Editor: Dianne P. O'Leary, oleary@cs.umd.edu

ELASTOPLASTIC TORSION: TWIST AND STRESS

By Dianne P. O'Leary

CONSIDER A LONG ROD MADE OF METAL, PLASTIC, RUBBER, OR SOME OTHER HOMOGENEOUS MATERIAL. HOLD THE ROD AT THE ENDS AND TWIST ONE END CLOCKWISE AND THE other end counterclockwise. This *torsion* (twisting) causes stresses in the rod. If the force we apply is small enough, the rod behaves as an *elastic body*: when we release it, it will return to its original state. But if we apply a lot of twisting force, we will eventually change the rod's structure: some portion of it will behave *plastically* and will be permanently changed. If the whole rod behaves elastically, or if it all behaves plastically, then modeling is rather easy. More difficult cases occur when there is a mixture of elastic and plastic behavior. Here, we'll investigate the rod's behavior over a full range of torsion.

The Elastic Model

As usual, we start with simplifying assumptions to make the computation tractable. We assume that the torsional force is evenly distributed throughout the rod, and that the rod has uniform cross-sections. Under these circumstances, we can understand the system by modeling the stress in any single cross-section. We'll call the interior of the 2D cross section D and its boundary $\hat{D}$.

The standard model involves the *stress function* $u(x, y)$ on D, where the quantities $-\partial u(x, y)/\partial x$ and $\partial u(x, y)/\partial y$ are the stress components. If we set the net force to zero at each point in the cross-section, we obtain

$$\nabla^2 u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \equiv u_{xx} + u_{yy} = -2G\theta \text{ in } D$$

$$u = 0 \text{ on } \hat{D},$$

where G is the *shear modulus* of the material, and θ (radians) is the angle of twist per unit length. To guarantee existence of a smooth solution to our problem, we'll assume that the boundary $\hat{D}$ is smooth; in fact, in our experiments, $\hat{D}$ will be an ellipse.

We can derive an alternate equivalent formulation by minimizing an energy function

$$E(u) = \frac{1}{2}\int\int_D |\nabla u(x, y)|\, dxdy$$
$$-2G\theta \int\int_D u(x, y)dxdy.$$

The magnitude of the gradient

$$|\nabla u(x, y)| = \sqrt{(\partial u(x, y)/\partial x)^2 + (\partial u(x, y)/\partial y)^2}$$

is the *shear stress* at the point (x, y), an important physical quantity. At any point where the shear stress exceeds the yield stress σ_0, the material becomes plastic, and our standard model is no longer valid.

For simple geometries (such as a circle), we can solve this problem analytically. But, for the sake of generality and in preparation for the more difficult elastoplastic problem, we will consider numerical methods. Discretization by *finite dif-*

This problem focuses on the stress induced in a rod by twisting it. We'll investigate two situations: first, when the stress is small enough that the rod behaves elastically, and second, when we pass the elastic–plastic boundary.

 Copublished by the IEEE CS and the AIP 1521-9615/04/$20.00 © 2004 IEEE

ferences would be a possibility, but the geometry makes the flexibility of *finite elements* attractive. We can use a finite-element package to formulate the matrix $\boldsymbol{K}$ that approximates the operator $-\nabla^2 u$ on D, and also assemble the right-hand side $\mathbf{b}$ so that the solution to the linear system $\boldsymbol{K}\mathbf{u} = \mathbf{b}$ is the approximation to $u(x, y)$ at the nodes (x_i, y_i) of the finite-element mesh. Because the boundary $\hat{D}$ and the forcing function $-2G\theta$ are smooth, we expect *optimal order approximation* of the finite-element solution to the true solution as the mesh is refined: for piecewise linear elements on triangles, for example, this means that the error is $O(h^2)$, where h is a measure of the triangles' size.

In Problem 1, we see what this model predicts for the sheer stress on our rod.

> **Problem 1.** Suppose that the rod's cross-section D is the interior of a circle of radius one, and let $G = 5$ and $\theta = 1$. Use a finite-element package to approximate the stress function. Plot the approximate solution and describe what it says about the stress. Solve again using a finer mesh and estimate the error in your approximation $1/2\mathbf{u}^T\boldsymbol{K}\mathbf{u} - \mathbf{b}^T\mathbf{u}$ to $E(u)$.

Note that by symmetry, we could reduce our computational domain in Problem 1 to a quarter circle, setting the normal derivative of u along the two straight edges to zero.

The Elastoplastic Model

As the value of θ is increased, the maximum value of the shear stress $|\nabla u(x, y)|$ increases, eventually exceeding the rod's yield stress, at which point our model breaks down because the rod is no longer behaving elastically. We can extend our model to this case by adding constraints: we still minimize the energy function, but we don't allow stresses larger than the yield stress:

$$\min_u E(u)$$

$$|\nabla u(x, y)| \le \sigma_0, \quad (x, y) \in D$$
$$u = 0 \text{ on } \hat{D}.$$

The new constraints $|\nabla u(x, y)| \le \sigma_0$ are nonlinear, but we can reduce them to linear by a simple observation: if we start at the boundary and work our way in, we see that the constraint is equivalent to saying that $|u(x, y)|$ is bounded by σ_0 times the (shortest) distance from (x, y) to the boundary.

So the next (and most challenging) ingredient in solving our problem is an algorithm for determining these distances. In the next two problems, we develop and implement such an algorithm.

> **Problem 2.** Derive an algorithm for finding the distance $d(\mathbf{z})$ between a given point $\mathbf{z} = [z_1, z_2]^T$ and an ellipse. In other words, solve the problem
>
> $$\min_{x, y} (x - z_1)^2 + (y - z_2)^2$$
>
> subject to
>
> $$\left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2 = 1,$$
>
> for given parameters α and β. Note that the distance is the square root of the optimal value of the objective function $(x - z_1)^2 + (y - z_2)^2$. The problem can be solved using Lagrange multipliers, as a calculus student would. You need only consider points $\mathbf{z}$ on or inside the ellipse, but handle all the special cases: $\alpha = \beta$, $\mathbf{z}$ has a zero coordinate, and so forth.

In Problem 2, we see that a rather simple sounding mathematical problem becomes complicated when we handle the special cases properly. When we consider the fact that computers do their arithmetic inexactly, we see that an algorithm for computing distances to an ellipse must also account for difficulties encountered, for example, when a component of $\mathbf{z}$ is near zero; we face the difficulties of this algorithm in Problem 3.

> **Problem 3.** Program your distance algorithm, document it, and produce a convincing validation of the code by designing a suitable set of tests and discussing the results.

Now we have the elements in place to solve our elastoplastic torsion problem. We discretize $E(u)$ using finite elements, and we use our distance function to form the constraints, resulting in the problem

$$\min_{\mathbf{u}} \; 1/2\, \mathbf{u}^T \boldsymbol{K} \mathbf{u} - \mathbf{b}^T\mathbf{u}$$
$$-\sigma_0\mathbf{d} \le \mathbf{u} \le \sigma_0\mathbf{d},$$

Tools

This project comes from a paper coauthored by Wei H. Yang in 1978.[1] At that time, we worked very hard to develop memory- and time-efficient algorithms to solve the elastoplastic problem so that we wouldn't need a supercomputer. Now, sufficient computational resources are available in most laptops.

A.P.S. Selvadurai[2] gives an excellent derivation of the elastic model equation. He also discusses the model's history, noting that several incorrect models existed before Barre de Saint-Venant proposed a correct one.

The solution to Problem 1 requires access to a package to generate finite-element meshes and stiffness matrices. You can use a stand-alone package such as PLTMG (www.scicomp.ucsd.edu/~reb/) or Matlab's PDE Toolbox routines (`initmesh`, `refinemesh`, `assempde`, `pdeplot`). An introduction to finite-element formulations appears elsewhere.[3,4]

Problem 2 is deceptively simple, but translating the algorithm into reliable software in Problem 3 requires a great deal of attention to details. Use a reliable rootfinder such as `fzero` to solve the nonlinear equation.

For Problem 4, you will need a quadratic programming algorithm, such as Matlab's `quadprog` from the Optimization Toolbox. Quadratic programming is discussed in nonlinear programming textbooks.[5]

References

1. D.P. O'Leary and W.H. Yang, "Elasto-Plastic Torsion by Quadratic Programming," *Computer Methods in Applied Mechanics and Eng.*, vol. 16, 1978, pp. 361–368.
2. A.P.S. Selvadurai, *Partial Differential Equations in Mechanics 2*, Springer, 2000, Section 9.9.
3. K.-J. Bathe, *Finite Element Procedures*, Prentice Hall, 1996.
4. S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer, 1996.
5. M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, 1992.

where $d_i = d(x_i, y_i)$, and the ith component of $\mathbf{u}$ approximates the solution at (x_i, y_i). Because the matrix $\boldsymbol{K}$ is symmetric positive definite (due to the differential equation's elliptic nature), the solution to the problem exists and is unique. This problem is a *quadratic programming problem*, so algorithms for solving it include *active set strategies* and the newer *interior point algorithms*.

Problem 4. Solve the elastoplastic problem on a mesh that you estimate will give an error of less than 0.1 in the function $E(u)$. Use the parameters $G = 1$, $\sigma_0 = 1$, and $\beta = 1$. Let $\alpha\,\theta = 0, 0.25, 0.50, \ldots, 5$ and $\beta/\alpha = 1, 0.8, 0.65, 0.5, 0.2$. Plot a few representative solutions. On a separate graph, for each value of β/α, plot a curve $T/(\sigma_0\alpha^3)$ versus $G\alpha\theta/\sigma_0$, where T is the estimate of the torque, the integral of u over the domain D. (This will give you five curves.) On the same plot, separate the elastic solutions (those for which no variable is at its bound) from the elastoplastic ones. Estimate the errors in your plot's data points.

We solved this problem on a rod with a simple cross-section. Think about how you could extend our methods to more complicated shapes!

Acknowledgments

I am grateful to Simon P. Schurr for helpful comments on this problem.

Partial Solution to Last Issue's Homework Assignment

FITTING EXPONENTIALS: AN INTEREST IN RATES

By Dianne P. O'Leary

SUPPOSE WE HAVE TWO CHEMICAL REACTIONS OCCURRING SIMULTANEOUSLY. THEN THE AMOUNT Y OF A REACTANT CHANGES DUE TO BOTH PROCESSES AND BEHAVES AS A FUNCTION OF time t as

$$y(t) = x_1 e^{\alpha_1 t} + x_2 e^{\alpha_2 t},$$

where x_1, x_2, α_1, and α_2 are fixed parameters. Typically, we observe the function $y(t)$ for m fixed t values, perhaps $t = 0$, Δt, $2\Delta t$, …, t_{final}. The residual vector is defined to be

$$\mathbf{r} = \mathbf{y} - A\mathbf{x},$$

where $A_{ij} = e^{\alpha_j t_i}$, $j = 1, 2$, $i = 1, \ldots, m$ and $y_i = y(t_i)$.

Let the singular value decomposition (SVD) of A be $U\Sigma V^T$, where the $m \times m$ matrix U satisfies $UU^T = U^T U = I$ (the $m \times m$ identity matrix), the $n \times n$ matrix V satisfies $VV^T = V^T V = I$, and the $m \times n$ matrix Σ is zero except for entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ on its main diagonal.

Problem 1.

a. The columns of the matrix $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ form an orthonormal basis for n-dimensional space. Let's express the solution $\mathbf{x}_{\text{true}}$ as

$$\mathbf{x}_{\text{true}} = w_1\mathbf{v}_1 + \ldots w_n\mathbf{v}_n.$$

Determine a formula for w_i ($i = 1, \ldots, n$) in terms of U, $\mathbf{y}_{\text{true}}$, and the singular values of A.

b. Justify the reasoning behind these two statements:

$A(\mathbf{x} - \mathbf{x}_{\text{true}}) = \mathbf{y} - \mathbf{y}_{\text{true}} - \mathbf{r}$ means

$$\|\mathbf{x} - \mathbf{x}_{\text{true}}\| \leq \frac{1}{\sigma_n}(\|\mathbf{y} - \mathbf{y}_{\text{true}} - \mathbf{r}\|)$$

$\mathbf{y}_{\text{true}} = A\mathbf{x}_{\text{true}}$ means $\|\mathbf{y}_{\text{true}}\| = \|A\mathbf{x}_{\text{true}}\| \leq \|A\|\,\|\mathbf{x}_{\text{true}}\|$.

c. Use these two statements and the fact that $\|A\| = \sigma_1$ to derive an upper bound on $\|\mathbf{x} - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$ in terms of the condition number $\kappa(A) \equiv \sigma_1/\sigma_n$ and $\|\mathbf{y} - \mathbf{y}_{\text{true}} - \mathbf{r}\|/\|\mathbf{y}_{\text{true}}\|$.

Answer:

a. We can solve the linear least-squares problem by minimizing the norm of $U^T\mathbf{r} = U^T\mathbf{y} - U^T A\mathbf{x} = \beta - \Sigma V^T\mathbf{x}$, where

$$\beta_i = \mathbf{u}_i^T\mathbf{y}, \qquad i = 1\ldots, m,$$

and $\mathbf{u}_i$ is the ith column of U. If we change the coordinate system by letting $\mathbf{w} = V^T\mathbf{x}$, then our problem is to minimize

$$(\beta_1 - \sigma_1 w_1)^2 + \ldots (\beta_n - \sigma_n w_n)^2 + \beta_{n+1}^2 + \ldots \beta_m^2.$$

From this expression, it is easy to see that the minimum is achieved by choosing $w_i = \beta_i/\sigma_i$, $i = 1, \ldots, n$, and thus setting $\mathbf{x} = V\mathbf{w}$.

b. From our SVD, we know that if $\mathbf{g} = U^T(\mathbf{y} - \mathbf{y}_{\text{true}} - \mathbf{r})$, then

$$\mathbf{x} - \mathbf{x}_{true} = \sum_{i=1}^{n} \frac{g_i}{\sigma_i}\boldsymbol{v}_i,$$

and because the vectors $\boldsymbol{v}_i$ are orthonormal,

$$\|\mathbf{x} - \mathbf{x}_{true}\|^2 = \sum_{i=1}^{n}\left[\frac{g_i}{\sigma_i}\right]^2$$

$$\leq \frac{1}{\sigma_n^2}\sum_{i=1}^{n} g_i^2$$

$$= \frac{1}{\sigma_n^2}\|\mathbf{g}\|^2$$

$$= \frac{1}{\sigma_n^2}\|\mathbf{y} - \mathbf{y}_{true} - \mathbf{r}\|^2.$$

The second inequality follows from the property of matrix norms that states that $\|A\mathbf{x}\| \leq \|A\|\,\|\mathbf{x}\|$ for any compatible A and $\mathbf{x}$.

c. If we divide the expressions derived in part b, we get

$$\frac{\|\mathbf{x} - \mathbf{x}_{true}\|}{\|\mathbf{x}_{true}\|} \leq \frac{\sigma_1}{\sigma_n}\frac{\|\mathbf{y} - \mathbf{y}_{true} - \mathbf{r}\|}{\|\mathbf{y}_{true}\|}.$$

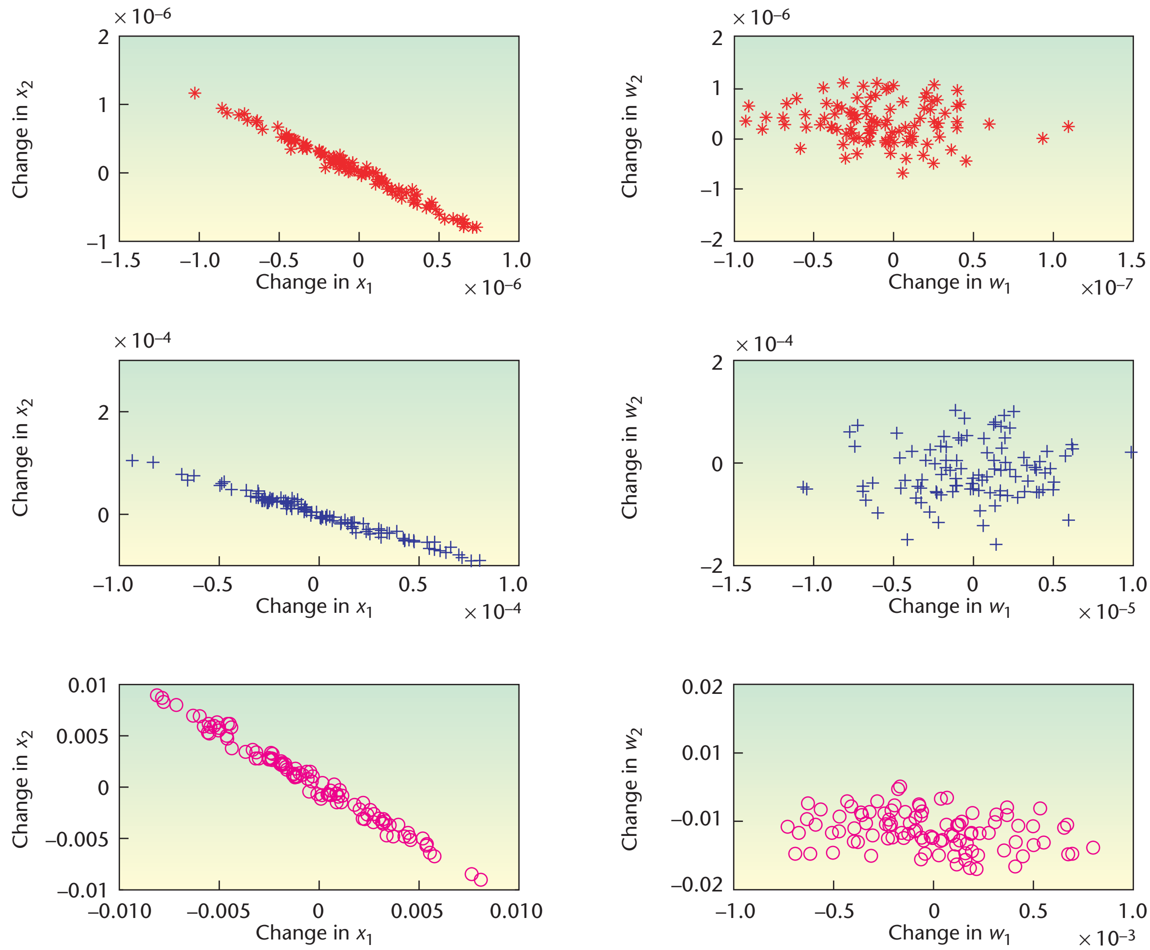

Figure 1. Solving problem 2. In this figure, $\alpha = [0.3, 0.4]$ and $\eta = 10^{-6}$ (top row), $\eta = 10^{-4}$ (middle row), and $\eta = 10^{-2}$ (bottom row). On the left, we plot the two components of $x - x_{true}$, and on the right $w - w_{true}$.

Note if the residual is zero (if the model fits the data exactly), then the relative change in **x** is bounded by the condition number of A times the relative change in **y**.

Problem 2. Generate 100 problems with data $x_{true} = [0.5, 0.5]^T$, $\alpha = [0.3, 0.4]$, and

$$\mathbf{y} = \mathbf{y}_{true} + \eta \mathbf{z},$$

where $\eta = 10^{-4}$, $\mathbf{y}_{true}$ contains the true observations $y(t)$, $t = 0, 0.01, \ldots, 6.00$, and the elements of the vector **z** are uniformly distributed on the interval [–1, 1]. In a figure, plot the computed solutions $\mathbf{x}^{(i)}$, $i = 1, \ldots, 100$ obtained via your SVD algorithm, assuming that α is known. In a second figure, plot the components $w^{(i)}$ of the solution in the coordinate system determined by V. Interpret these two plots using Problem 1's results. The points in the first figure are close to a straight line, but what determines the line's direction? What determines the shape and size of the second figure's point cluster? Verify your answers by repeating the experiment for $\alpha = [0.3, 0.31]$ and also try varying η to be $\eta = 10^{-2}$ and $\eta = 10^{-6}$.

Answer: Sample Matlab programs to solve this problem (and the others in this homework) are available at www.computer.org/cise/homework. The results are shown in Figures 1 and 2. Note that the shapes of the **w** clusters are rather circular; the sensitivity in the two components is approximately equal. This is not true of the **x** clusters; they are elongated in the direction corresponding to the eigenvector of the smallest singular value, because small changes in the data in this direction cause large changes in the solution. The length of the **x** cluster (and thus the solution's sensitivity) is greater in Figure 2 because the condition number is larger.

Problem 3. Suppose that the reaction results in

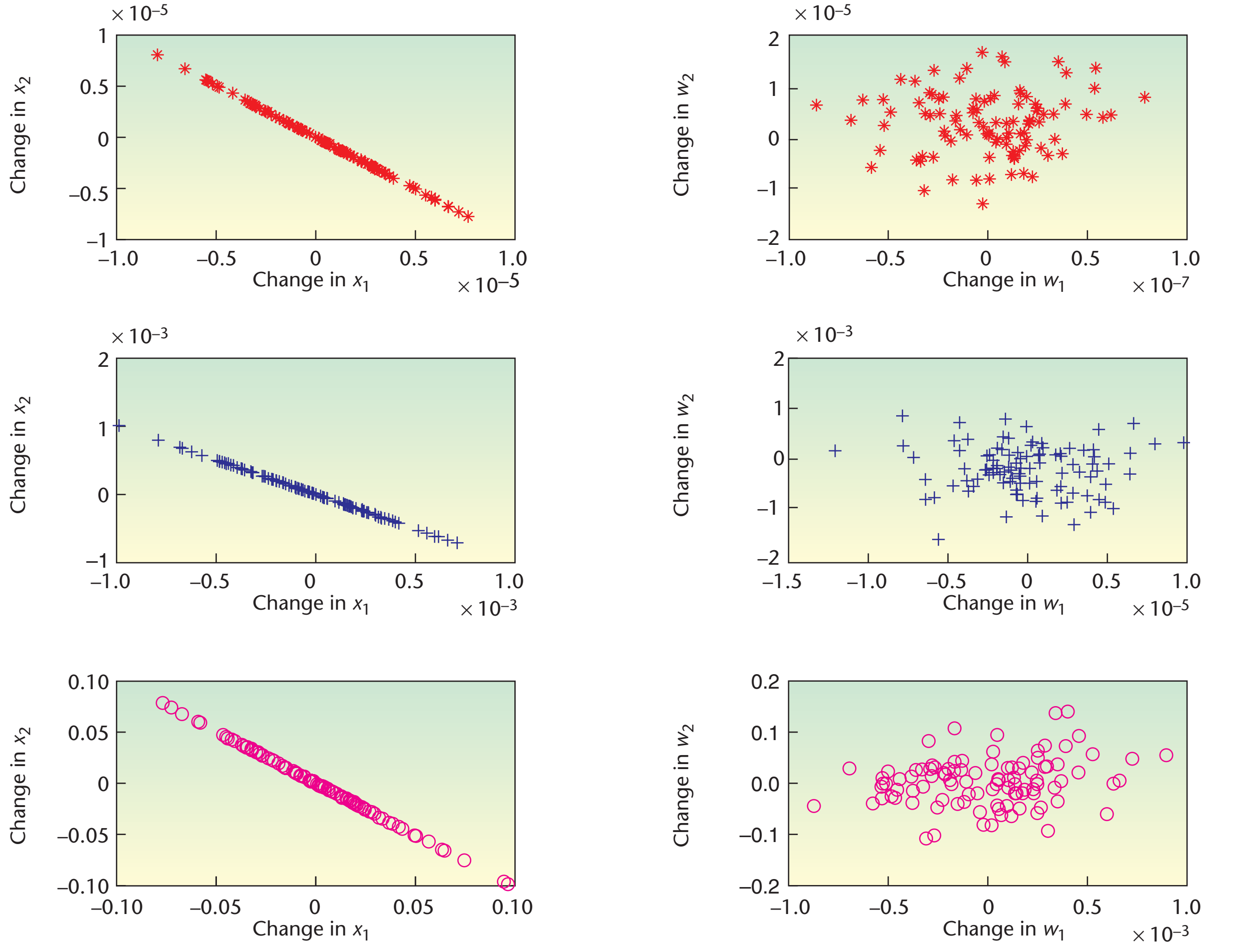

Figure 2. Solving problem 2. In this figure, $\alpha = [0.30, 0.31]$ and $\eta = 10^{-6}$ (top row), $\eta = 10^{-4}$ (middle row), and $\eta = 10^{-2}$ (bottom row). On the left, we plot the two components of $x - x_{true}$, and on the right $w - w_{true}$.

$$y(t) = 0.5e^{-0.3t} + 0.5e^{-0.7t}.$$

Next, suppose that we observe $y(t)$ for $t \in [0, t_{final}]$, with 100 equally spaced observations per second.

Compute the residual norm as a function of various α estimates, using the optimal values of x_1 and x_2 for each choice of α values. Make six contour plots of the log of the residual norm, letting the observation interval be $t_{final} = 1, 2, \ldots, 6$ seconds. Plot contours of –2, –6, and –10. How helpful is it to gather data for longer time intervals? How well determined are the α parameters?

Answer: Figure 3 shows the results. One thing to note is that the sensitivity is not caused by the conditioning of the *linear* parameters; as t_{final} is varied, the condition number $\kappa(A)$ varies from 62 to 146, which is quite small. But the plots dramatically illustrate the fact that a wide range of α values produce small residuals for this problem. This is an inherent limitation in the problem, and we cannot change it. It means, though, that we need to be very careful in computing and reporting results of exponential fitting.

One important requirement on the data is that there be a sufficiently large number of points in the range where each of the exponential terms is large.

Problem 4.

a. Use a nonlinear least-squares algorithm to determine the sum of two exponential functions that approximates the data set generated with $\alpha = [-0.3, -0.4]$, $\mathbf{x} = [0.5, 0.5]^T$, and normally distributed error with mean zero and standard deviation $\eta = 10^{-4}$. Provide 601 values of $(i, y(t))$ with $t = 0, 0.01, \ldots, 6.0$. Experiment with the initial guesses

$$\mathbf{x}^{(0)} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \ \alpha^{(0)} = [-1, -2]$$

and

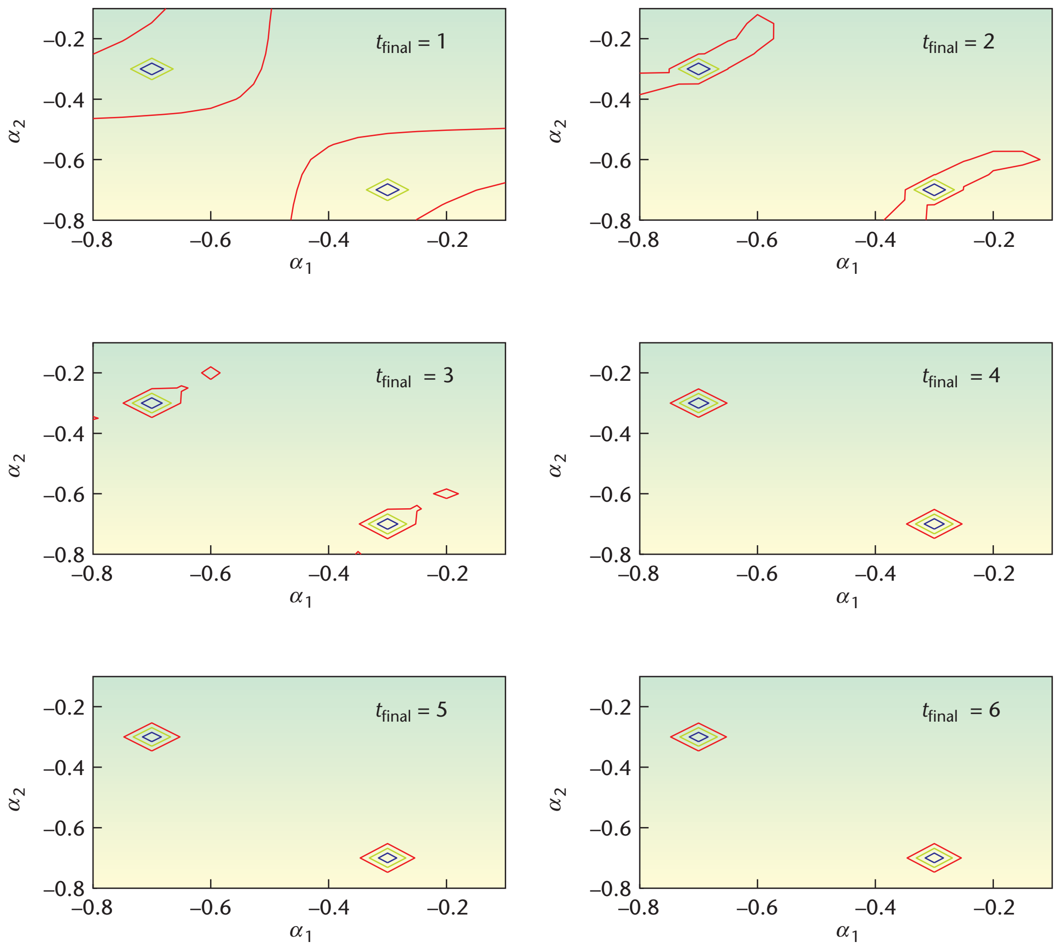

Figure 3. Solving problem 3. Contour plots of the residual norm as a function of the estimates of α for various values of t_{final}. The contours marked are 10^{-2}, 10^{-6}, and 10^{-10}.

$$\mathbf{x}^{(0)} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \alpha^{(0)} = [-5, -6] .$$

Next, plot the residuals obtained from each solution, and then repeat the experiment with $\alpha = [-0.30, -0.31]$. How sensitive is the solution to the starting guess?

b. Repeat the runs of part (a), but use variable projection to reduce to two parameters, the two components of α. Discuss the results.

Answer: When the true $\alpha = [-0.3, -0.4]$, the computations with four parameters produced unreliable results: [–0.343125 – 2.527345] for the first guess and [–0.335057 – 0.661983] for the second. The results for two parameters were somewhat better but still unreliable: [–0.328577 – 0.503422] for the first guess and [–0.327283 – 0.488988] for the second. Note that all the runs produced one significant figure for the larger of the rate constants but had more trouble with the smaller.

For the harder problem, when the true $\alpha = [-0.30, -0.31]$, the computations with four parameters produced [–0.304889 – 2.601087] for the first guess and [–0.304889 – 2.601087] for the second. The results for two parameters were again better but unreliable for the smaller rate constant: [–0.304889 – 0.866521] for the first guess and [–0.304889 – 0.866521] for the second.

The residuals for each of these fits are plotted in Figures 4 and 5. From the fact that none of the residuals from our computed solutions for the first problem resemble white noise, we can note that the solutions are not good approximations to the data. Troubles in the second problem are more difficult to diagnose, because the residual looks rather white. A single exponential function gives a good approximation to this data, and the second term has very little effect on the residual. This is true to a lesser extent for the first data set.

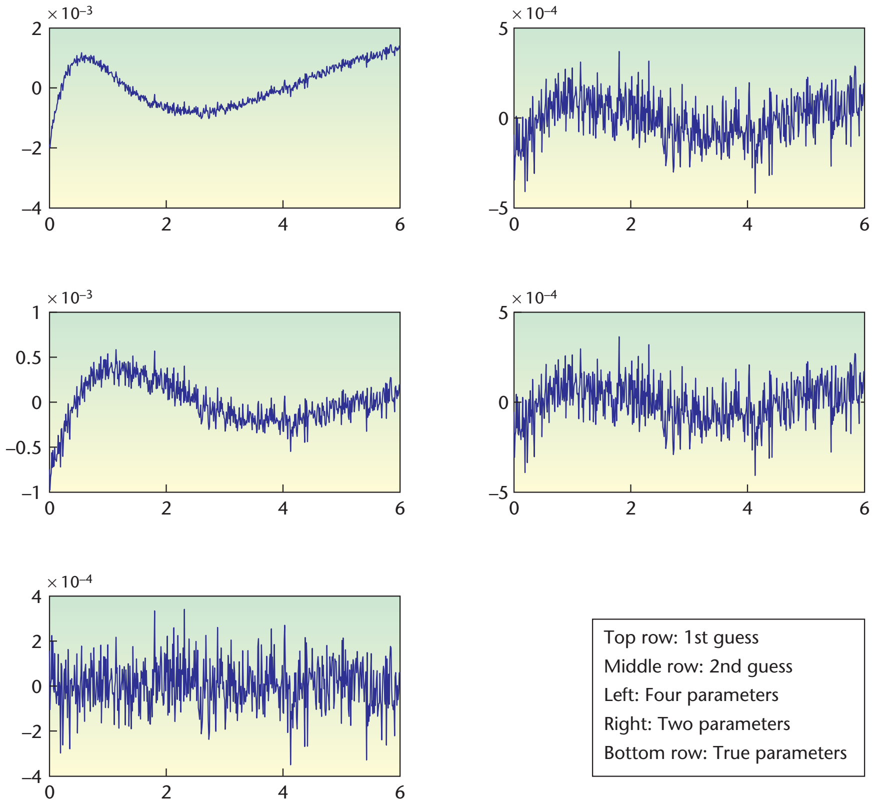

Figure 4. Solving problem 4. Residuals produced for the data with true α = [–0.3, –0.4] by minimizing with two or four parameters and two initial guesses, and the residual provided by the true parameter values.

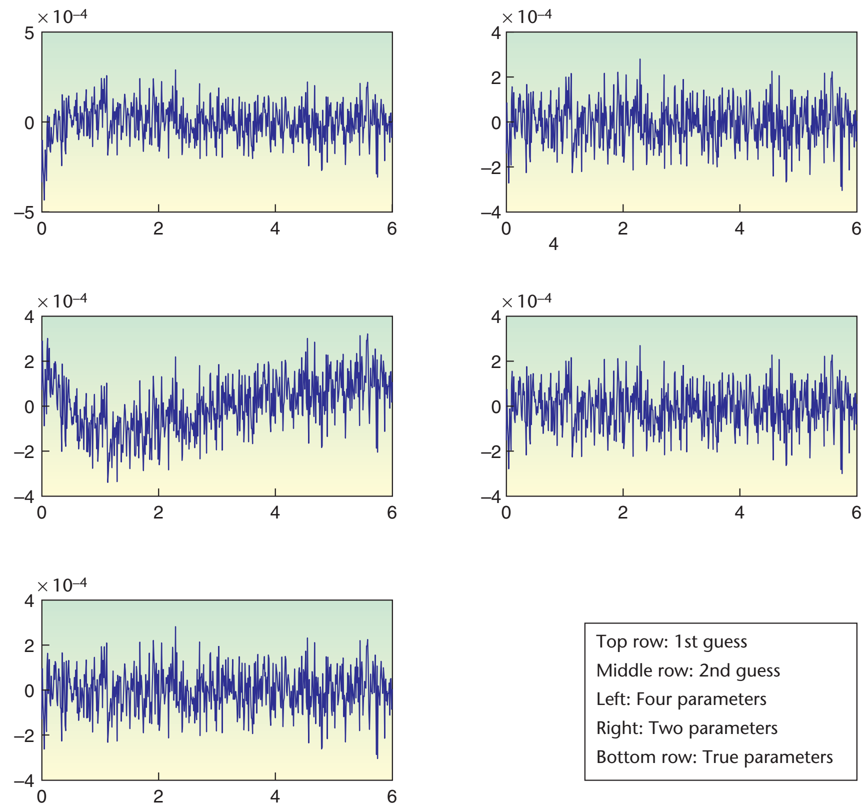

Figure 5. Solving problem 4. Residuals produced for the data with true α = [–0.30, –0.31] by minimizing with two or four parameters and two initial guesses, and the residual provided by the true parameter values.

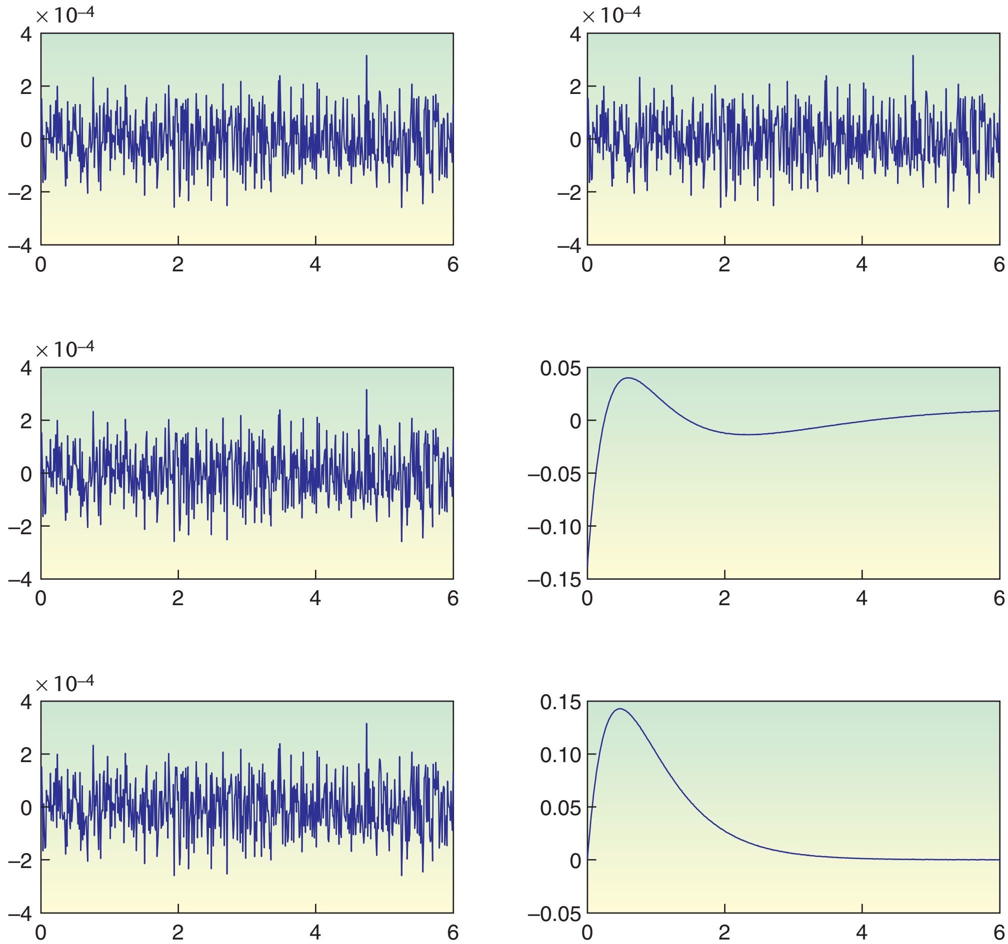

Figure 6. Solving problem 5. Residuals for the five computed solutions (residual component versus *t*), and, in the lower right, the data.

Results will vary with the particular sequence of random errors generated.

Problem 5. Suppose that we gather data from a chemical reaction involving two processes: one process produces a species and the other depletes it. We have measured the concentration of the species as a function of time. (If you prefer, consider the amount of a drug in a patient's bloodstream while the intestine is absorbing it and the kidneys are excreting it.) Figure 6 shows the data; it is also available at www.computer.org/cise/homework. Suppose your job (or even the patient's health) depends on determining the two rate constants and a measure of uncertainty in your estimates. Find the answer and document your computations and reasoning.

Answer: We solved this problem using Matlab's `lsqnonlin` and the two parameters α using several initial guesses: [–1, –2], [–5, –6], [–2, –6], [0, –6], and [–1, –3]. All runs except the fourth produced values α = [–1.6016, –2.6963] and a residual of 0.0024011. The fourth run produced a residual of 0.49631. Figure 6 shows the residuals for the five runs. The four "good" residuals look like white noise of size about 10^{-4}, giving some confidence in the fit.

We tested the sensitivity of the residual norm to changes in the parameters by creating a contour plot in the neighborhood of the optimal values computed earlier (see Figure 7). If the contours were square, then reporting the uncertainty in α as ± some value would be appropriate, but as we can see, this is far from the case. The log = –2.6 contour outlines a set of α values that changes the residual norm by less than 1 percent, the log = –2.5 contour denotes a change of less than 5 percent, and the log = –2.36 contour

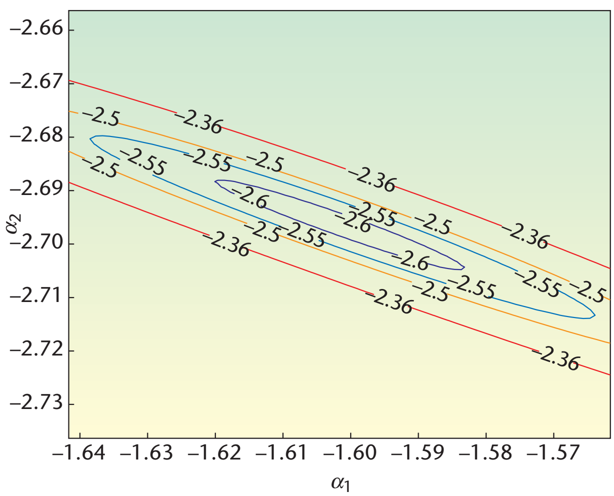

Figure 7. Solving problem 5. Contour plot of $\log_{10}$ of the norm of the residual for various values of the α parameters.

corresponds to a 10 percent change. The best value found was $\alpha = [-1.601660, -2.696310]$, with residual norm $0.002401137 = 10^{-2.6196}$. Our uncertainty in the rate constants is rather large.

The "true solution," the value used to generate the data, was $\alpha = [-1.6, -2.7]$ with $x_1 = -x_2 = 0.75$, and the standard deviation of the white noise was 10^{-4}.

Variants of Prony's method[1] provide alternate approaches to exponential fitting.

Exponential fitting is a very difficult problem, even when the number of terms n is known. It becomes even easier to get fooled when determining n is part of the problem!

Acknowledgments

I'm grateful to Bert Rust and Vadim Kavalerov for helpful comments on this problem.

Reference

1. M.R. Osborne and G.K. Smith, "A Modified Prony Algorithm for Exponential Function Fitting," *SIAM J. Scientific Computing*, vol. 16, no. 1, 1995, pp. 119–138.

Dianne P. O'Leary is a professor of computer science and a faculty member in the Institute for Advanced Computer Studies and the Applied Mathematics Program at the University of Maryland. Her interests include numerical linear algebra, optimization, and scientific computing. She received a BS in mathematics from Purdue University and a PhD in computer science from Stanford. She is a member of SIAM, ACM, and AWM. Contact her at oleary@cs.umd.edu; www.cs.umd.edu/users/oleary/.