_{\mathrm{v},C})$
	Output data
HP	$\mathbf{O}_{\mathtt{HP}} = \left(Q_{\mathtt{ao}}, Q_{\mathtt{ub}}, Q_{\mathtt{lb}}, \{Q_{\mathrm{v},i}\}_{i=1}^{C}\right)$
LP	$\mathbf{O}_{\mathtt{LP}} = (Q_{\mathtt{ao}}, P_{\mathtt{ub}}, P_{\mathtt{lb}})$
SV	$\mathbf{O}_{\mathtt{SV}} = (Q_{\mathrm{v},1}, Q_{\mathrm{v},2}, \ldots, Q_{\mathrm{v},C})$

it is well known that the most natural choice for the input data in the present setting is to take Neumann boundary conditions on every inlet/outlet of the domain. For the HP component, our previous experience dictates that the most convenient choice from the performance point of view is to take the pressure as input data [9]. Finally, for the LP component, the natural choice according to the morphology of the component (Figure 1) is to consider as input data the flow rate at the ub and lb connecting points and the pressure at the ao connecting point. These choices are summarized in Table I.

In abstract operator form, the governing equation for component X, $\mathtt{X} = \mathtt{HP}, \mathtt{LP}, \mathtt{SV}$, can be written as $\mathcal{R}_{\mathtt{X}}^{t}(\mathbf{X}_{\mathtt{X}}) = \mathcal{R}_{\mathtt{X}}^{t}(\mathbf{Q}_{\mathtt{X}}, \mathbf{P}_{\mathtt{X}}) = 0$. Reordering the functional dependence, we can write $\mathcal{F}_{\mathtt{X}}^{t}(\mathbf{O}_{\mathtt{X}}, \mathbf{I}_{\mathtt{X}}) = 0$, and in explicit form, each equation reads $\mathbf{O}_{\mathtt{X}} = \mathcal{G}_{\mathtt{X}}^{t}(\mathbf{I}_{\mathtt{X}})$. Hence, solving component X implies obtaining the output $\mathbf{O}_{\mathtt{X}}$ from the given input data $\mathbf{I}_{\mathtt{X}}$. The system of equations to be solved is coupled (Figure 1 and Table I) and can be written as

$$\begin{aligned} &\mathbf{O}_{\mathtt{HP}} - \mathcal{G}_{\mathtt{HP}}^{t}(\mathbf{I}_{\mathtt{HP}}) = 0, \\ &\mathbf{O}_{\mathtt{LP}} - \mathcal{G}_{\mathtt{LP}}^{t}(\mathbf{I}_{\mathtt{LP}}) = 0, \\ &\mathbf{O}_{\mathtt{SV}} - \mathcal{G}_{\mathtt{SV}}^{t}(\mathbf{I}_{\mathtt{SV}}) = 0. \end{aligned} \tag{1}$$

The number of coupling equations in (1) is $2N$, N being the number of coupling interfaces. Recall that the number of interface unknowns is also $2N$. This is because each component provides one equation per interface corresponding to the output data defined at such interface. So two equations are provided in total per interface. Thus, by construction, the system is always closed.

Remark 1

Operators $\mathcal{G}_{\mathtt{X}}^{t}$, $\mathtt{X} = \mathtt{HP}, \mathtt{LP}, \mathtt{SV}$, in (1), are time dependent and nonlinear, and entail dealing with ordinary differential equations, 1D and 3D partial differential equations, correspondingly. Thus, to solve each single component, it is necessary to apply both time discretization and linearization procedures for each component. In the HP and SV components, a spatial discretization is also needed.

2.3. Numerical approximation

In all components, time discretization is accomplished using the Crank–Nicolson method, and nonlinearities are treated using Picard iterations. For the HP component, spatial discretization is performed using finite elements for the equations written along the characteristics [3], whereas for the SV component, the spatial discretization is performed using linear finite elements in space with bubble enrichment for the velocity field [1, 3].

DOI: 10.1002/cnm

3. DECOMPOSITION STRATEGY

3.1. Broyden iterative algorithm

Our approach consists in using the Broyden-like methods to solve (1). Such methods rely on the evaluation of the residual of the system of equations. For a nonlinear equation $\mathcal{R}(x) = 0$, the plain Broyden method reads [27, 28]

Algorithm 1 (Broyden algorithm)

1. Given x^0 and B^0 do
2. Compute $r^0 = \mathcal{R}(x^0)$
3. For $j = 0, 1, \ldots$ until $\frac{\|r^j\|}{\|r^0\|} \leqslant \varepsilon$, or $\|r^j\| \leqslant \varepsilon^{\mathrm{a}}$, do
   4. $w^j = -(B^j)^{-1} r^j$
   5. $x^{j+1} = x^j + w^j$
   6. $r^{j+1} = \mathcal{R}(x^{j+1})$
   7. $B^{j+1} = B^j + \frac{r^{j+1}(w^j)^T}{(w^j)^T w^j}$
8. Enddo

In our context, $x^j = \mathbf{X}^j = (\mathbf{Q}^j, \mathbf{P}^j)$, and the algorithm requires a sufficiently good approximation of the Jacobian matrix as initial matrix B^0, after which a rank 1 update is performed at each iteration (see step 7). The convergence criterion depends on the relative residual (tolerance ε) and on the absolute residual (tolerance ε^{a}). Step 6 (and also the initial step 2) requires the evaluation of the residual r^j, which is

$$r^j = \begin{pmatrix} r^j_{\mathtt{HP}} \\ r^j_{\mathtt{LP}} \\ r^j_{\mathtt{SV}} \end{pmatrix} = \begin{pmatrix} \mathcal{R}^t_{\mathtt{HP}}(x^j) \\ \mathcal{R}^t_{\mathtt{LP}}(x^j) \\ \mathcal{R}^t_{\mathtt{SV}}(x^j) \end{pmatrix} = \begin{pmatrix} \mathbf{O}^j_{\mathtt{HP}} - \mathcal{G}^t_{\mathtt{HP}}\left(\mathbf{I}^j_{\mathtt{HP}}\right) \\ \mathbf{O}^j_{\mathtt{LP}} - \mathcal{G}^t_{\mathtt{LP}}\left(\mathbf{I}^j_{\mathtt{LP}}\right) \\ \mathbf{O}^j_{\mathtt{SV}} - \mathcal{G}^t_{\mathtt{SV}}\left(\mathbf{I}^j_{\mathtt{SV}}\right) \end{pmatrix}. \tag{2}$$

As the problem is time dependent, we have to go through this algorithm at each time step. Then we need to provide the matrix B^0 at each new time step $n+1$, say $B^{0,n+1}$. This is called the initialization of the Broyden algorithm. The strategy followed in the present work is to take $B^{0,n+1} = B^n$, which is equivalent to using a preconditioner for the nonlinear problem [28]. In the present work, at the first time step, the matrix B^0 is initialized with the Jacobian of the system computed by finite differences, that is,

$$B^0_{kl} = \frac{\mathcal{R}_k\left(x^0_l + \delta x^0_l\right) - \mathcal{R}_k\left(x^0_l\right)}{\delta x^0_l} \qquad k, l = 1, \ldots, 2N, \tag{3}$$

where $\mathcal{R}_k$ is the kth component of the vector operator $\mathcal{R}$ and $\delta x^0_l = \epsilon x^0_l$, $l = 1, \ldots, 2N$, with ϵ small enough.

3.2. Inner and outer iterations

The iterations performed by the Broyden algorithm are called *outer iterations* (or the Broyden iterations). Whenever a component is nonlinear, we need to perform the *inner iterations* to assess the residual in each specific equation. For example, given the datum at iteration j, $\mathbf{I}^j_{\mathtt{LP}}$, and the output at the same iteration j, $\mathbf{O}^j_{\mathtt{LP}}$, we have to iterate until we reach the corresponding fixed point in the `LP` component from which we retrieve the corresponding output $\mathcal{G}^t_{\mathtt{LP}}\left(\mathbf{I}^j_{\mathtt{LP}}\right)$, and then we measure the residual as in the second line of (2).

Remark 2
Evaluating the residual in a certain component is a local operation, that is, they do not depend on the rest of the components.

DOI: 10.1002/cnm

3.3. Local and global time stepping

After time discretization is performed, we introduce a global time step Δt_G, which determines the time instants at which the system of equations (1) is satisfied. Independently, we introduce a local time step $\Delta t_{L,\mathtt{X}}$, as the result of the time discretization in each component $\mathtt{X} = \mathtt{HP}, \mathtt{LP}, \mathtt{SV}$. This local time step may depend upon stability requirements of each specific component. Thus, we can weaken the continuity equations by enforcing them at larger time steps, say $\Delta t_G > \Delta t_{L,\mathtt{LP}} = \Delta t_{L,\mathtt{HP}} = \Delta t_{L,\mathtt{SV}}$; or we can have a different time step for different components, say $\Delta t_G = \Delta t_{L,\mathtt{SV}} > \Delta t_{L,\mathtt{HP}} = \Delta t_{L,\mathtt{LP}}$.

3.4. Generic algorithm

According to the previous sections, we set up a generic version of the decomposition strategy discriminating outer and inner iterations as well as global and local time steps.

Algorithm 2 (Broyden-based algorithm for coupling heterogeneous models)

1. Given x^0 and B^0 do
2. For $n = 0, 1, \ldots, n_{\max}$ do (time step: $\Delta t_G = t^{n+1} - t^n$)
   3. Set $x^{0,n+1} = x^n$ and $B^{0,n+1} = B^n$
   4. Evaluate $r^{0,n+1}(x^{0,n+1})$ through Algorithm 3
   5. For $j = 0, 1, \ldots$ until $\frac{\|r^{j,n+1}\|}{\|r^{0,n+1}\|} \leqslant \varepsilon_O$, or $\|r^{j,n+1}\| \leqslant \varepsilon^{\mathrm{a}}$, do
      6. $w^j = -(B^{j,n+1})^{-1} r^{j,n+1}$
      7. $x^{j+1,n+1} = x^{j,n+1} + w^j$
      8. Evaluate $r^{j+1,n+1}(x^{j+1,n+1})$ through Algorithm 3
      9. $B^{j+1,n+1} = B^{j,n+1} + \frac{r^{j+1,n+1}(w^j)^T}{(w^j)^T w^j}$
   10. Enddo
   11. Set $x^{n+1} = x^{j+1,n+1}$ and $B^{n+1} = B^{j+1,n+1}$
12. Enddo

This algorithm is complemented by the following procedure for the evaluation of the residual using substepping and inner nonlinear iterations. Thus, the full assembling of the system of equations is performed through the assembling of the equations corresponding to the local components.

Algorithm 3 (Residual evaluation and system assembling)

1. Given $x^{j,n+1}$ compute $r^{j,n+1}(x^{j,n+1})$ doing
2. For $\mathtt{X} \in \{\mathtt{HP}, \mathtt{LP}, \mathtt{SV}\}$ do
   3. Set $a_{\mathtt{X}} = \frac{\Delta t_G}{\Delta t_{L,\mathtt{X}}}$
   4. For $m_{\mathtt{X}} = \frac{1}{a_{\mathtt{X}}}, \frac{2}{a_{\mathtt{X}}}, \ldots, 1$ do (time step: $\Delta t_{L,\mathtt{X}} = t^{n+m_{\mathtt{X}}} - t^{n+m_{\mathtt{X}}-\frac{1}{a_{\mathtt{X}}}}$)
      5. Compute $r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},0} = \mathcal{R}_{\mathtt{X}}^{j,n+m_{\mathtt{X}},0}(x^{j,n+1})$
      6. For $k_{\mathtt{X}} = 0, 1, \ldots$ until $\frac{\|r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}}\|}{\|r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},0}\|} \leqslant \varepsilon_{I,\mathtt{X}}$, or $\|r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}}\| \leqslant \varepsilon^{\mathrm{a}}$, do
         7. $r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}+1} = \mathcal{R}_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}+1}(x^{j,n+1})$
      8. Enddo
      9. $r_{\mathtt{X}}^{j,n+m_{\mathtt{X}}} = r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}+1}$
   10. Enddo
   11. Assemble the contribution of $r_{\mathtt{X}}^{j,n+1}$ to $r^{j,n+1}$ (see (2))
12. Enddo

For outer iterations, the convergence tolerance is ε_O; and for inner iterations, the convergence tolerance is $\varepsilon_{I,\mathtt{X}}$, $\mathtt{X} = \mathtt{HP}, \mathtt{LP}, \mathtt{SV}$.

DOI: 10.1002/cnm

Figure 2 displays the flow chart for the iterative algorithm featuring the global (coupling) time step, the local (component) time steps, the outer (coupling) iterations, and the inner (component) iterations. In that figure, we considered a hypothetical situation with three components.

A more complex version of the iterative algorithm is achieved by incorporating interpolation between global time steps, that is, replacing Algorithm 3 by the following

Algorithm 4 (Residual evaluation and system assembling with interpolation)

1. Given $x^{j,n+1}$ compute $r^{j,n+1}(x^{j,n+1})$ doing
2. For $\mathtt{X} \in \{\mathtt{HP}, \mathtt{LP}, \mathtt{SV}\}$ do
   3. Set $a_{\mathtt{X}} = \frac{\Delta t_G}{\Delta t_{L,\mathtt{X}}}$
   4. For $m_{\mathtt{X}} = \frac{1}{a_{\mathtt{X}}}, \frac{2}{a_{\mathtt{X}}}, \ldots, 1$ do (time step: $\Delta t_{L,\mathtt{X}} = t^{n+m_{\mathtt{X}}} - t^{n+m_{\mathtt{X}}-\frac{1}{a_{\mathtt{X}}}}$)
      5. Set $x^{j,n+m_{\mathtt{X}}} = f_{L,a_{\mathtt{X}}}(x^n, x^{j,n+1})$
      6. Compute $r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},0} = \mathcal{R}_{\mathtt{X}}^{j,n+m_{\mathtt{X}},0}(x^{j,n+m_{\mathtt{X}}})$
      7. For $k_{\mathtt{X}} = 0, 1, \ldots$ until $\frac{\|r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}}\|}{\|r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},0}\|} \leqslant \varepsilon_{I,\mathtt{X}}$, or $\|r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}}\| \leqslant \varepsilon^{\mathrm{a}}$, do
         8. $r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}+1} = \mathcal{R}_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}+1}(x^{j,n+m_{\mathtt{X}}})$
      9. Enddo
      10. $r_{\mathtt{X}}^{j,n+m_{\mathtt{X}}} = r_{\mathtt{X}}^{j,n+m_{\mathtt{X}},k_{\mathtt{X}}+1}$
   11. Enddo
   12. Assemble the contribution of $r_{\mathtt{X}}^{j,n+1}$ to $r^{j,n+1}$ (see (2))
13. Enddo

In step 5 of Algorithm 4, function $f_{L,a_{\mathtt{X}}}$ provides an estimate of intermediate boundary conditions $x^{j,n+m_{\mathtt{X}}}$ for the components as a function of x^n and $x^{j,n+1}$. With this modification, it is expected to have a progressive variation of the datum for the intermediate local time steps (see step 5 in Algorithm 3).

3.5. Computational efficiency

Unlike classical domain decomposition methods, in this kind of problems, we deal with a distinctive feature: the number of interface unknowns in the problem is relatively small (two interface unknowns per coupling interface). Hence, the computational cost expended in carrying out the steps in Algorithm 2 and in Algorithm 2 is negligible (but for step 7 in Algorithm 3), when compared with the evaluation of the component residuals (step 7 in Algorithm 3). Also, the *message passage* cost is negligible, because the amount of data to be passed to and retrieved from the components is small. Under these considerations, there is almost no overhead, and the computational cost can be expressed only in terms of component iterations. More precisely, the cost of assessing the residual of the most expensive component (step 7 in Algorithm 3) is equivalent to the computational cost of a single outer iteration of the entire problem (steps 6–9 in Algorithm 2 including the message passage). This is why in the examples presented in the next section, we can understand the number of iterations as a measure of the performance of the methodology.

Let us analyze three common examples. Firstly, consider a simulation involving just the HP–LP components for the model of the CVS. In this case, the cost is driven by the HP component, whereas the cost of the LP component and of the subsidiary steps in Algorithm 2 are negligible. Secondly, consider the previous case coupled with a single SV component. Clearly, using the arguments as before, it is found that the computational cost is driven by the SV component. In these two scenarios, no load balancing problems arise. Finally, consider that we have more than one SV component coupled to the HP–LP system. In this situation, some overhead may occur depending upon the size of the different SV components and its relation to the computational resources allocated to each of them. Load balancing may be an issue when performing simulations involving multiple expensive SV components, and in a black-box setting, it has to be mitigated at the pre-processing stage when

DOI: 10.1002/cnm

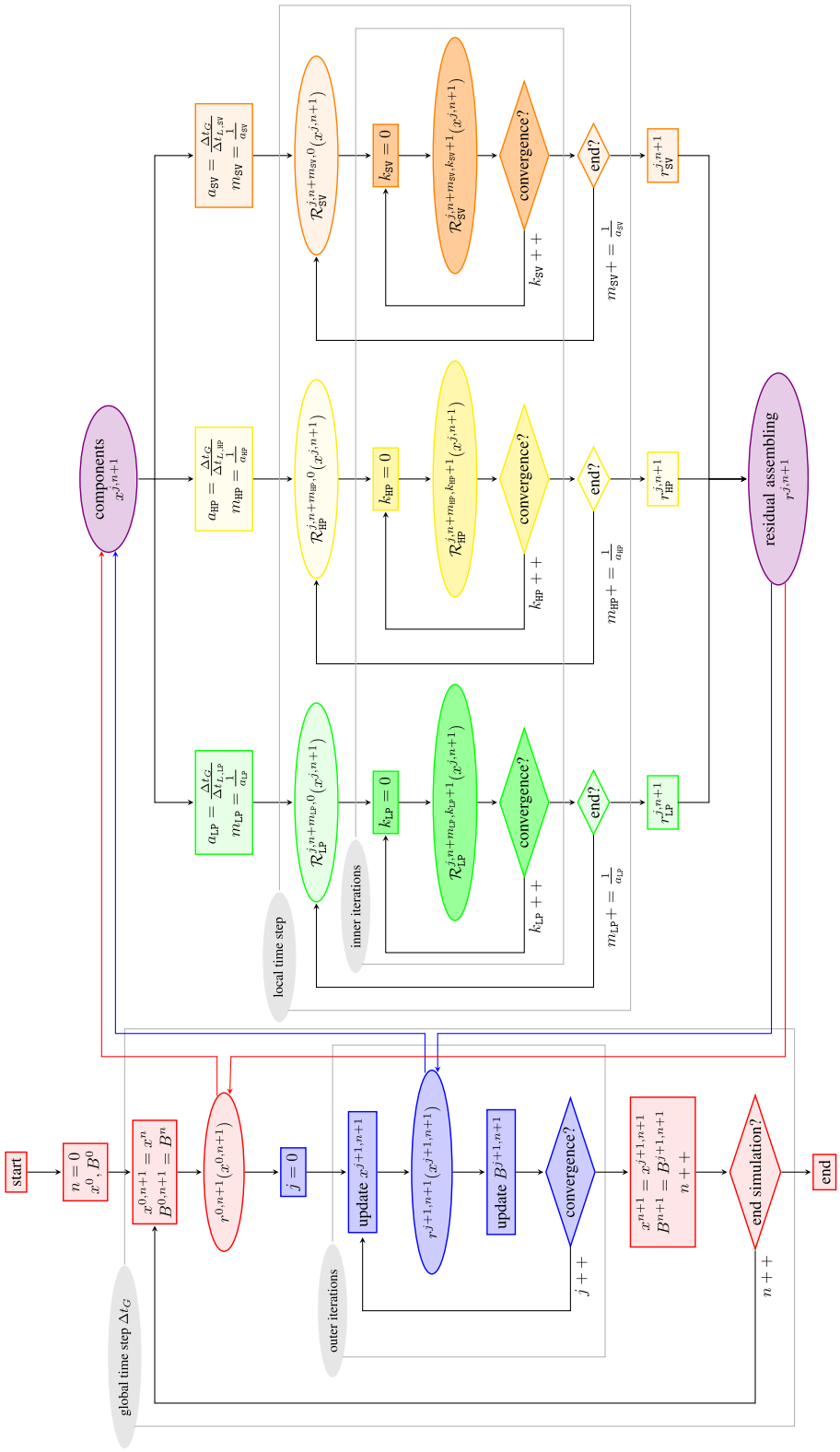

Figure 2. Iterative algorithm featuring global/local time steps and outer/inner iterations.

DOI: 10.1002/cnm

partitioning the network into the basic components and allocating the corresponding computational resources. This essential point is being a matter of current research.

4. HEMODYNAMICS SIMULATIONS

In all the simulations, we use Algorithm 2. Unless stated otherwise, four cardiac cycles were simulated. The convergence tolerances are $\varepsilon_O = 1 \cdot 10^{-6}$, $\varepsilon_{I,\mathtt{X}} = 1 \cdot 10^{-7}$, $\mathtt{X} = \mathtt{HP}, \mathtt{LP}, \mathtt{SV}$, and $\varepsilon^{\mathrm{a}} = 1 \cdot 10^{-14}$. Simulations are characterized by the global time step Δt_G and the local ratios $a_{\mathtt{X}}$, $\mathtt{X} = \mathtt{HP}, \mathtt{LP}, \mathtt{SV}$. In all cases, the cardiac period is $T = 1$s. The number of degrees of freedom in the HP component is 16,776, and in the LP component 46. For the HP and LP components, a direct solver is used to solve the algebraic system of equations, and for the SV component, a GMRES algorithm is used. Recall that the focus of the following applications is the analysis of the performance of Algorithm 2.

In all the examples, we fix the spatial discretization of the components (specifically of the HP and SV components) and perform sensitivity analyses with respect to the global and local time steps. Notice that, unlike classical domain decomposition methods, the number of interface unknowns in the system of equations (2) does not depend on the spatial discretization in the components. As a consequence, the proposed iterative method is insensitive to the number of internal degrees of freedom of each component. A mathematical proof of this result for linear problems is given in [29].

In the present work, we make use of implicit numerical methods for solving the time-advancing equations in the different component. Therefore, there is no CFL-like constraints over the local time steps.

4.1. 1D–0D blood flow modeling in the entire cardiovascular system

The first example is used to explore the iterative strategy for the closed-loop HP–LP system (no SV component is included in the present analysis).

In Tables II and III, the results of the simulations are summarized by combining Algorithms 2 and 3. We have included the case in which we solve the HP–LP system in a monolithic manner (considering $\Delta t = \Delta t_G$ and tolerance $\varepsilon = \varepsilon_O$). Notation $(\cdot)^{(L)}$ indicates that a single Picard inner iteration was performed. This is equivalent to considering the component as a linear one. Observe that this is related to the inner iterations, which are independent from the outer iterations that drive the convergence of the entire system (1). Running the HP component as a linear component is possible; on the contrary, running the LP component as a linear one makes the iterative method to diverge in all tested cases. This phenomenon is caused by the nondifferentiable behavior of cardiac valves around the opening/closing phases. Lack of convergence of the fixed point iterations in valve dynamics turns the LP component unstable, from which the iterative method for the coupled problem fails to converge.

Per cardiac beat, we have N_{ci} as the total number of coupling iterations between the two components, $N_{\mathrm{ss},\mathtt{X}}$ the number of times component X is solved, and $N_{\mathrm{ss},\mathtt{HP-LP}}$ the number of times that the monolithic HP–LP system is solved. A crashed simulation is indicated by '–' (LP inner iterations did not converge in all these cases). Analogously, notation $-^{(L)}$ indicates that the simulation crashed as a consequence of the lack of convergence of the LP component, which is also the same in the case of running the HP component as a linear one..

The main outcomes from Tables II and III are summarized here:

- The coupling iterations per global time step, that is, $\frac{N_{\mathrm{ci}}}{N_{\Delta t_G}}$, are almost insensitive to Δt_G, whereas they are sensitive to the nonlinear/linear character of the HP component.
- There is no relation between the number of times the HP and LP components are solved. The nonlinearities of the latter component are bounded to that component. Furthermore, it is possible to run the HP component as a linear component and reduce computational cost.

DOI: 10.1002/cnm

Table II. Iterative and monolithic solution of the HP–LP system for several combinations of global and local time steps (Δt_G, $a_{\mathtt{HP}}$, and $a_{\mathtt{LP}}$ specified in each case; in the monolithic case $\Delta t = \Delta t_G$).

Time stepping			Iterative solution			Monolithic solution
Δt_G [s]	$a_{\mathtt{HP}}$	$a_{\mathtt{LP}}$	N_{ci} $\left(\frac{N_{\mathrm{ci}}}{N_{\Delta t_G}}\right)$	$N_{\mathrm{ss,HP}}$	$N_{\mathrm{ss,LP}}$	$N_{\mathrm{ss,HP-LP}}$
0.0005	1	1	**8238** (4.12)	21,313	25,707	11,564
	1	1	10,414 (5.21)	**10,414**$^{(L)}$	30,581	
0.0010	1	1	4160 (4.16)	11,779	14,082	6519
	1	1	5912 (5.91)	5912$^{(L)}$	18,271	
	1	2	4071 (4.07)	11,336	46,952	
	1	2	5333 (5.33)	**5333**$^{(L)}$	62,011	
	2	2	**3766** (3.77)	30,954	43,829	
0.0020	1	1	2307 (4.61)	7368	7987	3750
	1	1	3034 (6.07)	3034$^{(L)}$	9585	
	1	2	2409 (4.82)	7324	31,194	
	1	2	3052 (6.10)	3052$^{(L)}$	39,755	
	2	2	**2057** (4.11)	18,217	26,689	
	1	4	2338 (4.68)	7186	53,960	
	1	4	2985 (5.97)	**2985**$^{(L)}$	69,337	
	2	4	2062 (4.12)	18,268	47,788	
	4	4	2063 (4.13)	33,928	47,830	

Per cardiac beat, N_{ci} denotes the total number of coupling iterations among components, $N_{\Delta t_G}$ is the number of global time steps $\left(\frac{T}{\Delta t_G}\right)$, $N_{\mathrm{ss,X}}$ is the number of times component X was solved, and $N_{\mathrm{ss,HP-LP}}$ is the number of times the coupled HP–LP system was solved. Notation $(\cdot)^{(L)}$ implies that the HP component was run as a linear component regarding the inner iterations (in such case $a_{\mathtt{HP}} = 1$ always). The symbol '–' denotes lack of convergence, and notation $(\cdot)^{(*)}$ indicates that the solution is far from the actual solution (because of the time-stepping strategy). For each Δt_G, the cheapest simulation is highlighted in bold.

- The nonlinearities of the LP are responsible for the lack of convergence of the monolithic system at large time steps. Substepping allows to circumvent this issue by setting adequate time steps according to each component requirements.
- Cheaper simulations are obtained using the largest possible time step for the most expensive component (in this case, the HP component), whereas the time step should be reduced in the rest of the components (in this case, the LP component).
- A measure of the computational cost for the solution of the coupled system is given, in this case, by the number of iterations $N_{\mathrm{ss,HP}}$. The physics captured by the different simulations are, roughly speaking, equivalent and independent of the time step. The lack of convergence of the monolithic strategy, which limits the time step to 0.0020 s, is thus just a numerical artifact. The proposed iterative method already reduces the cost from $N_{\mathrm{ss,HP}} = 3750$ to $N_{\mathrm{ss,HP}} = 2962$ for $\Delta t_G = 0.0020$ s (21% reduction, with interpolation; see succeeding discussions), but by exhibiting convergence up to $\Delta t_G = 0.0080$ s, it allows for further reduction of the cost to $N_{\mathrm{ss,HP}} = 1054$ (72% reduction, with interpolation; see succeeding discussions).

For the sake of completeness, we report in Table IV the wall time spent in solving four cardiac cycles by some representative cases taken from Tables II and III. The components run in parallel and communicate through Message Passing Interface (MPI) protocol. Each component was solved sequentially in an Intel(R) Core(TM) i7 Extreme 975 Processor running at 3.33 GHz. For each global time step, it is possible to accelerate the solution of the HP–LP system through the use of the iterative approach without deteriorating the quality of the solution. In Figure 3, a comparison between the iterative and monolithic solutions is performed. Several representative cases are displayed: with equal and different global and local time steps, and also with different time steps and

DOI: 10.1002/cnm

Table III. (Continuation of Table II) Iterative and monolithic solution of the HP–LP system for several combinations of global and local time steps (Δt_G, $a_{\mathtt{HP}}$, and $a_{\mathtt{LP}}$ specified in each case; in the monolithic case $\Delta t = \Delta t_G$).

Time stepping			Iterative solution			Monolithic solution
Δt_G [s]	$a_{\mathtt{HP}}$	$a_{\mathtt{LP}}$	N_{ci} $\left(\frac{N_{\mathrm{ci}}}{N_{\Delta t_G}}\right)$	$N_{\mathrm{ss,HP}}$	$N_{\mathrm{ss,LP}}$	$N_{\mathrm{ss,HP-LP}}$
0.0040	1	1	–	–	–	–
	1	2	1398 (5.59)	5154	20,528	
	1	2	1802 (7.21)	**1802**$^{(L)}$	26,679	
	2	2	1104 (4.41)	12,154	16,358	
	1	4	1418 (5.67)	5215	36,669	
	1	4	–	–$^{(L)}$	–	
	2	4	1080 (4.32)	11,886	28,145	
	4	4	1094 (4.38)	19,507	28,611	
	1	8	1415$^{(*)}$ (5.66)	5174	65,301	
	1	8	1783$^{(*)}$ (7.13)	1783$^{(L)}$	82,664	
	2	8	**1078** (4.31)	11,873	50,257	
	4	8	1083 (4.33)	19,295	50,475	
	8	8	1093 (4.37)	36,017	51,025	
0.0080	1	1	–	–	–	–
	2	2	–	–	–	
	1	4	788 (6.30)	3900	23,210	
	1	4	1156 (9.25)	**1156**$^{(L)}$	34,078	
	2	4	594 (4.75)	8132	17,769	
	4	4	**590** (4.72)	13,008	17,640	
	1	8	794 (6.35)	3893	41,270	
	1	8	1157$^{(*)}$ (9.26)	1157$^{(L)}$	60,083	
	2	8	594 (4.75)	8150	31,377	
	4	8	595 (4.76)	13,129	31,447	
	8	8	595 (4.76)	21,234	31,352	

See notation in Table II.

eliminating the inner iterations for the assessment of the residual in the HP component. Observe that even for quite large time steps, the model predicts the same behavior.

The counterpart to Tables II and III, now considering Algorithm 4, instead of Algorithm 3, with linear interpolation between global time steps, that is, $x^{j,n+m_{\mathtt{X}}} = \theta x^n + (1-\theta)x^{j,n+1}$, $\theta = 1-m_{\mathtt{X}}$, is presented in Tables V and VI for the cases in which $a_{\mathtt{X}} > 1$, $\mathtt{X} = \mathtt{HP}$ or $\mathtt{X} = \mathtt{LP}$.

Notice that all the numbers in the tables presented here are slightly smaller than the numbers reported in Tables II and III. Therefore, we conclude that there is room for gain by exploiting interpolation when performing multiple time-stepping strategies. In average, a reduction up to 10% was obtained in the computational cost when using interpolation. In terms of results, the solution obtained with Algorithms 2–4 is closer to the monolithic solution than the one obtained with Algorithms 2 and 3 when we make use of different time steps for the different components.

4.2. 3D–1D–0D blood flow modeling in a cerebral aneurysm

In this case, we use the same HP–LP system as in Section 4.1 and couple a SV component representing a patient-specific cerebral aneurism located at the middle cerebral artery. Clearly, the computational cost is driven by the SV component for which an adequate time step is $\Delta t_{L,\mathtt{SV}} \leqslant 0.002$. For this application, we limit ourselves to a few number of simulations to compare the performance, because the goal here is to show the applicability in cases of potential medical interest.

The geometry of the aneurism is reconstructed from a set of DICOM (Digital Imaging and Communication in Medicine) images following standard segmentation procedures. The number of degrees of freedom in the SV component is 165,046.

DOI: 10.1002/cnm

Table IV. Wall time taken by the iterative and monolithic approaches to solve the HP–LP system for several combinations of global and local time steps (Δt_G, $a_{\mathtt{HP}}$, and $a_{\mathtt{LP}}$ specified in each case; in the monolithic case $\Delta t = \Delta t_G$). For each Δt_G, the cheapest simulation is highlighted in bold.

Time stepping			Simulation time [s]	
Δt_G [s]	$a_{\mathtt{HP}}$	$a_{\mathtt{LP}}$	Iterative solution	Monolithic solution
0.0005	1	1	4369	2552
	1	1	**2119**$^{(L)}$	
0.0010	1	1	2311	1441
	1	2	2089	
	1	2	**1151**$^{(L)}$	
0.0020	1	1	1448	832
	1	2	1404	
	1	2	615$^{(L)}$	
	1	4	**604**$^{(L)}$	
0.0040	1	2	1006	–
	1	4	**971**	
	2	4	2271	
0.0080	1	4	760	–
	1	4	222$^{(L)}$	
	1	8	706	
	1	8	**216**$^{(L)}$	

Table VII summarizes the simulations performed in this case. The cheapest simulation, given that $a_{\mathtt{SV}} = 1$, is obtained when $a_{\mathtt{HP}} = 2$ and $a_{\mathtt{LP}} = 2$ for both time steps ($\Delta t = 0.001$ and $\Delta t = 0.002$), which is consistent with the conclusions obtained from the previous section.

The number of iterations showed to be insensitive to the multi-stepping strategy used (once Δt_G has been fixed), as concluded from Table VII. Moreover, the convergence history is approximately constant throughout the entire cardiac cycle. For $\Delta t_G = 0.001$, the number of iterations is smaller than for $\Delta t_G = 0.002$, but the computational cost turns to be less in the latter case because the reduction in $\frac{N_{\mathrm{ci}}}{\Delta t_G}$ does not make up for the increase in the number of times per cardiac cycle the SV component has to be solved (see last column in Table VII).

In this case, the measure of the computational cost for the solution of the coupled system is given by the number of iterations of the most expensive component, that is, $N_{\mathrm{ss,SV}}$.

Figure 4 shows the geometry of the patient-specific vessel and the results at all the coupling interfaces. These results correspond to the cases $a_{\mathtt{HP}} = 2$, $a_{\mathtt{LP}} = 2$, and $a_{\mathtt{SV}} = 1$ and to the last cardiac cycle.

Finally, in Figure 5, the local structure of blood flow is presented at three characteristic instants corresponding to the last cardiac cycle.

4.3. 3D–1D–0D blood flow modeling in the arm

This last example is used to show an application in which several SV components are considered within the HP–LP system seen before. These components correspond to the five arterial branchings taking place at one of the arms. Each bifurcation is constructed such that it matches the lumen radii and arterial wall parameters with the corresponding artery in the HP component. The aim here is to show the robustness in the solution process even for a large number of SV components (see Figure 6 for a schematic placement of the bifurcations in the region of interest). In this case, the computational cost should be carefully evaluated because the number of degrees of freedom in the SV components ranges between 85,365 and 136,941, and this can generate problems in the balance of load as already stated in Section 3.5. This is currently a matter of current research.

DOI: 10.1002/cnm

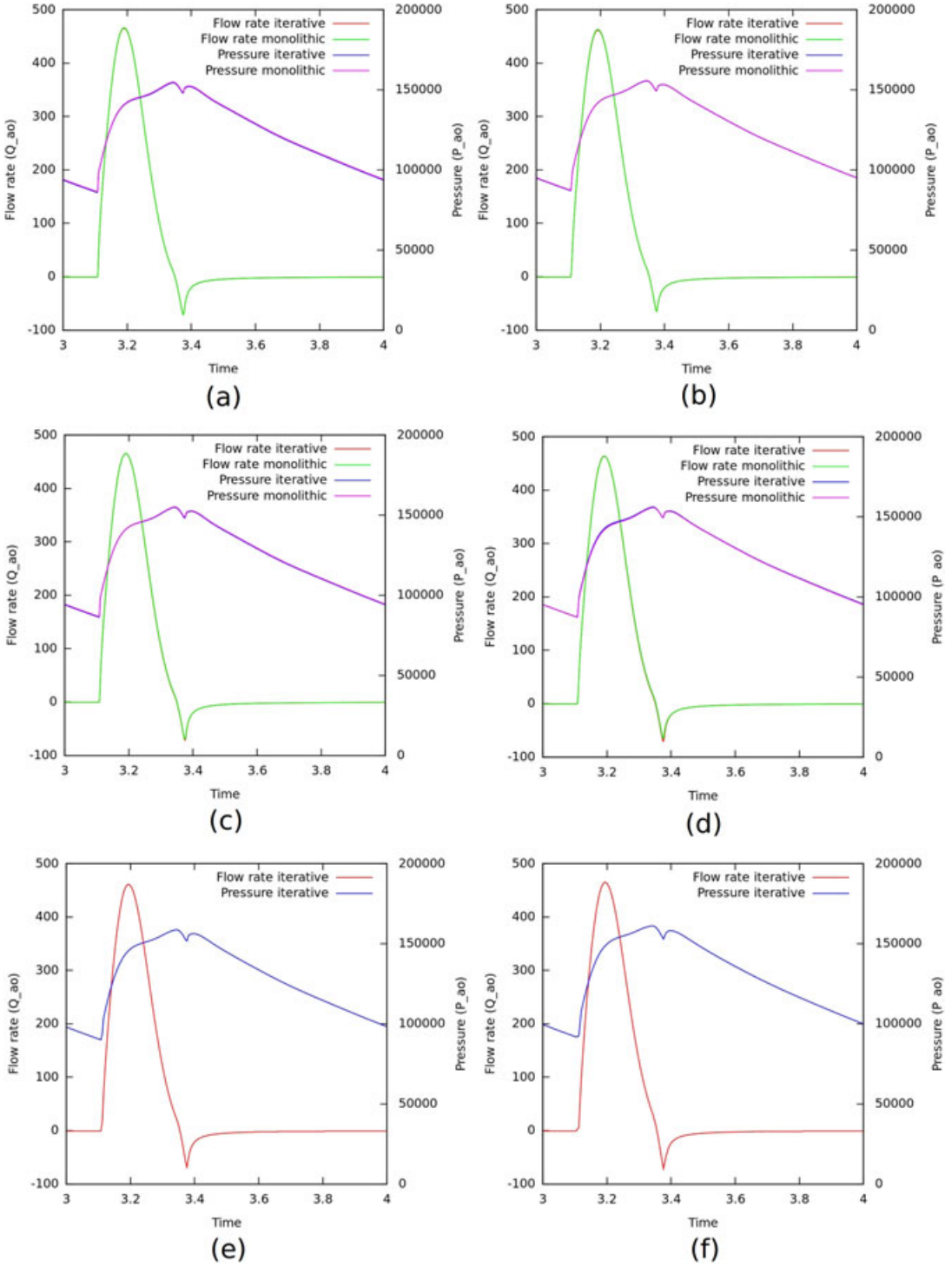

Figure 3. Flow rate and pressure at the aortic root. Comparison between monolithic and iterative solutions for several representative cases taken from Tables II and III (Δt_G, a_{HP} and a_{LP} specified in each case; in the monolithic case $\Delta t = \Delta t_G$). (a) $\Delta t_G = 0.0005$, $a_{\mathrm{HP}} = a_{\mathrm{LP}} = 1$; (b) $\Delta t_G = 0.0020$, $a_{\mathrm{HP}} = a_{\mathrm{LP}} = 1$; (c) $\Delta t_G = 0.0010$, $a_{\mathrm{HP}} = 1$, $a_{\mathrm{LP}} = 2$; (d) $\Delta t_G = 0.0020$, $a_{\mathrm{HP}} = 1^{(L)}$, $a_{\mathrm{LP}} = 4$; (e) $\Delta t_G = 0.0040$, $a_{\mathrm{HP}} = 1$, $a_{\mathrm{LP}} = 2$; (f) $\Delta t_G = 0.0080$, $a_{\mathrm{HP}} = 1^{(L)}$, $a_{\mathrm{LP}} = 4$. See notation in Table II.

We consider $\Delta t_G = 0.001$ and $a_{\mathrm{HP}} = a_{\mathrm{LP}} = a_{\mathrm{SV}} = 1$. In Figure 6, the results in terms of pressure and flow rate at the inlets of the SV components are shown. In turn, Figure 7 shows the

DOI: 10.1002/cnm

Table V. Iterative and monolithic solution of the HP–LP system for several combinations of global and local time steps considering interpolation (Δt_G, $a_{\mathtt{HP}}$, and $a_{\mathtt{LP}}$ specified in each case; in the monolithic case $\Delta t = \Delta t_G$).

Time stepping			Iterative solution with linear interpolation		
Δt_G [s]	$a_{\mathtt{HP}}$	$a_{\mathtt{LP}}$	N_{ci} ($\frac{N_{\mathrm{ci}}}{N_{\Delta t_G}}$)	$N_{\mathrm{ss,HP}}$	$N_{\mathrm{ss,LP}}$
0.0010	1	2	3808 (3.81)	10,540	44,066
	1	2	5250 (5.25)	**5250**$^{(L)}$	60,920
	2	2	**3563** (3.56)	29,290	41,361
0.0020	1	2	2302 (4.60)	7115	29,938
	1	2	2998 (6.00)	2998$^{(L)}$	39,116
	2	2	**2039** (4.08)	18,052	26,494
	1	4	2190 (4.38)	6981	50,800
	1	4	2962 (5.92)	**2962**$^{(L)}$	68,778
	2	4	2015 (4.03)	17,866	46,802
	4	4	2019 (4.04)	33,156	46,905

See notation in Table II.

Table VI. (Continuation of Table V) Iterative and monolithic solution of the HP–LP system for several combinations of global and local time steps considering interpolation (Δt_G, $a_{\mathtt{HP}}$, and $a_{\mathtt{LP}}$ specified in each case; in the monolithic case $\Delta t = \Delta t_G$).

Time stepping			Iterative solution with linear interpolation		
Δt_G [s]	$a_{\mathtt{HP}}$	$a_{\mathtt{LP}}$	N_{ci} $\left(\frac{N_{\mathrm{ci}}}{N_{\Delta t_G}}\right)$	$N_{\mathrm{ss,HP}}$	$N_{\mathrm{ss,LP}}$
0.0040	1	2	–	–	–
	1	2	–	–$^{(L)}$	–
	2	2	–	–	–
	1	4	1307$^{(*)}$ (5.23)	4876	33,910
	1	4	1662$^{(*)}$ (6.65)	1662$^{(L)}$	43,282
	2	4	1072 (4.29)	11,780	27,972
	4	4	1066 (4.27)	18,972	27,957
	1	8	1305$^{(*)}$ (5.22)	4852	60,033
	1	8	1691$^{(*)}$ (6.76)	1691$^{(L)}$	78,673
	2	8	**1061** (4.24)	**11,656**	49,369
	4	8	1062 (4.25)	18,884	49,452
	8	8	1062 (4.25)	34,947	49,494
0.0080	2	2	–	–	–
	1	4	760 (6.08)	3684	22,341
	1	4	1064 (8.51)	1064$^{(L)}$	31,401
	2	4	567 (4.54)	7764	16,866
	4	4	576 (4.61)	12,691	17,182
	1	8	746 (5.97)	3578	38,751
	1	8	1054 (8.43)	**1054**$^{(L)}$	54,984
	2	8	**560** (4.48)	7671	29,337
	4	8	567 (4.54)	12,491	29,822
	8	8	571 (4.56)	20,356	29,991

See notation in Table II.

convergence history in the last cardiac cycle and the change in the pressure wave at the inlets of the SV components as we move forward in the distal direction. Even though the system of nonlinear equations comprises 36 interface unknowns, the iterative algorithm takes just four to six iterations (occasionally seven and eight) to converge at each time step.

DOI: 10.1002/cnm

Table VII. Iterative solution of the HP–LP–SV system for some combinations of global and local time steps.

Time stepping				Iterative solution			
Δt_G [s]	a_{HP}	a_{LP}	a_{SV}	N_{ci} $\left(\frac{N_{\mathrm{ci}}}{N_{\Delta t_G}}\right)$	$N_{\mathrm{ss,HP}}$	$N_{\mathrm{ss,LP}}$	$N_{\mathrm{ss,SV}}$
0.0010	1	1	1	4239 (4.24)	14,044	13,912	11,130
	1	2	1	4289 (4.29)	14,089	49,612	11,135
	2	2	1	**4237** (4.24)	34,845	49,024	**11,110**
0.0020	1	1	1	2514 (5.03)	9719	8287	7517
	1	2	1	2549 (5.10)	9854	33,019	7610
	2	2	1	**2475** (4.95)	22,180	32,085	**7432**

See notation introduced in Tables II and III.

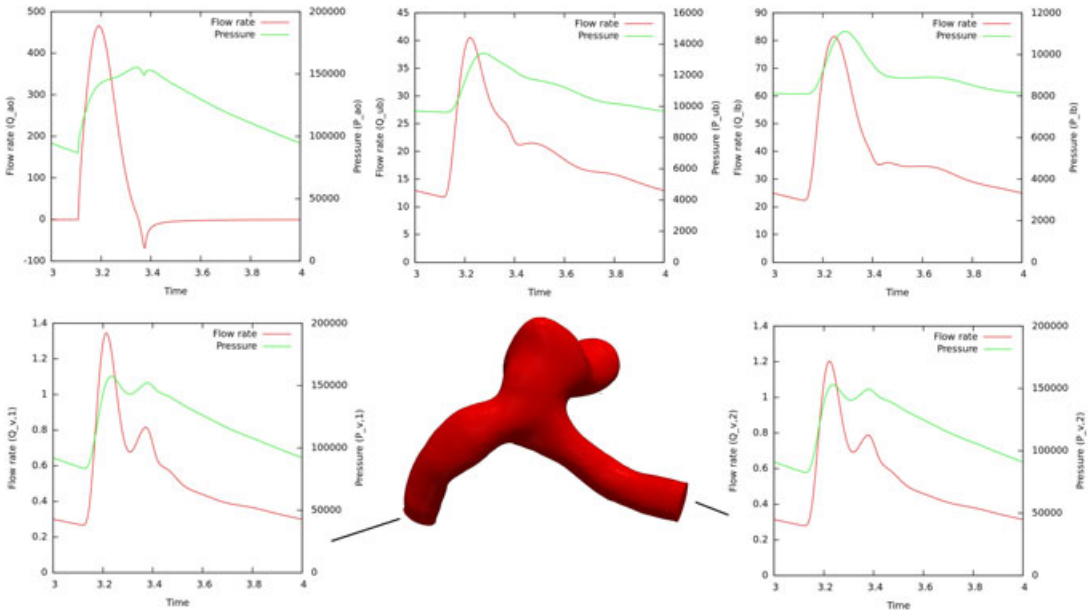

Figure 4. Aneurysm geometry and results (pressure and flow rate) at coupling interfaces. Units are dyn/cm^2 for pressure and cm^3/s for flow rate.

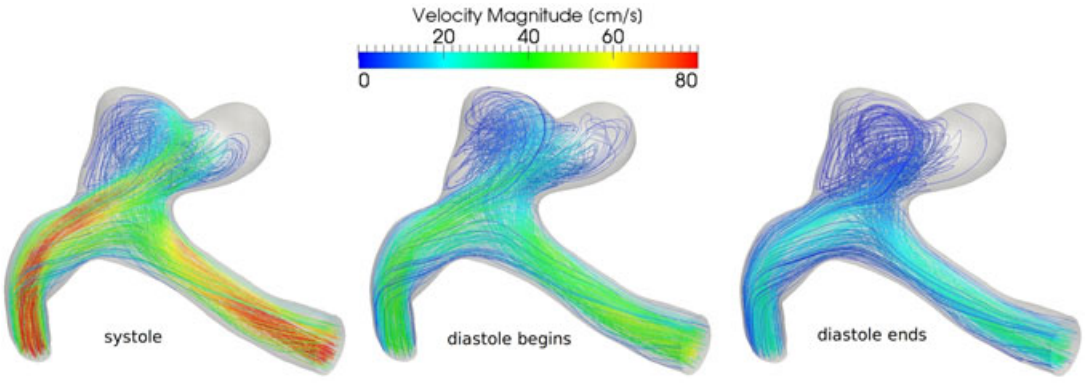

Figure 5. Local blood flow in the aneurysm visualized through streamlines at three instants within the cardiac cycle.

DOI: 10.1002/cnm

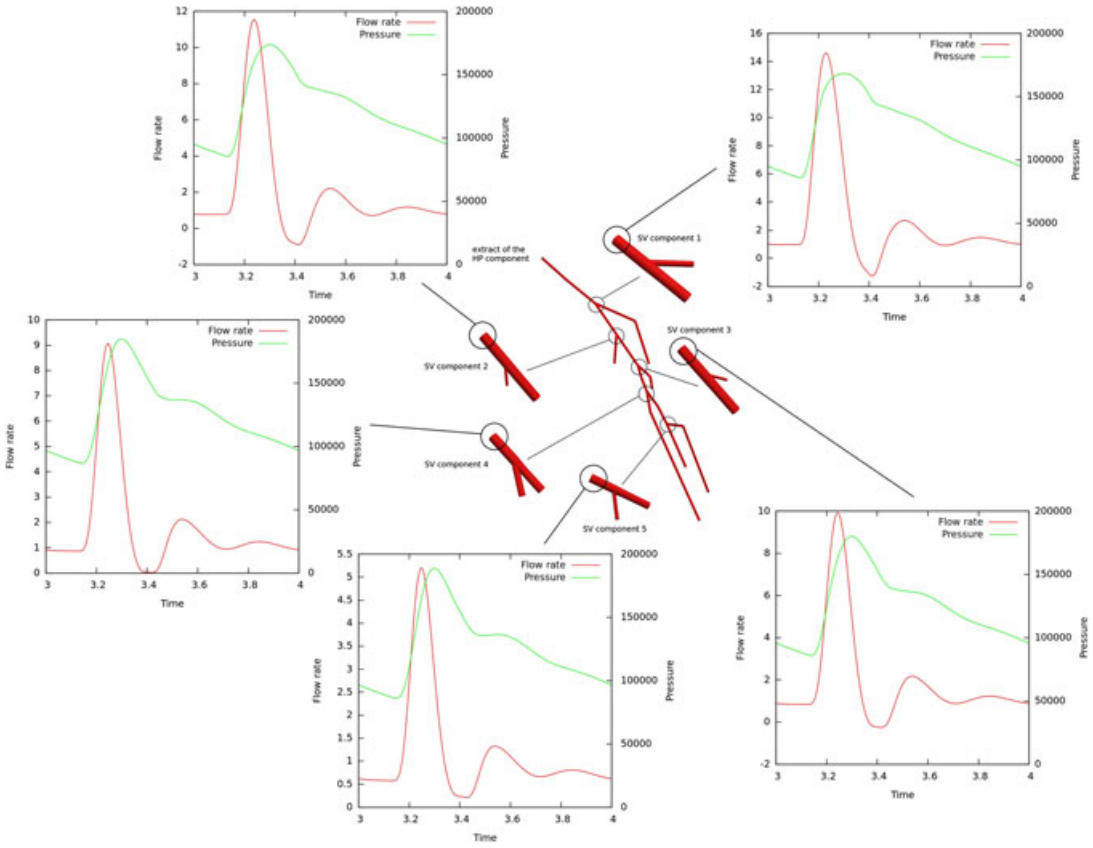

Figure 6. Embedding of five SV components to replace arterial branches of the arm in the HP component. Results, pressure, and flow rate, at coupling interfaces, are displayed. Units are dyn/cm^2 for pressure and cm^3/s for flow rate.

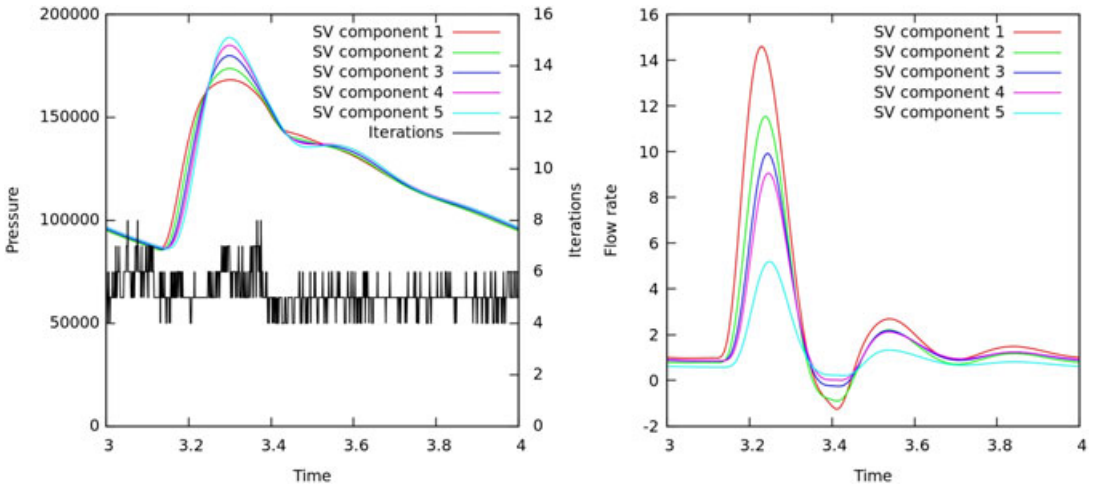

Figure 7. Convergence history for the blood flow simulation in the arm, and results at the inlets for all the SV components. Pressure units are dyn/cm^2 and flow rate units are cm^3/s.

Finally, Figure 8 presents the velocity magnitude inside each bifurcation at some time instants within the last cardiac cycle. Notice that each SV component is embedded in its corresponding hemodynamics environment, obtained from the closed-loop interaction provided by the HP–LP

DOI: 10.1002/cnm

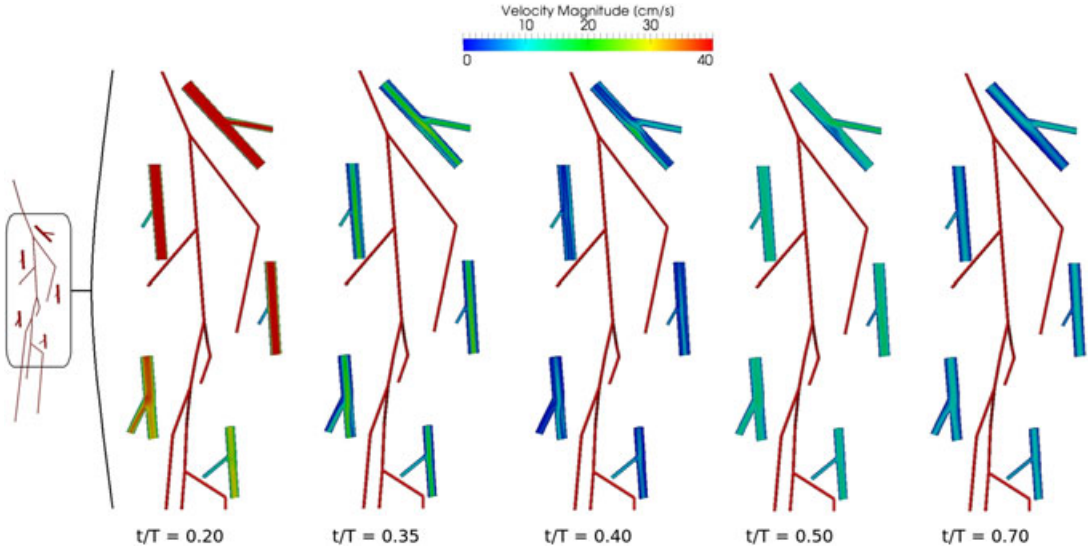

Figure 8. Slices showing the blood flow in the SV components at different time instants.

components. Clearly, a Womersley-like velocity develops after systole as a result of the inversion of the pressure gradient, and then the flow stabilizes its direction during diastole (see also Figure 6). In the fifth SV component, this phenomenon is less evident, because back flow is not present after systole as in the rest of the SV components (Figure 7).

This kind of application motivates the use of heterogeneous models in order to account for the effect of bifurcations (SV components) in the flow rate–pressure relation, for which simple and reliable models based on correlations are not available.

5. FINAL REMARKS

Let us now explain the idea in more general terms, so as to elucidate its wide applicability. Suppose you have a collection of systems (hydraulic, thermal, mechanical, electrical, etc.), each of which you model with a black-box code. Assume, further, that you control this code by a set of inputs at its boundaries and that each input of the set has some 'conjugate' or 'associated' output. This conjugacy between an input (**I**) and an output (**O**) is to be understood in the following sense: At each boundary, the black-box code allows you to impose the value of either **I** or **O**, but not both. Also, having imposed the value of **I** (pressure, temperature, displacement, voltage, etc.), the code provides, as an output, the value of **O** (flow rate, heat flux, force, current, etc.); and vice versa. For each model, you have a preferred set of inputs, which are easier to impose, or for which you have tuned some numerical parameters.

Assume now that you decide to *couple* these systems together into a larger system of which the aforementioned black-box models are *components*. In formal terms, to couple a set of components means that *the input and output values of component* `X` *at some boundary are required to equal some corresponding input/output values of another component* `Y`, which is happening simultaneously for all of the coupling interfaces in the system. *Solving the coupled system* amounts to finding the value of each input of each black-box component, such that by imposing these values and running all component codes, the output of each code at each interface matches the corresponding input/output variable of the neighbor component with which that interface is shared.

The essential point in the present approach is to regard the vector consisting of *all* the coupling variables of *all* interfaces as *interface unknowns* (global unknowns). Getting rid of the point of view in which, component-wise, inputs are data and outputs are results allows us to write the set of equations (1) in terms of inputs and outputs of each code, both being interface unknowns of the global coupled system.

DOI: 10.1002/cnm

Let us go back to the system of equations (1) and elaborate on a more intuitive interpretation. Assume we have a candidate solution vector $\mathbf{X}$. Some of the entries in $\mathbf{X}$ correspond to the input variables of component HP, for example. The rather abstract operator $\mathcal{G}^t_{\mathtt{HP}}$ in the first equation of (1) tells us, simply, 'run the code of component HP and compute its outputs'. These outputs also correspond to specific entries in $\mathbf{X}$, and that equation tells us to compare the values in $\mathbf{X}$ with those yielded by the HP code. If they are the same, the first equation of (1) is satisfied, and if this happens for all the components, the candidate vector $\mathbf{X}$ is indeed a solution of the *coupled system*. If not, a *residual* will appear from the set of equations, by simply subtracting values. This residual, or better the procedure for building it so described, corresponds to Algorithm 3 (condensed in step 8 from Algorithm 2) and is the only system-dependent step of the procedure proposed here. The rest is numerical technology for solving systems of equations having access just to the residuals, and the Broyden algorithm is not the only choice but instead the one that has proved to be more robust and effective in the (many) tests that we have conducted.

Our whole contribution, thus, is already contained in equations (1). In a nutshell, it amounts to separating the 'input versus output' reasoning, which we use to build the equations of each component, from the 'datum versus result' reasoning, which is overruled once the system is coupled, both inputs and outputs being, simply, *interface unknowns*.

Specifically, this black-box decomposition strategy allowed us to solve efficiently problems in the hemodynamics field, which consisted in the coupling of heterogeneous components. Its applicability to solve a closed-loop model of the CVS has been demonstrated through several detailed examples. Partitioned simulations with multi-time-stepping technique allowed us to reduce computational time by setting proper time steps for the different components. Also, we observed that even increasing the number of coupling interfaces, the performance of the iterative method remained bounded to a few number of coupling iterations resulting in high scalability with respect to the number of components in the system.

ACKNOWLEDGEMENTS

The first author and the third author acknowledge the support of the Brazilian agencies CNPq, FAPERJ, and FAPESP.

REFERENCES

1. Blanco P, Feijóo R, Urquiza S. A unified variational approach for coupling 3D–1D models and its blood flow applications. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:4391–4410.
2. Formaggia L, Gerbeau J, Nobile F, Quarteroni A. On the coupling of 3D and 1D Navier–Stokes equations for flow problems in compliant vessels. *Computer Methods in Applied Mechanics and Engineering* 2001; **191**:561–582.
3. Urquiza S, Blanco P, Vénere M, Feijóo R. Multidimensional modelling for the carotid artery blood flow. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:4002–4017.
4. Vignon-Clementel I, Figueiroa C, Jansen K, Taylor C. Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure waves in arteries. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3776–3996.
5. Formaggia L, Nobile F, Quarteroni A, Veneziani A. Multiscale modelling of the circulatory system: a preliminary analysis. *Computing and Visualization in Science* 1999; **2**:75–83.
6. Quarteroni A, Veneziani A. Analysis of a geometrical multiscale model based on the coupling of PDE's and ODE's for blood flow simulations. *SIAM Multiscale Modeling and Simulation* 2003; **1**:173–195.
7. Balossino R, Pennati G, Migliavacca F, Formaggia L, Veneziani A, Tuveri M, Dubini G. Computational models to predict stenosis growth in carotid arteries: which is the role of boundary conditions? *Computer Methods in Biomechanics and Biomedical Engineering* 2009; **12**:113–123.
8. Blanco P, Feijóo R. The role of the variational formulation in the dimensionally-heterogeneous modelling of the human cardiovascular system. In *Modeling of Physiological Flows*, Ambrosi D, Quarteroni A, Rozza G (eds). Springer: Italy, 2012; 251–288.
9. Blanco P, Leiva J, Feijóo R, Buscaglia G. Black-box decomposition approach for computational hemodynamics: one-dimensional models. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**:1389–1405.
10. Blanco P, Pivello M, Urquiza S, Feijóo R. On the potentialities of 3D–1D coupled models in hemodynamics simulations. *Journal of Biomechanics* 2009; **42**:919–930.
11. Blanco P, Trenhago P, Fernandes L, Feijóo R. On the integration of the baroreflex control mechanism in a heterogeneous model of the cardiovascular system. *International Journal for Numerical Methods in Biomedical Engineering* 2012; **28**:412–433.

DOI: 10.1002/cnm

12. Blanco P, Urquiza S, Feijóo R. Assessing the influence of heart rate in local hemodynamics through coupled 3D–1D–0D models. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **26**:890–903.
13. Grinberg L, Anor T, Madsen J, Yakhot A, Karniadakis G. Large-scale simulation of the human arterial tree. *Clinical and Experimental Pharmacology and Physiology* 2009; **36**:194–205.
14. Grinberg L, Cheever E, Anor T, Madsen J, Karniadakis G. Modeling blood flow circulation in intracranial arterial networks: a comparative 3D/1D simulation study. *Annals of Biomedical Engineering* 2011; **39**:297–309.
15. Kim H, Vignon-Clementel I, Figueroa C, LaDisa J, Jansen K, Feinstein J, Taylor C. On coupling a lumped parameter heart model and a three-dimensional finite element aorta model. *Annals of Biomedical Engineering* 2009; **37**:2153–2169.
16. Quarteroni A, Ragni S, Veneziani A. Coupling between lumped and distributed models for blood flow problems. *Computing and Visualization in Science* 2001; **4**:111–124.
17. Leiva J, Blanco P, Buscaglia G. Partitioned analysis for dimensionally-heterogeneous hydraulic networks. *SIAM Multiscale Modeling and Simulation* 2011; **9**:872–903.
18. Leiva J, Blanco P, Buscaglia G. Iterative strong coupling of dimensionally-heterogeneous models. *International Journal for Numerical Methods in Biomedical Engineering* 2010; **81**:1558–1580.
19. Hughes T. On the one-dimensional theory of blood flow in the larger vessels. *Mathematical Biosciences* 1973; **18**:161–170.
20. Stergiopulos N, Young D, Rogge T. Computer simulation of arterial flow with applications to arterial and aortic stenoses. *Journal of Biomechanics* 1992; **25**:1477–1488.
21. Wang J, Parker K. Wave propagation in a model of the arterial circulation. *Journal of Biomechanics* 2004; **37**:457–470.
22. Avolio A. Multi–branched model of the human arterial system. *Medical & Biological Engineering & Computing* 1980; **18**:709–718.
23. Liang F, Takagi S, Himeno R, Liu H. Multi-scale modeling of the human cardiovascular system with applications to aortic valvular and arterial stenoses. *Medical & Biological Engineering & Computing* 2009; **47**:743–755.
24. Korakianitis T, Shi Y. Numerical simulation of cardiovascular dynamics with healthy and diseased heart valves. *Journal of Biomechanics* 2006; **39**:1964–1982.
25. Cebral J, Yim P, Lohner R, Soto O, Choyke P. Blood flow modeling in carotid arteries with computational fluid dynamics and MR Imaging. *Academic Radiology* 2002; **9**:1286–1299.
26. Taylor C, Hughes T, Zarins C. Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering* 1998; **158**:155–196.
27. Dennis J, Schnabel R. *Numerical methods for unconstrained optimization and nonlinear equations*, Classics in Applied Mathematics. SIAM: Philadelphia, 1996.
28. Kelley C. *Iterative methods for linear and nonlinear equations*, Frontiers in Applied Mathematics. SIAM: Philadelphia, 1995.
29. Blanco P, Discacciati M, Quarteroni A. Modeling dimensionally-heterogeneous problems: analysis, approximation and applications. *Numerische Mathematik* 2011; **119**:299–335.

DOI: 10.1002/cnm