

PARAMETER IDENTIFICATION VIA THE ADJOINT METHOD: APPLICATION TO PROTEIN REGULATORY NETWORKS

Robin L. Raffard * **Keith Amonlirdviman** **
Jeffrey D. Axelrod *** **Claire J. Tomlin** ****

* *Department of Aeronautics and Astronautics, Stanford
University, CA 94305-4035 USA.*

`rraffard@stanford.edu`

** *Department of Aeronautics and Astronautics, Stanford
University, CA 94305-4035 USA. amon@stanford.edu*

*** *Department of Pathology, Stanford University School of
Medicine, Stanford CA 94305-5324 USA.*

`jaxelrod@stanford.edu`

**** *Department of Aeronautics and Astronautics, Stanford
University, CA 94305-4035 USA. tomlin@stanford.edu*

Abstract: An adjoint-based algorithm for performing automatic parameter identification on differential equation based models of biological systems is presented. The algorithm solves an optimization problem, in which the cost reflects the deviation between the observed data and the output of the parameterized mathematical model, and the constraints reflect the governing parameterized equations themselves. Preliminary results of the application of this algorithm to a previously presented mathematical model of planar cell polarity signaling in the wings of *Drosophila melanogaster* are presented.

1. INTRODUCTION

A key problem in systems biology is the identification of parameters in the mathematical models that describe biological systems. This problem is generally difficult due to both the number of state variables and parameters, and the fact that the governing equations are usually nonlinear functions of these states and parameters. It is also advantageous to perform the parameter identification problem relatively quickly, since this allows one to efficiently test the feasibility of different mathematical models.

In this paper, we present an algorithm for performing automatic parameter identification on differential equation based models of biological systems. The algorithm attempts to minimize an objective function which encodes the deviation between the observed data of the system and the output of the parameterized model, with the governing parameterized equations forming the constraints of this optimization problem. The algorithm relies on the adjoint method, which calculates the gradient of the objective function with respect to the unknown parameters, essentially describing analytically how to minimize the objective by varying the parameters. We augment this gradient based method by using additional information provided by the derivative of the gradient to give well-conditioned optimization even when the optimal parameter values are several

¹ This research was supported by DARPA under the BioComp program, by NIH under grant R01 GM075311-01, and by a Stanford Bio-X IIP Award.

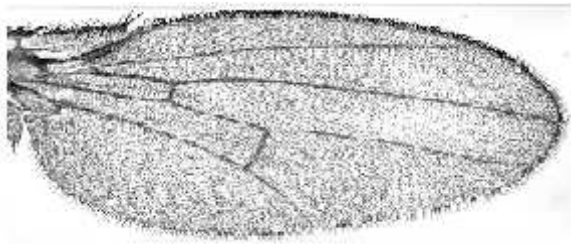


Fig. 1. *Drosophila* adult wing epithelium. Proximal edge is to the left, distal edge is to the right.

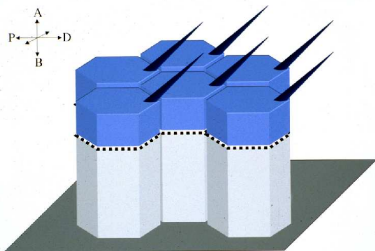


Fig. 2. Diagram shows that each epithelial cell constructs a hair that protrudes from its distal vertex and points distally, creating a virtually error free parallel array.

orders of magnitude different from each other. We present preliminary results of this algorithm on a previously described mathematical model (Amonlirdviman et al. (2005)) of the signaling network regulating the planar cell polarity of *Drosophila* wing epithelial cells orthogonal to their apical-basal axes. This network is termed planar cell polarity (PCP).

2. PLANAR CELL POLARITY (PCP)

In adult *Drosophila*, each epithelial cell on the wing produces a single hair, or trichome. The hairs grow from the distal edge (edge of the cell closest to the wing tip) of each cell and all point in the same direction, towards the wing tip, as shown in Figures 1 and 2 (note that all images in this paper follow the convention that the proximal side of the cell/wing is to the left of the image, distal is to the right). Genetic analyses have identified a group of proteins that are required to correctly polarize these arrays (Adler (2002), Strutt (2002)), and the regular array of hairs is caused by spatially asymmetric distributions of these proteins in the plane of the epithelium. The process by which the proteins controlling hair polarization localize to different areas within each cell during the development of the fly is called planar cell polarity (PCP) signaling. The wing epithelial cells aggregate in a hexagonal close-packed array (Figure 2).

In the presence of cell clones mutant for some PCP genes, the hair polarity in neighboring wild-type cells is disrupted, a phenomenon termed domineering non-autonomy. Domineering non-autonomy reverses hair orientation on either the proximal or distal side of the clone in a manner characteristic to the particular mutant protein. Based on the available biological data, a feedback loop mechanism describing the interaction of a group of PCP molecules was proposed to mediate PCP signaling (Axelrod (2001), and Tree et al. (2002)). The signaling diagram is drawn schematically in Figure 3, in which an arrow indicates a positive influence, and a line indicates a negative influence. The diagram describes the following: Frizzled (Fz), a membrane protein, promotes the localization of Disheveled (Dsh), a cytoplasmic protein, to a membrane; Dsh stabilizes Fz location; Fz promotes the localization of Van Gogh (Vang), a membrane protein, and Prickle (Pk), a cytoplasmic protein, on the membrane of a neighboring cell; Pk and Vang inhibit the recruitment of Dsh to a membrane. Experimentally, it has been observed that, in steady state, Dsh and Fz proteins localize to the distal edge and Pk and Vang to the proximal edge of all cells in the array, thus the large font indicates that the wild type protein localizes at this location. It is believed that the hair grows at the site of the highest concentration of Dsh protein.

A mathematical model based on the feedback loop model (Tree et al. (2002)) and a global directional cue that biases the direction toward which the feedback loop orients (Yang et al. (2002), Ma et al. (2003)) was used to demonstrate, through simulation, the feasibility of the model to reproduce all of the most characteristic PCP phenotypes (Amonlirdviman et al. (2005)). The logic of the feedback loop is encoded in the mathematical model by representing interactions as binding to form protein complexes. For example, the interaction between Fz and Dsh is represented as a reaction forming the complex DshFz, which can interact with other proteins and complexes, and it can undergo a backward reaction that separates it back into its components Fz and Dsh. The mathematical model includes the four original proteins, as well as six complexes, the last four of which form across the cell boundary with the adjacent cell: DshFz, VangPk, FzVang, DshFzVang, FzVangPk, DshFzVangPk. While positive influences are encoded by complex formation, negative influences are through terms that aid the reverse reaction. The state variables of the mathematical model are the local concentrations of these proteins (for example $[Fz]$ represents the concentration of Fz) which are assumed to be continuous. The mathematical model assumes that protein molecules move by diffusion: Dsh and Pk diffuse within the

cell interior, while Vang, Fz, and all the complexes diffuse only in the membrane (or shared membranes).

The mathematical model is represented by ten reaction-diffusion partial differential equations (PDEs). All of the model parameters, including reaction rates, diffusion constants and initial protein concentrations were not directly observable from the available data, so parameter values were identified by being constrained to result in the desired qualitative features of the hair pattern phenotypes. The Nelder-Mead simplex method (Nelder and Mead (1965)) was used to attempt to minimize an objective function composed of quadratic penalty functions corresponding to these feature constraints to produce a feasible solution set of parameters. The model includes 37 parameters, and each evaluation of the objective function required 13 runs of the model simulation corresponding to each of the experimental cases that the model was meant to reproduce. The complete development of the model and results of this analysis are available in Amonlirdviman et al. (2005).

3. THE PARAMETER IDENTIFICATION PROBLEM

In the current work, we strive to replace the simplex method with a more efficient optimization method. As we have described above, the governing equations for PCP consist of ten reaction-diffusion equations describing the time and space evolution of the concentrations of the four PCP proteins and six of their complexes. If $x(t, s) = ([\text{Dsh}], [\text{Pk}], [\text{Fz}], [\text{Vang}], [\text{DshFz}], [\text{VangPk}], [\text{FzVang}], [\text{DshFzVang}], [\text{FzVangPk}], [\text{DshFzVangPk}]) \in \mathbb{R}^{10}$ represents the vector of all protein concentrations, and if $s \in \Omega = \mathcal{V} \times \partial\mathcal{V}$ represents all space variables (covering both the cytoplasm \mathcal{V} and the membrane $\partial\mathcal{V}$), the governing equations can be written in the following compact form:

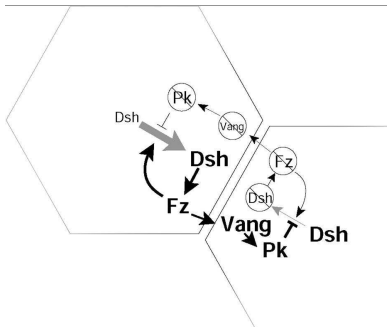


Fig. 3. Four protein PCP signaling network.

$$\frac{\partial x(t, s)}{\partial t} = P(s, x(t, s), \theta) + \mu(\theta) \Delta x(t, s), \quad \forall s \in \Omega \quad (1)$$

which means that the rate of change of each protein concentration is equal to its net rate of production $P(s, x(t, s), \theta)$, plus its rate of diffusion $\mu(\theta) \Delta x(t, s)$.

If protein i reacts with protein j to form complex k , P_i is a function of the type $R_i x_i x_j - \lambda_i x_k$ - it includes more reaction terms if protein i is present in more than one reaction. The forward rates of reaction R and the backward rates of reaction λ are stored in the parameter $\theta \in \mathbb{R}^{37}$, which has to be estimated. Finally, $\mu(\theta)$ is the constant of diffusion of each protein and Δ represents the Laplacian operator.

Eight of the ten proteins diffuse in the membrane. Therefore their reaction-diffusion equations are specified on a periodic domain and do not require boundary conditions. However, Dsh and Pk diffuse in the interior of the cell, which is a finite domain, and therefore require boundary conditions noted in compact form

$$\mu(\theta) \nabla_s x(t, s) \cdot n = CP(s, x(t, s), \theta), \quad \forall s \in \partial\Omega \quad (2)$$

in which n represents the unit normal vector to the membrane. The matrix C is a 10×10 matrix with all zero entries, except the first two diagonal elements (corresponding to Dsh and Pk) which are equal to one. C filters the last eight proteins, for which the diffusion across the membrane is zero. For Dsh and Pk, the rate of diffusion across the membrane is equal to their rate of production.

Experimental data consist of pictures of hair polarity which are provided at final time T , taken to be at the end of the signaling process. In our PCP model, hair polarity is predicted based on the Dsh concentration in the cells and is stored in a vector Y^{model} comprising as many entries as simulated cells, and calculated by

$$Y^{\text{model}} = \int_{\Omega} h(x(s, T), s) ds, \quad (3)$$

in which h is a differentiable function, which gives a score of 1 to a cell with Dsh localization on the distal side, -1 to a cell with Dsh localization on the proximal side and 0 with no Dsh localization. Similarly the data Y^{obs} is a vector with entries ranging from -1 to 1; -1 for cells with reverse polarity and 1 for cells with polarity. The problem of identifying the unknown parameters is the one of finding, among our parametrized set of PCP models, the model which best explains the experimental data. Therefore, it consists of minimizing the prediction error, *i.e.*, the deviation between the observed

data and the output of the parametrized model. Mathematically, it reads

$$\begin{aligned} & \text{minimize } J(\theta) = \|Y^{\text{model}} - Y^{\text{obs}}\| \\ & \text{subject to } \frac{\partial x(t, s)}{\partial t} = P(s, x(t, s), \theta) \\ & \quad \quad \quad + \mu(\theta)\Delta x(t, s) \end{aligned} \quad (4)$$

Usually, the norm $\|\cdot\|$ is chosen as a quadratic norm. Besides its mathematical convenience, such a norm is often chosen because it recovers the maximum likelihood criterion. Indeed, suppose measurements are stochastic data consisting of the sum of the true model outcome and normally distributed noise:

$Y^{\text{obs}} = \int_{\Omega} h(x^{\text{true}}(s, T), s) + v$, where v is a normal random variable with mean 0 and covariance Σ .

The likelihood of the observations is equal to

$$\begin{aligned} \text{PDF}(v = Y^{\text{obs}} - Y^{\text{model}}) &= \frac{1}{(2\pi)^N \det \Sigma} \\ \exp\left(-\frac{1}{2}(Y^{\text{obs}} - Y^{\text{model}})\Sigma^{-1}(Y^{\text{obs}} - Y^{\text{model}})\right) \end{aligned} \quad (5)$$

where PDF refers to probability density function and is a Gaussian in the present case. The parameter which maximizes the likelihood of the observations is then given by

$$\begin{aligned} \theta^* &= \arg \max \{ \exp(\|(Y^{\text{obs}} - Y^{\text{model}})\|_{\Sigma^{-1}}^2) \} \\ &= \arg \min \{ \|(Y^{\text{obs}} - Y^{\text{model}})\|_{\Sigma^{-1}}^2 \} \end{aligned} \quad (6)$$

We will assume, in the remainder of the paper, that Σ is the identity matrix.

4. SOLUTION METHOD VIA OPTIMAL CONTROL THEORY

The parameter identification problem consists of an optimization program in which the variables are constrained by a PDE. In this section, we will show how to efficiently solve such a problem.

4.1 Gradient computation

Many optimization algorithms rely on descent methods, which require the computation of the gradient of the objective function. For the case of PDE optimization programs, calculating the gradient can be efficiently done via a version of the adjoint method, which was developed by Jameson (1998) largely for use in nonlinear aerodynamic optimization problems. We will first review the adjoint method.

4.1.1. Adjoint method Let us consider an objective function J given by

$$J(\theta) = f(x, \theta) \quad (7)$$

where f is a differentiable function and x is the solution of a differential equation (DE), noted

$$D(x, \theta) = 0 \quad (8)$$

Under technical conditions (see Lions (1971) for more details), the function J is differentiable and

$$\lim_{h \rightarrow 0} \frac{J(\theta + h\tilde{\theta}) - J(\theta)}{h} = \nabla_x f(x, \theta)\tilde{x} + \nabla_{\theta} f(x, \theta)\tilde{\theta} \quad (9)$$

in which \tilde{x} is the solution of the linearized form of the original differential equation (8)

$$\nabla_x D(x, \theta)\tilde{x} + \nabla_{\theta} D(x, \theta)\tilde{\theta} = 0 \quad (10)$$

At this stage, computing the derivative in each direction, θ , requires one to solve the DE (10) for each of these directions and then form the derivative according to (9).

The adjoint method allows one to obtain the derivative in all directions – in other words, the gradient, by computing the solutions to only two DEs. It proceeds as follows: taking the inner product with an arbitrary costate q (lying in the same function space as x), we obtain

$$q \cdot \nabla_x D(x, \theta)\tilde{x} + q \cdot \nabla_{\theta} D(x, \theta)\tilde{\theta} = 0 \quad (11)$$

Adding this term to the derivative, we obtain,

$$\begin{aligned} \lim_{h \rightarrow 0} \frac{J(\theta + h\tilde{\theta}) - J(\theta)}{h} &= (\nabla_x f(x, \theta) + \\ & q \cdot \nabla_x D(x, \theta)\tilde{x} + (\nabla_{\theta} f(x, \theta) + q \cdot \nabla_{\theta} D(x, \theta))\tilde{\theta} \end{aligned} \quad (12)$$

Choosing q so as to cancel the effect of the state perturbation

$$\nabla_x f(x, \theta) + \nabla_x D(x, \theta) \cdot q = 0 \quad , \quad (13)$$

the derivative in any direction $\tilde{\theta}$ is $(\nabla_{\theta} f(x, \theta) + q \cdot \nabla_{\theta} D(x, \theta))\tilde{\theta}$ and therefore the gradient is

$$\nabla J(\theta) = \nabla_{\theta} f(x, \theta) + q \cdot \nabla_{\theta} D(x, \theta) \quad (14)$$

With the gradient in hand, it is now possible to perform a descent algorithm, called the quasi-Newton method, for which we will see an effective illustration in section 4.2.

4.1.2. *Adjoint equations for PCP* The method presented in the previous section is systematic and can be followed step by step for the PCP model. The regularity of the PCP partial differential equations (PDEs) provides us with enough technical conditions to compute the derivative of J as follows

$$\lim_{h \rightarrow 0} \frac{J(\theta + h\tilde{\theta}) - J(\theta)}{h} = 2 \left(\int_{\Omega} h(x(s, T), s) ds - Y^{\text{obs}} \right)^T \int_{\Omega} \nabla_x h(x(s, T), s) \tilde{x}(s, T) ds ; \quad (15)$$

in which \tilde{x} is the solution of the following linear PDE

$$\frac{\partial \tilde{x}}{\partial t}(t, s) = \nabla_x P(s, x(t, s), \theta) \tilde{x}(t, s) + \mu(\theta) \Delta \tilde{x}(t, s) + (\nabla_{\theta} P(s, x(t, s), \theta) + \Delta x(t, s) \nabla \mu(\theta)) \tilde{\theta} \quad (16)$$

With linear boundary conditions

$$\nabla \mu(\theta) \tilde{\theta} \nabla_s x(t, s) \cdot n + \mu(\theta) \nabla_s \tilde{x}(t, s) \cdot n = C(\nabla_x P(s, x(t, s), \theta) \tilde{x}(t, s) + \nabla_{\theta} P(s, x(t, s), \theta) \tilde{\theta}) \quad (17)$$

Taking the inner product of this linear PDE with an arbitrary costate q

$$\int_{\Omega} \int_0^T q^T \frac{\partial \tilde{x}}{\partial t}(t, s) = \int_{\Omega} \int_0^T q^T (\nabla_x P \tilde{x} + \mu(\theta) \Delta \tilde{x}(t, s)) + \int_{\Omega} \int_0^T q^T (\nabla_{\theta} P + \Delta x \nabla \mu(\theta)) \tilde{\theta} \quad (18)$$

Integrating by parts,

$$\begin{aligned} \int_{\Omega} q^T \tilde{x}(T) &= \int_{\Omega} \int_0^T \tilde{x}^T \left(\frac{\partial q}{\partial t} + \nabla_x P^T q + \mu(\theta) \Delta q \right) \\ &\quad + \int_{\Omega} \int_0^T q^T (\nabla_{\theta} P + \Delta x \nabla \mu(\theta)) \tilde{\theta} \\ &\quad + \int_{\partial \Omega} \int_0^T \tilde{x}^T (\nabla_x P^T C^T q - \mu(\theta) \nabla_s q \cdot n) \\ &\quad + \int_{\partial \Omega} q^T (\nabla_{\theta} P - \nabla_s x \cdot n \nabla \mu(\theta)) \tilde{\theta} \end{aligned} \quad (19)$$

We are now in a position to extract the gradient of J . Provided that q solves the following linear PDE

$$-\frac{\partial q}{\partial t} = \nabla_x P^T q + \mu(\theta) \Delta q \quad (20)$$

with boundary conditions

$$\mu(\theta) \nabla_s q \cdot n = \nabla_x P^T D^T q \quad (21)$$

and terminal condition

$$q(s, T) = 2 \begin{pmatrix} \int_{\Omega} h(x(s, T), s) - Y^{\text{obs}} \\ \nabla_x h(x(s, T), s) \end{pmatrix}^T \quad (22)$$

the gradient is

$$\begin{aligned} \nabla J &= \int_{\Omega} \int_0^T (\nabla_{\theta} P + \Delta x \nabla \mu(\theta))^T q \\ &\quad + \int_{\partial \Omega} (\nabla_{\theta} P - \nabla_s x \cdot n \nabla \mu(\theta))^T q \tilde{\theta} \end{aligned} \quad (23)$$

4.2 Second order method

The gradient algorithm is numerically efficient when the problem is well conditioned, meaning that the derivatives in all the directions have the same order of magnitude. In the case of PCP, the parameters are unknown and may range over several orders of magnitude. Therefore the problem is likely to be poorly conditioned, in which case a second order method is preferable. A second order method, such as the Newton method, rescales the variables so that in the new system of variables the problem is well conditioned and consequently the descent algorithm is fast, yet no tractable method currently exists for executing the Newton method in optimization programs involving general PDEs. However, it is possible to implement a quasi-Newton method (Gill et al. (1999)), in which the second order derivative of the objective function, called the Hessian, is computed via finite differences on the gradient.

By doing so, we can form an approximate Hessian H and the descent direction is taken as the one which minimizes the quadratic approximation of the objective function: $\delta \theta = -H^{-1} \nabla J$.

4.3 Summary of the algorithm

Algorithm 1. (2nd order adjoint based algo.). Start with an initial guess for the parameters θ^{guess} and an initial guess for the Hessian H^{guess} .

Repeat

- (1) Solve the governing equation (1) for x , using the current parameter vector θ .
- (2) Solve the adjoint equation (20) for q , using the current θ and x .
- (3) Determine the gradient ∇J according to equation (23).
- (4) Update the Hessian H via finite difference between the current gradient and the previous ones.
- (5) Form the descent direction $\Delta \theta = -H^{-1} \nabla J$.

- (6) Line search: compute $\beta > 0$ so that $J(\theta + \beta\Delta\theta)$ is minimized.
- (7) Update $\theta := \theta + \beta\Delta\theta$.

Terminate when $\nabla J^T H \nabla J$ is small

Return $\theta^* = \theta$.

4.4 Computational complexity

The adjoint method drastically reduces the complexity of the gradient computation. It only requires two PDE calculations, whereas calculating the gradient via finite difference would have required at least $d + 1$ PDE computations, in which d is the number of parameters to estimate ($d = 37$ in our PCP model). Each iteration of the algorithm moreover consists of a coarse one-dimensional minimization (line search), which is typically terminated after three to six PDE (1) computations. In total and from a conservative view-point, each iteration requires eight to ten times the computational time of running the governing PDE (1). Finally, in terms of convergence, the algorithm generally terminates after 50 iterations; therefore the algorithm requires on the order of 50×10 objection function evaluations.

5. PRELIMINARY SIMULATION RESULTS

An important validation of the algorithm is to make sure that it efficiently searches the parameter space. For this purpose, we present preliminary simulation results for the wild type case. For this simulation, we used a simplified version of the PDE model presented in Amonlirdviman et al. (2005), which assumes that the diffusion terms in the equations can be replaced by their quasi-steady-state solutions. This assumption permits elimination of the diffusion terms from the original PDEs, reducing the model to a system of ordinary differential equations (ODEs) in which the number of parameters to identify is reduced from 37 to 27. The complete ODE model development is presented in Amonlirdviman (2005) and Ma et al. (2005). The adjoint method and quasi-Newton algorithm applied is that presented in the previous section.

In these simulations, we assume some true values of the parameters, which are believed to generate a phenotype consistent with the characteristic PCP phenotypes (termed “true”). Then, we deviate from these parameters and we verify that the output of the search algorithm recovers the true phenotype, which we want to match. In practice, we set $\theta^{\text{guess}} = \theta^{\text{true}}(1 + \sigma\mathcal{N}(0, 1))$ with $\sigma = 1$ and we first run a simplex algorithm, not taking advantage of any gradient information, and second, our quasi-Newton method. The results are shown

in Figure 4. In this Figure, we display concentrations of Dsh protein, with “cool” colors representing relatively low concentrations, and “warm” colors representing relatively high concentrations. This result indicates that given the same amount of computational effort, the adjoint-based quasi-Newton method has almost recovered the true phenotype, whereas the simplex method is still far from converging.

For the quasi-Newton method, the computation involved for 30 function evaluations is 60 ODE calculations. We note that here, we are only matching the wild type phenotype. To match wild type and all mutant phenotypes as in Amonlirdviman et al. (2005), we would parallelize the computation; and the total computational time would be equal to the number of ODE computations multiplied by the time to compute the slowest phenotype.

This is the subject of our current work.

ACKNOWLEDGEMENTS

We would like to thank Professor Jonathan Goodman for his help on the optimization procedure.

REFERENCES

- P. N. Adler. Planar signaling and morphogenesis in *drosophila*. *Developmental Cell*, 2(5):525–535, 2002.
- K. Amonlirdviman. *Mathematical Modeling of Planar Cell Polarity Signaling in the Drosophila Melanogaster Wing*. PhD thesis, Stanford University, August 2005.
- K. Amonlirdviman, N. A. Khare, D. R. P. Tree, W.-S. Chen, J. D. Axelrod, and C. J. Tomlin. Mathematical modeling of planar cell polarity to understand domineering nonautonomy. *Science*, 307(5708):423–426, January 2005.
- J. D. Axelrod. Unipolar membrane association of Dishevelled mediates Frizzled planar cell polarity signaling. *Genes Dev.*, 15(10):1182–1187, May 2001.
- L. Evans. *Partial Differential Equations*. AMS Press, 2002.
- P. E. Gill and M. W. Leonard. Reduced-hessian quasi-newton methods for unconstrained optimization. *SIAM J. Optim. Vol. 12, No 1, pp.209-237*, 2001.
- P. E. Gill, W. Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press. Harcourt Brace and Company, 1999.
- A. Jameson. Aerodynamic design via control theory. *Princeton University Report MAE 1824, ICASE Report No. 88-64, November 1988, also, J. of Scientific Computing, Vol. 3, 1988, pp. 233-260*, 1998.

- J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. translated by S.K. Mitter, Springer Verlag, New York, 1971.
- D. Ma, K. Amonlirdviman, C. J. Tomlin, and J. D. Axelrod. Irregularities in cell packing necessitate a robust planar cell polarity signaling mechanism. 2005. Submitted.
- D. Ma, C.-H. Yang, H. McNeill, M. A. Simon, and J. D. Axelrod. Fidelity in planar cell polarity signalling. *Nature*, 421:543–547, 2003.
- J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7: 308–313, 1965.
- D. I. Strutt. The asymmetric subcellular localisation of components of the planar polarity pathway. *Semin. Cell Dev. Biology*, 13(3):225–231, 2002.
- D. R. Tree, J. M. Shulman, R. Rousset, M. P. Scott, D. Gubb, and J. D. Axelrod. Prickle mediates feedback amplification to generate asymmetric planar cell polarity signaling. *Cell*, 109 (3):371–381, May 2002.
- C.-H. Yang, J. D. Axelrod, and M. A. Simon. Regulation of frizzled by fat-like cadherins during planar polarity signaling in the *drosophila* compound eye. *Cell*, 108(5):675–688, 2002.

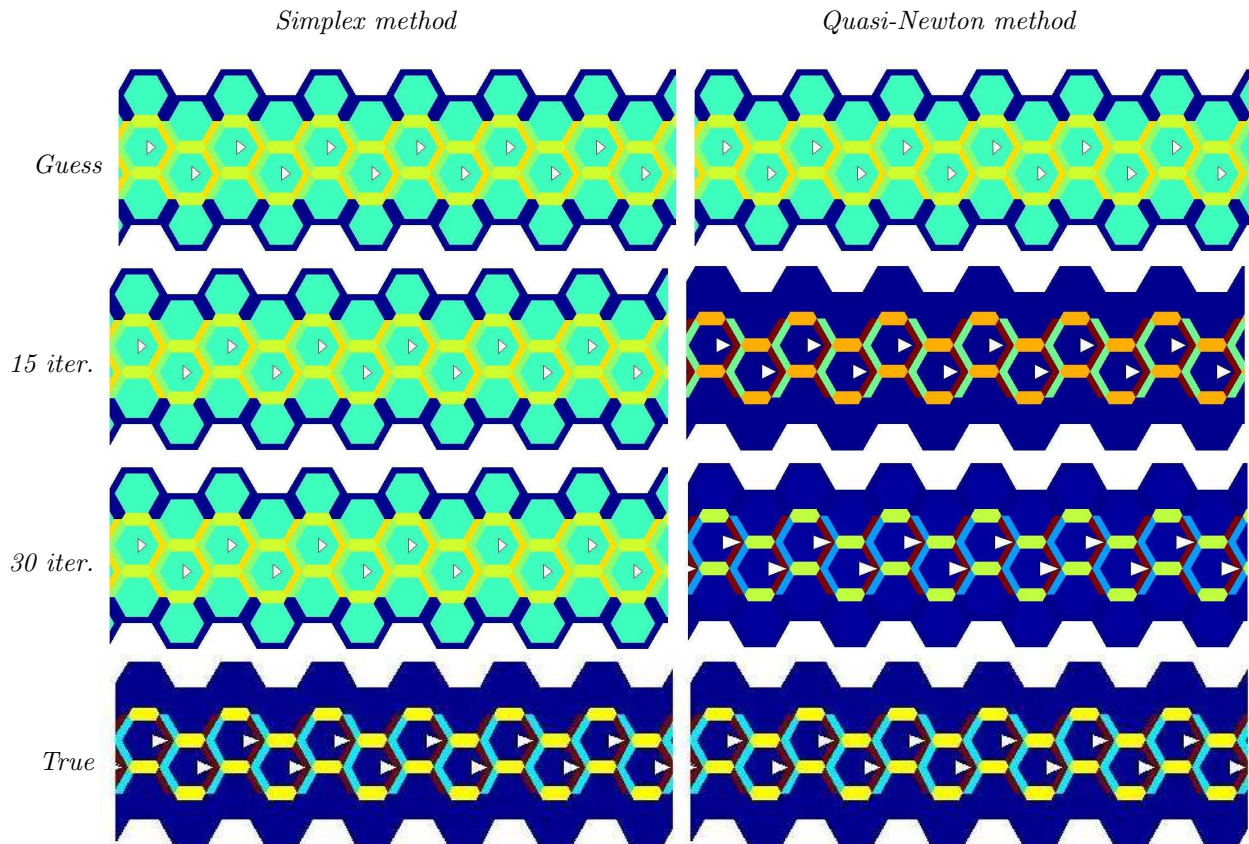


Fig. 4. Comparison between the simplex method and the quasi-Newton method described here for the parameter identification problem. After 30 iterations, the quasi-Newton method has almost recovered the true phenotype.