

# Numerical techniques for dynamic resistive networks

Richard G. Brown<sup>1</sup>

(Received 2 November 2012; revised 19 March 2013)

## Abstract

We consider approaches to the numerical difficulties posed by modelling resistive networks with dynamically changing resistances according to a set of coupled ordinary differential equations (ODEs). The prototype problem is autoregulation on a cerebral microvascular network. In this network the amount of perfusion, or tissue blood supply, is determined by the resistance of the vascular network. This resistance can be dynamically altered to regulate the amount of blood flow and hence maintain a balance of chemicals and nutrients in the tissue. The mechanisms responsible for this are primarily local, and are due to various chemical pathways to the vascular smooth muscle cells lining the arterioles and arteries in the vasculature. These cells contract and dilate, which alters the radii and hence resistance of the vessels.

---

<http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/6338>

gives this article, © Austral. Mathematical Soc. 2013. Published May 14, 2013, as part of the Proceedings of the 16th Biennial Computational Techniques and Applications Conference. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

Because of the global coupling induced by the resistive network, the resulting stiff system of ODEs has a dense Jacobian which precludes the direct application of traditional implicit numerical solution methods. We consider a means of remedying this problem by taking a block diagonal Jacobian approximation which allows use of an implicit method but retains the desirable property of explicit solvers of linear solution time scaling with problem size.

*Subject class:* 65L99

*Keywords:* resistive network, stiff, ordinary differential equations, brain, autoregulation

## Contents

<b>1</b>	<b>Introduction</b>	<b>C172</b>
<b>2</b>	<b>Block diagonal Jacobian direct solver</b>	<b>C175</b>
<b>3</b>	<b>Cerebrovascular autoregulation model</b>	<b>C177</b>
<b>4</b>	<b>Discussion and future work</b>	<b>C183</b>
	<b>References</b>	<b>C185</b>

## 1 Introduction

Motivated by the cerebrovascular autoregulation problem described by [David et al. \(2010\)](#), we consider network models whose potential and flow is analogous to those of a resistive network, but whose resistances are determined by a coupled system of ordinary differential equations (ODEs), which in turn depend on the potential and flow throughout the network tree.

Consider a network described by a directed graph. We prescribe a direction for the flows on each of the  $n$  edges and represent the graph by an incidence matrix  $\mathbf{A}$ , whose  $ij$ th entry is  $\mathbf{a}_{ij} = -1$  if edge  $j$  enters node  $i$ ,  $\mathbf{a}_{ij} = 1$  if edge  $j$  exits node  $i$ , and  $\mathbf{a}_{ij} = 0$  otherwise. Each edge  $j$  of the graph is assigned a conductance (reciprocal of resistance)  $\mathbf{g}_j$ . The constitutive relation giving the flow through the edge is Ohm's law: if the difference in potential between the start and end nodes of edge  $j$  is  $w_j$ , then the directed flow through edge  $j$  is  $q_j = \mathbf{g}_j w_j$ , or in matrix form  $\mathbf{q} = \mathbf{G}\mathbf{w} \in \mathbb{R}^n$ , where  $\mathbf{G} = \text{diag } \mathbf{g}$ .

To solve for pressure and flow throughout the network, we define a number of nodes as boundary nodes and prescribe fixed potentials at each of these. We remove the corresponding rows from the incidence matrix to obtain a modified incidence matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  where  $m$  is the number of internal nodes. The potential differences, written in matrix form, are

$$\mathbf{w} = \mathbf{A}^T \mathbf{p} + \mathbf{b}.$$

Flow conservation at each node yields conservation equations

$$\mathbf{A}\mathbf{q} = \mathbf{A}\mathbf{G}\mathbf{w} = 0,$$

and thus the vector of potentials  $\mathbf{p} \in \mathbb{R}^m$  is given by the positive definite linear system

$$\mathbf{A}\mathbf{G}\mathbf{A}^T \mathbf{p} = -\mathbf{A}\mathbf{G}\mathbf{b}. \quad (1)$$

The class of models we consider takes such a network and couples it to a system of ODEs. The differential equations for the state variables  $\mathbf{x} \in \mathbb{R}^l$  may depend on the potentials  $\mathbf{p}$  in the network. The system of ODEs may arise, for example, as a discretisation of a reaction-diffusion partial differential equation. Additionally, the conductances of the edges are algebraically dependent on the state variables of the ODE system, and the boundary potentials  $\mathbf{b}(t)$  may be time varying. We are thus considering semi-explicit index-1 DAE systems of the form

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{f}(t, \mathbf{x}, \mathbf{p}), \\ 0 &= \mathbf{A}\mathbf{G}(\mathbf{x}) [\mathbf{A}^T \mathbf{p} + \mathbf{b}(t)]. \end{aligned} \quad (2)$$

The system can naively be solved by a standard ODE solver by transforming it into an ODE system:

$$\frac{dx}{dt} = f\left(t, x, [AG(x)A^T]^{-1} AG(x)b(t)\right). \quad (3)$$

If the problem does not exhibit stiffness, then an explicit solver, such as a Runge–Kutta method can be used directly to solve (3). The system (1) is sparse and positive definite and, depending on its structure, is efficiently solved by the conjugate gradient method or direct sparse Cholesky factorisation. In other words, function evaluations for this system are relatively cheap. However, if the system is stiff, then explicit methods become highly inefficient and an implicit (or implicit-explicit) method is required, which entails solving a nonlinear system at each time step by Newton’s method. Gear (1971), Iserles (2009) and Shampine and Gordon (1975) provided detailed descriptions of computational considerations required to solve stiff ODEs. The Newton matrix for implicit multistep solvers is

$$M(t, x) = I - \gamma J(t, x), \quad (4)$$

where  $\gamma$  is proportional to the current time step length, and  $J$  is (an approximation to) the Jacobian of the system. Except in special cases where the Jacobian evaluation and solution of  $Mw = d$  is cheap, the modified Newton’s method is typically more computationally efficient (Iserles, 2009, §7.2), whereby  $J$  is held constant over the Newton iteration, and potentially a number of time steps, sacrificing quadratic convergence rather than reevaluating  $J$  and its factorisation, or preconditioner (if using Krylov methods).

The Jacobian of (3) is

$$J = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial p} \frac{\partial p}{\partial g} \frac{\partial g}{\partial x}. \quad (5)$$

While for a typical problem the terms  $\partial f/\partial x$ ,  $\partial f/\partial p$  and  $\partial g/\partial x$  are likely to be sparse, the term  $\partial p/\partial g$  is

$$\frac{\partial p}{\partial g} = -\{A \operatorname{diag}[g(x)] A^T\}^{-1} A \operatorname{diag}(A^T p + b), \quad (6)$$

a dense matrix. Therefore, the system Jacobian  $J$  is also dense, precluding its direct evaluation for a large scale problem.

There are two apparent approaches to modifying the problem to make it amenable to implicit differential equation solvers. The first is to take advantage of the fact that Newton's method tends to be relatively tolerant to errors in the Jacobian, and hence use a sparse approximation to the Jacobian in (4). As long as the approximation  $\hat{J}$  is sufficiently close to the true Jacobian, Newton's method still converges. A mathematical justification of this process is similar to that provided by [Iserles \(2009, §7.2\)](#) to explain the convergence of the modified Newton's method.

The second approach is to use Krylov subspace methods to solve the system  $Mw = d$  that arises in Newton's method using matrix-free Jacobian-vector product approximations ([Knoll and Keyes, 2004](#)) to avoid explicitly formulating the Jacobian. We consider the former approach, and show that for the cerebral autoregulation problem this approach leads to a fully implicit method that scales linearly with the problem size.

## 2 Block diagonal Jacobian direct solver

As already stated, for large scale problems it is not feasible to construct the full Jacobian of the system to use in a direct linear solver. One way to mitigate this problem is to use direct methods to solve the linear system but with a sparse or banded approximation of the Jacobian.

We choose the approximation of  $J$  to have block diagonal structure so the factorisation and solution decompose into as many independent tasks as there are blocks, and hence the decomposition yields a natural parallelisation of the problem. Because of the form of (5), we cannot evaluate block diagonal entries independently, so we form an approximation to the block diagonal entries of the Jacobian that are more efficiently computable.

To form the approximate Jacobian  $\hat{J}$  from some state  $\mathbf{x}$  at time  $\mathbf{t}$ , we first compute the conductances  $\mathbf{g}$  and potentials  $\mathbf{p}$  throughout the network by (1). Then, we group the edges of the graph into  $N$  distinct subsets  $E_1, \dots, E_N$ , and likewise partition the state variables into  $N$  corresponding subsets. The state variables are assumed to be ordered such that the indices of each partition are a convex subset of  $\{1, \dots, l\}$ . Then, each of the vertices which has connecting edges in two or more of the partitions is set to be a constant valued boundary vertex, and is set to the constant potential from the computed  $\mathbf{p}$ . This effectively decouples the network into a number of discrete pieces with constant potential boundaries. Let the indices of the newly formed boundary nodes be denoted  $\mathcal{B}$ , and the remaining active indices  $\mathcal{A}$ . Then let the corresponding vector of boundary potentials be denoted  $\mathbf{p}_{\mathcal{B}}$ , and the active nodes  $\mathbf{p}_{\mathcal{A}}$ . Likewise, denote  $\mathbf{A}_{\mathcal{A}}$  the matrix formed from the rows of  $\mathbf{A}$  corresponding to  $\mathcal{A}$ , and similarly define  $\mathbf{A}_{\mathcal{B}}$ . Then the potential difference equation becomes

$$\mathbf{w} = \mathbf{A}^T \mathbf{p} + \mathbf{b} = \mathbf{A}_{\mathcal{A}}^T \mathbf{p}_{\mathcal{A}} + (\mathbf{A}_{\mathcal{B}}^T \mathbf{p}_{\mathcal{B}} + \mathbf{b}) = \mathbf{A}_{\mathcal{A}}^T \mathbf{p}_{\mathcal{A}} + \hat{\mathbf{b}}. \quad (7)$$

Then, from (6), we form an approximation to  $\partial \mathbf{p} / \partial \mathbf{g}$ , whose rows are determined by

$$\begin{aligned} \frac{\partial \mathbf{p}_{\mathcal{A}}}{\partial \mathbf{g}} &= - \{ \mathbf{A}_{\mathcal{A}} \text{diag} [\mathbf{g}(\mathbf{x})] \mathbf{A}_{\mathcal{A}}^T \}^{-1} \mathbf{A}_{\mathcal{A}} \text{diag} (\mathbf{w}), \\ \frac{\partial \mathbf{p}_{\mathcal{B}}}{\partial \mathbf{g}} &= 0. \end{aligned} \quad (8)$$

We then form our approximate Jacobian from (5), using only the block diagonal entries of  $\partial \mathbf{f} / \partial \mathbf{x}$  and using the approximate  $\partial \mathbf{p} / \partial \mathbf{g}$ . From the method of construction, the product term in (5) automatically has the required block diagonal structure. The Jacobian approximation  $\hat{J}$  is then of the form

$$\hat{J} = \begin{bmatrix} \hat{J}_1 & & 0 \\ & \ddots & \\ 0 & & \hat{J}_N \end{bmatrix},$$

and each block is inverted independently.

For implicit methods using the modified Newton's method, when solving the nonlinear system of equations required to progress  $\mathbf{x}_k$  to  $\mathbf{x}_{k+1}$ , the Newton update is (Iserles, 2009)

$$\mathbf{w}[i + 1] = \mathbf{w}[i] - \mathbf{M}^{-1}\{\mathbf{w}[i] - \beta - \mathbf{h}\mathbf{g}(\mathbf{w}[i])\}, \quad i = 1, 2, \dots, \quad (9)$$

where  $\mathbf{h}$  is the step size. The evaluation of  $\mathbf{g}$  involves evaluating the right hand side of (3), and  $\mathbf{w}$  converges to  $\mathbf{x}_{k+1}$ . For the backward Euler method,

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{t}_{k+1}, \mathbf{x}) \quad \text{and} \quad \beta = \mathbf{x}_k.$$

This formulation is therefore well suited to a parallel environment, as  $\mathbf{M}$  is constructed and inverted independently in parallel, and the Newton iteration steps also proceed in parallel, except for a single function evaluation. So if the evaluation of the right hand sides of the full ODE system (3) is evaluated efficiently, then the problem has a significant amount of parallelism for exploitation.

The potential difficulty in this approach is the choice of partitioning. However, in some applications, such as where there are a number of subnetworks linked by a high conductance 'conduit' network (as is the case in various vascular networks in biology) the choice is straightforward.

### 3 Cerebrovascular autoregulation model

We apply the methodology to a model of cerebrovascular autoregulation. We use a simplified phenomenological model that exhibits qualitatively correct responses to changes in blood pressure, local metabolism, and arterial  $\text{CO}_2$ . This model serves as a test for the numerical procedure. The aim of the project is to eventually incorporate the important biochemical pathways involved in neurovascular coupling (Attwell et al., 2010) to study the function and

pathology of cerebral perfusion at the macroscopic scale. Here, we present an outline of the model structure, omitting the derivations of the equations. Full details of the phenomenological model will be presented in a forthcoming publication. [David and Moore \(2008\)](#), [Alastruey et al. \(2007\)](#), [David et al. \(2010\)](#), and [Farr and David \(2011\)](#) provide more background.

For this cerebrovascular autoregulation application, the network in question is a tree-like structure of blood vessels modelling a portion of the cerebral vasculature branching off the large cerebral arteries. The tree is assumed to be a binary tree, branching to progressively smaller and shorter vessels until reaching the terminal arterioles, of radius approximately  $10\ \mu\text{m}$ , which are then assumed to interface with the capillary bed, a mesh of fine capillaries, which is assumed to be at some constant pressure. The tree bifurcates in a space filling manner which is difficult to specify algorithmically with anatomically correct branching ratios and angles, so, as an approximation to a realistic network, we use a space filling H-tree to describe the geometry of the network. At each bifurcation the radius of the daughter vessels is  $1/\sqrt{2}$  the radius of the parent, and the length is divided by two every second bifurcation. The smallest vessels are defined to have radius  $10\ \mu\text{m}$  and length  $200\ \mu\text{m}$ , and the scale of the model is therefore determined by the number of levels in the tree. A tree of eleven levels, approximately representing a  $6 \times 6\ \text{mm}$  square  $0.2\ \text{mm}$  thick slice of tissue, is depicted in [Figure 1](#). The vessel scalings are of a similar order of magnitude to the more physiologically accurate model of [Ottesen et al. \(2004\)](#).

At the scales of the models, the Womersley number is very small and a Poiseuille approximation for blood flow in a tube is appropriate. In other words, the flow  $\mathbf{q}$  through a vessel is a function of vessel radius  $r$  and length  $l$ , and is linear in the pressure drop  $w$  over the length of the vessel, that is, satisfying Ohm's law and defining a resistive network. After appropriate nondimensionalisation the flow is

$$\mathbf{q} = g\mathbf{w}, \quad \text{where} \quad g = \frac{r^4}{l}. \quad (10)$$

The conductance is highly sensitive to the radius of the vessel. We assume

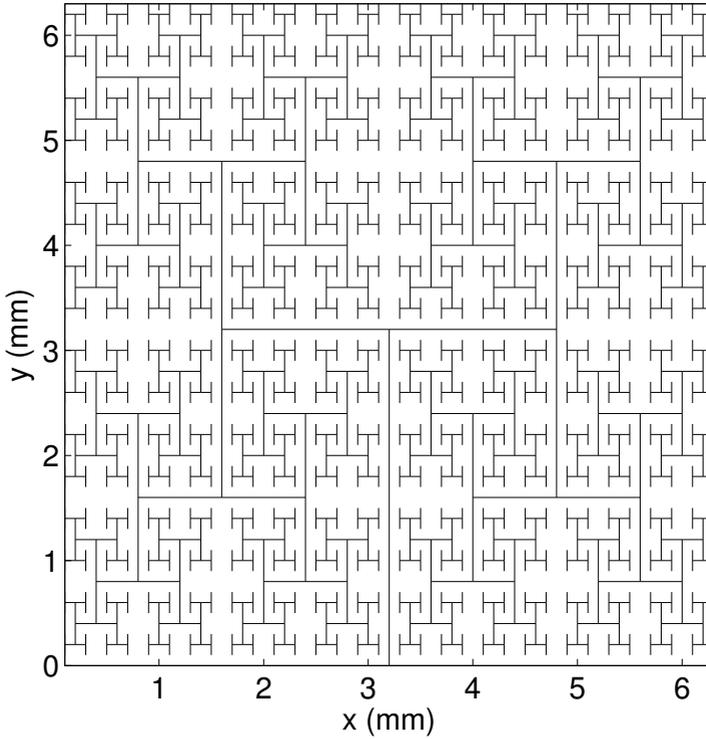


Figure 1:  $6 \times 6 \times 0.2$  mm tissue slice with H-tree vasculature.

a constant capillary bed pressure  $p_c$ , and a potentially time varying pressure  $p_0(t)$  at the root of the tree.

The tree is embedded in a two dimensional spatial domain which is discretised into a number of blocks, one for each terminal arteriole of the vascular network. Each block is modelled by a two-compartment  $\text{CO}_2$  exchange model, one compartment representing the volume fraction of the block occupied by the capillary network, and one compartment representing the tissue. Metabolic consumption of oxygen in the tissue yields  $\text{CO}_2$  as a byproduct, specified as a time and spatially varying rate of  $\text{CO}_2$  production  $\mu(t, x)$ . The capillary walls are permeable to  $\text{CO}_2$  and so we consider passive  $\text{CO}_2$  diffusion across

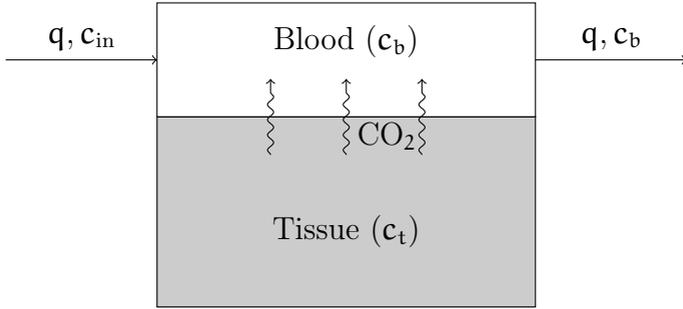


Figure 2: Two-compartment model of  $\text{CO}_2$  transfer in the capillary bed.

the total capillary wall surface area within the block, and transport in and out of the capillary compartment. The tissue model structure is depicted in Figure 2. The dimensionless equations for a tissue block are

$$\begin{aligned} \nu \dot{c}_b &= q(c_{in} - c_b) + \alpha(c_t - c_b), \\ (1 - \nu) \dot{c}_t &= \alpha(c_b - c_t) + \mu(t, x), \end{aligned} \quad (11)$$

where  $\mu$  represents a constant metabolic rate of  $\text{CO}_2$  production due to  $\text{O}_2$  metabolism and where  $q$  is the flow into that block from its associated arteriole.

We model the activity of the smooth muscle by the phenomenological model

$$\dot{f} = -\frac{f - f_\infty(c_t)}{\tau_m}, \quad (12)$$

where  $f_\infty(c_t) : \mathbb{R}^+ \rightarrow [0, 1]$  is a monotonically decreasing sigmoid, to represent that an increased buildup of tissue  $\text{CO}_2$  requires a relaxation of the vessel in order to allow increased convective removal of  $\text{CO}_2$ . The time constant  $\tau_m$  models the rate at which the smooth muscle is recruited or deactivated. Finally, the various components are linked together by a simple model of the arterial wall:

$$\dot{r} = -a_1 r + a_2 p_{\text{Tr}} - a_3 f + a_4. \quad (13)$$

This equation is derived from a balance of the thin walled Laplace stress with the viscoelastic and active stress developed in the wall by the elastic and smooth muscle components of the vessel tissue. Again, full details of the model will be presented in a forthcoming publication. The flow in the vascular tree is most sensitive to relative radius perturbations in the smallest vessels, as it is in these vessels that most of the pressure drop from the input to the capillary bed pressures occurs. We therefore model the autoregulation mechanism only on the smallest vessels, leaving the remaining vessels in the tree with constant radius.

To partition the tree for Jacobian approximation, we split the tree into a number of subtrees, setting the boundary nodes  $\mathbf{p}_B$  where the subtrees interface with the root subtree, as shown in Figure 3. For a large tree, the sparsity of the Jacobian is controlled by specifying the number of levels each subtree spans  $N_S$ .

To test the scalability of the method with problem size, without measuring the complexities of a full featured ODE solver, we ran a simple simulation using a fixed step backward Euler method as the numerical solution, with a constant number of Newton steps per time step, modelling the response of the network to a step increase in input pressure. The approximate Jacobian was created once at the beginning of the simulation and used for the remainder of the simulation. The algorithm was implemented in C, compiled using `gcc 4.7`, and used `CSparse` (Davis, 2006) for the sparse matrix operations (sparse Cholesky and sparse LU decomposition), and code was run in a single thread on a mobile Intel Core i5 (Sandy Bridge) processor with 8 GB of RAM, under Ubuntu 12.04.

We scaled the problem size, that is the total number of levels in the tree, keeping the subtree size constant, and measured the elapsed solution time. Both the Jacobian computation and the Newton iterates scale linearly with problem size (defined as the dimension  $l$  of the state space of (3)), as expected. The total solution time is plotted in Figure 4. Increasing the subtree size for a fixed problem size increases the computation time significantly. This

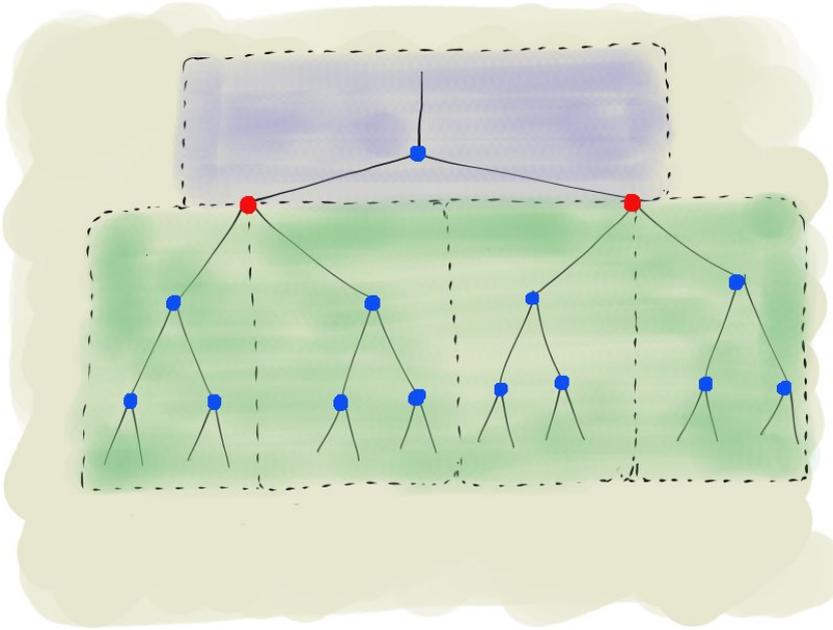


Figure 3: Partitioning of network for Jacobian approximation. The nodes marked in red are the boundary nodes  $\mathbf{p}_B$ , and the nodes marked in blue are the active nodes  $\mathbf{p}_A$ . For clarity, the tree is depicted using a more conventional layout.

is because of the increased memory and computation required to store and invert the larger Jacobian blocks. The curves for subtrees of three and five levels are very similar. This similarity is due to the choice of implementation; the penalty for having many blocks in this implementation is the overhead of calling multiple forward and backward solves, one for each block, rather than storing the larger block matrix as a single entity and calling a single forward/back solve.

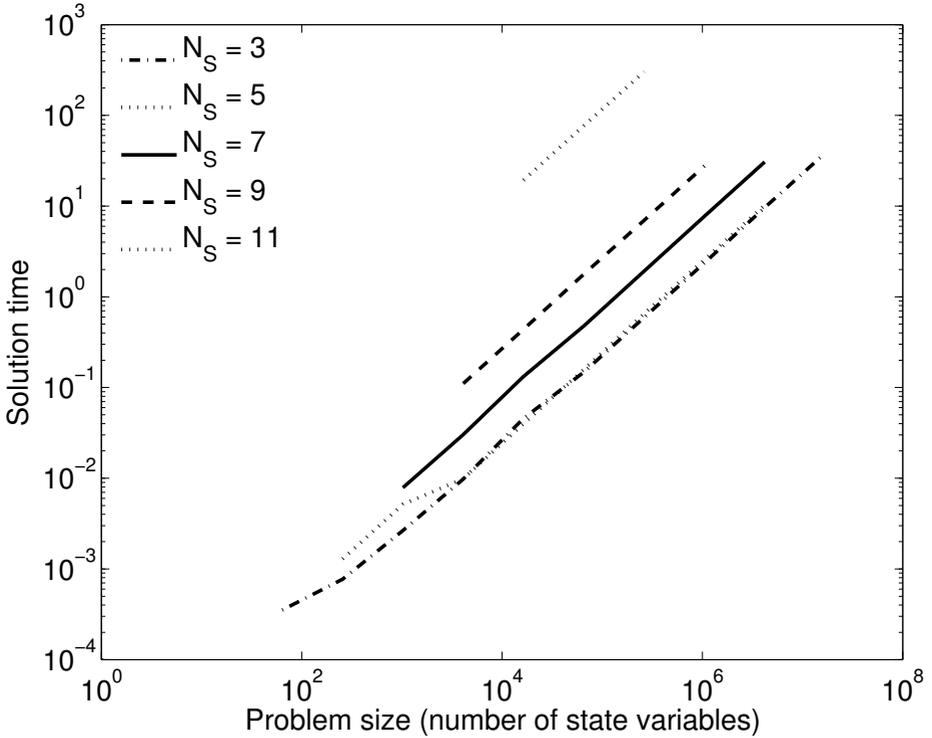


Figure 4: Linear scaling of solution time with problem size. Solution time includes a single Jacobian computation and a number of Newton iterations. The different lines represent different numbers of subtree levels  $N_S$ .

## 4 Discussion and future work

We showed that using a block diagonal Jacobian approximation formed from network partitioning is very effective in a direct linear solver for an implicit ODE method on a large scale resistively coupled stiff dynamic network. We have not investigated the use of Krylov subspace methods, which also avoid evaluating the full Jacobian by using matrix-free Jacobian-vector products. The Jacobian approximation developed in this work is likely to work effectively

as a preconditioner for methods such as those in Sundials CVODE (Cohen and Hindmarsh, 1996; Hindmarsh et al., 2005), and this approach may be well suited to networks where convergence of Newton’s method requires undesirably large Jacobian blocks. Another approach that warrants further exploration is the use of explicit exponential integrators. These are formulated in a Jacobian-free manner and were shown to perform similarly to traditional stiff solvers (Carr et al., 2013).

For biological problems involving dynamically varying vascular trees, the methods presented in this article present an appropriate paradigm, due to the weak coupling between vessels that are a long distance from each other in the tree. This weak coupling occurs because the vessels closer to the root of the tree are of relatively low resistance, so the change in pressure in an upstream vessel due to a downstream flow perturbation in a single vessel is very small, and hence has minimal influence on the other downstream vessels. However, this situation would be markedly different if we considered phenomena like blockage of upstream vessels.

In Section 2 we alluded to the potential for significant parallelism of the problem, and indeed work is in progress to implement a scalable parallel version of the code for a more realistic biological network including detailed cellular models, designed to run on up to 128 cores on a IBM Power 7 SMP machine. In addition to the parallelism inherent in a Newton step due to the Jacobian structure, it is possible to solve the pressure and flow equations (1) with high parallel efficiency with a modest number of computational nodes, but these considerations add considerable extra complexity to the implementation.

**Acknowledgements** I thank anonymous reviewers for their helpful feedback and suggestions.

## References

- J. Alastruey, S. M. Moore, K. H. Parker, T. David, J. Peir, and S. J... Sherwin. Reduced modelling of blood flow in the cerebral circulation : Coupling 1-D , 0-D and cerebral auto-regulation models. *International Journal for Numerical Methods in Fluids*, 2007. doi:[10.1002/fld.1606](https://doi.org/10.1002/fld.1606). C178
- David Attwell, Alastair M. Buchan, Serge Charpak, Martin Lauritzen, Brian a Macvicar, and Eric a Newman. Glial and neuronal control of brain blood flow. *Nature*, 468(7321):232–43, November 2010. doi:[10.1038/nature09613](https://doi.org/10.1038/nature09613). C177
- E. J. Carr, I. W. Turner, and P. Perré. A variable-stepsize Jacobian-free exponential integrator for simulating transport in heterogeneous porous media: Application to wood drying. *Journal of Computational Physics*, 233: 66–82, January 2013. doi:[10.1016/j.jcp.2012.07.024](https://doi.org/10.1016/j.jcp.2012.07.024). C184
- Scott D. Cohen and Alan C. Hindmarsh. CVODE, a stiff/nonstiff ODE solver in C. *Computers in Physics*, 10(2):138–143, 1996. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.5594>. C184
- T. David and S. Moore. Modeling perfusion in the cerebral vasculature. *Medical engineering & physics*, 30(10):1227–45, December 2008. doi:[10.1016/j.medengphy.2008.09.008](https://doi.org/10.1016/j.medengphy.2008.09.008). C178
- T. David, Thomas van Kempen, Huaxiong Huang, and Phillip Wilson. The geometry and dynamics of binary trees. *Mathematics and Computers in Simulation*, 81(7):1464–1481, 2010. doi:[10.1016/j.matcom.2010.04.020](https://doi.org/10.1016/j.matcom.2010.04.020). C172, C178
- Timothy A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial Mathematics, 2006. C181

- Hannah Farr and Tim David. Models of neurovascular coupling via potassium and EET signalling. *Journal of theoretical biology*, July 2011. doi:[10.1016/j.jtbi.2011.07.006](https://doi.org/10.1016/j.jtbi.2011.07.006). C178
- C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall PTR, 1971. C174
- Alan C. Hindmarsh, Peter N. Brown, Keith E. Grant, Steven L. Lee, Radu Serban, Dan E. Shumaker, and Carol S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005. doi:[10.1145/1089014.1089020](https://doi.org/10.1145/1089014.1089020). C184
- Arieh Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, Cambridge, 2 edition, 2009. C174, C175, C177
- D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, January 2004. doi:[10.1016/j.jcp.2003.08.010](https://doi.org/10.1016/j.jcp.2003.08.010). C175
- J. T. Ottesen, M. S. Olufsen, and J. K. Larsen. *Applied Mathematical Models in Human Physiology*. SIAM Monographs on Mathematical Modelling and Computation, 2004. C178
- Lawrence F. Shampine and Marilyn Kay Gordon. *Computer solution of ordinary differential equations : the initial value problem*. San Francisco : W.H. Freeman and Co., 1975. C174

## Author address

1. **Richard G. Brown**, Bluefern High Performance Computing Centre, University of Canterbury, Christchurch, New Zealand