Jim Lambers
CME 335
Spring Quarter 2010-11
Lecture 2 Notes

Matrices, Moments and Quadrature, cont'd

We have described how Jacobi matrices can be used to compute nodes and weights for Gaussian quadrature rules for a general positive, increasing measure $\alpha(\lambda)$, which ensures that the Jacobi matrix J_n is not only symmetric but also positive definite. Now, we consider the case of the specific inner product

$$\langle f, g\rangle = \int_a^b f(\lambda)g(\lambda)\,d\alpha(\lambda) = \mathbf{u}^T f(A)g(A)\mathbf{u},$$

with associated norm

$$\|f\|_\alpha = \langle f, f\rangle^{1/2} = (\mathbf{u}^T f(A)^2\mathbf{u})^{1/2}.$$

The underlying measure $\alpha(\lambda)$ allows us to represent the quadratic form $\mathbf{u}^T f(A)\mathbf{u}$ as a Riemann-Stieltjes integral that can be approximated via Gaussian quadrature.

Derivation of the Lanczos Algorithm

We now examine the computation of the required recursion coefficients

$$\alpha_j = \langle xq_{j-1}, q_{j-1}\rangle, \quad \beta_j = \langle p_j, p_j\rangle^{1/2}, \quad j \geq 1.$$

If we define the vectors

$$\mathbf{x}_j = q_{j-1}(A)\mathbf{u}, \quad \mathbf{r}_j = p_j(A)\mathbf{u}, \quad j \geq 1,$$

then it follows that

$$\alpha_j = \mathbf{x}_j^T A\mathbf{x}_j, \quad \beta_j = \|\mathbf{r}_j\|_2.$$

Furthermore,

$$\mathbf{r}_j = p_j(A)\mathbf{u} = (A - \alpha_j I)q_{j-1}(A)\mathbf{u} - \beta_{j-1}q_{j-2}(A)\mathbf{u} = (A - \alpha_j I)\mathbf{x}_j - \beta_{j-1}\mathbf{x}_{j-1}.$$

Putting all of these relations together yields the algorithm

$\mathbf{r}_0 = \mathbf{u}$
$\mathbf{x}_0 = \mathbf{0}$
for $j = 1, 2, \ldots, n$ **do**
  $\beta_{j-1} = \|\mathbf{r}_{j-1}\|_2$
  $\mathbf{x}_j = \mathbf{r}_{j-1}/\beta_{j-1}$

$\alpha_j = \mathbf{x}_j^T A \mathbf{x}_j$
$\mathbf{r}_j = (A - \alpha_j I)\mathbf{x}_j - \beta_{j-1}\mathbf{x}_{j-1}$
end

This is precisely the *Lanczos algorithm* that is often used to approximate extremal eigenvalues of A, and is closely related to the conjugate gradient method for solving symmetric positive definite systems. The vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are called the *Lanczos vectors*. The matrix $X_n = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_n \end{bmatrix}$ satisfies the relations

$$X_n^T X_n = I_n, \quad X_n^T A X_n = J_n.$$

The second relation follows from the above formula for α_j, as well as the relation

$$\beta_j = \langle x q_{j-1}, q_j \rangle = \mathbf{x}_{j-1}^T A \mathbf{x}_j, \quad j \geq 1.$$

The Lanczos vectors allow us to express our approximation of the quadratic form $\mathbf{u}^T f(A)\mathbf{u}$, that involves a function of an $N \times N$ matrix, in terms of a function of an $n \times n$ matrix. We have

$$\begin{aligned} \mathbf{u}^T f(A)\mathbf{u} &= (\beta_0 \mathbf{x}_1)^T f(A)(\beta_0 \mathbf{x}_1) \\ &= \beta_0^2 (X_n \mathbf{e}_1)^T f(A)(X_n \mathbf{e}_1) \\ &= \langle p_0, p_0 \rangle \mathbf{e}_1^T X_n^T f(A) X_n \mathbf{e}_1 \\ &\approx \langle 1, 1, \rangle \mathbf{e}_1^T f(X_n^T A X_n)\mathbf{e}_1 \\ &\approx \mathbf{u}^T \mathbf{u} \mathbf{e}_1^T f(J_n)\mathbf{e}_1 \\ &\approx \|\mathbf{u}\|_2^2 [f(J_n)]_{11}. \end{aligned}$$

It follows that if the particular function f is conducive to computing the $(1, 1)$ entry of a tridiagonal matrix efficiently, then there is no need to compute the nodes and weights for Gaussian quadrature explicitly.

Now, suppose that J_n has the spectral decomposition

$$J_n = U_n \Lambda_n U_n^T,$$

where U_n is an orthogonal matrix whose columns are the eigenvectors of J_n, and Λ_n is a diagonal matrix that contains the eigenvalues. Then we have

$$\begin{aligned} \mathbf{u}^T f(A)\mathbf{u} &\approx \|\mathbf{u}\|_2^2 \mathbf{e}_1^T f(U_n \Lambda_n U_n^T)\mathbf{e}_1 \\ &\approx \|\mathbf{u}\|_2^2 \mathbf{e}_1^T U_n f(\Lambda_n) U_n^T \mathbf{e}_1 \\ &\approx \|\mathbf{u}\|_2^2 \sum_{j=1}^n f(t_j) u_{1j}^2 \\ &\approx \sum_{j=1}^n f(t_j)\, w_j. \end{aligned}$$

Thus we have recovered the relationship between the quadrature weights and the eigenvectors of J_n.

If we let

$$I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda), \quad L_G[f] = \sum_{j=1}^n f(t_j)\, w_j,$$

then it follows from the fact that $L_G[f]$ is the exact integral of a polynomial that interpolates f at the nodes that the error in this quadrature rule is

$$I[f] - L_G[f] = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b \prod_{j=1}^n (\lambda - t_j)^2\, d\alpha(\lambda),$$

where ξ is an unknown point in (a, b). The above error formula can most easily be derived by defining $L_G[f]$ as the exact integral of the *Hermite* interpolating polynomial of f, rather than the Lagrange interpolant. This is possible because for a Gaussian rule, the weights satisfy

$$w_j = \int_a^b L_j(\lambda)\, d\alpha(\lambda) = \int_a^b L_j(\lambda)^2\, d\alpha(\lambda),$$

where L_j is the jth Lagrange polynomial for the interpolation points $t_1, \ldots, t_n$.

Now, suppose that the even-order derivatives of f are positive on (a, b). Then, the error is positive, which means that the Gaussian quadrature approximation is a *lower* bound for $I[f]$. This is the case, for example, if $f(\lambda) = \frac{1}{\lambda}$, or $f(\lambda) = e^{-\lambda t}$ where $t > 0$.

Prescribing Nodes

It is desirable to also obtain an upper bound for $I[f]$. To that end, we consider modifying the Gaussian rule to prescribe that one of the nodes be $\lambda = a$, the smallest eigenvalue of A. We assume that a good approximation of a can be computed, which is typically possible using the Lanczos algorithm, as it is well-suited to approximating extremal eigenvalues of a symmetric positive definite matrix.

We wish to construct an augmented Jacobi matrix J_{n+1} that has a as an eigenvalue; that is, $J_{n+1} - aI$ is a singular matrix. From the recurrence relation for the orthonormal polynomials that define the Jacobi matrix, we obtain the system of equations

$$(J_{n+1} - a_I)\mathbf{Q}_{n+1}(a) = \mathbf{0},$$

where the (unnormalized) eigenvector $\mathbf{Q}_{n+1}(a)$ is defined as $\mathbf{Q}_n$ was before. That is, $[Q_{n+1}(a)]_i = q_{i-1}(a)$, for $i = 1, 2, \ldots, n + 1$.

Decomposing this system of equations into blocks yields

$$\begin{bmatrix} J_n - aI & \beta_n \mathbf{e}_n \\ \beta_n \mathbf{e}_n^T & \alpha_{N+1} - a \end{bmatrix} \begin{bmatrix} \mathbf{Q}_n(a) \\ q_n(a) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}.$$

From the first n rows of this system, we obtain

$$(J_n - aI)\mathbf{Q}_n(a) = -\beta_n q_n(a)\mathbf{e}_n,$$

and the last row yields

$$\alpha_{n+1} = a - \frac{\beta_n q_{n-1}(a)}{q_n(a)}.$$

Now, if we solve the system

$$(J_n - aI)\delta(a) = \beta_n^2 \mathbf{e}_n,$$

we obtain

$$\mathbf{Q}_n(a) = -\frac{q_n(a)\delta(a)}{\beta_n},$$

which yields

$$\begin{aligned}
\alpha_{n+1} &= a - \frac{\beta_n q_{n-1}(a)}{q_n(a)} \\
&= a - \frac{\beta_n}{q_n(a)}\left(-\frac{q_n(a)[\delta(a)]_n}{\beta_n}\right) \\
&= a + [\delta(a)]_n.
\end{aligned}$$

Thus we have obtained a Jacobi matrix that has the prescribed node a as an eigenvalue.

The error in this quadrature rule, which is known as a *Gauss-Radau* rule, is

$$I[f] - L_{GR}[f] = \frac{f^{(2n+1)}(\xi)}{(2n+1)!}\int_a^b (\lambda - a)\prod_{j=1}^{n}(\lambda - t_j)^2\, d\alpha(\lambda).$$

Because the factor $\lambda - a$ is always positive on (a, b), it follows that this Gauss-Radau rule yields an *upper* bound on $I[f]$ if the derivatives of odd order are negative, which is the case for $f(\lambda) = \frac{1}{\lambda}$ and $f(\lambda) = e^{-\lambda t}$ where $t > 0$. Similarly, if we prescribe the node $\lambda = b$, then Gauss-Radau quadrature yield a lower bound, which may be sharper than the lower bound obtained from Gaussian quadrature.

Now, suppose we wish to prescribe *both* $\lambda = a$ and $\lambda = b$ as nodes. This yields what is known as a *Gauss-Lobatto* quadrature rule. To obtain such a rule, we again augment the Jacobi matrix J_n to obtain a new matrix J_{n+1} that has both a and b as eigenvalues. In contrast with Gauss-Radau rules, however, it is necessary to determine both β_n and α_{n+1} so that this is the case.

The recurrence relation for orthonormal polynomials, and the requirements that both a and b are roots of $q_{n+1}(x)$ yields the equations

$$0 = (a - \alpha_{n+1})q_n(a) - \beta_n q_{n-1}(a),$$

$$0 = (b - \alpha_{n+1})q_n(b) - \beta_n q_{n-1}(b),$$

or, in matrix-vector form,

$$\begin{bmatrix} q_n(a) & q_{n-1}(a) \\ q_n(b) & q_{n-1}(b) \end{bmatrix} \begin{bmatrix} \alpha_{n+1} \\ \beta_n \end{bmatrix} = \begin{bmatrix} aq_n(a) \\ bq_n(b) \end{bmatrix}.$$

The recurrence relation also yields the systems of equations

$$(J_n - aI)\mathbf{Q}_n(a) = -\beta_n q_n(a)\mathbf{e}_n, \quad (J_n - bI)\mathbf{Q}_n(b) = -\beta_n q_n(b)\mathbf{e}_n.$$

If we define $\delta(a)$ and $\delta(b)$ to be the solutions of

$$(J_n - aI)\delta(a) = \mathbf{e}_n, \quad (J_n - bI)\delta(b) = \mathbf{e}_n,$$

it follows that

$$\mathbf{Q}_n(a) = -\beta_n q_n(a)\delta(a), \quad \mathbf{Q}_n(b) = -\beta_n q_n(b)\delta(b),$$

and therefore

$$q_{n-1}(a) = -\beta_n q_n(a)[\delta(a)]_n, \quad q_{n-1}(b) = -\beta_n q_n(b)[\delta(b)]_n.$$

It follows from

$$\begin{bmatrix} 1 & \frac{q_{n-1}(a)}{q_n(a)} \\ 1 & \frac{q_{n-1}(b)}{q_n(b)} \end{bmatrix} \begin{bmatrix} \alpha_{n+1} \\ \beta_n \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

that the recursion coefficients β_n and α_{n+1} satisfy

$$\begin{bmatrix} 1 & -[\delta(a)]_n \\ 1 & -[\delta(b)]_n \end{bmatrix} \begin{bmatrix} \alpha_{n+1} \\ \beta_n \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}.$$

The error in this Gauss-Lobatto rule is

$$I[f] - L_{GL}[f] = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b (\lambda - b)(\lambda - a) \prod_{j=1}^{n-1} (\lambda - t_j)^2.$$

Because a and b are the endpoints of the interval of integration, and $(\lambda - b)(\lambda - a) < 0$ on (a, b), Gauss-Lobatto quadrature yields an *upper* bound if the even-order derivatives of f are positive on (a, b).

The Case of $\mathbf{u} \neq \mathbf{v}$

When using the above quadrature rules to approximate the bilinear form $\mathbf{u}^T f(A)\mathbf{v}$, where $\mathbf{u} \neq \mathbf{v}$, care must be exercised, because the underlying measure $\alpha(\lambda)$ is no longer guaranteed to be positive and increasing. This can lead to negative weights, which numerically destabilizes the quadrature rule.

To construct Gaussian, Gauss-Radau or Gauss-Lobatto quadrature rules, the *unsymmetric* Lanczos algorithm must be used, with left initial vector $\mathbf{u}$ and right initial vector $\mathbf{v}$. This algorithm yields two *biorthogonal* sequences of Lanczos vectors

$$\mathbf{x}_j = q_{j-1}(A)\mathbf{v}, \quad \mathbf{y}_j = \tilde{q}_{j-1}(A)\mathbf{u}, \quad j = 1, 2, \ldots, n,$$

such that

$$\mathbf{y}_i^T \mathbf{x}_j = \delta_{ij}.$$

However, because of the symmetry of A, we have

$$\tilde{q}_j(\lambda) = \pm q_j(\lambda), \quad j = 0, 1, \ldots, n-1,$$

so each underlying sequence of polymomials is still orthogonal with respect to the measure $\alpha(\lambda)$.

These sequences satisfy 3-term recurrence relations

$$\beta_j q_j(x) = p_j(x) = (x - \alpha_j) q_{j-1}(x) - \gamma_{j-1} q_{j-2}(x),$$

$$\gamma_j \tilde{q}_j(x) = \tilde{p}_j(x) = (x - \alpha_j) \tilde{q}_{j-1}(x) - \beta_{j-1} \tilde{q}_{j-2}(x),$$

for $j \geq 1$. It follows that

$$\alpha_j = \langle \tilde{q}_{j-1}, x q_{j-1} \rangle = \mathbf{y}_j^T A \mathbf{x}_j,$$

$$\gamma_j \beta_j = \langle \tilde{p}_j, p_j \rangle = \tilde{\mathbf{r}}_j^T \mathbf{r}_j,$$

where $\tilde{r}_j = \tilde{p}_j(A)\mathbf{u}$. The factorization of $\gamma_j \beta_j$ into γ_j and β_j is arbitrary, but is normally chosen so that $\gamma_j = \pm \beta_j$.

If, for any index $j = 0, 1, \ldots, n-1$, $\gamma_j = -\beta_j$, then the Jacobi matrix J_n is no longer symmetric. In this case, the eigenvalues of J_n are still the Gaussian quadrature nodes. They are still real and lie within the interval (a, b), because they are the roots of a member of a sequence of orthogonal polynomials. However, the weights are now the products of the first components of the left and right eigenvectors, which are no longer guaranteed to be the same because the Jacobi matrix is not symmetric. It follows that the weights are no longer guaranteed to be positive, as they are in the case of a symmetric Jacobi matrix.

As long as the unsymmetric Lanczos algorithm does not experience *serious breakdown*, in which $\gamma_j \beta_j = 0$, n-node Gaussian quadrature is exact for polynomials of degree up to $2n-1$, as in the case of quadratic forms. However, because the weights can be negative, typically alternative approaches are used to approximate bilinear forms using these quadrature rules. Some such approaches are:

- Using a perturbation of a quadratic form. The bilinear form is rewritten as

  $$\mathbf{u}^T f(A) \mathbf{v} = \frac{1}{\delta} [\mathbf{u}^T f(A)(\mathbf{u} + \delta \mathbf{v}) - \mathbf{u}^T f(A) \mathbf{u}],$$

  where the parameter δ is chosen sufficiently small so that the measure of the perturbed quadratic form is still positive and increasing.

- Using a difference of quadratic forms:

$$\mathbf{u}^T f(A)\mathbf{v} = \frac{1}{4}[(\mathbf{u}+\mathbf{v})^T f(A)(\mathbf{u}+\mathbf{v}) - (\mathbf{u}-\mathbf{v})^T f(A)(\mathbf{u}-\mathbf{v}),$$

  and the symmetric Lanczos algorithm can be used to approximate both quadratic forms.

- A block approach: the 2×2 matrix integral

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}^T f(A) \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}$$

  is instead approximated. The *block Lanczos algorithm*, due to Golub and Underwood, is applied to A with the initial block $\begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}$. The result is a $2n \times 2n$ block tridiagonal matrix with 2×2 blocks playing the role of recursion coefficients. The eigenvalues of this matrix serve as quadrature nodes, and the first two components of each eigenvector, taken in outer products with themselves, play the role of quadrature weights.

One important drawback of all of these approaches is that it is not possible to obtain upper or lower bounds for $\mathbf{u}^T f(A)\mathbf{v}$, whereas this is possible when it is approximated by a single quadrature rule. Gauss-Radau and Gauss-Lobatto rules can also be constructed in a similar manner as for quadratic forms. Details can be found in the paper of Golub and Meurant.