Designing Optical Fibers: Fitting the Derivatives of a Nonlinear PDE-Eigenvalue Problem


Computer Science Department, William Paterson University, Passaic, USA
Email: lkaufmang@comcast.net

Received July 3, 2012; revised September 15, 2012; accepted October 5, 2012

ABSTRACT
When trying to fit data to functions of the eigensystem of a PDE-eigenvalue problem, such as Maxwell’s equation, numerical differentiation is ineffective and analytic gradients must be supplied. In our motivating example of trying to determine the chemical composition of the layers of specialty optical fibers, the function involved fitting the higher order derivatives with respect to frequency of the positive eigenvalues. The computation of the gradient was the most time consuming part of the minimization problem. It was realized that if one interchanged the order of differentiation, and differentiated first with respect to the design parameters, fewer derivatives of the eigenvectors would be required and one could take full advantage that each grid point was affected by only a few variables. As the model was expanded to cover a fiber wrapped around a spool, the bandwidth of the linearized symmetric eigenvalue problem increased. At the heart of each of the iterative methods used to find the few positive eigenvalues was a symmetric, banded, indefinite matrix. Here we present an algorithm for this problem which reduces a symmetric banded matrix to a block diagonal matrix of $1 \times 1$ and $2 \times 2$ blocks. Fillin outside the band because of pivoting for stability is prevented by a sequence of planar transformations. Computationally the algorithm is compared to the block unsymmetric banded solver and the block positive definite symmetric band solver in LAPACK.

Keywords: Symmetric; Indefinite; Eigenvalue; Gradient; Optical Fiber Design

1. Introduction
In early 2000 the Lucent Fiber Division in Norcross, Georgia, announced the specifications for a specialty fiber the design of which, they estimated would take a year. They had software in which one inserted the widths of the various layers of a fiber and the refractive index profile, indicating the chemical composition of the layer, and the program would produce such optical properties as the dispersion and micro bendloss.

However, sitting in the audience the day of the announcement was a member of the Bell Labs group from Murray Hill, NJ, which had created a fiber design tool that treated the design process as an inverse problem- the optical properties and manufacturing constraints were input and the refractive index profile was output. The next morning he handed the Georgia group three designs that met the specification, possibly removing months from the design process. The story underlines the utility of treating designing optical fibers as an inverse problem. It also indicates one of the main difficulties in the design process; the existence of multiple solutions. In [1] the physicist W. A. Reed characterized the search for a solution to wandering on a beach with footprints and holes.

The model used by the Murray Hill group, which included the first author, assumed a fiber that was perfectly straight, circular, and uniform along its length, so that Maxwell’s equations for guided waves of the fiber could be reduced to a family of problems of the form

$$x = \beta^2 x.$$ 

In (1) $\alpha$ is a function of $\omega$ and the index of refraction profile $n(r)$, which in some regions was piecewise linear or constant as in Figure 1 and one wishes to determine the widths, slopes and concentrations that define $n(r)$ because they dictate the chemical composition of the various layers of the fiber. We will denote these parameters as $v$. Among the various users of the tool there was little agreement on the shape of the burn region near the center of fiber or whether they wanted piecewise linear or piecewise constants in the other regions. The quantity $\omega$ in (1) was a specified frequency.
and \( m \) was a specified mode number. For a typical problem there might be about 20 values of \( \omega \) and \( m \) might take on integer values between 0 and 2.

According to the Modified Chemical Vapor Deposition (MCVD) technique, a hollow cylinder several feet long and several inches in radius is put on a lathe and gases such as silicon dioxide, gallium and fluorine are introduced into this tube, according to the specification of the refractive index profile in \( n(r) \), and these gases deposit themselves on the interior of the cylinder until it is solid. The cylinder or preform is then put in a draw tower, its end is heated, and the fiber is drawn to the specified width. The modeling process here assumes that the chemicals on a given cross section of the preform is solid. The cylinder or preform is then put in a draw tower, its end is heated, and the fiber is drawn to the specified width. The modeling process here assumes that the chemicals on a given cross section of the preform will be found in the same arrangement on the fiber.

The waveform in (1) is truncated at some radius beyond the core of the fiber and the boundary condition is expressed as the \( m^n \) order modified Bessel function of the second kind [2]. A finite element approach can be used to approximate (1) and converts the family of differential equations in (1) to a family of symmetric generalized tridiagonal eigenvalue problems

\[
(A + s(\mu) e_q e_q^T)x = \mu x. \tag{2}
\]

where \( s(\mu) \) involves Bessel functions, \( A \) is a \( q \times q \) symmetric tridiagonal matrix for a finite element discretization, \( e_q \) is the last column of the \( q \times q \) Identity matrix, and \( M \) is the diagonal positive definite mass matrix which represents the inner product of the basis functions used in the finite element method. Thus the boundary condition changes the problem from finding the positive eigenvalues and corresponding eigenvectors of about 60 linear eigenvalue problems to 60 nonlinear eigenvalue problems per function evaluation.

Although solving the nonlinear eigenvalue problem is essential to designing the fiber, it is not the complete story. Depending on the ultimate use of the fiber (local area network, underwater transmissions, or to splice into an existing network to restore a degraded signal), after the appropriate eigenvalues and eigenvectors are determined, as explained in [3], a user might wish to evaluate the dispersion given by

\[
\frac{\partial^2 \beta}{\partial \lambda \partial \omega}, \tag{3}
\]

the dispersion slope,

\[
\frac{\partial^3 \beta}{\partial \lambda^2 \partial \omega}, \tag{4}
\]

where \( \lambda \) is the wavelength and \( 2\pi c = \lambda \omega \) where the frequency \( \omega \) is measured in radians per second and \( c \) is the speed of light, the effective area given by

\[
2\pi \int \frac{x^2 r \, dr}{x^2 r \, dr}, \tag{5}
\]

the Laplace Spot Size or the square of the second Peterman Radius given by

\[
2 \int \left( \frac{dx}{dr} \right)^2 \, r \, dr,
\]

which depends on the Laplace Spot size and the RMS Spot size = \( \frac{\int x^2 r \, dr}{\int x^2 r \, dr} \), the macrobendloss, or the cutoff ratio, the largest value of \( \omega \), for which there are no positive eigenvalues.

The design problem can thus be posed as a constrained nonlinear least squares problem of trying to meet target values of say the dispersion (3) and/or the effective area (5) for values of \( \omega \) while keeping the width of regions nonnegative and satisfying manufacturing constraints. Because of the complexity of the eigenvalue computation and the further construction of the optical properties, optimizers tend to balk with numerical differentiation.

There are many tantalizing aspects of this problem including the determination of an appropriate grid which could change as the width parameters of the layers changed, the solution of the nonlinear eigenvalue problem, and the characterization of the nature of the multiple solutions of the inverse problem. In this paper we will concentrate on some of the ideas that were not incorporated in the original package but were developed when the first author moved to academia and was no longer in the mode of trying to meet the immediate demands of customers.

Often the solution of the eigenvalue problem required only about 20 to 30 percent of the computation time. Calculating the quantities in (3)-(5) and their derivatives with respect to the design parameters took up most of the remainder of the time. Initially one should always use symbolic and/or automatic differentiators like ADIFOR [4] to make sure that the derivative are correct, but
we found that using these techniques we were spending much time computing quantities that were zero. In Section 2 we consider ways to reduce the computation cost of the derivatives. These techniques were suitable for undergraduates like the last 5 authors and hopefully these techniques might have more general application in other inverse problems. Specifically, we were concerned about whether it was best to interchange the order of differentiation if part of the objective function was a derivative as in (3), whether it paid to use the fact that for certain parameters, like the heights of the layers, the matrix \( A \) was only nonzero in the portion of its rows corresponding to the grid points of the current layer, and thirdly, how could one take advantage of the fact one could write the original matrix \( A \) as a linear combination of vectors where the coefficients were dependent only on \( \omega \) and the vectors were dependent only on \( n(r) \)—i.e. the variables separated.

Solving the fiber optics problem was not a one way street of technology transfer and hopefully this paper will serve to highlight some of the linear algebraic advances that were stimulated by this application. Several nonlinear eigensolvers that were applied to the problem were compared in [5], which in turn partly motivated the research in [6-10]. Each of these solvers required the solution of many symmetric indefinite tridiagonal systems. Augmenting the simple model in (1) to handle a fiber wrapped around a spool, changed the tridiagonal eigenvalue problem to a 7-diagonal eigenvalue problem with the largest elements on the main diagonal and the outermost diagonal. In [11], algorithms for solving symmetric indefinite banded tridiagonal and 5 diagonal systems are given. If the number of diagonals exceeds 5, the bandwidth could increase especially if the largest elements appeared on the outermost diagonal. The application motivated the first author of this paper, who was also a coauthor of [11] to investigate further the larger bandwidth problem. In Section 3 we discuss an algorithm for solving the symmetric indefinite banded system with bandwidths larger than 5 that was initially given in [12] and indicate how one can form a block version of the algorithm.

2. Obtaining Analytic Derivatives

In this section we look at techniques that can be used to speed up the computation of the derivatives. Table 1 gives the percentage of the computation time used to solve the eigenvalue problem (1) and then to compute the various quantities like the dispersion slope (4) and the effective area and their derivatives with respect to the 12 design parameters. In general each run involved about solving about 1800 eigenvalue systems assuming 30 function and/or gradient evaluations per run; Each function evaluation required finding the positive eigenvalues and their corresponding eigenvectors of about 60 systems for about 20 values of \( \omega \) and 3 of \( m \). The point of the table is that an efficient eigensolver only tackles some of the problem.

To get derivatives one can augment the system of differential equations to obtain the derivatives with respect to the design parameters but this would increase the dimension of the eigenvalue problem. A better idea would be to move immediately into the discrete domain. According to [2], if one uses a finite element method with a spacing \( \delta \), then the eigenvalue \( \mu \) of (2) satisfies \( \beta^2 = \mu \delta^2 + k_j \) where \( \beta \) is defined in (1) and \( k_j \) is the wave number of the cladding. In general the wave number is given by \( \omega n | \epsilon \).

To simplify our notation let \( B = A + s(\mu)ee^T \).

and let \( p \) be \( \omega \) or one of the design parameters in \( v \) used in defining \( n \). Modifying the classical formulation for the classical formulation for the derivative of the eigenvalues found in [13], and differentiating \( Bx = \mu Mx \), we get

\[
\frac{\partial \mu}{\partial p} = x^T \left( \frac{\partial A}{\partial p} + \frac{\partial s(\mu)}{\partial p} e e^T - \mu \frac{\partial M}{\partial p} \right) x / \sigma,
\]

where \( \sigma = 1 - s_p \epsilon x^T \), if one stipulates that \( x^T Mx = 1 \).

The formula given in (8) is basic to approximating the function value and some of the gradient of the function. The major part of the computation of the dispersion and dispersion slope required the second and third derivatives of \( \mu \) with respect of \( \omega \). If we let \( \mu' \) denote the derivative of \( \mu \) with respect to \( \omega \), and \( A' \) the derivative of \( A \) with respect to \( \omega \), then if we differentiate (2) twice with respect to \( \mu \), multiply the result by \( x^T \) and use the fact that \( M' = 0 \) we get the following formula to be used in the calculation of the dispersion:

\[
\mu'' = \left( x^T A'' x + 2 x^T B' x + \alpha \right) / \sigma,
\]

where \( \alpha = s_p \epsilon x^T x \), which calls for the determination of \( x' \), which can be determined by differentiating (2) yielding

\[
(\mu M - B) x' = - (\mu' M - B') x.
\]

and using the fact that \( x'^T M x = 0 \).

The quantity \( \mu'' \) required by the dispersion slope \( \frac{\partial \beta}{\partial \lambda^2 \partial \omega} \) can be derived similarly and is given by
\[
\mu'' = \left( x^T A''_p x + 3 x^T B'_p x' + 3 x^T B' x'' - 3 \mu' x^T M x'' + \rho \right) / \sigma \tag{11}
\]

where the quantity \( \rho \) has all the terms in \( \left( x_q \right)^2 s''(\mu) \) except \( \mu'' s_p x_q^2 \).

### 2.1. Interchanging the Order of Differentiation

Now let us consider formulae for the derivatives of (9) and (11) with respect to a parameter \( p \).

If one uses the convention that \( A_p \) is the derivative of the A matrix with respect to the \( p^{th} \) parameter and \( x_p \) is the derivative of the eigenvector \( x \) with respect to the \( p^{th} \) design parameter, then it has been shown in [14] that if \( \theta \) contains all the known terms in \( s(\mu)_p \theta^2 \) then \( \mu''_p \) can be expressed several ways including

\[
\mu''_p = \left( x^T A''_p x \right.
\]

\[
+ 2 \left( x^T B'_p x' + x^T B' x'' + x^T B x''' + x^T B' x'' + x^T B' x'' + x^T B' x'' \right) + \theta / \sigma \right) \theta
\]

where \( \theta \) collects the terms other than \( \mu''_p \left( x_q \right)^2 s_p \) in \( \left( x_q \right)^2 s(\mu)_p \) and \( x_p' \) satisfies

\[
\begin{align*}
\left( \mu M - B \right) x_p' &= - \left( \mu M - B \right)' x \\
- \left( \mu M - B \right)' x_p - \left( \mu M - B \right)' x_p &= \left( \mu M - B \right)' x
\end{align*}
\]

and the condition \( x^T M x_p' = -x^T M x_p \). Table 1 uses the formula in (12) which requires \( x \), \( x' \), and for the \( p^{th} \) design parameter \( x_p \) and \( x_p' \). For 12 design parameters one must solve for 26 vectors.

Another formula for the gradient vector of the dispersion can be derived by interchanging the order of differentiation. More specifically, one begins with (7) for the \( p^{th} \) design parameter, multiplies through by \( x^T \) to remove the term with the derivative of the vector and then continues to differentiate with respect to \( \omega \) yielding the equation

\[
\mu''_p = \left( x^T A''_p x + 4 x^T B'_p x' + 2 x^T B_p x'' \right.
\]

\[
+ 2 x^T B'_p x' - \mu'' x^T M x + \theta / \sigma \right) \theta
\]

where

\[
\begin{align*}
\left( \mu M - B \right) x'' &= -2 \left( \mu M - B \right)' x' \\
- \left( \mu M - B \right)' x'' &= \left( \mu M - B \right)' x
\end{align*}
\]

and \( x^T M x'' = -x^T M x' \).

The second formula for \( \mu''_p \) in (13) requires \( x \), \( x' \), \( x'' \). Thus for 12 design parameters the formula in (13) requires 3 vectors. Theoretically, (12) requires at least twice as much work as (13).

The gradient of the dispersion slope (4) with respect to the \( p^{th} \) design parameter requires the derivative of \( \mu'' \) with respect to that parameter. We have looked at several formulae for \( \mu''_p \) including

\[
\mu''_p = \left( x^T A''_p x - \mu'' x^T M x + 2 x^T B_p x'' + \psi \right.
\]

\[
+ 3 \left( x^T B'_p x'' + x^T B'_p x' + x^T B' x'' + x^T B'_p x'' + x^T B' x'' \right) + \theta / \sigma \right) \theta
\]

where \( \psi \) has all terms in \( x_q^2 s(\mu)_p \) except \( \mu''_p \left( x_q \right)^2 s_p \) and

\[
\mu''_p = \left( x^T A''_p x - \mu'' x^T M x + 2 x^T B_p x'' + \psi \right.
\]

\[
+ 3 \left( x^T B'_p x'' + x^T B'_p x' + x^T B' x'' + x^T B'_p x'' + x^T B' x'' \right) + \theta / \sigma \right) \theta
\]

where \( \psi \) has all terms in \( x_q^2 s(\mu)_p \) except \( \mu''_p \left( x_q \right)^2 s_p \) and

\[
\mu''_p = \left( x^T A''_p x - \mu'' x^T M x + 2 x^T B_p x'' + \psi \right.
\]

\[
+ 3 \left( x^T B'_p x'' + x^T B'_p x' + x^T B' x'' + x^T B'_p x'' + x^T B' x'' \right) + \theta / \sigma \right) \theta
\]

(14)

The main advantage of (16) is that one does not have to compute \( x_p \), \( x_p' \), and \( x_p'' \), which are the most time consuming portions of (15). It was derived by first differentiating with respect to the design parameter and then with respect to \( \omega \) three times. From Table 2 we see the advantage of using (16) and (13) over (15) and (12) just by changing the formulae. Using both sets of formulae the numerical values for the gradient of the dispersion and its slope are the same as long as the eigenvalues are accurate to full precision. Although \( x_p \) is not needed in (16), it is needed to compute the gradient of the effective area if that is part of the specifications of the fiber.

The theory and methodology given in (15) for (1) can be easily extended to spooled models or full vector models and other approximations of the differential operator such as those used in [15,16], or the plane-wave
Table 2. Time (sec) for one value of ω, one mode order, n = 736, 12 variables.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigen calculation (2 eigenvalues)</td>
<td>0.048</td>
</tr>
<tr>
<td>Determining x_p for all the</td>
<td>0.017</td>
</tr>
<tr>
<td>design variables</td>
<td></td>
</tr>
<tr>
<td>Dispersion and slope based on</td>
<td>0.063</td>
</tr>
<tr>
<td>(12) and (15)</td>
<td></td>
</tr>
<tr>
<td>Dispersion and slope based on</td>
<td>0.033</td>
</tr>
<tr>
<td>(13) and (16)</td>
<td></td>
</tr>
<tr>
<td>Dispersion and slope based on</td>
<td>0.015</td>
</tr>
<tr>
<td>(13) and (16)</td>
<td></td>
</tr>
<tr>
<td>taking advantage of the zero</td>
<td></td>
</tr>
<tr>
<td>structure of matrices in formulae</td>
<td></td>
</tr>
</tbody>
</table>

Another advantage of (16) is that all the matrices in the formulae involve derivatives with respect to the design variables \( v \) and all these parameters had a limited range of support. If the \( p^{th} \) design parameter denoted the height of a profile in a layer then \( M_p = 0 \) and \( A_p, A'_p, \) and \( A''_p \) were 0 for those grid points outside the layer. If the \( p^{th} \) design parameter denoted the width of a layer, the grid points corresponding to the layers closer to the center of the fiber were zero, but because the preform had a finite width, those grid points within the layer and further from the center contributed nonzero value to the matrices. Using the separable form of (17) greatly simplifies the construction of the matrices in (12). Table 3 gives the proportion of time spent in various sections in the final code. The big hit that was initially seen in Table 1 for the calculation of analytic derivatives has disappeared.

3. Fiber around a Spool-Solving Banded Symmetric Indefinite Linear Systems

For the model in (1) at each iteration one was concerned with getting the few positive eigenvalues and their eigenvectors. Usually a generalized Rayleigh quotient iteration that minimized \( \|A + s(\mu)\eta e_i^T\| x - \mu M x \| \) was effective.

**Nonlinear Rayleigh Quotient**

1) Determine a lower bound \( \mu_l \) and an upper bound \( \mu_u \) for the desired eigenvalue \( \mu \).

Set \( \mu_0 \) to \( \mu_l \).

2) Set \( x = e \), the vector of all 1’s if a good approximation to the eigenvector is not available from a previous iteration.

Set \( j \) to 0.

3) Until convergence.

a) Set \( B = A - s(\mu_j) e e^T \).

b) Solve \( (B - \mu_j I) y = x \) and determine the inertia of \( (B - \mu_j I) \).

c) If the inertia claims that \( \mu_j \) is less than \( \mu_l \), reset \( \mu_1 \) to \( \mu_l \).

d) If the inertia says that \( \mu_j \) is greater than \( \mu_l \), reset \( \mu_u \) to \( \mu_j \).

e) Set \( z = B y \).

f) Set \( \alpha = z^T y - s'(\mu) y^T y \).

g) If \( \alpha < \mu_u \), set \( \mu_{j+1} = 0.9 \mu_l + 0.1 \mu_u \).

If \( \alpha > \mu_l \), set \( \mu_{j+1} = 0.9 \mu_u + 0.1 \mu_l \).
Table 3. Speedup in derivative calculations using ideas in Section 2 for one mode order, one $\omega$, 736 gridpoints, 12 variables.

<table>
<thead>
<tr>
<th>Speedup</th>
<th>Percentage of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building matrices</td>
<td>2.76</td>
</tr>
<tr>
<td>Eigenvalue problem</td>
<td>1.0</td>
</tr>
<tr>
<td>Dispersion and slope</td>
<td>4.3</td>
</tr>
<tr>
<td>Gradient of the eigensystem</td>
<td>1.75</td>
</tr>
<tr>
<td>Effective area</td>
<td>2.26</td>
</tr>
<tr>
<td>Total</td>
<td>1.96</td>
</tr>
</tbody>
</table>

If $\mu_l \leq \alpha \leq \mu_u$, set $\mu_{i+1} = \alpha$.

h) set $x = y / (y^T y)$.

increment $j$.

The inertia is a triple of the number of positive, negative, and zero eigenvalues. To get the bounds in step 1, to solve for $y$ in step 3b) and to calculate the inertia, the factorization algorithm for tridiagonal matrices in [11] was used. This algorithm reduced a matrix to a block diagonal form of $1 \times 1$ and $2 \times 2$ block. Since each $2 \times 2$ block corresponded to a positive negative pair, by adding the number of $2 \times 2$ blocks to the number of positive $1 \times 1$ blocks, one could determine the inertia of the system.

One of the concerns of fiber designers is how wrapping the fiber around a spool would affect the optical properties of a fiber. Decades ago Marcuse [20] suggested a model that considered the deformation of the modes (the eigenvectors) of a spooled fiber. For a given frequency one would add a coupling term that was based on the ratio $\varepsilon$ of the radius of the fiber to the radius of the spool. With $\varepsilon = 0$, one would get the solutions for various values of $m$ that were obtained in (2) and one wished to track these modes as $\varepsilon$ increased. If at some $\varepsilon_2$ no mode is found whose innerproduct with a specific node at say $\varepsilon_1$ is within 0.9 of that at $\varepsilon_1$, then a new value of $\varepsilon$, the geometric mean of $\varepsilon_1$ and $\varepsilon_2$ was used.

In (2) the matrix $A$ had a tridiagonal structure, but for the spooled problem, if say 3 modes were found for a given $\omega$, the matrix $A$ would have the form

$$
\begin{pmatrix}
T_1 & C & H \\
C & T_2 & G \\
H & G & T_3
\end{pmatrix},
$$

where the $C$, $H$, and $G$ matrices were diagonal and the $T$ matrices were tridiagonal. If one denoted the $i$th diagonal of $C$, $H$, $G$, as $c_i$, $h_i$, $g_i$, respectively then $0(c_i) = \varepsilon$, $0(g_i) = \varepsilon$ and $0(h_i) = \varepsilon^2$, i.e. small. Let $d_{i,j}$ denote the $i$th diagonal of $T_j$ and $e_{i,j}$ the $i$th diagonal of $T_j$. These elements were usually of $O(1)$. To reduce its bandwidth the $A$ matrix for the spooled matrix could be permuted into a 7 diagonal matrix given in (18) with the largest off diagonal elements in magnitude appearing on the outermost band.

Bunch and Kaufman [11] gave algorithms for factoring dense matrices, tridiagonal and 5 diagonal matrices. They did not give an algorithm for a 7 diagonal matrix or indeed for a general banded matrix. To bound the element growth of the final block diagonal matrix, Bunch and Kaufman insisted that at each stage whenever 2 rows and columns were to be eliminated leading to a $2 \times 2$ diagonal block, the rows and columns of the matrix had to be permuted so that the (2,1) element be the largest off diagonal element in magnitude in its column. The permutation and the elimination of the 2 rows and columns could upset the band structure. For a matrix like that in (18) where the largest would have been in the fourth row, the permuted matrix would have a zero structure like

$$
\begin{pmatrix}
d_{1,1} & c_1 & h_1 & e_{2,1} \\
c_1 & d_{1,2} & g_1 & e_{2,2} \\
h_1 & g_1 & d_{1,3} & e_{2,3} \\
e_{2,3} & d_{2,1} & c_2 & h_2 & e_{3,1} \\
e_{2,2} & c_2 & d_{2,2} & g_2 & e_{3,2} \\
e_{2,3} & h_2 & g_2 & d_{2,3} & e_{3,3} \\
e_{3,3} & d_{3,1} & c_3 & h_3 \\
e_{3,2} & c_3 & d_{3,2} & g_3 \\
e_{3,3} & h_3 & g_3 & d_{3,3}
\end{pmatrix}.
$$
The retraction algorithm proceeds with rows

\[
\begin{pmatrix}
d & e & y & y \\
e & d & y & y & y & y & y \\
y & y & x & x & x & x & x \\
y & y & x & x & x & x & x \\
y & y & x & x & x & x & x \\
\end{pmatrix}
\]

(19)

The reason why one of the \( y \) elements is denoted by \( \hat{y} \) will become clear later.

Eliminating the \( y \) elements of (19), would leave a matrix whose zero structure would look like

\[
\begin{pmatrix}
d & e \\
e & d \\
x & x & x & x & f \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
x & x & x & x & x \\
\end{pmatrix}
\]

(20)

where \( f \) lay outside the original band structure so that one had a 9 diagonal matrix rather than a 7 diagonal matrix.

If again a \( 2 \times 2 \) pivot was chosen and \( f \) was the largest offdiagonal element, the second and fifth rows and columns of the reduced matrix would be permuted and after elimination of the unwanted off diagonal elements, the matrix would be 13 diagonal. The process of increasing the band width could continue. In fact that was just what observed when the algorithm was applied to the spooled problem. The factorization given in [11] could not be used as originally given.

If one partitions (19) into

\[
A = \begin{pmatrix} D & Y^T \\ Y & B \end{pmatrix}
\]

(21)

where \( D \) is the top \( 2 \times 2 \) submatrix, \( Y \) has 2 columns that contain the \( y \) elements below \( D \) in (19) and \( B \) is the portion with the \( x \)'s. Let \( F \) be that portion of (20) having the \( x \)'s and \( f \). In block form

\[
F = B - YZ
\]

where \( Z = D^{-1}Y^T \). The matrix \( Z \) has 2 rows and if one writes it as

\[
Z = \begin{pmatrix} z & v & w & w & w \\ s & t & u & u & u \end{pmatrix}
\]

(22)

then the unwanted element \( f \) in (20) is \( f = \hat{y}s \) where \( \hat{y} \) is given in (19) and \( s \) is given in (22). If \( s \) in (22) had been 0, there would have been no fill and one could have continued on as a 7-diagonal matrix and not a 9-diagonal matrix.

If one did a planar transformation on the first 2 columns of \( Z \) to annihilate \( s \) in (22) the fillin would be prevented. Applying this planar transformation to the first two rows and columns of \( B \) in (21) lengthens the short second rows and columns of \( B \) by one element but no nonzero element is introduced outside the band. Thus making \( s \) zero by applying a transformation to \( Y, Z \) and \( B \) means that \( f \) in (20) will be zero and we have an algorithm that can be used to factor a 7-diagonal matrix. Using this symmetric banded algorithm rather than a specifically tuned unsymmetric band solver reduced the computation time for a linear Rayleigh quotient iteration for the spooled problem by about 37 percent. In a problem with 12 variables, producing the dispersion at 5 specific values of \( \epsilon \) required eigen solutions at 25 values of \( \epsilon \) in order to follow the mode. Thus decreasing the eigendecomposition time was significant.

The algorithm can be generalized to larger bandwidths. Assume we are working with a matrix \( A \) with bandwidth \( 2g + 1 \) and that the Bunch-Kaufman algorithm version C has dictated using a \( 2 \times 2 \) pivot requiring the permutation of rows and columns 2 and \( r \) so that the second row and column are long and the \( r \)th row and column are short so that \( Y \) in (21) can be written as

\[
Y = \begin{pmatrix} y_1 & y_2 \\ 0 & \hat{y} \\ 0 & 0 \end{pmatrix}
\]

(23)

where the length of \( \hat{y} \) is \( r - 3 \) and \( Z \) in (21) as

\[
Z = \begin{pmatrix} z^T & v & w^T \\ s^T & t & u^T \end{pmatrix}
\]

(24)

The length of \( s^T \) is \( r - 3 \) so \( t \) would be aligned with the short column of the permuted \( A \).

When the new matrix \( F = B - YZ \) is formed, elements will appear outside the original bandwidth by the multiplication of \( \hat{y}s^T \). If however, \( s^T = 0 \) in (24), no elements would appear outside the band in \( F \). Annihilating \( s^T \) using a single Householder transformation introduces elements outside the band structure in \( F \). But a sequence of \( r - 3 \) planar transformations in planes \( (1, r - 2), (2, r - 2), \ldots, (r - 3, r - 2) \) always involve the short column of \( B \) and there is no fillin.

For a symmetric banded matrix \( A^{(k)} \) with \( k \) rows and bandwidth \( 2g + 1 \) the retraction algorithm proceeds.
as follows

1) Let $x^{(k)} = |a_{k,1}^{(k)}| = \text{maximum element in absolute value in the first column.}$

2) If $|a_{k,1}^{(k)}| \geq \gamma x^{(k)}$, use a $1 \times 1$ pivot to obtain $A^{(k-1)}$, decrease $k$ by 1) and return to 1). Here $\gamma$ is a scalar of about 1/3 to balance element growth.

else

3) Determine $\sigma^{(k)}$, largest element in absolute value of the $r$th column.

4) If $\gamma x^{(k)} \leq |a_{r,1}^{(k)}| \sigma^{(k)}$, use a $1 \times 1$ pivot to obtain $A^{(k-1)}$, decrease $k$ by 1) and return to 1).

else

5) Interchange the $r$th and second rows and columns of $A^{(k)}$ and partition it like (21).

6) Form the $Z$ matrix $Z = D^{-1}Y^T$.

7) For $i = 1, \ldots, \min(r-3, k-r)$.

Do a sequence of planar rotations in planes $(i, r-2)$ to eliminate elements in the second row of $Z$ and apply these transformations to the corresponding rows of $Y$, and $A^{(k)}$.

8) Perform a 2) by 2) pivot to obtain $A^{(k-2)}$, decrease $k$ by 2) and return to 1). 

For a matrix $A$ of $n$ rows, the retraction algorithm outlined above using stabilized elementary transformations for the planar rotations requires between $1/2 \cdot gn^2$ and $5/4 \cdot gn^2$ multiplications compared to between $gn^2$ and $2gn^2$ multiplications required by unsymmetric Gaussian elimination. It requires $(2g+1)n$ elements to hold the $Z$ matrices compared to $(3g+1)n$ operations. As reported in [12] where the planar transformations were introduced not as a preventative measure but as a way of cleaning up elements that appeared outside the original band structure, an implementation of the algorithm was faster than LAPACK’s DGBTF2 [21] for nonsymmetric banded matrices. However, when $g$ was sufficiently large so that it was advantageous to use a block structured algorithm, the retraction algorithm could not compete with LAPACK’s DGBTRF.

Since the publication of [12], it was realized that it was possible to use the extra space required for the $Z$ matrices to generate a block approach for those portions of the algorithm whenever the planar transformations were not required. By embedding the matrix in the larger $(2g+1)\times n$ array one can avoid some of the difficulties in the block symmetric codes in LAPACK which store and access only the lower triangular portion of the matrix. Similarly one can avoid a few of the problems associated with the banded codes of continually copying and retrieving a block or a triangle to a scratch space. The sketch in Figure 2 indicates how we partitioned the matrix with the dashed lines indicating the structure of

the mathematical object and the sequences of rectangles the computational object. When applying transformations matrix-matrix operations could be used exclusively.

In Figure 3 the times for the symmetric indefinite block version of the retraction algorithm is compared to DPBTRF, Lapack’s positive definite banded routine block version, and DGBTRF for various bandwidths. For the retraction algorithm the block size is 16 while for that of the other algorithms, it is set at 32. ATLAS BLAS were used in FORTRAN.

In Figure 4 the matrices all have the same bandwidth, but the outer diagonal and the main diagonal are varied to produce matrices that require various number of planar transformations. In this version, a block concept of the retraction algorithm is pursued until one hits a 2 by 2 pivot that requires planar transformations. At the crossover point about 60 percent of the rows are involved in $2 \times 2$ blocks. It is obvious that the more transformations, the greater the time.

4. Conclusions

Although this research was directed at the fiber optic industry, we hope its impact will have a significant wider breath to cover other PDE-eigenvalue problems with design parameters. Specifically, this paper should be applicable to inverse problems where finding the gradient is absolutely necessary but seems to involve a tremendous amount of computation. We have shown that it pays to take advantage of local support of variables, to write the function in a separable form if possible and if the function is itself a derivative, to ask whether it pays to change the order of differentiation when producing the gradient.

Trying to solve the spooled fiber optics problem suggested to the first author that there are linear systems that are symmetric indefinite and banded. Her first attempts seemed to suggest that only for examples where the inertia is definitely required or the bandwidth is not large, that the retraction algorithm is useful. However, recently she has realized that the extra space required by the algorithm to store information for solving linear systems could be used to produce a viable block version.
5. Acknowledgements

All coauthors were supported in part by a National Science Grant for research at undergraduate institutions.

REFERENCES


326-328. doi:10.1049/el:19780222
