
PDE-based Parameter Reconstruction through Schur and Schwarz Decompositions

Yuan He and David E. Keyes*

Department of Applied Physics & Applied Mathematics, Columbia University,
New York NY, 10027 {yh2030,kd2112}@columbia.edu

1 Introduction

We consider in this work a distributed parameter identification problem for the FitzHugh-Nagumo system of equations of electrocardiology [8]. Specifically, we consider the two-component reaction-diffusion system

$$\begin{aligned} \partial_t u &= \mu \Delta u + u(u - \alpha)(1 - u) - v, & \text{in } Q, \\ \partial_t v &= \kappa \Delta v + \epsilon(\vartheta u - \gamma v), & \text{in } Q, \\ u(\mathbf{x}, 0) &= 0, \quad v(\mathbf{x}, 0) = 0 & \text{in } \Omega, \\ \mathbf{n} \cdot \nabla u(\mathbf{x}, t) &= I(\mathbf{x}, t), \quad \mathbf{n} \cdot \nabla v(\mathbf{x}, t) = 0, & \text{on } \partial Q, \end{aligned} \tag{1}$$

where $\Omega \subset \mathbb{R}^n$, with $n = 2$ for the results in section 4. Q and ∂Q are defined as $Q \equiv \Omega \times (0, T)$ and $\partial Q \equiv \partial\Omega \times (0, T)$, respectively. See [8] for details on system parameters. The objective of the parameter identification is to reconstruct the reactive coefficient $\alpha(\mathbf{x})$ in the first equation from boundary measurements of the electrical potential u .

Our aim here is to present a numerical algorithm that can solve the reconstruction problem in large-scale (parallel) environments. The algorithm is of Newton-Krylov-Schur-Schwarz type; it combines Newton's method for numerical optimization with Krylov subspace solvers for the resulting reduced Karush-Kuhn-Tucker (KKT) systems. Schwarz preconditioning is used to solve the partial differential equations that are involved in the inversion procedure.

2 PDE-constrained Optimization

The parameter identification problem for the FitzHugh-Nagumo system is to reconstruct the physical coefficient $\alpha(\mathbf{x})$ from the knowledge of $h = u(\mathbf{x}, t)$ on the boundary of the domain. Since one can measure the boundary potential for various applied current stimuli I , one thus has access to the time-dependent partial

* This work is supported in part by the U. S. National Science Foundation under CCF-03-52334

Neumann-to-Dirichlet map: $\Lambda : I(\mathbf{x}, t) \rightarrow h$. The parameter identification problem employs knowledge of this map Λ to recover function $\alpha(\mathbf{x})$.

We solve the inverse problem by formulating it as a PDE-constrained optimization problem [1, 4, 5, 6]:

$$\begin{aligned} & \min_{\alpha, u} \mathcal{F}(\alpha, u) \\ & \text{subject to} \quad \mathcal{C}_s(I_s, \alpha, u_s, v_s) = 0, \quad s = 1, 2, \dots, N_s. \end{aligned} \tag{2}$$

where $\mathcal{C}_s(I_s, \alpha, u_s, v_s) = 0$ is abstract notation for (1) with source I_s . N_s is the number of source scenarios producing detectable measurements. The functional to be minimized is defined as

$$\mathcal{F}(\alpha, u) := \frac{1}{2} \sum_{s=1}^{N_s} \sum_{j=1}^{N_d} \int_0^T \int_{\partial\Omega} (u_s - h_s)^2 \delta(\mathbf{x} - \mathbf{x}_j) \, d\sigma(\mathbf{x}) dt + \rho \mathcal{R}(\alpha), \tag{3}$$

with h_s the measurement corresponding to source I_s . \mathbf{x}_j , $j = 1, \dots, N_d$, are detector positions. To simplify notation, we write $u = (u_1, \dots, u_s, \dots, u_{N_s})$. $d\sigma$ denotes the surface measure on $\partial\Omega$. $\mathcal{R}(\alpha)$ is a regularization functional, and the regularization parameter ρ controls the strength of regularization.

The Lagrangian functional for the above minimization problem is

$$\mathcal{L}(u, v, \alpha, \lambda, \eta) = \mathcal{F}(\alpha, u) + \sum_{s=1}^{N_s} \langle (\lambda_s, \eta_s), \mathcal{C}_s \rangle, \tag{4}$$

where again sets of variables corresponding to the full set of different sources, such as $v = (v_1, \dots, v_s, \dots, v_{N_s})$, are implied. λ_s and η_s denote the Lagrangian multipliers (adjoint variables) corresponding to u_s and v_s , respectively. The solution to the constrained minimization problem satisfies the first-order optimality conditions of the Lagrangian functional, which are

$$\begin{aligned} \mathcal{L}_\lambda(u, v, \alpha, \lambda, \eta) &= 0, & \mathcal{L}_\eta(u, v, \alpha, \lambda, \eta) &= 0, \\ \mathcal{L}_u(u, v, \alpha, \lambda, \eta) &= 0, & \mathcal{L}_v(u, v, \alpha, \lambda, \eta) &= 0, \\ \mathcal{L}_\alpha(u, v, \alpha, \lambda, \eta) &= 0. \end{aligned} \tag{5}$$

Denoting $(u, v, \alpha, \lambda, \eta)$ by \mathbf{u} , we can recast (5) as the root-finding problem:

$$\mathcal{L}_{\mathbf{u}}(\mathbf{u}) = 0. \tag{6}$$

3 The Newton-Krylov-Schur-Schwarz Algorithm

In order to solve the optimality equations, a hybrid set of algebraic equations and quasilinear partial differential equations of reaction-diffusion type, we need a discretization of the PDEs and an algebraic solver for the resulting large nonlinear algebraic system. The Newton-Krylov family of methods provides an efficient way to solve such PDE systems [9].

3.1 The Newton-Krylov Method

Newton methods for solving (6) follow the iteration

$$\mathbf{u}_{k+1} = \mathbf{u}_k + l_k \delta \mathbf{u}_k, \tag{7}$$

with some initial guess \mathbf{u}_0 , until convergence criteria are satisfied. The update direction $\delta \mathbf{u}_k$ at Newton iteration k is given by solving the saddle point problem

$$\mathcal{L}_{\mathbf{u}\mathbf{u}}(\mathbf{u}_k) \delta \mathbf{u}_k = -\mathcal{L}_{\mathbf{u}}(\mathbf{u}_k). \tag{8}$$

Here the step length l_k is given by a line search or other globalization technique. The nested iteration is called Newton-Krylov when Krylov subspace methods are used to solve the inner KKT system. The method has received wide attention from practitioners in recent years; see the references cited in [9].

For the FitzHugh-Nagumo model we consider here, the KKT system has the form

$$\begin{pmatrix} \mathcal{L}_{uu} & 0 & \mathcal{L}_{u\alpha} & \mathcal{L}_{u\lambda} & \mathcal{L}_{u\eta} \\ 0 & 0 & 0 & \mathcal{L}_{v\lambda} & \mathcal{L}_{v\eta} \\ \mathcal{L}_{\alpha u} & 0 & \mathcal{L}_{\alpha\alpha} & \mathcal{L}_{\alpha\lambda} & 0 \\ \mathcal{L}_{\lambda u} & \mathcal{L}_{\lambda v} & \mathcal{L}_{\lambda\alpha} & 0 & 0 \\ \mathcal{L}_{\eta u} & \mathcal{L}_{\eta v} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \\ \delta\alpha \\ \delta\lambda \\ \delta\eta \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_u \\ \mathcal{L}_v \\ \mathcal{L}_\alpha \\ \mathcal{L}_\lambda \\ \mathcal{L}_\eta \end{pmatrix}, \tag{9}$$

where $\delta u = [\delta u_1, \delta u_2, \dots, \delta u_{N_s}]^T$, and $\delta v, \delta\lambda, \delta\eta$ and $\mathcal{L}_u, \mathcal{L}_v, \mathcal{L}_\lambda, \mathcal{L}_\eta$ are similarly defined. Because the forward problems for different sources are decoupled, the operator \mathcal{L}_{uu} has diagonal structure:

$$\mathcal{L}_{uu} = \text{diag}\{\mathcal{L}_{u_1 u_1}, \mathcal{L}_{u_2 u_2}, \dots, \mathcal{L}_{u_{N_s} u_{N_s}}\} \tag{10}$$

and similarly for operators $\mathcal{L}_{u\lambda}, \mathcal{L}_{u\eta}, \mathcal{L}_{v\lambda}$, and $\mathcal{L}_{v\eta}$ and their adjoint operators, $\mathcal{L}_{\lambda u}, \mathcal{L}_{\eta u}, \mathcal{L}_{\lambda v}$, and $\mathcal{L}_{\eta v}$. Operators $\mathcal{L}_{u\alpha}$ and $\mathcal{L}_{\lambda\alpha}$ have the structure that $\mathcal{L}_{u\alpha} = [\mathcal{L}_{u_1\alpha}, \mathcal{L}_{u_2\alpha}, \dots, \mathcal{L}_{u_{N_s}\alpha}]^T$ and $\mathcal{L}_{\lambda\alpha} = [\mathcal{L}_{\lambda_1\alpha}, \mathcal{L}_{\lambda_2\alpha}, \dots, \mathcal{L}_{\lambda_{N_s}\alpha}]^T$. $\mathcal{L}_{\alpha u}$ and $\mathcal{L}_{\alpha\lambda}$ are their adjoint operators, respectively.

3.2 The Schur Complement Reduced Space Method

To avoid a huge storage requirement, we do not solve the KKT system (9) directly. Instead, in each Newton iteration, for a given α , we first solve the FitzHugh-Nagumo system (2), which turns out to be the first two equations in (5). We then solve the adjoint problem, (the fourth and fifth equations in (5)), thereupon, the terms $\mathcal{L}_u, \mathcal{L}_v, \mathcal{L}_\lambda$ and \mathcal{L}_η vanish in the KKT system (9). The KKT system thus becomes

$$\begin{pmatrix} \mathcal{L}_{uu} & 0 & \mathcal{L}_{u\alpha} & \mathcal{L}_{u\lambda} & \mathcal{L}_{u\eta} \\ 0 & 0 & 0 & \mathcal{L}_{v\lambda} & \mathcal{L}_{v\eta} \\ \mathcal{L}_{\alpha u} & 0 & \mathcal{L}_{\alpha\alpha} & \mathcal{L}_{\alpha\lambda} & 0 \\ \mathcal{L}_{\lambda u} & \mathcal{L}_{\lambda v} & \mathcal{L}_{\lambda\alpha} & 0 & 0 \\ \mathcal{L}_{\eta u} & \mathcal{L}_{\eta v} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \\ \delta\alpha \\ \delta\lambda \\ \delta\eta \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ \mathcal{L}_\alpha \\ 0 \\ 0 \end{pmatrix}. \tag{11}$$

We can now build the Schur complement of (11) by eliminating $\delta u, \delta v, \delta\lambda$ and $\delta\eta$. We then obtain

$$\mathcal{H}_{red} \delta\alpha = -\mathcal{L}_\alpha, \tag{12}$$

where the reduced gradient \mathcal{L}_α is given by

$$\mathcal{L}_\alpha = \sum_{s=1}^{N_s} \int_0^T \lambda_s u_s (1 - u_s) dt + \rho \mathcal{R}'(\alpha) \tag{13}$$

and the reduced Hessian \mathcal{H}_{red} (Schur complement of the KKT) is given by

$$\mathcal{H}_{red} = \mathcal{L}_{\alpha\alpha} - \mathcal{L}_{\alpha u} W - W^* \mathcal{L}_{u\alpha} + W^* \mathcal{L}_{uu} W, \tag{14}$$

with W defined as $W = [\mathcal{L}_{\lambda u}^{-1} + \mathcal{L}_{\lambda u}^{-1} (\mathcal{L}_{\eta v} - \mathcal{L}_{\eta u} \mathcal{L}_{\lambda u}^{-1})^{-1} \mathcal{L}_{\eta u} \mathcal{L}_{\lambda u}^{-1}] \mathcal{L}_{\lambda \alpha}$. Here W^* denotes the adjoint of W . The reduced Hessian \mathcal{H}_{red} has a much smaller size (and is much denser) than the original Hessian \mathcal{L}_{uu} . It can be verified that $\mathcal{H}_{red} = \mathcal{H}_{red}^*$, that is, \mathcal{H}_{red} is self-adjoint.

One can obtain the Gauss-Newton approximation by dropping second derivative information in the $\mathcal{L}_{\alpha u}$ and $\mathcal{L}_{u\alpha}$ terms [7, 10], resulting in the reduced Hessian:

$$\mathcal{H}_{red}^{GN} = \mathcal{L}_{\alpha\alpha} + W^* \mathcal{L}_{uu} W. \tag{15}$$

Table 1. The reduced-space Newton algorithm

Algorithm 1: Reduced-space Newton algorithm

```

set  $k_{max}, \varepsilon_1, \varepsilon_2$ 
guess  $\alpha_0(\mathbf{x})$ ; set  $k = 0$ 
evaluate  $\mathcal{F}(\alpha_0)$ 
while ( $k < k_{max}$  &  $\frac{\|\mathcal{L}_{\alpha k}\|}{\|1 + \mathcal{F}(\alpha_k)\|} > \varepsilon_1, \frac{\mathcal{F}(\alpha_k)}{\mathcal{F}(\alpha_0)} > \varepsilon_2$ )
    evaluate  $\mathcal{L}_{\alpha k}$  by (13)
    compute  $\delta\alpha_k$  by (12)
    compute  $l_k$  by a line search
     $\alpha_{k+1} = \alpha_k + l_k \delta\alpha_k$ 
    evaluate  $\mathcal{F}(\alpha_{k+1})$ 
     $k = k + 1$ 
end while

```

We thus obtain the following Newton-Krylov-Schur (reduced-space) method as described in Table 1. For full space methods of similar type, see [5, 6].

3.3 The Schwarz Decomposition PDE Solver

In the aforementioned Newton-Krylov-Schur inversion procedure, at each Newton step, many time-dependent PDEs need to be solved. Some of those PDEs are quasi-linear (the FitzHugh-Nagumo system), others are linear (the adjoint equations). The efficiency of the inversion algorithm depends strongly on the efficiency of the algebraic solvers that are used. Our strategy for building an efficient parallel solver is based on the parallel solver toolkit PETSc from Argonne National Laboratory [2]. All the PDEs are passed to the SNES solver in PETSc after being discretized in time by implicit Euler.

4 Numerical Simulations

We present in this section some performance analysis for the algorithm presented above. For detailed analysis on the quality of reconstructions and its relationship with various algorithmic parameters, we refer interested readers to [8]. All the results shown in this section are obtained on the Mac cluster *System X* at the Virginia Polytechnic Institute and State University.

4.1 Performance of Different Solver-preconditioner Combinations

In the first study, we compare the performance of different algebraic solvers and preconditioning methods on our forward model problem.

The algebraic solvers considered here are all Krylov subspace methods (KSP), including the generalized minimal residual (GMRES), modified GMRES, flexible GMRES, conjugate gradient (CG), bi-conjugate gradient (BiCG), and the stabilized version of bi-conjugate gradient squared (BCGS). We refer to [3] for details of those methods. The preconditioning methods we considered include the Jacobi, block Jacobi and the additive Schwarz method. We present in Table 2 the execution time of different combinations. Since many linear systems we encounter in the solution of the forward and inverse problems are indefinite, we use the classical GMRES method with additive Schwarz as the preconditioner in the following sections although there are other combinations that can achieve similar performance as indicated in Table 2.

Table 2. Execution time for the forward model using different KSP accelerators with different preconditioners

| | none | Jacobi | bJacobi | ASM (basic) |
|----------------------|------|--------|---------|-------------|
| GMRES (classical GS) | 89.5 | 90.0 | 81.3 | 67.9 |
| GMRES (modified GS) | 94.7 | 74.2 | 84.5 | 87.2 |
| f GMRES | 91.0 | 77.0 | 68.0 | 87.7 |
| CG | 96.1 | 66.3 | 63.8 | 66.9 |
| BiCG | 88.8 | 67.3 | 80.5 | 88.8 |
| BCGS | 83.6 | 66.3 | 66.4 | 63.0 |

4.2 Scalability Results on the Forward Solver

We now consider parallel performance of the algorithm we have developed on the forward problem. We show in Fig. 1 some fixed-size scaling results obtained by increasing the number of processors with a fixed grid size. The strong speedup and efficiency results based on execution time for two different spatial grid size, 128×128 and 256×256 are presented. As expected, speedup and efficiency improve with problem size.

In Table 3 we show the results on execution time and implementation efficiency ε [5] which is based on the average Mflop/s. We find that the implementation efficiency

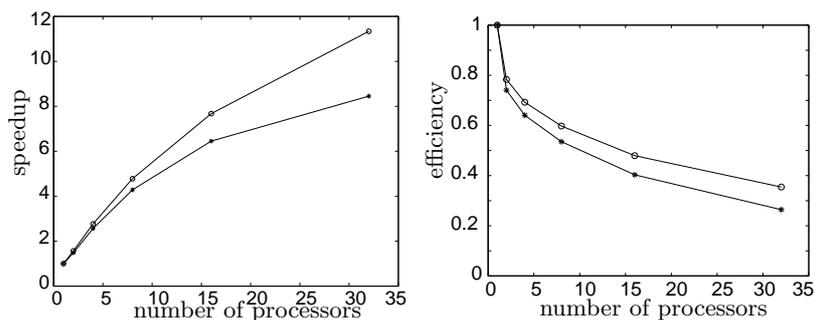


Fig. 1. Strong speedup (left) and efficiency (right) for the forward solver on 128×128 (\star) and 256×256 (\circ) spatial grids, respectively.

Table 3. Performance analysis of the forward solver. NP denotes the number of processors; ε is the implementation efficiency.

| NP | 128×128 | | 256×256 | |
|----|------------------|---------------|------------------|---------------|
| | execution time | ε | execution time | ε |
| 1 | 312.1 | 1.00 | 1881.3 | 1.00 |
| 2 | 219.8 | 0.89 | 1241.0 | 0.79 |
| 4 | 121.8 | 0.84 | 679.4 | 0.73 |
| 8 | 72.9 | 0.75 | 393.6 | 0.62 |
| 16 | 48.4 | 0.58 | 245.2 | 0.50 |
| 32 | 36.9 | 0.54 | 166.0 | 0.36 |

of small size problem is slightly better than the implementation efficiency of the problem of large size.

4.3 Scalability Results on the Inversion Algorithm

We present in Figure 2 the strong speedup and efficiency results for the inversion algorithm for up to 32 processors. We observe by comparing Fig. 2 and Fig. 1 that the scalability of the inversion algorithm is slightly better than that of the forward solver. One explanation of this phenomenon is the forward solver deals with only nonlinear problems (the FitzHugh-Nagumo model), while the inverse solver deals with both nonlinear (forward problem) and linear (adjoint problem). The performance from the linear problem part is better than that from the nonlinear problem part.

In Table 4 we show some results on execution time and the implementation efficiency ε for the inversion algorithm. The implementation efficiency of small problem size is fairly independent of problem size. Again, by comparing with Table 3, the implementation efficiency of the inversion algorithm is slightly better than the implementation efficiency of forward model.

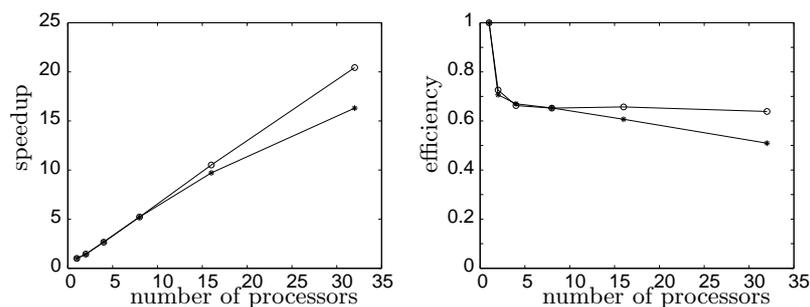


Fig. 2. Strong speedup (left) and efficiency (right) for the inversion algorithm on 128×128 (\star) and 256×256 (\circ) spatial grids, respectively.

Table 4. Performance analysis of the inverse solver. NP denotes the number of processors; ε is the implementation efficiency.

| NP | 128×128 | | 256×256 | |
|----|------------------|---------------|------------------|---------------|
| | execution time | ε | execution time | ε |
| 1 | 9914.0 | 1.00 | 72728.9 | 1.00 |
| 2 | 7015.5 | 0.79 | 50101.0 | 0.70 |
| 4 | 3701.6 | 0.85 | 27743.1 | 0.73 |
| 8 | 1897.7 | 0.89 | 13937.0 | 0.75 |
| 16 | 1021.2 | 0.85 | 6916.2 | 0.79 |
| 32 | 608.2 | 0.75 | 3558.1 | 0.76 |

5 Conclusion

We have presented in limited space a parallel numerical algorithm for a PDE-based distributed parameter reconstruction problem. This Newton-Krylov algorithm combines Newton's method for numerical optimization with Krylov subspace solvers for the resulting KKT system. We have also discussed the performance of both the forward solver and the inversion algorithm. Physical results of the inversion are available in [8].

Future research will focus on accelerating the current code, extending it to three dimensions on much larger numbers of processors, and comparing simulations on more realistic geometries with experimental measurements.

References

- [1] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloemen Waanders. Parallel algorithms for PDE-constrained optimization. In M. Heroux, P. Raghaven, and H. Simon, editors, *Frontiers of Parallel Computing*. SIAM, 2006.
- [2] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Homepage, 2007. <http://www.mcs.anl.gov/petsc>.

- [3] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, 2nd edition, 1994.
- [4] L. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, editors. *Large-Scale PDE-Constrained Optimization*. Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin, 2003.
- [5] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. *SIAM J. Sci. Comput.*, 27:687–713, 2005.
- [6] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange-Newton solver and its application to optimal control of steady viscous flows. *SIAM J. Sci. Comput.*, 27:714–739, 2005.
- [7] E. Haber, U. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16:1263–1280, 2000.
- [8] Y. He and D. E. Keyes. Reconstructing parameters of the FitzHugh-Nagumo system from boundary potential measurements. *J. Comput. Neurosci.*, 23(2), 2007.
- [9] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. *J. Comput. Phys.*, 193:357–397, 2004.
- [10] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.