Accepted Manuscript

Novel semi-implicit, locally conservative Galerkin (SILCG) methods: Application to blood flow in a systemic circulation

Hayder M. Hasan, Alberto Coccarelli, Perumal Nithiarasu

PII: S0045-7825(16)31288-9
DOI: https://doi.org/10.1016/j.cma.2017.12.017
Reference: CMA 11709

To appear in: *Comput. Methods Appl. Mech. Engrg.*

Received date : 2 October 2016
Revised date : 6 December 2017
Accepted date : 12 December 2017

Please cite this article as: H.M. Hasan, A. Coccarelli, P. Nithiarasu, Novel semi-implicit, locally conservative Galerkin (SILCG) methods: Application to blood flow in a systemic circulation, *Comput. Methods Appl. Mech. Engrg.* (2017), https://doi.org/10.1016/j.cma.2017.12.017

Novel semi-implicit, locally conservative Galerkin (SILCG) methods: Application to blood flow in a systemic circulation☆

Hayder M. Hasan[a,b], Alberto Coccarelli[a], Perumal Nithiarasu[a,*]

[a]*Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea SA2 8PP, United Kingdom.*
[b]*Shatra Technical Institute, Southern Technical University, Ministry of Higher Education and Scientific Research, Iraq.*

Abstract

Three novel, locally conservative Galerkin (LCG) methods in their semi-implicit form are proposed for 1D blood flow modelling in arterial networks. These semi-implicit discretizations are: the second order Taylor expansion (SILCG-TE) method, the streamline upwind Petrov-Galerkin (SILCG-SUPG) procedure and the forward in time and central in space (SILCG-FTCS) method. In the LCG method, enforcement of the flux continuity condition at the element interfaces allows to solve the discretized system of equations at element level. For problems with a large number of degrees of freedoms, this offers a significant advantage over the standard continuous Galerkin (CG) procedure. The well established fully explicit LCG method is used for assessing the accuracy of the proposed new methods. Results presented in this work demonstrate that the proposed SILCG methods are stable and as accurate as the explicit LCG method. Among the three methods proposed, the SILCG-FTCS method requires considerably lower number of iterations per element, and thus requires lowest amount of CPU time. On the other hand, the SILCG-TE and SILCG-SUPG methods are stable and accurate for larger time step sizes. Although the standard Newton method requires evaluation of both the Jacobian matrix and the residual for every single iteration, which may be expensive for standard implicit solvers, the computed results show that the maximum number of iterations per element for SILCG-TE and SILCG-SUPG is less than unity (less than 0.3 and 0.7 respectively). Also, numerical experiments show that

*Corresponding author
Email address: P.Nithiarasu@Swansea.ac.uk (Perumal Nithiarasu)

the Jacobian matrix can be calculated only once per time step, allowing to save a significant amount of computational time.

Keywords: Semi-implicit, locally conservative Galerkin (SILCG) methods, SILCG-TE, SILCG-SUPG and SILCG-FTCS methods, elastic tubes, systemic circulation, arterial flow

1. Introduction

One dimensional flow modelling is a popular approach for computing arterial flow in the human systemic circulation and it represents a very useful tool for studying clinical problems, such as detection of aneurysms [1, 2], fractional flow reserve [3, 4] and other problems of societal and practical relevance [5]. In the last three decades a large number of one-dimensional models has been proposed for predicting the blood pressure and flow along the expanded arterial trees. In some specific cases with simplified assumptions such as axisymmetric velocity profiles, the solution has been calculated in an analytical way by means of a Fourier technique [6]. This method has been adopted in other studies [7, 8, 9], and has also been combined with one-dimensional electrical analogy [10]. However, such approaches have not accounted for non-linear effects [11, 12, 13]. The non-linear governing equations have been solved using the Finite Difference method [14, 15, 16], system of characteristics [17], Galerkin Least Squares [18], Discontinuous Galerkin methods [19, 20, 13] and Finite Volume methods [21, 22]. Some of the notable studies in this area include [7, 23]. Recently, much effort has been also made to analyse blood flow in anatomically very detailed networks [24]. With regard to the numerical strategy, there is no a clear evidence exists on the most suitable method for dealing with one dimensional blood flow modelling. The algorithms available in the literature have also been generally classified according to the time integration scheme adopted. In a fully explicit scheme, the time step must be selected in such a way that the CFL condition is satisfied. This is also the case for prescribing the required boundary conditions [25, 1]. It was demonstrated in [26] that some explicit methods may represent, in terms of accuracy, a valuable alternative to implicit schemes. Implicit methods have become very popular as they do not have restriction on the time step. However the accuracy of the transient solution may be compromised if a significantly larger time step than the one allowed by CFL condition is combined with a low order time discretization. In implicit methods, difficulties related to their implementation may arise, due to the high non-linearity of the governing equations. The design of the iterative numer-

ical procedure for computing the solution is crucial, as it may lead to inefficient schemes [27]. In addition, with an implicit method, a realistic representation of the complex boundary conditions encountered in systemic circulation problems is often difficult. For example, the use of the system of characteristics (Riemann method) for prescribing boundary condition may be computationally expensive as unconditional stability may be compromised. The alternative is to include such boundary conditions into the implicit formulation (matrix) along with the main variables. This, however, may add extra costs to the solution process. Some authors have developed mixed time discretization methods between explicit and implicit schemes in which the main domain has been solved explicitly whereas the bifurcation points have been treated implicitly to ensure strong coupling of the solution [28, 29]. Another valuable method has been proposed in [30], where the domain decomposition does not impose any stability restriction on the time step size. In large scale problems, implicit schemes lead to assembly of large global matrices, leading to potential memory storage issues and reduction of computational efficiency. This may also be associated to the poor convergence of the Newton's method for systems with a large number of degrees of freedoms [31]. Such issue has been the driving motivation behind the development of quasi-Newton methods, such as Broyden's method. For avoiding an excessive computational cost, some of these methods approximate the Jacobian matrix of the system by means of the finite difference method.

In the present work, we propose a family of new semi-implicit approaches that are able to simplify the numerical procedure without compromising the solution accuracy. This semi-implicit, locally conservative Galerkin (SILCG) methods limit the maximum matrix sizes to the element matrix sizes. With the introduction of LCG discretization, the iterative numerical procedure is substantially simplified. The three versions introduced here are the second order Taylor expansion based (SILCG-TE) method, the streamline upwind Petrov-Galerkin (SILCG-SUPG) scheme and the forward in time and central in space (SILCG-FTCS) method. Since SILCG uses only element matrix in the formulation, the resulting system size, if linear elements are employed, is 4 x 4. This small matrix size allows us to easily compute the exact Jacobian matrix every time step. The use of element matrices allows us to increase the mesh size (number of elements) without enlarging the matrix size. The original LCG method treats each element as an independent domain, while the information between subdomains are explicitly transferred via fluxes. This method was first introduced by [32] for convection-diffusion problems and then generalised to other problems including incompressible Navier-Stokes equations and systemic circulation prob-

lems [25, 1, 33, 34, 35, 36, 37, 5]. The fully explicit version of the LCG method computes multiple solutions at the node shared between two elements. To obtain a unique solution, *a posteriori* averaging of the multiple solution was carried out [32]. The proposed SILCG methods also follow an averaging technique that is identical to that of the explicit LCG method. The main difference between the proposed semi-implicit methods and the fully explicit scheme is that all the terms, except for the boundary fluxes, are treated implicitly. The original contributions of the present work includes the introduction of the SILCG method in three versions for systemic blood circulation, demonstration of its simplicity and accuracy against the fully explicit LCG and other implicit methods. The paper is organised into the following sections. In section 2 we present the governing equations, numerical procedure and boundary conditions. In section 3, a detailed demonstration of the accuracy, robustness and computational efficiency of the proposed methodology is provided. Finally, in section 4, the concluding remarks are reported.

2. Mathematical formulation of the problem

2.1. Governing equations

In the current study, the flow is assumed to be incompressible, Newtonian and laminar. The mass and momentum conservation equations for one dimensional flow in compliant vessels may be written, respectively, as [1, 38]

$$\frac{\partial A}{\partial t} + \frac{\partial (Au)}{\partial x} = 0, \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x} - \frac{f}{\rho} = 0, \tag{2}$$

where A is the cross sectional area, u is the average velocity over such section and ρ is the fluid density. The term f represents the friction forces due to viscosity and it is given as [1]

$$f = \frac{-8\pi\mu u}{A}, \tag{3}$$

in which μ is the fluid viscosity. The pressure p is linked to the area via a non-linear relationship, given as [25, 39, 40, 41]

$$p = p_{ext} + \beta(\sqrt{A} - \sqrt{A_0}), \tag{4}$$

where p_{ext} is the pressure from the surrounding tissues, A_0 is the area at zero transmural pressure and β accounts for the material properties of the elastic vessel

and given by

$$\beta = \frac{\sqrt{\pi}hE}{A_0(1-\sigma^2)}, \tag{5}$$

in which h is the vessel wall thickness, E is the Young's modulus and σ is the Poisson's ratio, assumed to be 0.5 (i.e. the vessel wall is incompressible). The system (1) and (2) can be presented in the following compact form [25, 1]

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{S} - \frac{\partial \mathbf{F}}{\partial x}, \tag{6}$$

where

$$\mathbf{U} = \begin{bmatrix} A \\ u \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} Au \\ \frac{u^2}{2} + \frac{1}{\rho}\left[p_{ext} + \beta\left(\sqrt{A} - \sqrt{A_0}\right)\right] \end{bmatrix} \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} 0 \\ -\frac{8\pi\mu}{\rho}\frac{u}{A} \end{bmatrix}. \tag{7}$$

2.2. Numerical schemes

Here, the three variants of the SILCG formulation are presented.

2.2.1. SILCG-TE

The proposed methodology is based on a LCG spatial discretization scheme, stabilised with a Taylor expansion based method [25, 1, 26]. Differentiating Eq. (6) with respect to time and applying the chain rule yields

$$\frac{\partial^2 \mathbf{U}}{\partial t^2} = \mathbf{C}\frac{\partial \mathbf{U}}{\partial t} - \frac{\partial}{\partial x}\left(\mathbf{H}\frac{\partial \mathbf{U}}{\partial t}\right), \tag{8}$$

where $\mathbf{C}{=}\partial\mathbf{S}/\partial\mathbf{U}$ and $\mathbf{H}{=}\partial\mathbf{F}/\partial\mathbf{U}$ are the Jacobian matrices. The Taylor expansion in time of the primitive variables (A and u), restricted to the second order terms, reads

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \frac{\partial \mathbf{U}}{\partial t}^{n+\theta} + \frac{\Delta t^2}{2}\frac{\partial^2 \mathbf{U}}{\partial t^2}^{n+\theta}, \tag{9}$$

where θ is a parameter ($0 \leq \theta \leq 1$) controlling the time integration scheme. By inserting (6) and (8) into Eq. (9), it is possible to obtain the following semi-discrete form [25]

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = \left\{\mathbf{S} - \frac{\partial \mathbf{F}}{\partial x} + \frac{\Delta t}{2}\left[-\mathbf{C}\left(\frac{\partial \mathbf{F}}{\partial x} - \mathbf{S}\right)\frac{\partial}{\partial x} + \mathbf{H}\left(\frac{\partial \mathbf{F}}{\partial x} - \mathbf{S}\right)\right]\right\}^{n+\theta}. \tag{10}$$

The primitive variables are approximated in space by linear standard finite element shape functions, i.e.

$$A = \mathbf{N}\tilde{\mathbf{A}}, \quad u = \mathbf{N}\tilde{\mathbf{u}}. \tag{11}$$

In the above equation

$$\mathbf{N} = [N_1 \;\; N_2 \;\; ... \;\; N_m]^T, \quad \tilde{\mathbf{A}} = [\tilde{A}_1 \;\; \tilde{A}_2 \;\; ... \;\; \tilde{A}_m], \quad \tilde{\mathbf{u}} = [\tilde{u}_1 \;\; \tilde{u}_2 \;\; ... \;\; \tilde{u}_m], \tag{12}$$

where $\mathbf{N}$ contains m nodal shape functions defined over a spatial domain Ω, whilst $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{u}}$ are the nodal values of the primitive variables [42]. For the sake of simplicity, Eq. (11) can be rewritten in the compact form as $\mathbf{U} = \mathbf{N}\tilde{\mathbf{U}}$. The same discretization is carried out for the vectors $\mathbf{F} = \mathbf{N}\tilde{\mathbf{F}}$ and $\mathbf{S} = \mathbf{N}\tilde{\mathbf{S}}$. Applying the Galerkin weighting to Eq. (10), expressed over an elemental sub-domain Ω_e with boundary Γ_e, yields

$$[\mathbf{M}_e]\Delta\tilde{\mathbf{U}}^n - \Delta t\left([\mathbf{K}_e]\tilde{\mathbf{F}}^{n+\theta} + [\mathbf{L}_e]\tilde{\mathbf{S}}^{n+\theta} + \mathbf{f}_{\Gamma_e}^{n+\theta}\right) = 0, \tag{13}$$

where $\mathbf{M}_e$, $\mathbf{K}_e$ and $\mathbf{L}_e$ are respectively the element mass, convective and source matrices, whilst $\mathbf{f}_{\Gamma_e}$ is the boundary flux term. We notice that the subscript e refers to the element domain. The fully discrete form of the semi-implicit method, with $\theta = 0$ for the boundary fluxes and $\theta = 1$ elsewhere, becomes

$$[\mathbf{M}_e]\Delta\tilde{\mathbf{U}}^n - \Delta t\left([\mathbf{K}_e]\tilde{\mathbf{F}}^{n+1} + [\mathbf{L}_e]\tilde{\mathbf{S}}^{n+1} + \mathbf{f}_{\Gamma_e}^{n}\right) = 0. \tag{14}$$

As seen in the above equation, the interface LCG fluxes are treated at time level n and thus the method is semi-implicit. Making these fluxes implicit may require the solution of additional equations, and the introduced approximations may lead to inaccuracies. The mass matrix in the above equation is defined as

$$\mathbf{M}_e = \frac{l_e}{6}\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}, \tag{15}$$

and it can be lumped for simplifying the solution procedure [43]

$$\mathbf{M}_e = \frac{l_e}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

where l_e is the element length. In the current study, lumped mass matrices are used in all the calculations. The other matrices introduced are defined as

$$\mathbf{K}_e = \frac{1}{2}\begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} + \frac{\Delta t}{4}\mathbf{C}\begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$-\frac{\Delta t}{2l_e}\mathbf{H}\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \tag{17}$$

$$\mathbf{L}_e = \frac{l_e}{6}\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix} + \frac{\Delta t\, l_e}{12}\mathbf{C}\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$
$$+\frac{\Delta t}{4}\mathbf{H}\begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{18}$$

The vectors of Eq. (14) are expressed as

$$\Delta\tilde{\mathbf{U}}^n = \begin{Bmatrix} \Delta\tilde{A}_1 \\ \Delta\tilde{A}_2 \\ \Delta\tilde{u}_1 \\ \Delta\tilde{u}_2 \end{Bmatrix}^n, \quad \tilde{\mathbf{F}}^{n+1} = \begin{Bmatrix} \left(\tilde{A}\tilde{u}\right)_1 \\ \left(\tilde{A}\tilde{u}\right)_2 \\ \frac{\tilde{u}_1^2}{2} + \frac{1}{\rho}\left[p_{ext} + \beta\left(\sqrt{\tilde{A}} - \sqrt{A_0}\right)\right]_1 \\ \frac{\tilde{u}_2^2}{2} + \frac{1}{\rho}\left[p_{ext} + \beta\left(\sqrt{\tilde{A}} - \sqrt{A_0}\right)\right]_2 \end{Bmatrix}^{n+1},$$

$$\tilde{\mathbf{S}}^{n+1} = \begin{Bmatrix} 0 \\ 0 \\ -\frac{8\pi\mu}{\rho}\frac{\tilde{u}_1}{\tilde{A}_1} \\ -\frac{8\pi\mu}{\rho}\frac{\tilde{u}_2}{\tilde{A}_2} \end{Bmatrix}^{n+1}. \tag{19}$$

The subscripts 1 and 2 refer respectively to first and second nodes of the element. The Jacobian matrices can be written as

$$\mathbf{H} = \begin{bmatrix} u_{ave} & 0 & A_{ave} & 0 \\ 0 & u_{ave} & 0 & A_{ave} \\ \frac{\beta_1}{2\rho\sqrt{A_{ave}}} & 0 & u_{ave} & 0 \\ 0 & \frac{\beta_2}{2\rho\sqrt{A_{ave}}} & 0 & u_{ave} \end{bmatrix}, \tag{20}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{8\pi\mu}{\rho}\frac{u_{ave}}{A_{ave}^2} & 0 & -\frac{8\pi\mu}{\rho}\frac{1}{A_{ave}} & 0 \\ 0 & \frac{8\pi\mu}{\rho}\frac{u_{ave}}{A_{ave}^2} & 0 & -\frac{8\pi\mu}{\rho}\frac{1}{A_{ave}} \end{bmatrix}. \tag{21}$$

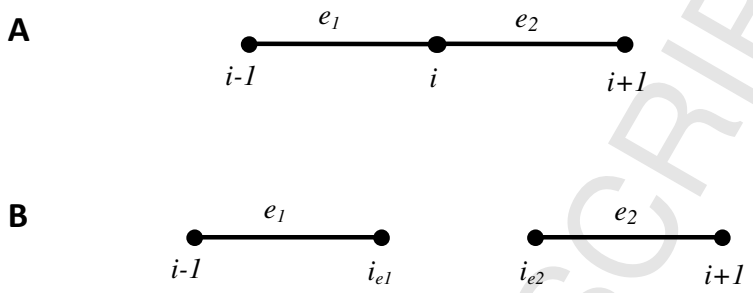

Figure 1: One-dimensional elements in CG (A) and LCG methods (B).

Note that the entries of the Jacobian matrices (**H** and **C**) are averaged over the element. For example, for calculating A_1, the first line of the matrix **H** is used as $u_{ave} = (\tilde{u}_1 + \tilde{u}_2)/2$ and $A_{ave} = \left(\tilde{A}_1 + \tilde{A}_2\right)/2$. Similar procedure is applied for the other quantities in the matrix **H**. The same approach is used for computing matrix **C**. We notice that, in CG method, the flux term in Eq. (14) is equal to 0, except at the boundaries. In the LCG method however, we use this term to link elements and thus the normal assembly procedure is avoided here. This is why the LCG method is explicitly, locally conservative rather than implicitly conservative as in the case of CG methods [36]. The definition of fluxes $\tilde{\mathbf{F}}$ is given in Eq. (19). We use appropriate outward pointing normals to define the fluxes at both ends of an element, i.e. $\mathbf{f}^n_{\Gamma_e} = \tilde{\mathbf{F}}^n$. These fluxes must be recalled from the previous time step (i.e. n time level) for maintaining continuity (see [32, 33] for more details). An alternative would be introducing very expensive additional implicit approximations for the fluxes to maintain explicit local conservation. Thus, the methods proposed at the best can only be semi-implicit. Figure 1 shows, respectively, the CG and LCG element discretizations. While CG method uses a continuous discretization between elements, the LCG method breaks the link, which is then reestablished through edge fluxes *a posteriori*. Eq. (14) is solved independently over the elements e_1 and e_2 and a unique nodal solution is obtained by averaging the variables on the intermediate nodes as [32, 33]

$$\tilde{\mathbf{U}}_i = \frac{\tilde{\mathbf{U}}_{ie1} + \tilde{\mathbf{U}}_{ie2}}{2}, \tag{22}$$

where subscripts $ie1$ and $ie2$ refer to the two solutions obtained at node i.

2.2.2. SILCG-SUPG

With the streamline upwind Petrov-Galerkin (SUPG) method, the test function is modified in order to stabilise the oscillations due to the convective term

$$\mathbf{W} = \mathbf{N} + \frac{\alpha\, l_e}{2} \frac{\tilde{u}_i}{|u_{ave}|} \frac{d\mathbf{N}}{dx}, \tag{23}$$

where $\mathbf{W}$ is a modified shape function of the linear weighting function $\mathbf{N}$. The value of α varies depending on local Peclet number Pe and it can be optimally calculated as [44]

$$\alpha = coth(Pe) - \frac{1}{Pe} \quad \texttt{and} \quad Pe = \frac{|u_{ave}|\, l_e}{2\nu}, \tag{24}$$

in which ν is the kinematic viscosity of the fluid. The rest of the formulation is identical to that of the SILCG-TE method presented in the previous subsection.

2.2.3. SILCG-FTCS

The last method we introduce is the forward in time and central (standard Galerkin method) in space (FTCS) method in its semi-implicit form. The motivation for this comes from the fact that the implicit form of such method for linear convection equations is unconditionally stable. The idea here is therefore to investigate the limits of such a method in the context of the semi-implicit discretization. The method is identical to either SILCG-TE or SILCG-SUPG method without the additional stabilisation terms. In other words, the matrices are reduced by taking both matrices $\mathbf{H}$ and $\mathbf{C}$ equal to 0.

2.3. Solution procedure

The solution of the system defined in Eq. (14) is sought by employing the Newton's method [44, 45]. The iterative solution of the unknown variables vector $\tilde{\mathbf{U}}$ (i.e. $\tilde{\mathbf{U}} = \left[\tilde{A}_1 \;\; \tilde{A}_2 \;\; \tilde{u}_1 \;\; \tilde{u}_2\right]^T$) at the iteration $k+1$ is given by

$$\tilde{\mathbf{U}}^{k+1} = \tilde{\mathbf{U}}^k + \delta\tilde{\mathbf{U}}^k. \tag{25}$$

The update $\delta\tilde{\mathbf{U}}^k$ is estimated via

$$\mathbf{J}\left(\tilde{\mathbf{U}}^k\right)\delta\tilde{\mathbf{U}}^k = -\mathbf{R}\left(\tilde{\mathbf{U}}^k\right), \tag{26}$$

where $\mathbf{J}\left(\tilde{\mathbf{U}}^k\right)$ is a 4 x 4 Jacobian matrix and $\mathbf{R}\left(\tilde{\mathbf{U}}^k\right)$ is the residual vector of Eq. (14). Eq. (26) is iteratively solved until convergence is reached. If the initial guess is far from the root, Newton method may not converge. In the current work, the variables are initialised at each iteration with the values of the previous time step [46]. A linear representation of Eq. (26) gives four linear algebraic equations per element (i.e. $R_{A_1}, R_{A_2}, R_{u_1}$ and R_{u_2}). The detailed form may now be written as

$$\begin{bmatrix} \frac{\partial R_{A_1}}{\partial A_1} & \frac{\partial R_{A_1}}{\partial A_2} & \frac{\partial R_{A_1}}{\partial u_1} & \frac{\partial R_{A_1}}{\partial u_2} \\ \frac{\partial R_{A_2}}{\partial A_1} & \frac{\partial R_{A_2}}{\partial A_2} & \frac{\partial R_{A_2}}{\partial u_1} & \frac{\partial R_{A_2}}{\partial u_2} \\ \frac{\partial R_{u_1}}{\partial A_1} & \frac{\partial R_{u_1}}{\partial A_2} & \frac{\partial R_{u_1}}{\partial u_1} & \frac{\partial R_{u_1}}{\partial u_2} \\ \frac{\partial R_{u_2}}{\partial A_1} & \frac{\partial R_{u_2}}{\partial A_2} & \frac{\partial R_{u_2}}{\partial u_1} & \frac{\partial R_{u_2}}{\partial u_2} \end{bmatrix} \begin{Bmatrix} \delta\tilde{A}_1 \\ \delta\tilde{A}_2 \\ \delta\tilde{u}_1 \\ \delta\tilde{u}_2 \end{Bmatrix} = - \begin{Bmatrix} R_{A_1} \\ R_{A_2} \\ R_{u_1} \\ R_{u_2} \end{Bmatrix}. \tag{27}$$

The linear system presented above is solved by using LU factorisation [47]. The global system solution is computed at each time step via the following algorithm

---

Algorithm 1

---

1. do i=1,m m=1,2,.....,elements
2. do j=1,2 nodes
3. initialise $\tilde{\mathbf{U}}^{k,n+1} = \tilde{\mathbf{U}}^{k,n}$
4. end do
5. do k=1,number of iterations
6. compute $\mathbf{J}\left(\tilde{\mathbf{U}}^k\right)$ and $\mathbf{R}\left(\tilde{\mathbf{U}}^k\right)$
7. call LU and solve for $\delta\tilde{\mathbf{U}}^k$ through Eq. (26)
8. evaluate, if $\delta\tilde{\mathbf{U}}^k \leq$ *tolerance*

yes → go step 10

no → put $\tilde{\mathbf{U}}^{k+1} = \tilde{\mathbf{U}}^{k}$
9. end do
10. update through Eq. (25)
11. end do

Steps 2-4 represent the element initialisation procedure (where SILCG holds both elements and nodes in different loops whereas CG has only nodes). Steps from 5 to 10 refer to the Newton iteration. The stopping tolerance for the algorithms is set equal to 10^{-4}.

2.4. Network boundary conditions

For prescribing boundary conditions, the methodology proposed in [25, 1] is followed. With regard to the inlet flow condition, two types of boundaries can generally be used. The reflective boundaries, where values of A, u, and p (or commonly used flow rate $Q = Au$) are directly prescribed at the inlet node. Usually, any of these variables can be expressed as a function of time (see for instance [26, 30]). Alternatively, the non-reflective boundaries where characteristic variables are used. The latter methods are more realistic and here the boundary conditions are calculated as a combination of outside and inside conditions by implementing Riemann method. Since the wave nature of the problem is exploited by the latter method, we use this approach for specifying the boundary conditions. As a part of the wave comes from the domain interior, the boundaries need to be extrapolated from the previous time step. The problem is initialised using a specified wave of $p_{int} = 50\ mmHg$ [1]. The prescribed wave from outside the domain is the pressure wave form as shown in Figure 2. By assuming $\mathbf{S} \approx 0$ near the boundaries and applying chain rule to the spatial derivative, Eq. (6) becomes

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{H}\frac{\partial \mathbf{U}}{\partial x} = 0, \tag{28}$$

The characteristic speeds for the system are the eigenvalues of the matrix $\mathbf{H}$, which can be obtained by solving $|\mathbf{H} - \lambda\mathbf{I}| = 0$, with $\mathbf{I}$ being the identity matrix. The characteristic speeds are given as

$$\lambda_{f,b} = u \pm c, \tag{29}$$

where the subscripts f and b refer to forward and backward travelling directions, and c is the pulse wave speed which is given as

$$c = \sqrt{\frac{\beta}{2\rho}} A^{1/4}. \tag{30}$$

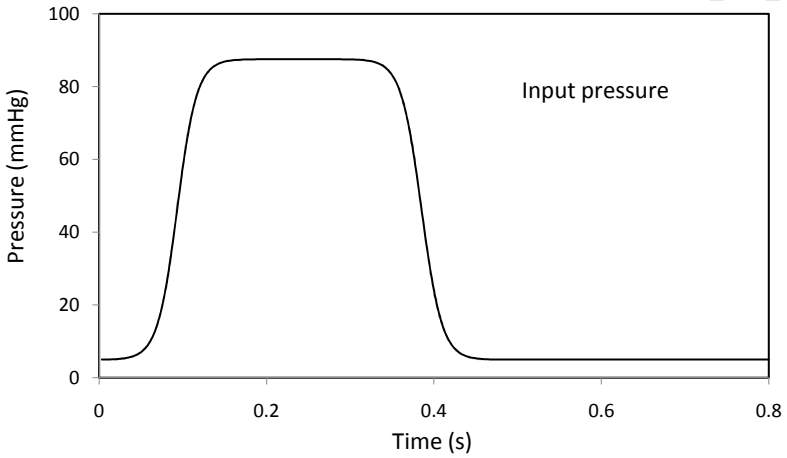

Figure 2: Input pressure wave.

The wave speed c is much greater than fluid velocity u under physiological flow conditions, and therefore $\lambda_f = u + c > 0$ while $\lambda_b = u - c < 0$. These two eigenvalues are real, so the system is strictly hyperbolic. This needs only one boundary condition at each inlet and exit of the domain [25]. The matrix $\mathbf{L}$ of the left eigenvalues of $\mathbf{H}$ can be written as

$$\mathbf{L} = \frac{\partial \mathbf{V}}{\partial \mathbf{U}} = \begin{bmatrix} \frac{c}{A} & 1 \\ -\frac{c}{A} & 1 \end{bmatrix}, \tag{31}$$

where $\mathbf{V} = \begin{bmatrix} w_f & w_b \end{bmatrix}^T$, and then

$$\mathbf{LH} = \Lambda \mathbf{L}, \tag{32}$$

$$\Lambda = \begin{bmatrix} \lambda_f & 0 \\ 0 & \lambda_b \end{bmatrix}. \tag{33}$$

Multiplying Eq. (28) by the matrix $\mathbf{L}$ and substituting $\mathbf{LH}$ from Eq. (32) gives

$$\mathbf{L}\frac{\partial \mathbf{U}}{\partial t} + \Lambda \mathbf{L}\frac{\partial \mathbf{U}}{\partial x} = 0. \tag{34}$$

Note that $\mathbf{H} = \mathbf{L}^{-1}\Lambda\mathbf{L}$, and thus the above equation can be rewritten as

$$\frac{\partial \mathbf{V}}{\partial t} + \Lambda \frac{\partial \mathbf{V}}{\partial x} = 0. \tag{35}$$

The characteristic variables can now be determined as

$$w_{f,b} = u \pm 4c. \tag{36}$$

The primitive variables may be estimated from the characteristic variables as [25, 48]

$$A = \frac{\left(w_f - w_b\right)^4}{1024}\left(\frac{\rho}{\beta}\right)^2, \tag{37}$$

$$u = \frac{w_f + w_b}{2}. \tag{38}$$

The equations above show that a pair of the characteristic variables is necessary for specifying one boundary condition. At the inlet, w_f implies the imposed pressure shown in Figure 2; thus rearranging Eq. (37) gives [25, 1]

$$w_f^{n+1} = w_b^0 + 8A^{1/4}\sqrt{\frac{\beta}{2\rho}}, \tag{39}$$

where w_b^0 is the initial backward variable, and it is equal to w_b at any time (i.e. assume no backward waves reach the inlet [25]). This is possible at the inlet as p is given (see Figure 2). Similarly, at first time step where p_{int} is known and thus initial A can be computed by rearranging Eq. (4). For prescribing the velocity, similar procedure may be applied to Eq. (38), so that

$$w_f^{n+1} = 2u - w_b^0. \tag{40}$$

The velocity at the beginning is set equal to 0 for the first time step. Backward variable w_b carries information from domain interior, so it is extrapolated according to the following equation [25, 1]

$$w_b^{n+1}|_{x=x_0} = w^n_{b|_{x=x_0} - \lambda_b^n \Delta t}. \tag{41}$$

With regard to the terminal boundaries, w_b represents the impedance, and is calculated by using the tapering vessel approach. In this approach, the area A of an arterial segment decreases gradually along the length. Pure reflections are assumed, and thus w_b can be determined as

$$w_b^{n+1} = w_b^0. \tag{42}$$

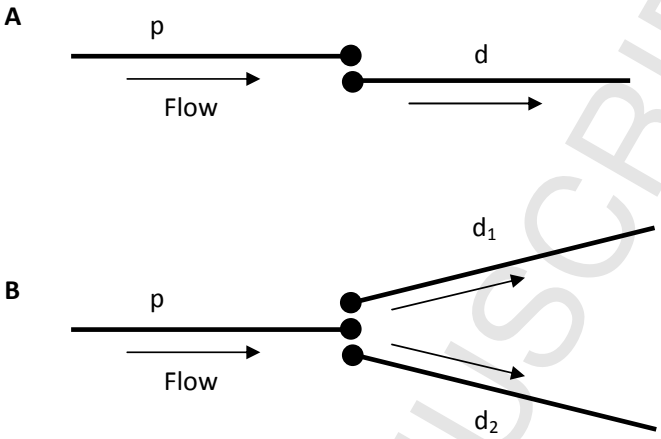

Figure 3: Different types of arterial tree discontinuities.

Meanwhile, w_f is extrapolated by adopting [25, 1]

$$w_f^{n+1}|_{x=x_L} = w^n_{f|_{x=x_L} - \lambda_f^n \Delta t}. \tag{43}$$

Moreover, the characteristic system is useful to enforce quantities at bifurcations and other discontinuities. Here, we consider two cases as shown in Figure 3. Figure 3**A** shows a case of single tube with a sudden change in the A or β, representing a discontinuity. Figure 3**B** shows the case of a parent vessel branches and its two daughter branches.

For each branching location, there are six unknowns, which are A and u for parent (p) and daughter (d) segments. Such variables are computed by assembling a system of equations accounting for mass and momentum conservation, and definitions of characteristic speeds [25]. The conservation of mass can be simply written as

$$Q_p = \sum_{i=1}^{N} Q_i, \tag{44}$$

where N is number of daughter vessels. In Eq. (44) the flow direction is assumed to be from parent to daughter vessels. The continuity of total pressure (momentum conservation) between the parent and daughter vessels may be expressed as

$$\frac{\rho u_p^2}{2} + p_{s_p} = \frac{\rho u_i^2}{2} + p_{s_i}, \tag{45}$$

where $\rho u^2/2$ is the dynamic pressure and p_s represents the static pressure given in Eq. (4). The characteristic equations for the parent and daughter vessels are

$$w_{f_p} = u_p + 4A_p^{1/4}\sqrt{\frac{\beta_p}{2\rho}}, \tag{46}$$

$$w_{b_i} = u_i - 4A_i^{1/4}\sqrt{\frac{\beta_i}{2\rho}}. \tag{47}$$

3. Results and discussions

In this section, the accuracy and computational efficiency of the proposed semi-implicit methods are discussed. The well established explicit Taylor-LCG method [25, 1] is used as a benchmark method to compare the results. For computing the solution, the maximum admissible time step size is adopted. The spatial step used is $l_e = 0.25\ cm$ (corresponding to 6288 elements) for all cases unless otherwise stated. The arterial network used in the current study is shown in Figure 4. The model consists of 63 arterial segments.

The duration of a cardiac cycle is assumed to be $\approx 0.8\ s$ or $70 - 75\ beats/min$ and the opening ventricular valve duration is $0.057\ s$ while the closing duration is $0.039\ s$. A coronary model as in [25] is also included. The blood viscosity and density are set equal to, respectively, $0.035\ g/(cm\ s)$ and $1.06\ g/cm^3$. More details on the morphology and structural properties of arterial system can be found in [25, 1].

3.1. Wave-forms along the network

Figures 5 and 6 show, for all the SILCG methods, the pressure and flow time evolution at different locations along the arterial tree. The monitored segments are labelled 9, 13, 23, 15, 29, 60, 54, 35 and 49 according to Figure 4. Results obtained by using the explicit LCG are also included. For the semi-implicit methods, the Jacobian matrix is updated at each iteration. The solutions computed with new schemes agree very well with each other and with the one obtained with the explicit method. The calculations show that the SILCG-FTCS admits a maximum time step equal to $\Delta t = 0.1\ ms$, while for the SILCG-TE and SILCG-SUPG methods larger time step sizes can be selected. As expected, the additional stabilisation improves the stability and allows for larger time step values.

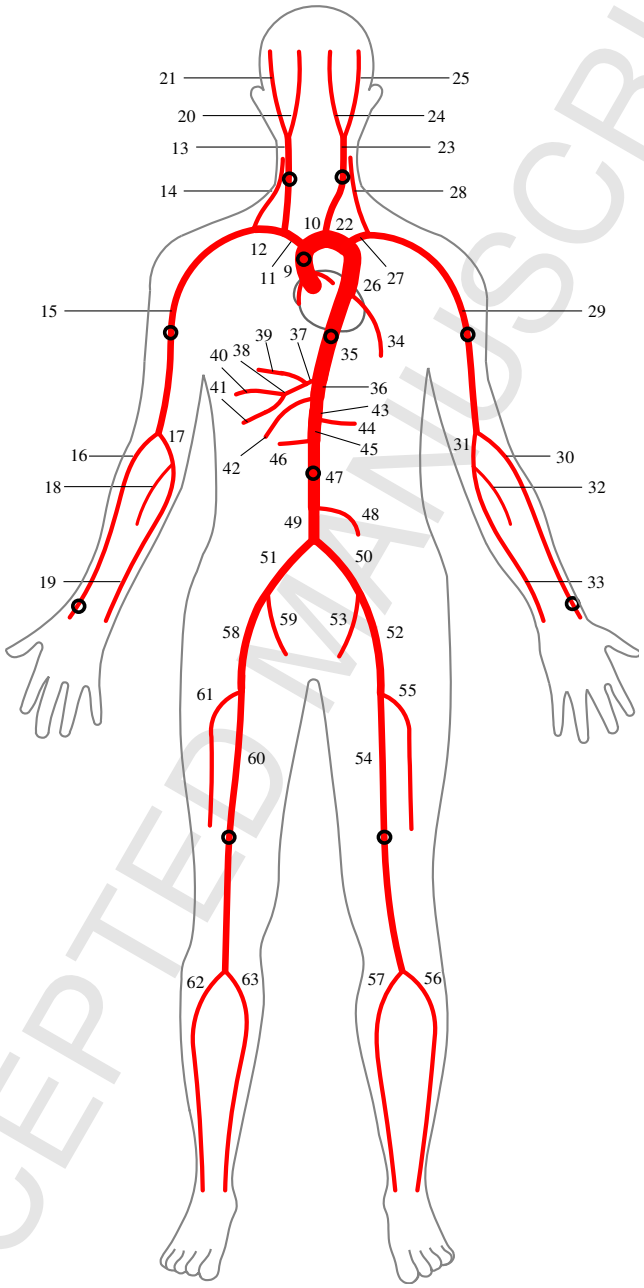

Figure 4: Systemic circulation model [1].

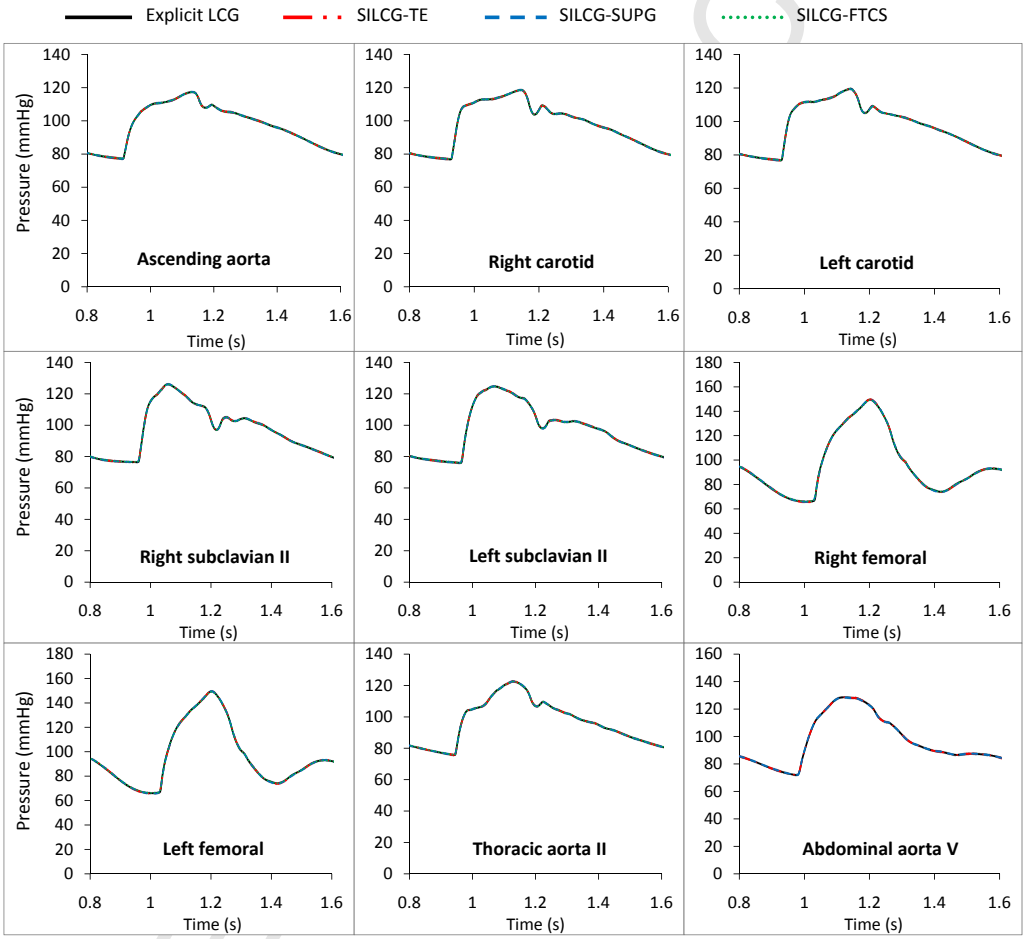

Figure 5: Pressure variation in different arterial segments.

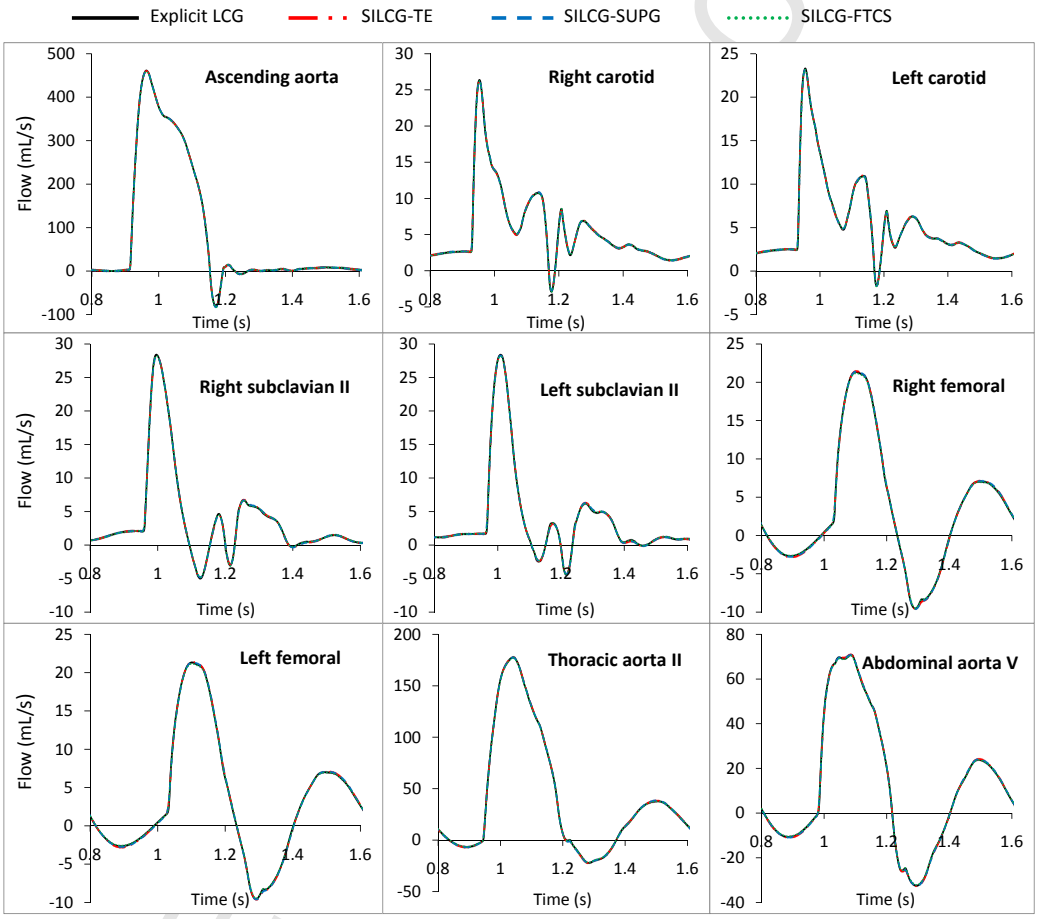

Figure 6: Flow variation in various segments.

At this stage, we estimate the root mean square errors (*RMSE*) for pressure and flow as shown in Table 1. These errors are defined as [26]

$$RMSE_p = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{p^i_{SILCG} - p^i_{LCG}}{p^i_{LCG}}\right)^2}, \tag{48}$$

$$RMSE_F = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{F^i_{SILCG} - F^i_{LCG}}{F^i_{LCG_{max}}}\right)^2}, \tag{49}$$

where the subscripts p and F refer to pressure and flow respectively. p^i_{SILCG} and F^i_{SILCG} are the estimated values from all proposed approaches at a specific location and for different times (i=1,.....,n). p^i_{LCG} and F^i_{LCG} are the reference values considered from [1, 25] (at the same location and time). The table shows that the maximum $RMSE_p$ is always below 0.4%, whilst $RMSE_F$ is less than 0.7%. Both percentage errors are widely within the range of accuracy expected.

In Figures 7 and 8 we show the average number of iterations for reaching convergence against the number of elements employed. In Figure 7 the Jacobian matrix is calculated at each iteration, whilst in Figure 8 the Jacobian matrix is only calculated once each time step. As seen, the iterations to number of elements ratio is substantially lower than unity. Similar observations were found in [29]. Among the methods presented, the SILCG-FTCS is the fastest.

3.2. Spatial convergence

The effect of mesh refinement on the accuracy is highlighted for various meshes in Figure 9. The results are plotted for mid-point of the ascending aorta (i.e. segment 9). A total of three different meshes with element sizes 0.25, 0.33 and 0.5 *cm* respectively are tested on SILCG-TE. All other methods give similar results and thus not plotted. The results are compared against a fine mesh solution obtained by using the explicit LCG method. The plot shows that the solution, as the mesh is refined, converges to the results obtained with the explicit LCG method.

3.3. Computational efficiency

Finally, we evaluate the CPU time for all proposed methods as shown in Table 2, by considering the maximum admissible time step sizes. The number of elements used is 6288, and the results are presented for two cases of Jacobian matrix calculations. In case 1, the Jacobian matrix is updated at every iteration, while for case 2, the matrix is only factorised once per time step. Table 2 shows that the

Table 1: Percentage $RMSE_p$ and $RMSE_F$ for all cases shown in Figures 5 and 6.

Location	Error %	SILCG-TE	SILCG-SUPG	SILCG-FTCS
segment: 9	$RMSE_p$	0.038	0.052	0.087
	$RMSE_F$	0.151	0.154	0.199
segment: 13	$RMSE_p$	0.069	0.078	0.098
	$RMSE_F$	0.371	0.380	0.293
segment: 23	$RMSE_p$	0.070	0.078	0.100
	$RMSE_F$	0.386	0.399	0.321
segment: 15	$RMSE_p$	0.133	0.136	0.122
	$RMSE_F$	0.406	0.411	0.307
segment: 29	$RMSE_p$	0.146	0.151	0.133
	$RMSE_F$	0.437	0.441	0.316
segment: 60	$RMSE_p$	0.360	0.344	0.280
	$RMSE_F$	0.651	0.617	0.434
segment: 54	$RMSE_p$	0.359	0.342	0.280
	$RMSE_F$	0.650	0.616	0.434
segment: 35	$RMSE_p$	0.090	0.097	0.109
	$RMSE_F$	0.250	0.240	0.219
segment: 49	$RMSE_p$	0.152	0.157	0.145
	$RMSE_F$	0.580	0.529	0.401

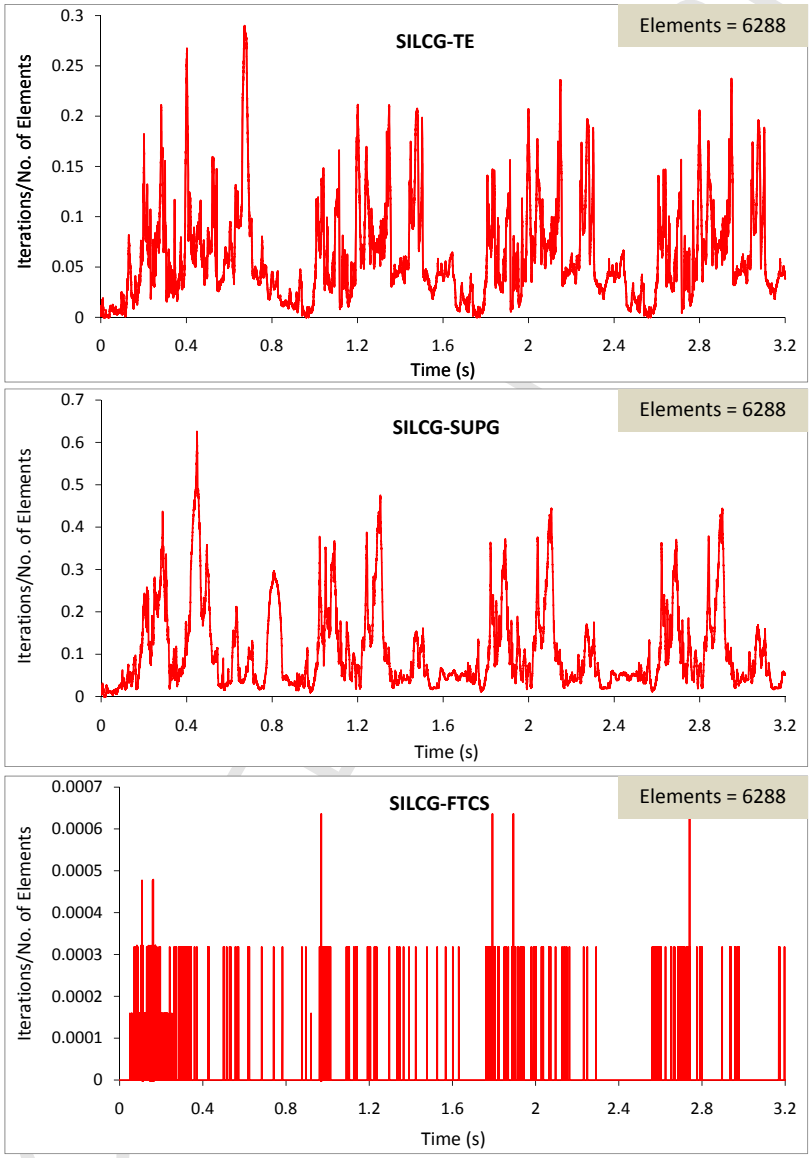

Figure 7: Number of iterations to number of elements ratio for for all three approaches. Jacobian matrix is calculated at every iteration.

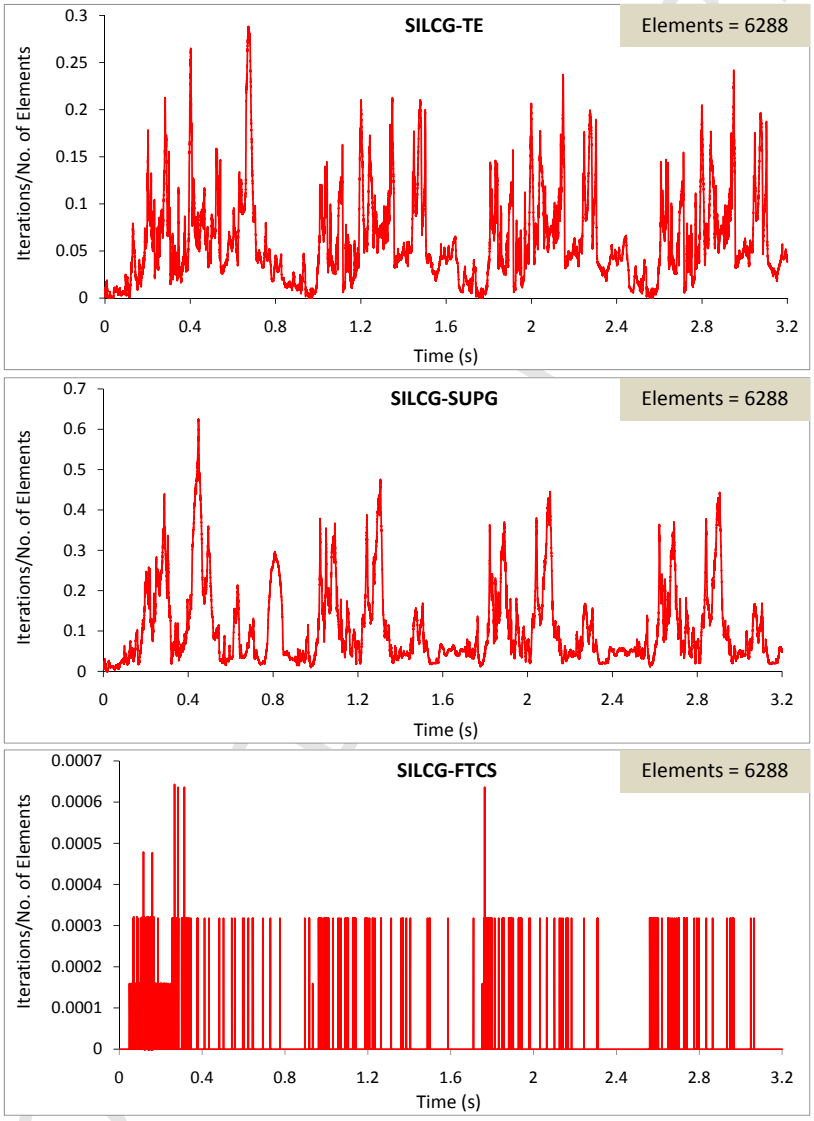

Figure 8: Number of iterations to number of elements ratio for for all three approaches. Jacobian matrix is calculated at every time step.

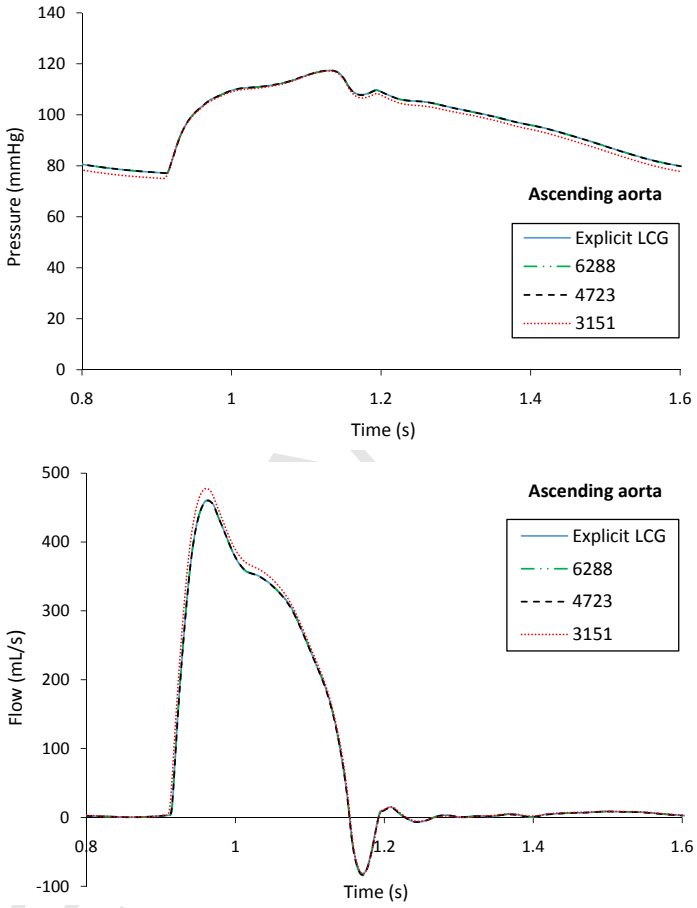

Figure 9: Mesh convergence of SILCG-TE method.

Table 2: Computational speed CPU time per cardiac cycle, $l_e = 0.25\ cm$.

Jacobian matrix status	SILCG-TE (min)	SILCG-SUPG (min)	SILCG-FTCS (min)
case 1	29.63	33.19	3.40
case 2	13.05	14.20	3.06

SILCG-FTCS is the fastest one among the methods proposed due to the significant simplification in the numerical formulation discussed in subsection *2.2*. For this reason, the CPU time is almost independent of the way in which Jacobian matrix is computed. For the other methods, continuous Jacobian matrix updating (case 1) significantly affects the computational performances. Overall, the maximum CPU time taken is less than 34 minutes and minimum is just over 3 minutes per cardiac cycle.

4. Conclusions

A set of semi-implicit, locally conservative Galerkin methods (SILCG) have been proposed and tested for blood flow in a human arterial system. The three novel methods proposed include a Taylor expansion based method, a streamline upwind Petrov-Galerkin method and a FTCS type method. Although the FTCS type method is the most straightforward and simple method, it is unstable beyond a time step $\Delta t = 0.1\ ms$. This is not a major disadvantage as the method can be very fast without the additional stabilisation terms. The other two methods can admit larger time steps but the accuracy may deteriorate with increase in time step values. None of the new methods are unconditionally stable as the characteristic variables and boundary fluxes are used from the previous time step. Overall, the proposed semi-implicit methods represent a promising alternative to the existing explicit LCG method, as the time step is not restricted by the CFL condition. Moreover, the proposed methodology has the advantage of avoiding large matrix assembly, as it requires only factorisation at element level. Further research is required to make the proposed method fully implicit and more computationally efficient.

References

[1] K. Low, R. van Loon, I. Sazonov, R. L. T. Bevan, P. Nithiarasu, An improved baseline model for a human arterial network to study the impact of aneurysms on pressure-flow waveforms, International Journal for Numerical Methods in Biomedical Engineering 28 (2012) 1224–1246.

[2] I. Sazonov, S. Khir, W. Hacham, E. Boileau, J. Carson, R. van Loon, C. Ferguson, P. Nithiarasu, A novel method for non-invasively detecting the severity and location of aortic aneurysms., Biomechanics and modeling in mechanobiology DOI 10.1007/s10237-017-0884-8.

[3] E. Boileau, P. Nithiarasu, One-dimensional modelling of the coronary circulation. application to noninvasive quantification of fractional flow reserve (ffr), in: J. Tarves, R. Jorge (Eds.), Computational and experimental biomedical sciences: Methods and Applications, Springer, 2013, pp. 137–156.

[4] E. Boileau, S. Pant, C. Roobottom, I. Sazonov, J. Deng, X. Xie, P. Nithiarasu, Estimating the accuracy of a reduced-order model for the calculation of fractional flow reserve (ffr), International Journal for Numerical Methods in Biomedical Engineering 10.1002/cnm.2908 e2908.

[5] A. Coccarelli, E. Boileau, D. Parthimos, P. Nithiarasu, An advanced computational bioheat transfer model for a human body with an embedded systemic circulation, Biomechanics and Modeling in Mechanobiology 15(5) (DOI: 10.1007/s10237-015-0751-4) (2016) 1173–1190.

[6] J. R. Womersley, An elastic tube theory of pulse transmission and oscillatory flow in mammalian arteries, Tech. Rep. TR 56-614, WADC Technical Report (1957).

[7] A. Avolio, Multi-branched model of the human arterial system, Medical and Biological Engineering and Computing 18 (6) (1980) 709–718.

[8] D. Brown, Input impedance and reflection coefficient in fractal-like models of asymmetrically branching compliant tubes, IEEE Transactions on Biomedical Engineering 43 (7) (1996) 715–722.

[9] Y. Huo, G. Kassab, A hybrid one-dimensional/womersley model of pulsatile blood flow in the entire coronary arterial tree, American Journal of Physiology 292 (2007) 2623–2633.

[10] N. Westerhof, F. Bosman, C. De Vries, A. Noordergraaf, Analog studies of the human systemic arterial tree, Journal of Biomechanics 2 (2) (1969) 121–134.

[11] W. Bodley, The non-linearities of arterial blood flow, Physics in Medicine and Biology 16 (4) (1971) 663–672.

[12] P. Reuderink, H. Hoogstraten, P. Sipkema, B. Hillen, N. Westerhof, Linear and nonlinear one-dimensional models of pulse wave transmission at high womersley numbers, Journal of Biomechanics 22 (8-9) (1989) 819–827.

[13] S. J. Sherwin, V. Franke, J. Peiró, K. Parker, One-dimensional modelling of a vascular network in space-time variables, Journal of Engineering Mathematics 47 (2003) 217–250. doi:10.1023/B:ENGI.0000007979.32871.e2.

[14] J. Raines, M. Jaffrin, A. Shapiro, A computer simulation of arterial dynamics in the human leg, Journal of Biomechanics 7 (1) (1974) 77–91.

[15] N. Stergiopulos, D. Young, T. Rogge, Computer simulation of arterial flow with applications to arterial and aortic stenoses, Journal of Biomechanics 25 (12) (1992) 1477–1488.

[16] C. Sheng, S. Sarwal, K. Watts, A. Marble, Computational simulation of blood flow in human systemic circulation incorporating an external force field, Medical and Biological Engineering and Computing 33 (1) (1995) 8–17.

[17] B. Schaaf, P. Abbrecht, Digital computer simulation of human systemic arterial pulse wave transmission: A nonlinear model, Journal of Biomechanics 5 (4) (1972) 345–364.

[18] S. A. Urquiza, P. J. Blanco, M. J. Vénere, R. A. Feijóo, Multidimensional modelling for the carotid artery blood flow, Computer Methods in Applied Mechanics and Engineering 195 (2006) 4002–4017.

[19] J. Wan, B. Steele, S. Spicer, S. Strohband, G. Feijóo, T. Hughes, C. Taylor, A one-dimensional finite element method for simulation-based medical planning for cardiovascular disease, Computer Methods in Biomechanics and Biomedical Engineering 5 (3) (2002) 195 – 206.

[20] L. Formaggia, F. Nobile, A. Quarteroni, A one-dimensional model for blood flow: application to vascular prosthesis, in: I. Babuška, et al. (Eds.), Mathematical Modeling and Numerical Simulation in Continuum Mechanics, Vol. 19 of Lecture Notes in Computational Science and Engineering, 2002, pp. 137—153.

[21] J. Carson, R. Van Loon, An implicit solver for 1d arterial network models, International Journal for Numerical Methods in Biomedical Engineering 10.1002/cnm.2837.

[22] C. Contarino, E. F. Toro, G. I. Montecinos, R. Borsche, J. Kall, Junction-generalized riemann problem for stiff hyperbolic balance laws in networks: An implicit solver and ader schemes, Journal of Computational Physics 315 (2016) 409433.

[23] N. Bessonov, A. Sequeira, S. Simakov, Y. Vassilevskii, , V. Volpert, Methods of blood flow modelling, Math. Model. Nat. Phenom. 11 (1) (2016) 1 – 25.

[24] P. J. Blanco, S. M. Watanabe, E. A. Dari, M. A. R. F. Passos, R. A. Feijóo, Blood flow distribution in an anatomically detailed arterial network model: criteria and algorithms, Biomech Model Mechanobiol 13 (2014) 1303–1330.

[25] J. Mynard, P. Nithiarasu, A 1D arterial blood flow model incorporating ventricular pressure, aortic valve and regional coronary flow using locally conservative Galerkin (LCG) method., Communications in Numerical Methods in Engineering 24 (2008) 367–417.

[26] E. Boileau, P. Nithiarasu, J. Blanco, L. Muller, F. E. Fossans, L. Helleviks, W. Doners, W. Huberts, M. Willemet, J. Alastruey, A benchmark study of 1-d numerical schemes for arterial blood flow modelling, International Journal for Numerical Methods in Biomedical Engineering e02732 (DOI: 10.1002/cnm.2732).

[27] A. Brooks, T. Hughes, Streamline upwind/Petrov–Galerkin formulation for convection dominated flows with particular emphasis on the incompressible navier stokes equation, Computer Methods in Applied Mechanics and Engineering 32 (1982) 199–259.

[28] S. S. Simakov, A. S. Kholodov, Computational study of oxygen concentration in human blood under low frequency disturbances, Mathematical

Models and Computer Simulations 1 (DOI:10.1134/S2070048209020112) (2009) 283–295.

[29] A. C. I. Malossi, P. J. Blanco, 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 (2012) 212–226.

[30] P. J. Blanco, J. S. Leiva, R. A. Feijóo, G. C. Buscaglia, Black-box decomposing approach for computational hemodynamic: One-dimensional model, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1389–1405.

[31] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in Fortran 77: the art of scientific computing, second-volume 1 of fortran numerical recipes Edition, Cambridge University Press, 1996-1997.

[32] P. Nithiarasu, A simple locally conservative galerkin (LCG) finite-element method for transient conservation equations, Numerical Heat Transfer Part B - Fundamentals 46 (4) (2004) 357–370.

[33] C. G. Thomas, P. Nithiarasu, R. L. T. Bevan, The locally conservative galerkin (lcg) method for solving the incompressible navier-stokes equations, International Journal for Numerical Methods in Fluids 57 (2008) 1771–1792.

[34] C. G. Thomas, P. Nithiarasu, An element-wise, locally conservative galerkin (lcg) method for solving diffusion and convection-diffusion problems, International Journal for Numerical Methods in Engineering 73 (2008) 642–664.

[35] R. Bevan, P. Nithiarasu, R. V. Loon, I. Sazonov, H. Luckraz, A. Garnham, Application of a locally conservative Galerkin (LCG) method for modelling blood flow through a patient-specific carotid bifurcation, International Journal for Numerical Methods in Fluids 64 (2010) 1274–1295.

[36] N. Zhang, Z. Huang, J. Y. School, Locally conservative galerkin and finite volume methods for two-phase flow in porous media, J. Comput. Phys. 254 (2013) 39–51.

[37] A. Coccarelli, P. Nithiarasu, A robust finite element modelling approach to conjugate heat transfer in flexible elastic tubes and tube networks., Numerical Heat Transfer, Part A Applications 67 (2015) 513–530.

[38] S. Sherwin, V. Franke, J. Peiró, K. Parker, One-dimensional modelling of a vascular network in space-time variables, Journal of Engineering Mathematics 47 (3) (2003) 217–250.

[39] L. Formaggia, F. Nobile, A. Quarteroni, A one dimensional model for blood flow: Application to vascular prosthesis, in: B. I., M. T., C. P.G. (Eds.), Mathematical Modeling and Numerical Simulation in Continuum Mechanics, Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2002.

[40] M. Olufsen, C. Peskin, W. Kim, E. Pedersen, A. Nadim, J. Larsen, Numerical simulation and experimental validation of blood flow in arteries with structured-tree outflow conditions, Annals of Biomedical Engineering 28 (2000) 1281–1299.

[41] S. Acosta, C. Puelz, B. Riviere, D. J. Pennya, C. G. Rusin, Numerical method of characteristics for one-dimensional blood flow, J. Comput. Phys. 294 (2015) 96–109.

[42] P. Nithiarasu, R. Lewis, K. Seetharamu, Fundamentas of the finite element method for heat, mass and fluid flow, 2nd Edition, Wiley, 2016.

[43] R. Lewis, P. Nithiarasu, K. Seetharamu, Fundamentals of the finite element method for heat and fluid flow, John Wiley and Sons, 2004.

[44] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, The Finite Element Method for Fluid Dynamics, seventh Edition, Elsevier, 2014.

[45] O. Zienkiewicz, R. Taylor, D. Fox, The finite element method for solid and structural mechanics, seventh Edition, Elsevier, 2014.

[46] R. H. W. Hoppea, C. Linsenmann, An adaptive newton continuation strategy for the fully implicit finite element immersed boundary method, J. Comput. Phys. 231 (2012) 4676–4693.

[47] C. D. Meyer, Matrix analysis and applied linear algebra, SIAM, 2000.

[48] S. Sherwin, L. Formaggia, J. Peiró, V. Franke, Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system, International Journal for Numerical Methods in Fluids 43 (6-7) (2003) 673–700.