A. Further Background on Gauss Quadrature

We present below a more detailed summary of material on Gauss quadrature to make the paper self-contained.

A.1. Selecting weights and nodes

We've described that the Riemann-Stieltjes integral could be expressed as

$$I[f] := Q_n + R_n = \sum\nolimits_{i=1}^{n} \omega_i f(\theta_i) + \sum\nolimits_{i=1}^{m} \nu_i f(\tau_i) + R_n[f],$$

where Q_n denotes the nth degree approximation and R_n denotes a remainder term. The weights $\{\omega_i\}_{i=1}^n$, $\{\nu_i\}_{i=1}^m$ and nodes $\{\theta_i\}_{i=1}^n$ are chosen such that for all polynomials of degree less than $2n+m-1$, denoted $f \in \mathbb{P}^{2n+m-1}$, we have *exact* interpolation $I[f] = Q_n$. One way to compute weights and nodes is to set $f(x) = x^i$ for $i \leq 2n+m-1$ and then use this exact nonlinear system. But there is an easier way to obtain weights and nodes, namely by using polynomials orthogonal with respect to the measure α. Specifically, we construct a sequence of *orthogonal polynomials* $p_0(\lambda), p_1(\lambda), \ldots$ such that $p_i(\lambda)$ is a polynomial in λ of degree exactly k, and p_i, p_j are orthogonal, i.e., they satisfy

$$\int_{\lambda_{\min}}^{\lambda_{\max}} p_i(\lambda) p_j(\lambda) d\alpha(\lambda) = \begin{cases} 1, & i = j \\ 0, & \text{otherwise.} \end{cases}$$

The roots of p_n are distinct, real and lie in the interval of $[\lambda_{\min}, \lambda_{\max}]$, and form the nodes $\{\theta_i\}_{i=1}^n$ for Gauss quadrature (see, e.g., (Golub & Meurant, 2009, Ch. 6)).

Consider the two *monic polynomials* whose roots serve as quadrature nodes:

$$\pi_n(\lambda) = \prod\nolimits_{i=1}^{n} (\lambda - \theta_i), \quad \rho_m(\lambda) = \prod\nolimits_{i=1}^{m} (\lambda - \tau_i),$$

where $\rho_0 = 1$ for consistency. We further denote $\rho_m^+ = \pm\rho_m$, where the sign is taken to ensure $\rho_m^+ \geq 0$ on $[\lambda_{\min}, \lambda_{\max}]$. Then, for $m > 0$, we calculate the quadrature weights as

$$\omega_i = I\left[\frac{\rho_m^+(\lambda)\pi_n(\lambda)}{\rho_m^+(\theta_i)\pi_n'(\theta_i)(\lambda - \theta_i)}\right], \quad \nu_j = I\left[\frac{\rho_m^+(\lambda)\pi_n(\lambda)}{(\rho_m^+)'(\tau_j)\pi_n(\tau_j)(\lambda - \tau_j)}\right],$$

where $f'(\lambda)$ denotes the derivative of f with respect to λ. When $m = 0$ the quadrature degenerates to Gauss quadrature and we have

$$\omega_i = I\left[\frac{\pi_n(\lambda)}{\pi_n'(\theta_i)(\lambda - \theta_i)}\right].$$

Although we have specified how to select nodes and weights for quadrature, these ideas cannot be applied to our problem because the measure α is unknown. Indeed, calculating the measure explicitly would require knowing the entire spectrum of A, which is as good as explicitly computing $f(A)$, hence untenable for us. The next section shows how to circumvent the difficulties due to unknown α.

A.2. Gauss Quadrature Lanczos (GQL)

The key idea to circumvent our lack of knowledge of α is to recursively construct polynomials called *Lanczos polynomials*. The construction ensures their orthogonality with respect to α. Concretely, we construct Lanczos polynomials via the following three-term recurrence:

$$\begin{aligned} &\beta_i p_i(\lambda) = (\lambda - \alpha_i) p_{i-1}(\lambda) - \beta_{i-1} p_{i-2}(\lambda), \quad i = 1, 2, \ldots, n \\ &p_{-1}(\lambda) \equiv 0; \quad p_0(\lambda) \equiv 1, \end{aligned} \tag{A.1}$$

while ensuring $\int_{\lambda_{\min}}^{\lambda_{\max}} d\alpha(\lambda) = 1$. We can express (A.1) in matrix form by writing

$$\lambda P_n(\lambda) = J_n P_n(\lambda) + \beta_n p_n(\lambda) e_n,$$

where $P_n(\lambda) := [p_0(\lambda), \ldots, p_{n-1}(\lambda)]^\top$, e_n is nth canonical unit vector, and J_n is the tridiagonal matrix

$$J_n = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}. \tag{A.2}$$

This matrix is known as the *Jacobi matrix*, and is closed related to Gauss quadrature. The following well-known theorem makes this relation precise.

Theorem 10 ((Wilf, 1962; Golub & Welsch, 1969))**.** *The eigenvalues of J_n form the nodes $\{\theta_i\}_{i=1}^n$ of Gauss-type quadratures. The weights $\{\omega_i\}_{i=1}^n$ are given by the squares of the first elements of the normalized eigenvectors of J_n.*

Thus, if J_n has the eigendecomposition $J_n = P_n^\top \Gamma P_n$, then for Gauss quadrature Theorem 10 yields

$$Q_n = \sum\nolimits_{i=1}^n \omega_i f(\theta_i) = e_1^\top P_n^\top f(\Gamma) P_n e_1 = e_1^\top f(J_n) e_1. \tag{A.3}$$

Specialization. We now specialize to our main focus, $f(A) = A^{-1}$, for which we prove more precise results. In this case, (A.3) becomes $Q_n = [J_n^{-1}]_{1,1}$. The task now is to compute Q_n, and given A, u to obtain the Jacobi matrix J_n.

Fortunately, we can efficiently calculate J_n iteratively using the *Lanczos Algorithm* (Lanczos, 1950). Suppose we have an estimate J_i, in iteration $(i+1)$ of Lanczos, we compute the tridiagonal coefficients α_{i+1} and β_{i+1} and add them to this estimate to form J_{i+1}. As to Q_n, assuming we have already computed $[J_i^{-1}]_{1,1}$, letting $j_i = J_i^{-1} e_i$ and invoking the Sherman-Morrison identity (Sherman & Morrison, 1950) we obtain the recursion:

$$[J_{i+1}^{-1}]_{1,1} = [J_i^{-1}]_{1,1} + \frac{\beta_i^2([j_i]_1)^2}{\alpha_{i+1} - \beta_i^2 [j_i]_i}, \tag{A.4}$$

where $[j_i]_1$ and $[j_i]_i$ can be recursively computed using a Cholesky-like factorization of J_i (Golub & Meurant, 2009, p.31).

For Gauss-Radau quadrature, we need to modify J_i so that it has a prescribed eigenvalue. More precisely, we extend J_i to J_i^{lr} for left Gauss-Radau (J_i^{rr} for right Gauss-Radau) with β_i on the off-diagonal and α_i^{lr} (α_i^{rr}) on the diagonal, so that J_i^{lr} (J_i^{rr}) has a prescribed eigenvalue of $\lambda_{\min}$ ($\lambda_{\max}$).

For Gauss-Lobatto quadrature, we extend J_i to J_i^{lo} with values β_i^{lo} and α_i^{lo} chosen to ensure that J_i^{lo} has the prescribed eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$. For more detailed on the construction, see (Golub, 1973).

For all methods, the approximated values are calculated as $[(J_i')^{-1}]_{1,1}$, where $J_i' \in \{J_i^{\mathrm{lr}}, J_i^{\mathrm{rr}}, J_i^{\mathrm{lo}}\}$ is the modified Jacobi matrix. Here J_i' is constructed at the i-th iteration of the algorithm.

The algorithm for computing Gauss, Gauss-Radau, and Gauss-Lobatto quadrature rules with the help of Lanczos iteration is called *Gauss Quadrature Lanczos* (GQL) and is shown in (Golub & Meurant, 1997). We recall its pseudocode in Algorithm 1 to make our presentation self-contained (and for our proofs in Section 4).

The error of approximating $I[f]$ by Gauss-type quadratures can be expressed as

$$R_n[f] = \frac{f^{(2n+m)}(\xi)}{(2n+m)!} I[\rho_m \pi_n^2],$$

for some $\xi \in [\lambda_{\min}, \lambda_{\max}]$ (see, e.g., (Stoer & Bulirsch, 2013)). Note that ρ_m does not change sign in $[\lambda_{\min}, \lambda_{\max}]$; but with different values of m and τ_j we obtain different (but fixed) signs for $R_n[f]$ using $f(\lambda) = 1/\lambda$ and $\lambda_{\min} > 0$. Concretely, for Gauss quadrature $m = 0$ and $R_n[f] \geq 0$; for left Gauss-Radau $m = 1$ and $\tau_1 = \lambda_{\min}$, so we have $R_n[f] \leq 0$; for right Gauss-Radau we have $m = 1$ and $\tau_1 = \lambda_{\max}$, thus $R_n[f] \geq 0$; while for Gauss-Lobatto we have $m = 2$, $\tau_1 = \lambda_{\min}$ and $\tau_2 = \lambda_{\max}$, so that $R_n[f] \leq 0$. This behavior of the errors clearly shows the ordering relations between the target values and the approximations made by the different quadrature rules. Lemma 2 (see e.g., (Meurant, 1997)) makes this claim precise.

Algorithm 5 Gauss Quadrature Lanczos (GQL)

Input: u and A the corresponding vector and matrix, $\lambda_{\min}$ and $\lambda_{\max}$ lower and upper bounds for the spectrum of A
Output: g_i, g_i^{rr}, g_i^{lr} and g_i^{lo} the Gauss, right Gauss-Radau, left Gauss-Radau and Gauss-Lobatto quadrature computed at i-th iteration
Initialize: $u_{-1} = 0$, $u_0 = u/\|u\|$, $\alpha_1 = u_0^\top A u_0$, $\beta_1 = \|(A - \alpha_1 I)u_0\|$, $g_1 = \|u\|/\alpha_1$, $c_1 = 1$, $\delta_1 = \alpha_1$, $\delta_1^{\text{lr}} = \alpha_1 - \lambda_{\min}$, $\delta_1^{\text{rr}} = \alpha_1 - \lambda_{\max}$, $u_1 = (A - \alpha_1 I)u_0/\beta_1$, $i = 2$
while $i \le N$ **do**
  $\alpha_i = u_{i-1}^\top A u_{i-1}$ {Lanczos Iteration}
  $\tilde{u}_i = A u_{i-1} - \alpha_i u_{i-1} - \beta_{i-1} u_{i-2}$
  $\beta_i = \|\tilde{u}_i\|$
  $u_i = \tilde{u}_i/\beta_i$
  $g_i = g_{i-1} + \frac{\|u\|\beta_{i-1}^2 c_{i-1}^2}{\delta_{i-1}(\alpha_i\delta_{i-1} - \beta_{i-1}^2)}$ {Update g_i with Sherman-Morrison formula}
  $c_i = c_{i-1}\beta_{i-1}/\delta_{i-1}$
  $\delta_i = \alpha_i - \frac{\beta_{i-1}^2}{\delta_{i-1}}$, $\delta_i^{\text{lr}} = \alpha_i - \lambda_{\min} - \frac{\beta_{i-1}^2}{\delta_{i-1}^{\text{lr}}}$, $\delta_i^{\text{rr}} = \alpha_i - \lambda_{\max} - \frac{\beta_{i-1}^2}{\delta_{i-1}^{\text{rr}}}$
  $\alpha_i^{\text{lr}} = \lambda_{\min} + \frac{\beta_i^2}{\delta_i^{\text{lr}}}$, $\alpha_i^{\text{rr}} = \lambda_{\max} + \frac{\beta_i^2}{\delta_i^{\text{rr}}}$ {Solve for J_i^{lr} and J_i^{rr}}
  $\alpha_i^{\text{lo}} = \frac{\delta_i^{\text{lr}}\delta_i^{\text{rr}}}{\delta_i^{\text{rr}} - \delta_i^{\text{lr}}}\left(\frac{\lambda_{\max}}{\delta_i^{\text{lr}}} - \frac{\lambda_{\min}}{\delta_i^{\text{rr}}}\right)$, $(\beta_i^{\text{lo}})^2 = \frac{\delta_i^{\text{lr}}\delta_i^{\text{rr}}}{\delta_i^{\text{rr}} - \delta_i^{\text{lr}}}(\lambda_{\max} - \lambda_{\min})$ {Solve for J_i^{lo}}
  $g_i^{\text{lr}} = g_i + \frac{\beta_i^2 c_i^2 \|u\|}{\delta_i(\alpha_i^{\text{lr}}\delta_i - \beta_i^2)}$, $g_i^{\text{rr}} = g_i + \frac{\beta_i^2 c_i^2 \|u\|}{\delta_i(\alpha_i^{\text{rr}}\delta_i - \beta_i^2)}$, $g_i^{\text{lo}} = g_i + \frac{(\beta_i^{\text{lo}})^2 c_i^2 \|u\|}{\delta_i(\alpha_i^{\text{lo}}\delta_i - (\beta_i^{\text{lo}})^2)}$ {Update g_i^{rr}, g_i^{lr} and g_i^{lo} with Sherman-Morrison formula}
  $i = i + 1$
end while

Lemma 11. *Let g_i, g_i^{lr}, g_i^{rr}, and g_i^{lo} be the approximations at the i-th iteration of Gauss, left Gauss-Radau, right Gauss-Radau, and Gauss-Lobatto quadrature, respectively. Then, g_i and g_i^{rr} provide lower bounds on $u^\top A^{-1} u$, while g_i^{lr} and g_i^{lo} provide upper bounds.*

The final connection we recall as background is the method of conjugate gradients. This helps us analyze the speed at which quadrature converges to the true value (assuming exact arithmetic).

A.3. Relation with Conjugate Gradient

While Gauss-type quadratures relate to the Lanczos algorithm, Lanczos itself is closely related to conjugate gradient (CG) (Hestenes & Stiefel, 1952), a well-known method for solving $Ax = b$ for positive definite A.

We recap this connection below. Let x_k be the estimated solution at the k-th CG iteration. If x^* denotes the true solution to $Ax = b$, then the *error* ε_k and *residual* r_k are defined as

$$\varepsilon_k := x^* - x_k, \qquad r_k = A\varepsilon_k = b - Ax_k, \tag{A.5}$$

At the k-th iteration, x_k is chosen such that r_k is orthogonal to the k-th *Krylov space*, i.e., the linear space $\mathcal{K}_k$ spanned by $\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$. It can be shown (Meurant, 2006) that r_k is a scaled Lanczos vector from the k-th iteration of Lanczos started with r_0. Noting the relation between Lanczos and Gauss quadrature applied to appoximate $r_0^\top A^{-1} r_0$, one obtains the following theorem that relates CG with GQL.

Theorem 12 (CG and GQL; (Meurant, 1999))**.** *Let ε_k be the error as in (A.5), and let $\|\varepsilon_k\|_A^2 := \varepsilon_k^T A \varepsilon_k$. Then, it holds that*

$$\|\varepsilon_k\|_A^2 = \|r_0\|^2([J_N^{-1}]_{1,1} - [J_k^{-1}]_{1,1}),$$

where J_k is the Jacobi matrix at the k-th Lanczos iteration starting with r_0.

Finally, the rate at which $\|\varepsilon_k\|_A^2$ shrinks has also been well-studied, as noted below.

Theorem 13 (CG rate, see e.g. (Shewchuk, 1994))**.** *Let ε_k be the error made by CG at iteration k when started with x_0.*

Let κ be the condition number of A, i.e., $\kappa = \lambda_1/\lambda_N$. Then, the error norm at iteration k satisfies

$$\|\varepsilon_k\|_A \leq 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^k \|\varepsilon_0\|_A.$$

Due to these explicit relations between CG and Lanczos, as well as between Lanczos and Gauss quadrature, we readily obtain the following convergence rate for relative error of Gauss quadrature.

Theorem 14 (Gauss quadrature rate)**.** *The i-th iterate of Gauss quadrature satisfies the relative error bound*

$$\frac{g_N - g_i}{g_N} \leq 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^i.$$

Proof. This is obtained by exploiting relations among CG, Lanczos and Gauss quadrature. Set $x_0 = 0$ and $b = u$. Then, $\varepsilon_0 = x^*$ and $r_0 = u$. An application of Theorem 12 and Theorem 13 thus yields the bound

$$\begin{aligned}\|\varepsilon_i\|_A^2 &= \|u\|^2([J_N^{-1}]_{1,1} - [J_i^{-1}]_{1,1}) = g_N - g_i \\ &\leq \quad 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^i \|\varepsilon_0\|_A = 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^i u^\top A^{-1}u = 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^i g_N\end{aligned}$$

where the last equality draws from Lemma 15. □

In other words, Theorem 14 shows that the iterates of Gauss quadrature converge linearly.

B. Proofs for Main Theoretical Results

We begin by proving an exactness property of Gauss and Gauss-Radau quadrature.

Lemma 15 (Exactness)**.** *With A being symmetric positive definite with simple eigenvalues, the iterates g_N, g_N^{lr}, and g_N^{rr} are exact. Namely, after N iterations they satisfy*

$$g_N = g_N^{lr} = g_N^{rr} = u^\top A^{-1}u.$$

Proof. Observe that the Jacobi tridiagonal matrix can be computed via Lanczos iteration, and Lanczos is essentially essentially an iterative tridiagonalization of A. At the i-th iteration we have $J_i = V_i^\top A V_i$, where $V_i \in \mathbb{R}^{N\times i}$ are the first i Lanczos vectors (i.e., a basis for the i-th Krylov space). Thus, $J_N = V_N^\top A V_N$ where V_N is an $N \times N$ orthonormal matrix, showing that J_N has the same eigenvalues as A. As a result $\pi_N(\lambda) = \prod_{i=1}^N(\lambda - \lambda_i)$, and it follows that the remainder

$$R_N[f] = \frac{f^{(2N)}(\xi)}{(2N)!} I[\pi_N^2] = 0,$$

for some scalar $\xi \in [\lambda_{\min}, \lambda_{\max}]$, which shows that g_N is exact for $u^\top A^{-1}u$. For left and right Gauss-Radau quadrature, we have $\beta_N = 0$, $\alpha_N^{\text{lr}} = \lambda_{\min}$, and $\alpha_N^{\text{rr}} = \lambda_{\max}$, while all other elements of the $(N+1)$-th row or column of J_N' are zeros. Thus, the eigenvalues of J_N' are $\lambda_1, \ldots, \lambda_N, \tau_1$, and $\pi_N(\lambda)$ again equals $\prod_{i=1}^N(\lambda - \lambda_i)$. As a result, the remainder satisfies

$$R_N[f] = \frac{f^{(2N)}(\xi)}{(2N)!} I[(\lambda - \tau_1)\pi_N^2] = 0,$$

from which it follows that both g_N^{rr} and g_N^{lr} are exact. □

The convergence rate in Theorem 13 and the final exactness of iterations in Lemma 15 does not necessarily indicate that we are making progress at each iterations. However, by exploiting the relations to CG we can indeed conclude that we are making progress in each iteration in Gauss quadrature.

Theorem 16. *The approximation g_i generated by Gauss quadrature is monotonically nondecreasing, i.e.,*

$$g_i \leq g_{i+1}, \quad \textit{for } i < N.$$

Proof. At each iteration r_i is taken to be orthogonal to the i-th Krylov space: $\mathcal{K}_i = \mathrm{span}\{u, Au, \ldots, A^{i-1}u\}$. Let Π_i be the projection onto the complement space of $\mathcal{K}_i$. The residual then satisfies

$$\begin{aligned}
\|\varepsilon_{i+1}\|_A^2 &= \varepsilon_{i+1}^T A \varepsilon_{i+1} = r_{i+1}^\top A^{-1} r_{i+1} \\
&= (\Pi_{i+1} r_i)^\top A^{-1} \Pi_{i+1} r_i \\
&= r_i^\top (\Pi_{i+1}^\top A^{-1} \Pi_{i+1}) r_i \le r_i A^{-1} r_i,
\end{aligned}$$

where the last inequality follows from $\Pi_{i+1}^\top A^{-1} \Pi_{i+1} \preceq A^{-1}$. Thus $\|\varepsilon_i\|_A^2$ is monotonically nonincreasing, whereby $g_N - g_i \ge 0$ is monotonically decreasing and thus g_i is monotonically nondecreasing. □

Before we proceed to Gauss-Radau, let us recall a useful theorem and its corollary.

Theorem 17 (Lanczos Polynomial (Golub & Meurant, 2009))**.** *Let u_i be the vector generated by Algorithm 1 at the i-th iteration; let p_i be the Lanczos polynomial of degree i. Then we have*

$$u_i = p_i(A) u_0, \quad \textit{where } p_i(\lambda) = (-1)^i \frac{\det(J_i - \lambda I)}{\prod_{j=1}^i \beta_j}.$$

From the expression of Lanczos polynomial we have the following corollary specifying the sign of the polynomial at specific points.

Corollary 18. *Assume $i < N$. If i is odd, then $p_i(\lambda_{\min}) < 0$; for even i, $p_i(\lambda_{\min}) > 0$, while $p_i(\lambda_{\max}) > 0$ for any $i < N$.*

Proof. Since $J_i = V_i^\top A V_i$ is similar to A, its spectrum is bounded by $\lambda_{\min}$ and $\lambda_{\max}$ from left and right. Thus, $J_i - \lambda_{\min}$ is positive semi-definite, and $J_i - \lambda_{\max}$ is negative semi-definite. Taking $(-1)^i$ into consideration we will get the desired conclusions. □

We are ready to state our main result that compares (right) Gauss-Radau with Gauss quadrature.

Theorem 19 (Theorem 4 in the main text)**.** *Let $i < N$. Then, g_i^{rr} gives better bounds than g_i but worse bounds than g_{i+1}; more precisely,*

$$g_i \le g_i^{rr} \le g_{i+1}, \quad i < N. \tag{B.1}$$

Proof. We prove inequality (B.1) using the recurrences satisfied by g_i and g_i^{rr} (see Alg. 1)

Upper bound: $g_i^{rr} \le g_{i+1}$. The iterative quadrature algorithm uses the recursive updates

$$\begin{aligned}
g_i^{\mathrm{rr}} &= g_i + \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i^{\mathrm{rr}} \delta_i - \beta_i^2)}, \\
g_{i+1} &= g_i + \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_{i+1} \delta_i - \beta_i^2)}.
\end{aligned}$$

It suffices to thus compare α_i^{rr} and α_{i+1}. The three-term recursion for Lanczos polynomials shows that

$$\begin{aligned}
\beta_{i+1} p_{i+1}(\lambda_{\max}) &= (\lambda_{\max} - \alpha_{i+1}) p_i(\lambda_{\max}) - \beta_i p_{i-1}(\lambda_{\max}) > 0, \\
\beta_{i+1} p_{i+1}^*(\lambda_{\max}) &= (\lambda_{\max} - \alpha_i^{\mathrm{rr}}) p_i(\lambda_{\max}) - \beta_i p_{i-1}(\lambda_{\max}) = 0,
\end{aligned}$$

where p_{i+1} is the original Lanczos polynomial, and p_{i+1}^* is the modified polynomial that has $\lambda_{\max}$ as a root. Noting that $p_i(\lambda_{\max}) > 0$, we see that $\alpha_{i+1} \le \alpha_i^{\mathrm{rr}}$. Moreover, from Theorem 16 we know that the g_i's are monotonically increasing, whereby $\delta_i(\alpha_{i+1}\delta_i - \beta_i^2) > 0$. It follows that

$$0 < \delta_i(\alpha_{i+1}\delta_i - \beta_i^2) \le \delta_i(\alpha_i^{\mathrm{rr}}\delta_i - \beta_i^2),$$

and from this inequality it is clear that $g_i^{\mathrm{rr}} \le g_{i+1}$.

Lower-bound: $g_i \le g_i^{rr}$. Since $\beta_i^2 c_i^2 \ge 0$ and $\delta_i(\alpha_i^{\mathrm{rr}}\delta_i - \beta_i^2) \ge \delta_i(\alpha_{i+1}\delta_i - \beta_i^2) > 0$, we readily obtain

$$g_i \le g_i + \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i^{\mathrm{rr}}\delta_i - \beta_i^2)} = g_i^{\mathrm{rr}}.$$ □

Combining Theorem 19 with the convergence rate of relative error for Gauss quadrature (Theorem 14) immediately yields the following convergence rate for right Gauss-Radau quadrature:

Theorem 20 (Relative error of right Gauss-Radau, Theorem 5 in the main text)**.** *For each i, the right Gauss-Radau g_i^{rr} iterates satisfy*

$$\frac{g_N - g_i^{rr}}{g_N} \leq 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^i.$$

This results shows that with the same number of iterations, right Gauss-Radau gives superior approximation over Gauss quadrature, though they share the same relative error convergence rate.

Our second main result compares Gauss-Lobatto with (left) Gauss-Radau quadrature.

Theorem 21 (Theorem 6 in the main text)**.** *Let $i < N$. Then, g_i^{lr} gives better upper bounds than g_i^{lo} but worse than g_{i+1}^{lo}; more precisely,*

$$g_{i+1}^{lo} \leq g_i^{lr} \leq g_i^{lo}, \quad i < N.$$

Proof. We prove these inequalities using the recurrences for g_i^{lr} and g_i^{lo} from Algorithm 5.

$g_i^{lr} \leq g_i^{lo}$: From Algorithm 5 we observe that $\alpha_i^{\mathrm{lo}} = \lambda_{\min} + \frac{(\beta_i^{\mathrm{lo}})^2}{\delta_i^{\mathrm{lr}}}$. Thus we can write g_i^{lr} and g_i^{lo} as

$$\begin{aligned} g_i^{\mathrm{lr}} &= g_i + \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i^{\mathrm{lr}}\delta_i - \beta_i^2)} = g_i + \frac{\beta_i^2 c_i^2}{\lambda_{\min}\delta_i^2 + \beta_i^2(\delta_i^2/\delta_i^{\mathrm{lr}} - \delta_i)} \\ g_i^{\mathrm{lo}} &= g_i + \frac{(\beta_i^{\mathrm{lo}})^2 c_i^2}{\delta_i(\alpha_i^{\mathrm{lo}}\delta_i - (\beta_i^{\mathrm{lo}})^2)} = g_i + \frac{(\beta_i^{\mathrm{lo}})^2 c_i^2}{\lambda_{\min}\delta_i^2 + (\beta_i^{\mathrm{lo}})^2(\delta_i^2/\delta_i^{\mathrm{lr}} - \delta_i)} \end{aligned}$$

To compare these quantities, as before it is helpful to begin with the original three-term recursion for the Lanczos polynomial, namely

$$\beta_{i+1}p_{i+1}(\lambda) = (\lambda - \alpha_{i+1})p_i(\lambda) - \beta_i p_{i-1}(\lambda).$$

In the construction of Gauss-Lobatto, to make a new polynomial of order $i+1$ that has roots $\lambda_{\min}$ and $\lambda_{\max}$, we add $\sigma_1 p_i(\lambda)$ and $\sigma_2 p_{i-1}(\lambda)$ to the original polynomial to ensure

$$\begin{cases} \beta_{i+1}p_{i+1}(\lambda_{\min}) + \sigma_1 p_i(\lambda_{\min}) + \sigma_2 p_{i-1}(\lambda_{\min}) &= 0, \\ \beta_{i+1}p_{i+1}(\lambda_{\max}) + \sigma_1 p_i(\lambda_{\max}) + \sigma_2 p_{i-1}(\lambda_{\max}) &= 0. \end{cases}$$

Since β_{i+1}, $p_{i+1}(\lambda_{\max})$, $p_i(\lambda_{\max})$ and $p_{i-1}(\lambda_{\max})$ are all greater than 0, $\sigma_1 p_i(\lambda_{\max}) + \sigma_2 p_{i-1}(\lambda_{\max}) < 0$. To determine the sign of polynomials at $\lambda_{\min}$, consider the two cases:

1. Odd i. In this case $p_{i+1}(\lambda_{\min}) > 0$, $p_i(\lambda_{\min}) < 0$, and $p_{i-1}(\lambda_{\min}) > 0$;
2. Even i. In this case $p_{i+1}(\lambda_{\min}) < 0$, $p_i(\lambda_{\min}) > 0$, and $p_{i-1}(\lambda_{\min}) < 0$.

Thus, if $S = (\mathrm{sgn}(\sigma_1), \mathrm{sgn}(\sigma_2))$, where the signs take values in $\{0, \pm 1\}$, then $S \neq (1,1)$, $S \neq (-1,1)$ and $S \neq (0,1)$. Hence, $\sigma_2 \leq 0$ must hold, and thus $(\beta_i^{\mathrm{lo}})^2 = (\beta_i - \sigma_2)^2 \geq \beta_i^2$ given that $\beta_i^2 > 0$ for $i < N$.

Using $(\beta_i^{\mathrm{lo}})^2 \geq \beta_i^2$ with $\lambda_{\min} c_i^2(\delta_i)^2 \geq 0$, an application of monotonicity of the univariate function $g(x) = \frac{ax}{b+cx}$ for $ab \geq 0$ to the recurrences defining g_i^{lr} and g_i^{lo} yields the desired inequality $g_i^{\mathrm{lr}} \leq g_i^{\mathrm{lo}}$.

$g_{i+1}^{lo} \leq g_i^{lr}$: From recursion formulas we have

$$\begin{aligned} g_i^{\mathrm{lr}} &= g_i + \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i^{\mathrm{lr}}\delta_i - \beta_i^2)}, \\ g_{i+1}^{\mathrm{lo}} &= g_{i+1} + \frac{(\beta_{i+1}^{\mathrm{lo}})^2 c_{i+1}^2}{\delta_{i+1}(\alpha_{i+1}^{\mathrm{lo}}\delta_{i+1} - (\beta_{i+1}^{\mathrm{lo}})^2)}. \end{aligned}$$

Establishing $g_i^{\text{lr}} \geq g_{i+1}^{\text{lo}}$ thus amounts to showing that (noting the relations among g_i, g_i^{lr} and g_i^{lo}):

$$
\begin{aligned}
& \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i^{\text{lr}}\delta_i - \beta_i^2)} - \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_{i+1}\delta_i - \beta_i^2)} \quad \geq \quad \frac{(\beta_{i+1}^{\text{lo}})^2 c_{i+1}^2}{\delta_{i+1}(\alpha_{i+1}^{\text{lo}}\delta_{i+1} - (\beta_{i+1}^{\text{lo}})^2)} \\
\iff \quad & \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i^{\text{lr}}\delta_i - \beta_i^2)} - \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_{i+1}\delta_i - \beta_i^2)} \quad \geq \quad \frac{(\beta_{i+1}^{\text{lo}})^2 c_i^2 \beta_i^2}{(\delta_i)^2\delta_{i+1}(\alpha_{i+1}^{\text{lo}}\delta_{i+1} - (\beta_{i+1}^{\text{lo}})^2)} \\
\iff \quad & \frac{1}{\alpha_i^{\text{lr}}\delta_i - \beta_i^2} - \frac{1}{\alpha_{i+1}\delta_i - \beta_i^2} \quad \geq \quad \frac{(\beta_{i+1}^{\text{lo}})^2}{\delta_i\delta_{i+1}(\alpha_{i+1}^{\text{lo}}\delta_{i+1} - (\beta_{i+1}^{\text{lo}})^2)} \\
\iff \quad & \frac{1}{(\alpha_{i+1} - \delta_{i+1}^{\text{lr}}) - \beta_i^2/\delta_i} - \frac{1}{\alpha_{i+1} - \beta_i^2/\delta_i} \quad \geq \quad \frac{1}{\delta_{i+1}(\alpha_{i+1}^{\text{lo}}\delta_{i+1}/(\beta_{i+1}^{\text{lo}})^2 - 1)} \qquad \text{(Lemma 23)} \\
\iff \quad & \frac{1}{\delta_{i+1} - \delta_{i+1}^{\text{lr}}} - \frac{1}{\delta_{i+1}} \quad \geq \quad \frac{1}{\delta_{i+1}\left(\frac{\lambda_{\min}\delta_{i+1}}{(\beta_{i+1}^{\text{lo}})^2} + \frac{\delta_{i+1}}{\delta_{i+1}^{\text{lr}}} - 1\right)} \\
\iff \quad & \frac{\lambda_{\min}\delta_{i+1}}{(\beta_{i+1}^{\text{lo}})^2} + \frac{\delta_{i+1}}{\delta_{i+1}^{\text{lr}}} - 1 \geq \frac{\delta_{i+1}}{\delta_{i+1}^{\text{lr}}} - 1 \\
\iff \quad & \frac{\lambda_{\min}\delta_{i+1}}{(\beta_{i+1}^{\text{lo}})^2} \geq 0,
\end{aligned}
$$

where the last inequality is obviously true; hence the proof is complete. □

In summary, we have the following corollary for all the four quadrature rules:

Corollary 22 (Monotonicity of Lower and Upper Bounds, Corr. 7 in the main text)**.** *As the iteration proceeds, g_i and g_i^{rr} gives increasingly better asymptotic lower bounds and g_i^{lr} and g_i^{lo} gives increasingly better upper bounds, namely*

$$
\begin{aligned}
g_i \leq g_{i+1}; \quad g_i^{rr} \leq g_{i+1}^{rr} \\
g_i^{lr} \geq g_{i+1}^{lr}; \quad g_i^{lo} \geq g_{i+1}^{lo}.
\end{aligned}
$$

Proof. Directly drawn from Theorem 16, Theorem 19 and Theorem 21. □

Before proceeding further to our analysis of convergence rates of left Gauss-Radau and Gauss-Lobatto, we note two technical results that we will need.

Lemma 23. *Let α_{i+1} and α_i^{lr} be as in Alg. 1. The difference $\Delta_{i+1} = \alpha_{i+1} - \alpha_i^{lr}$ satisfies $\Delta_{i+1} = \delta_{i+1}^{lr}$.*

Proof. From the Lanczos polynomials in the definition of left Gauss-Radau quadrature we have

$$
\begin{aligned}
\beta_{i+1}p_{i+1}^*(\lambda_{\min}) &= \big(\lambda_{\min} - \alpha_i^{\text{lr}}\big)p_i(\lambda_{\min}) - \beta_i p_{i-1}(\lambda_{\min}) \\
&= \big(\lambda_{\min} - (\alpha_{i+1} - \Delta_{i+1})\big)p_i(\lambda_{\min}) - \beta_i p_{i-1}(\lambda_{\min}) \\
&= \beta_{i+1}p_{i+1}(\lambda_{\min}) + \Delta_{i+1}p_i(\lambda_{\min}) = 0.
\end{aligned}
$$

Rearrange this equation to write $\Delta_{i+1} = -\beta_{i+1}\frac{p_{i+1}(\lambda_{\min})}{p_i(\lambda_{\min})}$, which can be further rewritten as

$$
\Delta_{i+1} \overset{\text{Theorem 17}}{=} -\beta_{i+1}\frac{(-1)^{i+1}\det(J_{i+1} - \lambda_{\min}I)/\prod_{j=1}^{i+1}\beta_j}{(-1)^i\det(J_i - \lambda_{\min}I)/\prod_{j=1}^{i}\beta_j} = \frac{\det(J_{i+1} - \lambda_{\min}I)}{\det(J_i - \lambda_{\min}I)} = \delta_{i+1}^{\text{lr}}.
$$
□

Remark 24. Lemma 23 has an implication beyond its utility for the subsequent proofs: it provides a new way of calculating α_{i+1} given the quantities δ_{i+1}^{lr} and α_i^{lr}; this saves calculation in Algorithm 5.

The following lemma relates δ_i to δ_i^{lr}, which will prove useful in subsequent analysis.

Lemma 25. *Let δ_i^{lr} and δ_i be computed in the i-th iteration of Algorithm 1. Then, we have the following:*

$$
\delta_i^{lr} < \delta_i, \tag{B.2}
$$

$$
\frac{\delta_i^{lr}}{\delta_i} \leq 1 - \frac{\lambda_{\min}}{\lambda_N}. \tag{B.3}
$$

Proof. We prove (B.2) by induction. Since $\lambda_{\min} > 0$, $\delta_1 = \alpha_1 > \lambda_{\min}$ and $\delta_1^{\text{lr}} = \alpha - \lambda_{\min}$ we know that $\delta_1^{\text{lr}} < \delta_1$. Assume that $\delta_i^{\text{lr}} < \delta_i$ is true for all $i \leq k$ and considering the $(k+1)$-th iteration:

$$\delta_{k+1}^{\text{lr}} = \alpha_{k+1} - \lambda_{\min} - \frac{\beta_k^2}{\delta_k^{\text{lr}}} < \alpha_{k+1} - \frac{\beta_k^2}{\delta_k} = \delta_{k+1}.$$

To prove (B.3), simply observe the following

$$\frac{\delta_i^{\text{lr}}}{\delta_i} = \frac{\alpha_i - \lambda_{\min} - \beta_{i-1}^2/\delta_{i-1}^{\text{lr}}}{\alpha_i - \beta_{i-1}^2/\delta^{i-1}} \overset{\text{(B.2)}}{\leq} \frac{\alpha_i - \lambda_{\min}}{\alpha_i} \leq 1 - \frac{\lambda_{\min}}{\lambda_N}.$$ □

With aforementioned lemmas we will be able to show how fast the difference between g_i^{lr} and g_i decays. Note that g_i^{lr} gives an upper bound on the objective while g_i gives a lower bound.

Lemma 26. *The difference between g_i^{lr} and g_i decreases linearly. More specifically we have*

$$g_i^{lr} - g_i \leq 2\kappa^+ (\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^i g_N$$

where $\kappa^+ = \lambda_N/\lambda_{\min}$ and κ is the condition number of A, i.e., $\kappa = \lambda_N/\lambda_1$.

Proof. We rewrite the difference $g_i^{\text{lr}} - g_i$ as follows

$$\begin{aligned}
g_i^{\text{lr}} - g_i &= \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i^{\text{lr}}\delta_i - \beta_i^2)} \\
&= \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_{i+1}\delta_i - \beta_i^2)} \frac{\delta_i(\alpha_{i+1}\delta_i - \beta_i^2)}{\delta_i(\alpha_i^{\text{lr}}\delta_i - \beta_i^2)} \\
&= \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_{i+1}\delta_i - \beta_i^2)} \frac{1}{\left(\alpha_i^{\text{lr}} - \beta_i^2/\delta_i\right)/\left(\alpha_{i+1} - \beta_i^2/\delta_i\right)} \\
&= \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i\delta_i - \beta_i^2)} \frac{1}{1 - \Delta_{i+1}/\delta_{i+1}},
\end{aligned}$$

where $\Delta_{i+1} = \alpha_{i+1} - \alpha_i^{\text{lr}}$. Next, recall that $\frac{g_N - g_i}{g_N} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i$. Since g_i lower bounds g_N, we have

$$\begin{aligned}
\left(1 - 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\right) g_N \leq g_i \leq g_N, \\
\left(1 - 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{i+1}\right) g_N \leq g_{i+1} \leq g_N.
\end{aligned}$$

Thus, we can conclude that

$$\frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i\delta_i - \beta_i^2)} = g_{i+1} - g_i \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i g_N.$$

Now we focus on the term $\left(1 - \Delta_{i+1}/\delta_{i+1}\right)^{-1}$. Using Lemma 23 we know that $\Delta_{i+1} = \delta_{i+1}^{\text{lr}}$. Hence,

$$\begin{aligned}
1 - \Delta_{i+1}/\delta_{i+1} &= 1 - \delta_{i+1}^{\text{lr}}/\delta_{i+1} \\
&\geq 1 - (1 - \lambda_{\min}/\lambda_N) = \lambda_{\min}/\lambda_N \triangleq \frac{1}{\kappa^+}.
\end{aligned}$$

Finally we have

$$g_i^{\text{lr}} - g_i = \frac{\beta_i^2 c_i^2}{\delta_i(\alpha_i\delta_i - \beta_i^2)} \frac{1}{1 - \Delta_{i+1}/\delta_{i+1}} \leq 2\kappa^+ \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i g_N.$$ □

Theorem 27 (Relative error of left Gauss-Radau, Theorem 8 in the main text)**.** *For left Gauss-Radau quadrature where the preassigned node is $\lambda_{\min}$, we have the following bound on relative error:*

$$\frac{g_i^{lr} - g_N}{g_N} \le 2\kappa^+ \Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^i,$$

where $\kappa^+ := \lambda_N/\lambda_{\min},\ i < N$.

Proof. Write $g_i^{\mathrm{lr}} = g_i + (g_i^{\mathrm{lr}} - g_i)$. Since $g_i \le g_N$, using Lemma 26 to bound the second term we obtain

$$g_i^{\mathrm{lr}} \le g_N + 2\kappa^+ \Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^i g_N,$$

from which the claim follows upon rearrangement. □

Due to the relations between left Gauss-Radau and Gauss-Lobatto, we have the following corollary:

Corollary 28 (Relative error of Gauss-Lobatto, Corr. 9 in the main text)**.** *For Gauss-Lobatto quadrature, we have the following bound on relative error:*

$$\frac{g_i^{lo} - g_N}{g_N} \le 2\kappa^+ \Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^{i-1}, \tag{B.4}$$

where $\kappa^+ := \lambda_N/\lambda_{\min}$ and $i < N$.

C. Generalization: Symmetric Matrices

In this section we consider the case where u lies in the column space of several top eigenvectors of A, and discuss how the aforementioned theorems vary. In particular, note that the previous analysis assumes that A is positive definite. With our analysis in this section we relax this assumption to the more general case where A is symmetric with simple eigenvalues, though we require u to lie in the space spanned by eigenvectors of A corresponding to positive eigenvalues.

We consider the case where A is symmetric and has the eigendecomposition of $A = Q\Lambda Q^\top = \sum_{i=1}^N \lambda_i q_i q_i^\top$ where λ_i's are eigenvalues of A increasing with i and q_i's are corresponding eigenvectors. Assume that u lies in the column space spanned by top k eigenvectors of A where all these k eigenvectors correspond to positive eigenvalues. Namely we have $u \in \mathrm{Span}\{\{q_i\}_{i=N-k+1}^N\}$ and $0 < \lambda_{N-k+1}$.

Since we only assume that A is symmetric, it is possible that A is singular and thus we consider the value of $u^\top A^\dagger u$, where $A^\dagger$ is the pseudo-inverse of A. Due to the constraints on u we have

$$u^\top A^\dagger u = u^\top Q \Lambda^\dagger Q^\top u = u^\top Q_k \Lambda_k^\dagger Q_k^\top u = u^\top B^\dagger u,$$

where $B = \sum_{i=N-k+1}^N \lambda_i q_i q_i^\top$. Namely, if u lies in the column space spanned by the top k eigenvectors of A then it is equivalent to substitute A with B, which is the truncated version of A at top k eigenvalues and corresponding eigenvectors.

Another key observation is that, given that u lies only in the space spanned by $\{q_i\}_{i=N-k+1}^N$, the Krylov space starting at u becomes

$$\mathrm{Span}\{u, Au, A^2u, \dots\} = \mathrm{Span}\{u, Bu, B^2u, \dots, B^{k-1}u\} \tag{C.1}$$

This indicates that Lanczos iteration starting at matrix A and vector u will finish constructing the corresponding Krylov space after the k-th iteration. Thus under this condition, Algorithm 1 will run at most k iterations and then stop. At that time, the eigenvalues of J_k are exactly the eigenvalues of B, thus they are exactly $\{\lambda_i\}_{i=N-k+1}^N$ of A. Using similar proof as in Lemma 15, we can obtain the following generalized exactness result.

Corollary 29 (Generalized Exactness)**.** *g_k, g_k^{rr} and g_k^{lr} are exact for $u^\top A^\dagger u = u^\top B^\dagger u$, namely*

$$g_k = g_k^{rr} = g_k^{lr} = u^\top A^\dagger u = u^\top B^\dagger u.$$

The monotonicity and the relations between bounds given by various Gauss-type quadratures will still be the same as in the original case in Section 4, but the original convergence rate cannot apply in this case because now we probably have $\lambda_{\min}(B) = 0$, making κ undefined. This crash of convergence rate results from the crash of the convergence of the corresponding conjugate gradient algorithm for solving $Ax = u$. However, by looking at the proof of, e.g., (Shewchuk, 1994), and by noting that $\lambda_1(B) = \ldots = \lambda_{N-k}(B) = 0$, with a slight modification of the proof we actually obtain the bound

$$\|\varepsilon^i\|_A^2 \leq \min_{P_i} \max_{\lambda \in \{\lambda_i\}_{i=N-k+1}^N} [P_i(\lambda)]^2 \|\varepsilon^0\|_A^2,$$

where P_i is a polynomial of order i. By using properties of Chebyshev polynomials and following the original proof (e.g., (Golub & Meurant, 2009) or (Shewchuk, 1994)) we obtain the following lemma for conjugate gradient.

Lemma 30. *Let ε^k be as before (for conjugate gradient). Then,*

$$\|\varepsilon^k\|_A \leq 2\Big(\frac{\sqrt{\kappa'}-1}{\sqrt{\kappa'}+1}\Big)^k \|\varepsilon_0\|_A, \quad \textit{where } \kappa' := \lambda_N/\lambda_{N-k+1}.$$

Following this new convergence rate and connections between conjugate gradient, Lanczos iterations and Gauss quadrature mentioned in Section 4, we have the following convergence bounds.

Corollary 31 (Convergence Rate for Special Case)**.** *Under the above assumptions on A and u, due to the connection Between Gauss quadrature, Lanczos algorithm and Conjugate Gradient, the relative convergence rates of g_i, g_i^{rr}, g_i^{lr} and g_i^{lo} are given by*

$$\begin{aligned}
\frac{g_k - g_i}{g_k} &\leq 2\Big(\frac{\sqrt{\kappa'}-1}{\sqrt{\kappa'}+1}\Big)^i \\
\frac{g_k - g_i^{rr}}{g_k} &\leq 2\Big(\frac{\sqrt{\kappa'}-1}{\sqrt{\kappa'}+1}\Big)^i \\
\frac{g_i^{lr} - g_k}{g_k} &\leq 2\kappa'_m\Big(\frac{\sqrt{\kappa'}-1}{\sqrt{\kappa'}+1}\Big)^i \\
\frac{g_i^{lo} - g_k}{g_k} &\leq 2\kappa'_m\Big(\frac{\sqrt{\kappa'}-1}{\sqrt{\kappa'}+1}\Big)^i,
\end{aligned}$$

where $\kappa'_m = \lambda_N/\lambda'_{\min}$ and $0 < \lambda'_{\min} < \lambda_{N-k+1}$ is a lowerbound for nonzero eigenvalues of B.

D. Accelerating MCMC for k-DPP

We present details of a *Retrospective Markov Chain Monte Carlo (MCMC)* in Algorithm 6 and Algorithm 7 that samples for efficiently drawing samples from a k-DPP, by accelerating it using our results on Gauss-type quadratures.

Algorithm 6 Gauss-kDPP (L, k)

Input: L the kernel matrix we want to sample DPP from, k the size of subset and $\mathcal{Y} = [N]$ the ground set
Output: Y sampled from exact kDPP (L) where $|Y| = k$
Randomly Initialize $Y \subseteq \mathcal{Y}$ where $|Y| = k$
while not mixed **do**
  Pick $v \in Y$ and $u \in \mathcal{Y}\backslash Y$ uniformly randomly
  Pick $p \in (0, 1)$ uniformly randomly
  $Y' = Y\backslash\{v\}$
  Get lower and upper bounds $\lambda_{\min}, \lambda_{\max}$ of the spectrum of $L_{Y'}$
  if k-DPP-JudgeGauss($pL_{v,v} - L_{u,u}, p, L_{Y',u}, L_{Y',v}, \lambda_{\min}, \lambda_{\max}$) = *True* **then**
    $Y = Y' \cup \{u\}$
  end if
end while

Algorithm 7 kDPP-JudgeGauss($t, p, u, v, A, \lambda_{\min}, \lambda_{\max}$)

Input: t the target value, p the scaling factor, u, v and A the corresponding vectors and matrix, $\lambda_{\min}$ and $\lambda_{\max}$ lower and upper bounds for the spectrum of A
Output: Return *True* if $t < p(v^\top A^{-1} v) - u^\top A^{-1} u$, *False* if otherwise
$u_{-1} = 0$, $u_0 = u/\|u\|$, $i^u = 1$, $\beta_0^u = 0$, $d^u = \infty$
$v_{-1} = 0$, $v_0 = v/\|v\|$, $i^v = 1$, $\beta_0^v = 0$, $d^v = \infty$
while *True* **do**
  if $d^u > pd^v$ **then**
    Run one more iteration of Gauss-Radau on $u^\top A^{-1} u$ to get tighter $(g^{\mathrm{lr}})^u$ and $(g^{\mathrm{rr}})^u$
    $d^u = (g^{\mathrm{lr}})^u - (g^{\mathrm{rr}})^u$
  else
    Run one more iteration of Gauss-Radau on $v^\top A^{-1} v$ to get tighter $(g^{\mathrm{lr}})^v$ and $(g^{\mathrm{rr}})^v$
    $d^v = (g^{\mathrm{lr}})^v - (g^{\mathrm{rr}})^v$
  end if
  if $t < p\|v\|^2 (g^{\mathrm{rr}})^v - \|u\|^2 (g^{\mathrm{lr}})^u$ **then**
    Return *True*
  else if $t \geq p\|v\|^2 (g^{\mathrm{lr}})^v - \|u\|^2 (g^{\mathrm{rr}})^u$ **then**
    Return *False*
  end if
end while

E. Accelerating Stochastic Double Greedy

We present details of *Retrospective Stochastic Double Greedy* in Algorithm 8 and Algorithm 9 that efficiently select a subset $Y \in \mathcal{Y}$ that approximately maximize $\log\det(L_Y)$.

Algorithm 8 Gauss-DG (L)

Input: L the kernel matrix and $\mathcal{Y} = [N]$ the ground set
Output: $X \in \mathcal{Y}$ that approximately maximize $\log\det(L_Y)$
$X_0 = \emptyset$, $Y_0 = \mathcal{Y}$
for $i = 1, 2, \ldots, N$ **do**
  $Y_i' = Y_{i-1} \backslash \{i\}$
  Sample $p \in (0, 1)$ uniformly randomly
  Get lower and upper bounds $\lambda_{\min}^-, \lambda_{\max}^-, \lambda_{\min}^+, \lambda_{\max}^+$ of the spectrum of $L_{X_{i-1}}$ and $L_{Y_i'}$ respectively
  if DG-JudgeGauss($L_{X_{i-1}}, L_{Y_i'}, L_{X_{i-1},i}, L_{Y_i',i}, L_{i,i}, p, \lambda_{\min}^-, \lambda_{\max}^-, \lambda_{\min}^+, \lambda_{\max}^+$) = *True* **then**
    $X_i = X_{i-1} \cup \{i\}$
  else
    $Y_i = Y_i'$
  end if
end for

Algorithm 9 DG-JudgeGauss($A, B, u, v, t, p, \lambda_{\min}^{A}, \lambda_{\max}^{A}, \lambda_{\min}^{B}, \lambda_{\max}^{B}$)

Input: t the target value, p the scaling factor, u, v, A and B the corresponding vectors and matrix, $\lambda_{\min}^{A}$, $\lambda_{\max}^{A}$, $\lambda_{\min}^{B}$, $\lambda_{\max}^{B}$ lower and upper bounds for the spectrum of A and B
Output: Return *True* if $p|\log(t - u^\top A^{-1}u)|_+ \leq (1-p)| - \log(t - v^\top B^{-1}v)|_+$, *False* if otherwise
$d^u = \infty$, $d^v = \infty$
while *True* **do**
  if $pd^u > (1-p)d^v$ **then**
    Run one more iteration of Gauss-Radau on $u^\top A^{-1}u$ to get tighter lower and upper bounds l^u, u^u for $|\log(t - u^\top A^{-1}u)|_+$
    $d^u = u^u - l^u$
  else
    Run one more iteration of Gauss-Radau on $v^\top B^{-1}v$ to get tighter lower and upper bounds l^v, u^v for $|\log(t - v^\top B^{-1}v)|_+$
    $d^v = u^v - l^v$
  end if
  if $pu^u \leq (1-p)l^v$ **then**
    Return *True*
  else if $pl^u > (1-p)u^v$ **then**
    Return *False*
  end if
end while