Reconstruction of tridiagonal matrices from spectral data

Ricardo S. Leite, Nicolau C. Saldanha and Carlos Tomei *

August 3, 2005

Abstract

Jacobi matrices are parametrized by their eigenvalues and norming constants (first coordinates of normalized eigenvectors): this coordinate system breaks down at reducible tridiagonal matrices. The set of real symmetric tridiagonal matrices with prescribed simple spectrum is a compact manifold, admitting an open covering by open dense sets $\mathcal{U}_\Lambda^\pi$ centered at diagonal matrices Λ^π, where π spans the permutations. *Bidiagonal coordinates* are a variant of norming constants which parametrize each open set $\mathcal{U}_\Lambda^\pi$ by the Euclidean space.

The reconstruction of a Jacobi matrix from inverse data is usually performed by an algorithm introduced by de Boor and Golub. In this paper we present a reconstruction procedure from bidiagonal coordinates and show how to employ it as an alternative to the de Boor-Golub algorithm. The inverse bidiagonal algorithm rates well in terms of speed and accuracy.

Keywords: Jacobi matrix, inverse eigenvalue problem, bidiagonal coordinates.
MSC-class: 65F18; 15A29.

1 Introduction

Recall that a real tridiagonal symmetric matrix

$$T = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & b_{n-1} & a_n \end{pmatrix}$$

is a *Jacobi matrix* if $b_i > 0$ for all i. Jacobi matrices have simple spectrum and their eigenvectors have nonzero first and last coordinates. Thus, a Jacobi matrix T diagonalizes uniquely as $T = Q^* \Lambda Q$, $\Lambda = \mathrm{diag}(\lambda_1 < \cdots < \lambda_n)$, provided we demand that the *norming constants* $w_i = Q_{i1}$ be positive for all i. Let $\mathcal{J}_\Lambda^0$ be the set of Jacobi matrices with given simple spectrum Λ and $\mathbb{S}_+^{n-1} = \{(w_1, \ldots, w_n) \mid w_i > 0, w_1^2 + \cdots + w_n^2 = 1\}$, the open positive octant of the unit sphere $\mathbb{S}^{n-1} \subset \mathbb{R}^n$. Define the *map of norming constants* $\omega_\Lambda : \mathcal{J}_\Lambda^0 \to \mathbb{S}_+^{n-1}$ by $\omega_\Lambda(T) = w = (w_1, \ldots, w_n)$. Moser ([9]) proved that ω_Λ is a diffeomorphism.

On another route, numerical analysts considered the problem of reconstructing a Jacobi matrix T from its eigenvalues λ_i, $i = 1 \ldots n$, and the eigenvalues μ_i, $i = 1 \ldots n-1$, of its bottom principal $(n-1) \times (n-1)$ minor. The interlacing

*The authors acknowledge support from CNPq, IM-AGIMB and FAPERJ.

theorem requires that $\lambda_1 < \mu_1 < \lambda_2 < \mu_2 < \cdots < \mu_{n-1} < \lambda_n$. Existence and uniqueness of T, continuous dependence on λ's and μ's and an iterative algorithm to obtain T were obtained by Hochstadt, Gray, Wilson and Hald ([6], [7], [4], [5]); A direct, stable algorithm was obtained by de Boor and Golub ([1]). The first step of their algorithm is the computation of the norming constants w in terms of λ's and μ's. The problem then boils down to computing the inverse map $\omega_\Lambda^{-1}(w)$, as will be described in section 2. For a survey of the Jacobi reconstruction problem, see [3].

Let $\mathcal{J}_\Lambda$ be the closure of $\mathcal{J}_\Lambda^0$, the set of tridiagonal symmetric matrices T with spectrum Λ and $b_i \geq 0$. The map of norming constants ω_Λ extends continuously to $\mathcal{J}_\Lambda$ but this extension is no longer injective. Indeed, for $n = 3$ and $\Lambda = \mathrm{diag}(1, 2, 4)$ we have

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3-\cos 2t & \sin 2t \\ 0 & \sin 2t & 3+\cos 2t \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos t & \sin t \\ 0 & -\sin t & \cos t \end{pmatrix} \Lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos t & -\sin t \\ 0 & \sin t & \cos t \end{pmatrix}$$

and therefore $\omega_\Lambda(T) = (1, 0, 0)$, $\mu_1 = 4$ and $\mu_2 = 6$ for all such T. Thus, in some sense, any reconstruction algorithm either from λ's and μ's or from λ's and w's must degenerate at some points of the boundary of $\mathcal{J}_\Lambda$.

In [8], the authors introduced *bidiagonal coordinates*, a variant of norming constants which behaves well at the boundary. In this paper we provide a direct reconstruction algorithm from bidiagonal coordinates with good behavior at boundary points, where norming constants break down. The conversion from norming constants to bidiagonal coordinates is simple, and the resulting algorithm is comparable in time and space with that of de Boor and Golub, being more accurate in many cases.

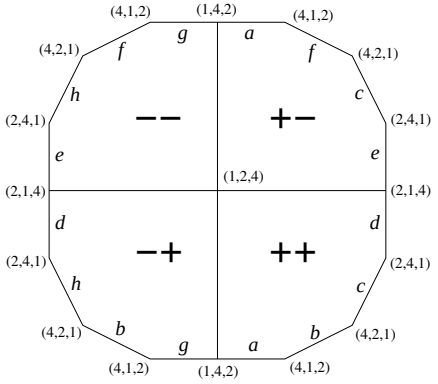

Figure 1: The manifold $\mathcal{I}_\Lambda$ for $\Lambda = \mathrm{diag}(1, 2, 4)$

Matters become clearer with some geometric vocabulary. Let $\mathcal{I}_\Lambda \supset \mathcal{J}_\Lambda^0$ be the set of tridiagonal symmetric matrices with spectrum Λ. As proved in [10], $\mathcal{I}_\Lambda$ is a compact oriented manifold of dimension $n-1$. The closure $\mathcal{J}_\Lambda \subset \mathcal{I}_\Lambda$ of $\mathcal{J}_\Lambda^0$ is homeomorphic to the convex polytope $\mathcal{P}_\Lambda$ with vertices $\Lambda^\pi = \mathrm{diag}(\lambda_{\pi(1)}, \ldots, \lambda_{\pi(n)})$ where π spans the set of permutations of $\{1, 2, \ldots, n\}$ ([10], [2]). Each of the 2^{n-1} possible choices of signs for the entries b_i define a closed subset of $\mathcal{I}_\Lambda$ which is isomorphic to $\mathcal{J}_\Lambda$: as is well known, dropping the signs of the off-diagonal entries of a tridiagonal symmetric matrix does not change its spectrum. Thus, $\mathcal{I}_\Lambda$ can be constructed by glueing 2^{n-1} copies of $\mathcal{P}_\Lambda$ along faces consisting of reducible tridiagonal matrices (i.e., matrices for which some b_i is zero). In figure 1 we show what happens for $n = 3$. The polytope $\mathcal{P}_\Lambda$ is a hexagon and in each of its copies

we indicate the signs of b_1 and b_2. Vertices are diagonal matrices and edges consist of reducible matrices; edges with the same label are glued. It follows that $\mathcal{I}_\Lambda$ is a bitorus for $n = 3$.

Removal of the outer boundary edges in the picture yields an open dense subset of $\mathcal{I}_\Lambda$ centered at Λ. As we shall see, bidiagonal coordinates can be smoothly defined on this set, yielding an explicit diffeomorphism with $\mathbb{R}^2$. More generally, for each Λ and each permutation π, we define an open dense subset $\mathcal{U}_\Lambda^\pi$ of $\mathcal{I}_\Lambda$. The complement of $\mathcal{U}_\Lambda^\pi$ consists of matrices T for which there exist $i \le k < n$ with $b_k = 0$ and $\lambda_{\pi(i)}$ belongs to the spectrum of the bottom principal $(n-k) \times (n-k)$ minor. As in the example above, $\mathcal{U}_\Lambda^\pi$ is centered at Λ^π and bidiagonal coordinates provide a diffeomorphism between $\mathcal{U}_\Lambda^\pi$ and $\mathbb{R}^{n-1}$. Also, the sets $\mathcal{U}_\Lambda^\pi$ form an open cover of $\mathcal{I}_\Lambda$: this is the crucial property for the local study of iterations preserving tridiagonality and spectrum as performed in [8].

2 The de Boor-Golub algorithm

In this section we present part of the contents of [1] phrased in such a way as to emphasize the differences and similarities between this more well known algorithm and the inverse bidiagonal algorithm, to be presented in section 4. We assume that the off-diagonal entries b_i of T are positive so that T is a Jacobi matrix. Write $T = Q^*\Lambda Q$ where $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ (in this section, the simple eigenvalues λ_i are taken in an arbitrary order), and Q is orthogonal with positive first column. For $D_b = \mathrm{diag}(1, b_1, b_1b_2, \dots, b_1b_2\cdots b_{n-1})$ and $D_w = \mathrm{diag}(Q_{11}, Q_{21}, \dots, Q_{n1}) = \mathrm{diag}(w_1, w_2, \dots, w_n)$, write $\tilde{P} = D_b Q^* D_w^{-1}$ so that

$$\tilde{T} = D_b T D_b^{-1} = \begin{pmatrix} a_1 & 1 & & & \\ b_1^2 & a_2 & 1 & & \\ & b_2^2 & a_3 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & b_{n-1}^2 & a_n \end{pmatrix} = \tilde{P}\Lambda\tilde{P}^{-1}.$$

Let $\tilde{p}_{k-1}^* = e_k^*\tilde{P}$ be the k-th row of $\tilde{P}$; in particular, $\tilde{p}_0^* = (1, 1, \dots, 1)$. We have $\tilde{T}\tilde{P} = \tilde{P}\Lambda$ and the rows of $\tilde{P}$ satisfy the recursion

$$\begin{aligned} \tilde{p}_1^* &= \tilde{p}_0^*\Lambda - a_1\tilde{p}_0^*, \\ \tilde{p}_{k+1}^* &= \tilde{p}_k^*\Lambda - a_{k+1}\tilde{p}_k^* - b_k^2\tilde{p}_{k-1}^*, \quad 0 < k < n-1, \\ 0 &= \tilde{p}_{n-1}^*\Lambda - a_n\tilde{p}_{n-1}^* - b_{n-1}^2\tilde{p}_{n-2}^*. \end{aligned}$$

Furthermore, the vectors $\tilde{p}_k$ form an orthogonal basis under the inner product $\langle\langle u, v\rangle\rangle = \langle u, D_w^2 v\rangle$. For $0 \le k < n$, let $\check{p}_k$ be the unique polynomial of degree less than n satisfying $\check{p}_k(\lambda_j) = (\tilde{p}_k)_j$: we have $\check{p}_0 = 1$,

$$\check{p}_1 = t\check{p}_0 - a_1\check{p}_0, \quad \check{p}_{k+1} = t\check{p}_k - a_{k+1}\check{p}_k - b_k^2\check{p}_{k-1}, \quad 0 < k < n,$$

so that $\check{p}_k$ is a monic polynomial of degree k.

In the notation of [1], let $p_k(t) = \det(tI - T_k)$ where T_k is the principal minor of T consisting of the first k rows and columns; set also $p_0 = 1$. The expansion of the determinant along the last row of each minor yields

$$p_1 = tp_0 - a_1p_0, \quad p_{k+1} = tp_k - a_{k+1}p_k - b_k^2p_{k-1}, \quad 0 < k < n \tag{1}$$

and therefore $p_k = \check{p}_k$ since $p_0 = \check{p}_0$ and both sequences satisfy the same recurrence. Equivalently, the j-th coordinate of the vector $\tilde{p}_k^*$ is $p_k(\lambda_j)$.

Summing up, assume Λ and $D_w = \mathrm{diag}(w_1, \ldots, w_n)$ given. The linear bijection between the space of real polynomials of degree less than n and $\mathbb{R}^n$ given by evaluation on the λ_j's allows us to pull back the inner product $\langle\langle \cdot, \cdot \rangle\rangle$ giving rise to an inner product on polynomials:

$$\langle\langle q_1, q_2 \rangle\rangle = \sum_{j=1}^{n} w_j^2 q_1(\lambda_j) q_2(\lambda_j).$$

The de Boor-Golub algorithm now constructs the monic polynomials p_k using the orthogonality condition $\langle\langle p_k, p_{k'} \rangle\rangle = 0$ for $k \neq k'$. Recursion 1 for the polynomials p_k obtains the entries of T.

3 Bidiagonal coordinates

We quote some of the results and notations of [8] to be used in the inverse bidiagonal algorithm. Diagonalize $T \in \mathcal{I}_\Lambda$ as $T = Q^* \Lambda Q$, $\Lambda = \mathrm{diag}(\lambda_1 < \lambda_2 < \cdots < \lambda_n)$, and factor $Q = PLU$ where P is a permutation matrix, L is lower unipotent and U is upper triangular. For a permutation π, let P_π be the permutation matrix with (i, j) entry equal 1 iff $i = \pi(j)$ (thus $P_{\pi_1 \pi_2} = P_{\pi_1} P_{\pi_2}$ and $P_\pi e_i = e_{\pi(i)}$). The PLU factorization can usually be done for several choices of the permutation matrix: it turns out that we can take $P = P_\pi$ if and only if $T \in \mathcal{U}_\Lambda^\pi$, the open dense subset of $\mathcal{I}_\Lambda$ presented in the introduction. Write $Q_\pi = E P_\pi^{-1} Q = L_\pi U_\pi$ where E is a diagonal matrix with diagonal entries equal to 1 and -1, L_π is lower unipotent and U_π is upper triangular with positive diagonal. The rows of Q_π are eigenvectors of T but their first coordinates are not necessarily nonnegative: instead, signs are determined from the fact that the determinants of leading principal minors of Q_π are positive.

Let $B_\pi = L_\pi^{-1} \Lambda L_\pi = R_\pi^{-1} T R_\pi$ where $R_\pi = U_\pi^{-1}$ so that $L_\pi = Q_\pi R_\pi$. From the first formula, B_π is lower triangular; from the second, it is upper Hessenberg; thus, B_π is lower bidiagonal:

$$B_\pi = \begin{pmatrix} \lambda_1^\pi & & & & \\ \beta_1^\pi & \lambda_2^\pi & & & \\ & \beta_2^\pi & \lambda_3^\pi & & \\ & & \ddots & \ddots & \\ & & & \beta_{n-1}^\pi & \lambda_n^\pi \end{pmatrix}.$$

The map $\psi_\pi : \mathcal{U}_\Lambda^\pi \to \mathbb{R}^{n-1}$ taking T to the *π-bidiagonal coordinates* $(\beta_1^\pi, \ldots, \beta_{n-1}^\pi)$ is a diffeomorphism. Indeed, start from an explicit formula for the matrix L_π in terms of bidiagonal coordinates:

$$L_\pi = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{\beta_1^\pi}{\lambda_2^\pi - \lambda_1^\pi} & 1 & 0 & \cdots & 0 \\ \frac{\beta_1^\pi \beta_2^\pi}{(\lambda_3^\pi - \lambda_1^\pi)(\lambda_3^\pi - \lambda_2^\pi)} & \frac{\beta_2^\pi}{\lambda_3^\pi - \lambda_2^\pi} & 1 & & 0 \\ \vdots & \vdots & & \ddots & \\ \frac{\beta_1^\pi \beta_2^\pi \cdots \beta_{n-1}^\pi}{(\lambda_n^\pi - \lambda_1^\pi)(\lambda_n^\pi - \lambda_2^\pi) \cdots (\lambda_n^\pi - \lambda_{n-1}^\pi)} & \frac{\beta_2^\pi \cdots \beta_{n-1}^\pi}{(\lambda_n^\pi - \lambda_2^\pi) \cdots (\lambda_n^\pi - \lambda_{n-1}^\pi)} & & & 1 \end{pmatrix}.$$

Given L_π, its QR factorization yields Q_π and R_π, from which one obtains $T = R_\pi B_\pi R_\pi^{-1}$. A straightforward computation shows that b_i and β_i^π have the same

sign and that near a diagonal matrix, β_i^π equals b_i to first order; more, for the inverse map $\phi_\pi = (\psi_\pi)^{-1} : \mathbb{R}^{n-1} \to \mathcal{U}_\Lambda^\pi \subset \mathcal{I}_\Lambda$,

$$\phi_\pi(0,\dots,0) + D\phi_\pi(0,\dots,0)(u_1,\dots,u_{n-1}) = \begin{pmatrix} \lambda_1^\pi & u_1 & & \\ u_1 & \lambda_2^\pi & u_2 & \\ & u_2 & \lambda_3^\pi & \\ & & & \ddots \end{pmatrix}.$$

For any permutation π and any $T \in \mathcal{U}_\Lambda^\pi$, the norming constants $w_i^\pi = w_{\pi(i)}$ and the π-bidiagonal coordinates β_i^π are related by

$$w_i^\pi = w_1^\pi \frac{\beta_1^\pi \cdots \beta_{i-1}^\pi}{(\lambda_i^\pi - \lambda_1^\pi)\cdots(\lambda_i^\pi - \lambda_{i-1}^\pi)}, \quad 2 \le i \le n,$$

$$\beta_i^\pi = \frac{(\lambda_{i+1}^\pi - \lambda_1^\pi)\cdots(\lambda_{i+1}^\pi - \lambda_i^\pi) w_{i+1}^\pi}{(\lambda_i^\pi - \lambda_1^\pi)\cdots(\lambda_i^\pi - \lambda_{i-1}^\pi) w_i^\pi}, \quad 1 \le i \le n-1.$$

The inverse bidiagonal algorithm, presented in the next section, obtains the matrix R_π in another way, closer in spirit to the recursions in section 2. The basic version of this algorithm receives as input a permutation π, eigenvalues λ_i^π and bidiagonal coordinates β_i^π and returns the corresponding tridiagonal matrix $T \in \mathcal{U}_\Lambda^\pi$. The de Boor-Golub algorithm, instead, receives as input the eigenvalues λ_i and the norming constants w_i: in this case a simultaneous permutation π is innocuous, at least with exact arithmetic.

4 The inverse bidiagonal algorithm

We first describe a preliminary version of the algorithm, which only works in the irreducible case, where all β_k^π (or, equivalently, all b_k) are nonzero. Write $\tilde{T} = \hat{R} B_\pi \hat{R}^{-1}$ where $\hat{R} = D_b R_\pi$ is an upper triangular matrix with rows $\hat{r}_k^*$. Clearly, $\hat{r}_1^* = e_1^* \hat{R} = e_1^* D_b R_\pi = e_1^* R_\pi = e_1^* Q_\pi^* L_\pi = (Q_\pi e_1)^* L_\pi = (L_\pi U_\pi e_1)^* L_\pi = u_{11}(L_\pi e_1)^* L_\pi = u_{11} e_1^* L_\pi^* L_\pi$ and therefore $\hat{r}_1 = c L_\pi^* L_\pi e_1$, the value of $c = u_{11} > 0$ being irrelevant throughout the algorithm.

Equate rows in $\tilde{T}\hat{R} = \hat{R} B_\pi$,

$$\begin{pmatrix} a_1 & 1 & & & \\ b_1^2 & a_2 & 1 & & \\ & b_2^2 & a_3 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & b_{n-1}^2 & a_n \end{pmatrix} \begin{pmatrix} \hat{r}_1^* \\ \hat{r}_2^* \\ \hat{r}_3^* \\ \vdots \\ \hat{r}_n^* \end{pmatrix} = \begin{pmatrix} \hat{r}_1^* \\ \hat{r}_2^* \\ \hat{r}_3^* \\ \vdots \\ \hat{r}_n^* \end{pmatrix} \begin{pmatrix} \lambda_1^\pi & & & & \\ \beta_1^\pi & \lambda_2^\pi & & & \\ & \beta_2^\pi & \lambda_3^\pi & & \\ & & \ddots & \ddots & \\ & & & \beta_{n-1}^\pi & \lambda_n^\pi \end{pmatrix},$$

to obtain $\hat{r}_{k+1}^* = \hat{r}_k^* B_\pi - a_k \hat{r}_k^* - b_{k-1}^2 \hat{r}_{k-1}^*$. Since $\hat{R}$ is known to be upper triangular, this recursion, together with the initial term $\hat{r}_1^*$, allows us to compute the coefficients a_i, $i = 1,\dots,n$ and b_i^2, $i = 1,\dots,n-1$. The numbers b_i and β_i^π have the same sign in $\mathcal{U}_\Lambda^\pi$: this completes the preliminary version of the reconstruction algorithm for irreducible matrices.

We need to modify the algorithm in order to extend it to the general case. For

an integer $k \geq 0$, set

$$L_{\pi,k} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{(\beta_1^\pi)^k}{\lambda_2^\pi - \lambda_1^\pi} & 1 & 0 & \cdots & 0 \\ \frac{(\beta_1^\pi \beta_2^\pi)^k}{(\lambda_3^\pi - \lambda_1^\pi)(\lambda_3^\pi - \lambda_2^\pi)} & \frac{(\beta_2^\pi)^k}{\lambda_3^\pi - \lambda_2^\pi} & 1 & & 0 \\ \vdots & \vdots & & \ddots & \\ \frac{(\beta_1^\pi \beta_2^\pi \cdots \beta_{n-1}^\pi)^k}{(\lambda_n^\pi - \lambda_1^\pi)(\lambda_n^\pi - \lambda_2^\pi)\cdots(\lambda_n^\pi - \lambda_{n-1}^\pi)} & \frac{(\beta_2^\pi \cdots \beta_{n-1}^\pi)^k}{(\lambda_n^\pi - \lambda_2^\pi)\cdots(\lambda_n^\pi - \lambda_{n-1}^\pi)} & & & 1 \end{pmatrix} \quad (2)$$

and $B_{\pi,k} = L_{\pi,k}^{-1} \Lambda^\pi L_{\pi,k}$, or, more explicitly,

$$B_{\pi,k} = \begin{pmatrix} \lambda_1^\pi & & & & \\ (\beta_1^\pi)^k & \lambda_2^\pi & & & \\ & (\beta_2^\pi)^k & \lambda_3^\pi & & \\ & & \ddots & \ddots & \\ & & & (\beta_{n-1}^\pi)^k & \lambda_n^\pi \end{pmatrix}.$$

Still in the irreducible case, define $D_\beta = \mathrm{diag}(1, \beta_1^\pi, \beta_1^\pi \beta_2^\pi, \ldots, \beta_1^\pi \beta_2^\pi \cdots \beta_{n-1}^\pi)$ and $\tilde{R} = c^{-1} \hat{R} D_\beta^{-1}$ with rows $\tilde{r}_k^*$ so that $B_{\pi,2} = D_\beta B_\pi D_\beta^{-1}$ and $\tilde{T}\tilde{R} = \tilde{R} B_{\pi,2}$. Straightforward computations verify that $\tilde{r}_1 = L_{\pi,2}^* L_{\pi,0} e_1$. Expanding the matrix products as above we obtain the recursion $\tilde{r}_{k+1}^* = \tilde{r}_k^* B_{\pi,2} - a_k \tilde{r}_k^* - b_{k-1}^2 \tilde{r}_{k-1}^*$. Thus, from $\tilde{r}_{k-1}^*$ and $\tilde{r}_k^*$ we compute $\tilde{r}_k^* B_{\pi,2}$, then b_{k-1} and a_k and finally $\tilde{r}_{k+1}^*$. This completes the description of the *inverse bidiagonal algorithm* for irreducible matrices; we now prove that this procedure works for any $\beta^\pi \in \mathbb{R}^{n-1}$, obtaining all matrices $T \in \mathcal{U}_\Lambda^\pi$. Let $\mathrm{Upper}^+(\mathbb{R}, n)$ be the group of upper triangular matrices with positive diagonal.

Proposition 4.1 *There is a smooth function $\rho : \mathcal{U}_\Lambda^\pi \to \mathrm{Upper}^+(\mathbb{R}, n)$ satisfying $\rho(T) = (R_\pi)_{11} D_b R_\pi D_\beta^{-1}$ for all irreducible matrices $T \in \mathcal{U}_\Lambda^\pi$.*

Here, as in section 3, $T = Q_\pi^* \Lambda^\pi Q_\pi$, $L_\pi = Q_\pi R_\pi$, Q_π orthogonal, L_π lower unipotent and $R_\pi \in \mathrm{Upper}^+(\mathbb{R}, n)$. The purpose of this proposition is to make sense of $\tilde{R}$ for reducible matrices T (or, equivalently, for β^π with some zero coordinate). The formula in the statement defines $\rho(T)$ as $\tilde{R}$ for irreducible T but otherwise involves divisions by zero.

Proof: Define $\tilde{\rho} : \mathcal{U}_\Lambda^\pi \to \mathbb{R}^{n \times n}$ row by row: let $\tilde{\rho}_k^*$ denote the k-th row of $\tilde{\rho}(T)$ and set $\tilde{\rho}_1 = L_{\pi,2}^* L_{\pi,0} e_1$, $\tilde{\rho}_{k+1}^* = \tilde{\rho}_k^* B_{\pi,2} - a_k \tilde{\rho}_k^* - b_{k-1}^2 \tilde{\rho}_{k-1}^*$. The function $\tilde{\rho}$ is clearly smooth in $\mathcal{U}_\Lambda^\pi$. Also, we proved above that $\tilde{\rho}(T) = \rho(T) = \tilde{R}$ for irreducible T. Thus, by continuity, $\tilde{\rho}(T)$ is always upper triangular with nonnegative diagonal entries $\tilde{\rho}_{k,k}$. In the irreducible case,

$$\tilde{\rho}_{k,k} = \frac{b_1 b_2 \cdots b_{k-1}}{c \beta_1^\pi \beta_2^\pi \cdots \beta_{k-1}^\pi} (R_\pi)_{k,k}.$$

It remains to prove that $\tilde{\rho}_{k,k} \neq 0$ for reducible T so that we can then set $\rho = \tilde{\rho}$.

One way of completing the proof is recalling from [8] that the quotients b_j / β_j^π are smooth positive functions on $\mathcal{U}_\Lambda^\pi$. Alternatively, from $B_{\pi,2} = L_{\pi,2}^{-1} \Lambda L_{\pi,2}$, we can write $(B_{\pi,2}^*)^{k-1} \tilde{\rho}_1 = L_{\pi,2}^* \Lambda^{k-1} L_{\pi,0} e_1$. The coordinates of $L_{\pi,0} e_1$ are all nonzero by equation 2 and, from the standard Vandermonde argument, the vectors $\Lambda^{k-1} L_{\pi,0} e_1$, $k = 1, \ldots, n$, form a basis; since $L_{\pi,2}^*$ is invertible, so do the vectors $(B_{\pi,2}^*)^{k-1} \tilde{\rho}_1$. From the recursion formula, so do the vectors $\tilde{\rho}_k$ and we are done. ■

In general, we start from $\tilde{r}_1^*$ and use the recursive formula $\tilde{r}_{k+1}^* = \tilde{r}_k^* B_{\pi,2} - a_k \tilde{r}_k^* - b_{k-1}^2 \tilde{r}_{k-1}^*$. More precisely, assume by induction that $\tilde{r}_{k-1}^*$ and $\tilde{r}_k^*$ are known. From the proposition, $\tilde{r}_{k,k}$ and $\tilde{r}_{k-1,k-1}$ are positive. The first nonzero coordinate of $\tilde{r}_k^* B_{\pi,2}$ occupies position $k-1$ and equals $(\beta_{k-1}^\pi)^2 \tilde{r}_{k,k}$. The algorithm then calculates $b_{k-1} = \beta_{k-1}^\pi \sqrt{\tilde{r}_{k,k}/\tilde{r}_{k-1,k-1}}$, so that the square root is evaluated at a strictly positive number. Notice that the algorithm treats uniformly all $\beta^\pi \in \mathbb{R}^{n-1}$, i.e., there is no checking of signs or division into cases. The values of a_k and of $\tilde{r}_{k+1}^*$, the $(k+1)$-th row of the upper triangular matrix $\tilde{R}$, are now easily obtained, concluding the computation of $\phi_\pi(\beta_1^\pi, \ldots, \beta_{n-1}^\pi)$ and the description of the inverse bidiagonal algorithm.

5 Accuracy and tight permutations

Empirical evidence indicates that a good choice of the permutation π is extremely important for the accuracy of the inverse bidiagonal algorithm. One is reminded of Gaussian elimination, where pivoting strategies have a similar effect. There is a crucial difference, however. In Gaussian elimination, the permutation is chosen along the process; the inverse bidiagonal algorithm admits no easy way to accomodate a change of permutation in mid-flight. As to estimating accuracy with respect to the choice of permutation, our theoretical understanding is limited and we provide instead a simple numerical experiment [1]. We start with random inverse data for an 8×8 matrix and perform the inverse bidiagonal algorithm for each of the 8! permutations with 8 digits of precision. Results are then compared with the "correct" answer, computed with an exaggerated number of digits. Different permutations yield very different errors: the smallest error is $2.0 \cdot 10^{-7}$, there are 19 other permutations with error smaller than $3 \cdot 10^{-7}$ and there are 8 permutations with error greater then $7 \cdot 10^{-2}$. The error here is defined as

$$\sum_{i=1}^{n} |a_i - \tilde{a}_i| + \sum_{i=1}^{n-1} |b_i - \tilde{b}_i|$$

where a_i and b_i are the "correct" values and $\tilde{a}_i$ and $\tilde{b}_i$ are the computed values. All entries of T have absolute value smaller than 1. In this section, we present a strategy for choosing π.

Let τ_i be the transposition $(k, k+1)$ in cycle notation. Two permutations π_0 and π_1 *differ by* τ_k if $\pi_1 = \pi_0 \circ \tau_k$. Thus, Λ^{π_1} is obtained from Λ^{π_0} by interchanging the (k,k) and $(k+1,k+1)$ entries. Bidiagonal coordinates $\beta_i^{\pi_0}$ and $\beta_i^{\pi_1}$ are equal except for

$$\beta_{k-1}^{\pi_1} = q_k^{\pi_0} \beta_{k-1}^{\pi_0}, \quad \beta_k^{\pi_1} = -(q_k^{\pi_0})^{-2} \beta_k^{\pi_0}, \quad \beta_{k+1}^{\pi_1} = -q_k^{\pi_0} \beta_{k+1}^{\pi_0}, \tag{3}$$

where

$$q_k^{\pi_0} = \frac{\beta_k^{\pi_0}}{\lambda_{k+1}^{\pi_0} - \lambda_k^{\pi_0}},$$

as can be proved from the formulae relating β's and w's in section 3.

Given inverse data λ_i and w_i, we call a permutation π *tight* if $|q_i^\pi| \leq 1$ for all $i = 1, \ldots, n-1$ and we say that the transposition τ_k is π-*tightening* if $|q_k^\pi| > 1$. From equation 3, it is easy to see that if τ_k is π-tightening then $|q_k^{\pi \circ \tau_k}| < 1$. Clearly, π is tight if and only if there are no π-tightening transpositions. A *tightening sequence* is a maximal sequence (π_m) of permutations such that π_{m+1} differs from π_m by a π_m-tightening transposition τ_{k_m}. Thus, a tightening sequence is either infinite or ends at a tight permutation.

[1] Maple worksheets for all experiments in this paper are available at `http://www.mat.puc-rio.br/~nicolau/papers/invbi-mw`.

Lemma 5.1 *Tightening sequences are finite.*

Proof: Assume by contradiction that there exists an infinite tightening sequence: clearly, there exist $m_0 < m_1$ with $\pi_{m_0} = \pi_{m_1}$. We show that there are no such cycles.

Set $p_{k,\pi} = \prod_{i \geq k} |\beta_i^\pi|$ and use the lexicographical order to define a total order in the permutation group: $\pi_0 \prec \pi_1$ if and only if there exists k' such that $p_{k,\pi_0} = p_{k,\pi_1}$ for $k < k'$ and $p_{k',\pi_0} < p_{k',\pi_1}$. For any m, it follows from equation 3 that $p_{k,\pi_{m+1}} = p_{k,\pi_m}$ for $k < k_m$ and $p_{k_m,\pi_{m+1}} < p_{k_m,\pi_m}$, implying $\pi_{m+1} \prec \pi_m$. By transitivity, $\pi_{m_0} \prec \pi_{m_1} = \pi_{m_0}$, a contradiction. ■

In the example discussed above, there were 4 tight permutations with errors between $2.8 \cdot 10^{-7}$ and $5.3 \cdot 10^{-7}$. Empirical evidence shows that this is frequent: tight permutations usually yield small errors in the inverse bidiagonal algorithm. Thus, upon receiving inverse data λ_i and $w_i > 0$ for a Jacobi matrix T, we first order the w's in decreasing order to obtain a permutation π_0 and then apply tightening transpositions until we reach a tight permutation. Experiments suggest that this takes approximately $n/2$ sweeps.

6 Operational costs

We estimate the number of operations (or flops) and the amount of memory necessary to execute the inverse bidiagonal algorithm. As usual, we only keep track of the number of products and quotients.

The process of finding a tight permutation will not be carefully examined: suffice it to say that, from empirical evidence, the number of operations is approximately Cn^2 where $C < 1/2$.

The matrices $B_{\pi,2}$ and $L_{\pi,2}$ will come up along the algorithm and it is therefore convenient to compute and keep the squares $(\beta_i^\pi)^2$, with an initial cost of n operations and n storage units. The first major step of the algorithm is the computation of $\tilde{r}_1 = L_2^* L_0 e_1$. For $n = 4$, after reordering terms, $\tilde{r}_1$ becomes

$$\left(\begin{array}{l} \dfrac{(\beta_1^\pi)^2(\beta_2^\pi)^2(\beta_3^\pi)^2}{(\lambda_4^\pi - \lambda_1^\pi)^2(\lambda_4^\pi - \lambda_2^\pi)^2(\lambda_4^\pi - \lambda_3^\pi)^2} + \dfrac{(\beta_1^\pi)^2(\beta_2^\pi)^2}{(\lambda_3^\pi - \lambda_1^\pi)^2(\lambda_3^\pi - \lambda_2^\pi)^2} + \dfrac{(\beta_1^\pi)^2}{(\lambda_2^\pi - \lambda_1^\pi)^2} + 1, \\ \dfrac{(\beta_2^\pi)^2(\beta_3^\pi)^2}{(\lambda_4^\pi - \lambda_1^\pi)(\lambda_4^\pi - \lambda_2^\pi)^2(\lambda_4^\pi - \lambda_3^\pi)^2} + \dfrac{(\beta_2^\pi)^2}{(\lambda_3^\pi - \lambda_1^\pi)(\lambda_3^\pi - \lambda_2^\pi)^2} + \dfrac{1}{(\lambda_2^\pi - \lambda_1^\pi)}, \\ \dfrac{(\beta_3^\pi)^2}{(\lambda_4^\pi - \lambda_1^\pi)(\lambda_4^\pi - \lambda_2^\pi)(\lambda_4^\pi - \lambda_3^\pi)^2} + \dfrac{1}{(\lambda_3^\pi - \lambda_1^\pi)(\lambda_3^\pi - \lambda_2^\pi)}, \\ \dfrac{1}{(\lambda_4^\pi - \lambda_1^\pi)(\lambda_4^\pi - \lambda_2^\pi)(\lambda_4^\pi - \lambda_3^\pi)} \end{array} \right).$$

These terms are computed from bottom to top of each column, following the obvious patterns, with a cost of approximately $3n^2/2$ operations and n storage units.

The recursion formula which obtains b_{k-1}, a_k and $\tilde{r}_{k+1}$ only requires $\tilde{r}_{k-1}$ and $\tilde{r}_k$ so that we only need to keep at most three rows of the triangular matrix $\tilde{R}$ at any given time. The number of operations is approximately $2n^2$; also, $n-1$ square roots are needed.

Summing up, given a tight permutation π and the values of β_i^π, a run takes approximately $7n^2/2$ operations, n square roots and $4n$ storage units (provided some units do double duty, first as entries of $\tilde{R}$ and later as a's or b's).

7 Benchmarks

In this section, we compare the de Boor-Golub and inverse bidiagonal algorithms in a few scenarios. This is only possible for irreducible matrices since otherwise, as we saw, the norming constants w break down. The inverse bidiagonal algorithm receives as input permuted eigenvalues λ_i^π, $i = 1, \ldots, n$, and bidiagonal coordinates β_i^π, $i = 1, \ldots, n-1$. In order to allow for comparisons, we must step back and provide as input the eigenvalues λ_i and the norming constants w_i: we then obtain a tight permutation π and compute β_i^π.

It is a common feature of both algorithms that the coefficients a_i, $i = 1, \ldots, n$ and b_i, $i = 1, \ldots, n-1$, are obtained in the order $a_1, b_1, a_2, b_2, a_3, \ldots$. Also, both algorithms admit a reversal by conjugation. More precisely, let P_ρ be the permutation matrix with $(P_\rho)_{ij} = 1$ if and only if $i + j = n + 1$. Let λ_i and w_i be the inverse data for a Jacobi matrix T: the inverse data for $\tilde{T} = P_\rho T P_\rho$ is λ_i and

$$\tilde{w}_i = \frac{c}{w_i \prod_{j \neq i} |\lambda_i - \lambda_j|}$$

for some positive normalizing constant c ([1]). From data λ_i and $\tilde{w}_i$ either algorithm obtains, in this order, $\tilde{a}_1 = a_n, \tilde{b}_1 = b_{n-1}, \tilde{a}_2 = a_{n-1}, \ldots$. Experiments show that it is far wiser to do both things, i.e., to compute the top half of T directly from w_i and the bottom half from $\tilde{w}_i$. In the examples below, this strategy, the *two-sided* algorithms, is always adopted.

We implement in a Maple worksheet both the two-sided de Boor-Golub (BG) and the two-sided inverse bidiagonal (BI) algorithms, generate a sequence of random inverse problems and compare errors for different values of the dimension and of the number of significative digits.

In the first class of examples, random real symmetric tridiagonal matrices T are obtained as follows: the nonzero entries are independent random variables with a Gaussian distribution centered at 0 with variance 1. We then compute the inverse variables λ_i and w_i of T and test the algorithms with these inputs: the norming constants w_i typically span several orders of magnitude. There are cases where either algorithm outperforms the other, but in the average BI fares decisively better than BG. In the worksheet, we repeated this experiment 40 times with dimension $n = 40$, working with 12 significant digits; errors were measured as in the previous section. The run is declared a failure if the error exceeds 0.1: there were two runs where both BG and BI failed, another 30 failed runs for BG and none other for BI.

We next consider matrices near T_0, the Jacobi matrix with diagonal entries equal to 0 and off-diagonal entries equal to 1: for our purposes, T_0 is as good as the free Laplacian. It turns out that T_0 is special from several points of view: the values of both λ_i and w_i can be obtained explicitly, there are no small gaps in the spectrum and all norming constants w_i have roughly the same size. These features, particularly the last one, seem to favor BG. Indeed, for $n = 40$, working with 12 digits, Gaussian perturbations of T_0 with small variance in the off-diagonal entries favors BG: among 40 examples, there are no failures of either algorithm but the errors are smaller for BG than for BI.

References

[1] de Boor, C. and Golub, G. H., The numerically stable reconstruction of a Jacobi matrix from spectral data, Linear Algebra and its Applications 21, 245-260, 1978.

[2] Bloch, A. M., Flaschka, H. and Ratiu, T., A convexity theorem for isospectral manifolds of Jacobi matrices in a compact Lie algebra, Duke Math. J., 61, 41-65, 1990.

[3] Erra, R. and Phillipe, B., On some structured inverse eigenvalue problems, Numerical Algorithms, 15, 15-35, 1997.

[4] Gray, L. J. and Wilson, D. G., Construction of Jacobi matrix from spectral data, Linear Algebra Appl. 14, 131-134, 1976.

[5] Hald, O. H., Inverse eigenvalue problems for Jacobi matrices, Linear Algebra Appl. 14, 63-85, 1976.

[6] Hochstadt, H., On some inverse problems in matrix theory, Arch. Math. 18, 201-207, 1967.

[7] Hochstadt, H., On the construction of a Jacobi matrix from spectral data, Linear Algebra Appl. 8, 435-446, 1974.

[8] Leite, R. S., Saldanha, N. C. and Tomei, C., New inverse data for tridiagonal matrices and the asymptotics of Wilkinson's shift iteration, preprint, `www.arxiv.org/abs/math.NA/0412493`.

[9] Moser, J., Finitely many mass points on the line under the influence of an exponential potential, In: Dynamic systems theory and applications, (ed. J. Moser) 467-497, New York, 1975.

[10] Tomei, C., The Topology of Manifolds of Isospectral Tridiagonal Matrices, Duke Math. J., 51, 981-996, 1984.

Ricardo S. Leite, Departamento de Matemática, UFES
Av. Fernando Ferrari, 514, Vitória, ES 29075-910, Brazil
Nicolau C. Saldanha and Carlos Tomei, Departamento de Matemática, PUC-Rio
R. Marquês de S. Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil

rsleite@cce.ufes.br
nicolau@mat.puc-rio.br; http://www.mat.puc-rio.br/∼nicolau/
tomei@mat.puc-rio.br