Finding selected eigenvalues of the lattice Dirac operator using the Lánczos algorithm with selective orthogonalization

A D Kennedy

SUPA & School of Physics & Astronomy
University of Edinburgh

Cray-TIFR Workshop on High-Performance Computing in Physics, 2011

Introduction

- We want to find some of the eigenvalues and eigenvectors of large sparse Hermitian (symmetric) matrices.
- We have implemented the **LANSO** (Lánczos algorithm with selective orthogonalization) of Parlett and Scott[1] in the **Chroma** software system.[2]
- We shall discuss:
  - Why we are interested in the problem;
  - How to find eigenpairs of small matrices;
  - How eigenpairs are related to those of submatrices (Ritz pairs);
  - Krylov spaces and the Lánczos method;
  - Diseases and benefits of using finite-precision floating-point arithmetic;
  - Bounds on convergence rate, especially for inner eigenvalues near to voids in the spectrum;
  - Implementation issues for large parallel architectures.
- This work was done in collaboration with Chris Johnson (EPCC, University of Edinburgh) under the NAIS project (`www.nais.org.uk`).

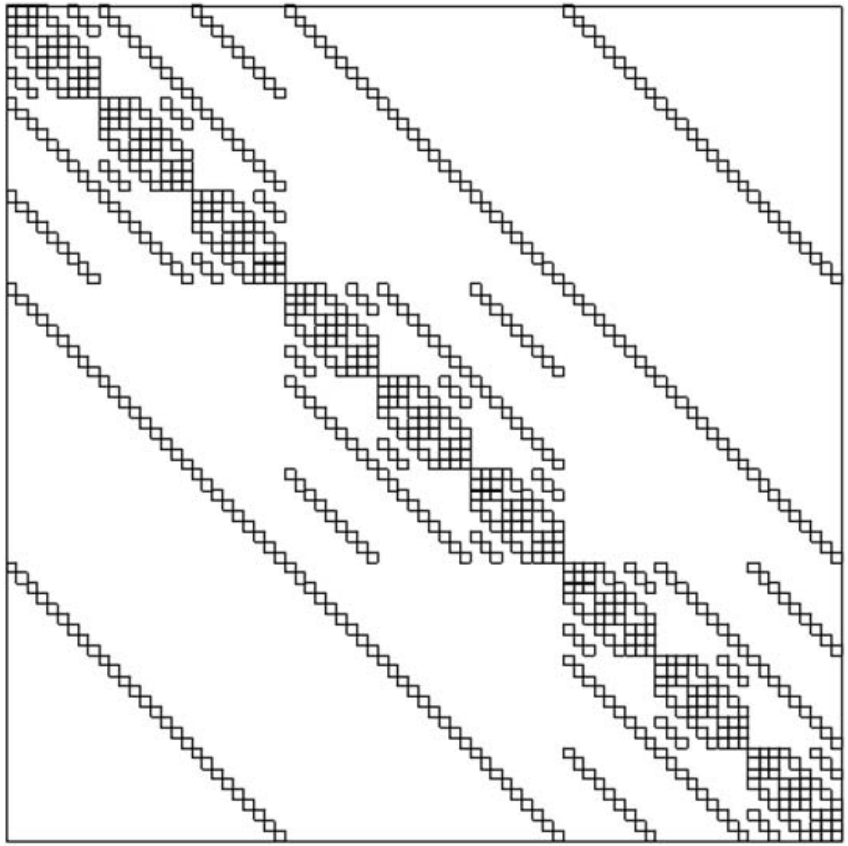
[1] B. N. Parlett and D. S. Scott, Mathematics of Computation, **33** 145, 217–238 (1979)
[2] Robert G. Edwards and Bálint Joó, Nucl. Phys. Proc. Suppl. **140** 832 (2005)

Application — Chiral Lattice Quarks

- The problem is common to many areas of computational science, but the particular application we are interested in is in computing the Neuberger operator for lattice QCD (Quantum Chromodynamics being the quantum field theory of the strong nuclear force).
- This requires us to evaluate the **sgn** function of the "Hermitian Dirac operator" $\gamma_5 D$, which is defined by diagonalizing this matrix and taking the **sgn** (± 1) of each of its eigenvalues.
- It is far too expensive to carry out the full diagonalization, so we use a Zolotarev rational approximation for the **sgn** function as this can be evaluated just using matrix addition, multiplication, and inversion (using a multi-shift solver for its stable partial fraction expansion).
- The approximation is expensive for very small eigenvalues of $\gamma_5 D$, and as there are only a relatively small number of these we want to deflate (project them out) and take their sign explicitly (which is very easy to do).

Dirac Operator

- The lattice Wilson–Dirac operator is large and sparse, the sparsity corresponding to it being the discretization of a local differential operator on space-time.
- Our sample operator acts on vectors with $24^3 \times 48 \times 3 \times 4 = 7,962,624$ complex components, which is moderate by today's standards.
- Despite its size it is fairly well-conditioned, basically because it is an approximation to a well-behaved linear operator on a infinite dimensional Hilbert space.



Structure of the Wilson–Dirac operator; *S. M. Pickles, University of Edinburgh Ph.D. thesis (1998)*

Basic Properties of Symmetric Matrices

- A matrix A is Hermitian (with respect to a sesquilinear inner product) if $A = A^\dagger$, which means

$$(u, Av) = (A^\dagger u, v) = (Au, v) = (v, Au)^*,$$

or equivalently

$$u^\dagger \cdot Av = (A^\dagger u)^\dagger \cdot v = (Au)^\dagger \cdot v = (v^\dagger \cdot Au)^*.$$

- An eigenvalue λ of A satisfies $Az = \lambda z$ where $z \neq 0$ is the corresponding eigenvector.
- The eigenvalues are real

$$\lambda = \frac{(z, Az)}{(z, z)} = \frac{(z, A^\dagger z)}{(z, z)} = \frac{(z, Az)^*}{(z, z)^*} = \lambda^*.$$

Basic Properties of Symmetric Matrices

- The eigenvectors belonging to different eigenvalues are orthogonal

$$\lambda(z', z) = (z', Az) = (Az, z')^* = (z, Az')^* = \lambda'^*(z, z')^* = \lambda'(z', z),$$

  hence $(\lambda' - \lambda)(z', z) = 0$, so $\lambda \neq \lambda' \Rightarrow (z', z) = 0$.
- We can choose eigenvectors belonging to the same eigenvalue to be orthogonal, for example by using the Gram–Schmidt procedure, as any linear combination of such eigenvectors is also an eigenvector.
- The set of eigenvectors is a complete basis for the linear space. It follows any matrix can be reduced to triangular form T by a unitary (orthogonal) transformation[3] (change of basis), $A = UTU^{-1} = UTU^\dagger$, and

$$T^\dagger = (U^\dagger AU)^\dagger = U^\dagger A^\dagger U = U^\dagger AU = T.$$

  The columns of U furnish the orthonormal eigenvectors.

[3] This is "Schur normal form," which follows from the Cayley–Hamilton theorem that every matrix satisfies its characteristic equation, and the fundamental theorem of algebra which states that the characteristic polynomial $p(\lambda) = \det(A - \lambda)$ has exactly $N = \dim(A)$ complex roots, counting multiplicity.

Power Method

- How do we find eigenvalues and eigenvectors numerically? One obvious approach is the **Power Method**.
- Start with an arbitrary vector which can, in theory, be expanded in the orthonormal eigenvector basis $\{z_j\}$, $u_0 = \sum_j z_j (z_j, u_0)$.
- Apply A to u_0 and normalize the result to get u_1, and so forth: $u_{k+1} = Au_k / \|Au_k\|$, where the norm is $\|x\| = \sqrt{(x,x)}$.
- We then find that $u_k \propto \sum_j A^k z_j (z_j, u_0) = \sum_j \lambda_j^k z_j (z_j, u_0) \to \lambda_1^k z_1 (z_1, u_0) \propto z_1$, where we label the eigenpairs such that $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_N|$.
- If the eigenvalue λ_1 is degenerate then u_k converges to the eigenvector parallel to u_0.
- The rate of convergence is governed by $|\lambda_2/\lambda_1|^k = e^{-k(\ln|\lambda_1| - \ln|\lambda_2|)}$.
- If we shift the matrix A by a constant then we just shift its eigenvalues by the same constant and leave the eigenvectors unchanged. However such a shift *does* change the rate of convergence of the power method.

Inverse Iteration

- Another fairly obvious method is to use **Inverse Iteration**.
- This is similar to the power method, except that we apply the inverse of A at each step, $u_{k+1} = A^{-1}u_k / \|A^{-1}u_k\|$.
- Clearly, this process converges to an eigenvector belonging to λ_N, the eigenvalue of smallest absolute value.
- The advantage of inverse iteration is that we may use a shift close to $-\lambda_N$ leading to an almost infinite convergence rate. With a good choice of shift only a few iterations are required.
- The disadvantage is that it is expensive to apply A^{-1}. If A is small then we can solve the linear equations $Av = u_k$ by factoring A into triangular factors $A = LU$ (Gaussian elimination) or into a product of unitary (orthogonal) and triangular factors $A = QR$.
- Why an upper triangular matrix is called U (upper) in one case and R (right) in the other is one of the unfathomable mysteries of numerical analysis. There is also a QL variant, but who knows whether L stands for left or lower?

Givens Rotations

- It is useful to understand how such a **QR** decomposition is effected.
- We eliminate the non-zero subdiagonal elements of A successively using 2×2 unitary (orthogonal) transformations called **Givens rotations**.
- We may eliminate the elements in the order

$$\begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ 3 & \bullet & \bullet & \bullet \\ 2 & 5 & \bullet & \bullet \\ 1 & 4 & 6 & \bullet \end{pmatrix}$$

where at the fifth step, for example, we have

$$Q_5^\dagger A_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c^* & s^* & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ 0 & 5 & \bullet & \bullet \\ 0 & 0 & 6 & \bullet \end{pmatrix} = \begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \\ 0 & 0 & 6 & \bullet \end{pmatrix} = A_6,$$

where $c \cdot (A_5)_{32} = s \cdot (A_5)_{22}$ and $|c|^2 + |s|^2 = 1$. The notation just indicates which elements are non-zero, the actual values may change at each step.

QR Decomposition

- We obtain

$$R = A_7 = Q_6^\dagger A_6 = Q_6^\dagger Q_5^\dagger A_5 = \cdots = Q_6^\dagger \cdots Q_1^\dagger A_1 = Q^\dagger A$$

  where the matrix $Q = Q_1 \cdots Q_6$ is unitary (orthogonal).
- We can now solve $Ax = QRx = y$ as $Rx = Q^\dagger y$, which is easily done by starting from the bottom row and working upwards (back substitution).
- This method is particularly efficient if A is of **Hessenberg** form

$$\begin{pmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{pmatrix},$$

  where it only requires $N-1$ steps rather than $\frac{1}{2}N(N-1)$.

QR Algorithm

- Numerical matrix diagonalization, unlike matrix inversion, has to be an iterative process. If the initial matrix elements are rational the eigenvalues, being roots of the characteristic polynomial, will in general be algebraic numbers and cannot be computed exactly in a finite number of rational operations.
- The preferred method for diagonalizing small matrices is the **QR Algorithm**, which iterates the basic step

$$A_{k+1} = R_k Q_k \qquad \text{where} \qquad A_k = Q_k R_k;$$

in other words

$$A_{k+1} = {Q_k}^\dagger A_k Q_k = \bar{Q}_k^\dagger A \bar{Q}_k$$

where $\bar{Q}_k = Q_1 \cdots Q_k$.

- This process converges to A_∞ which is diagonal.

QR Algorithm

- To understand why this works note that $A_{k+1} = \bar{Q}_k^\dagger A \bar{Q}_k$ means that

$$\bar{Q}_k A_{k+1} = A \bar{Q}_k;$$

hence

$$\bar{Q}_{k+1} R_{k+1} = \bar{Q}_k Q_{k+1} R_{k+1} = \bar{Q}_k A_{k+1} = A \bar{Q}_k.$$

- The leftmost column of this equation is[4]

$$\bar{Q}_{k+1} R_{k+1} e_1 = (R_{k+1})_{11} \bar{Q}_{k+1} e_1 = A \bar{Q}_k e_1$$

whose norm tells us that $(R_{k+1})_{11} = \|A \bar{Q}_k e_1\|$.

- In other words the sequence of vectors $u_k = \bar{Q}_k e_1$ are just the iterates of the power method, so $(R_k)_{11} \to \lambda_1$.

[4] e_k is a unit vector in direction k, i.e., $(e_k)_j = \delta_{jk}$.

QR Algorithm

- Likewise, the rightmost column corresponds to inverse iteration.
- The conjugate of $\bar{Q}_{k+1} R_{k+1} = A \bar{Q}_k$ is $R^\dagger_{k+1} \bar{Q}^\dagger_{k+1} = \bar{Q}^\dagger_k A$, and hence

$$\bar{Q}_k R^\dagger_{k+1} = A \bar{Q}_{k+1}.$$

- Noting that $R^\dagger_{k+1}$ is a lower or left triangular matrix, we see that the rightmost column of this equation is

$$\bar{Q}_k R^\dagger_{k+1} e_N = (R^\dagger_{k+1})_{NN} \bar{Q}_k e_N = A \bar{Q}_{k+1} e_N$$

or equivalently $A^{-1} u_k = u_{k+1} / (R^\dagger_{k+1})_{NN}$ where here $u_k = \bar{Q}_k e_N$ is an iterate of the inverse iteration algorithm with initial vector e_N, and $(R^\dagger_{k+1})_{NN} = 1 / \| A^{-1} u_k \| \to \lambda_N \in \mathbb{R}$.

QR Shifts

- We can apply the QR algorithm to $A - \lambda$ for any shift λ, and if this shift is close to an eigenvalue of A then $A - \lambda$ is almost singular and the last column will converge to a corresponding eigenvector very rapidly.
- Indeed, we can adjust the shift at every iteration; this is known as a *non-stationary* iteration. With a good choice of shifts we usually only need a few iterations to obtain each eigenvalue to machine precision.
- If we have prior knowledge of some eigenvalues then we can make good use of this information.
- Once the Hermitian matrix has been reduced to Hessenberg (and hence **tridiagonal** form) the QR iteration preserves this structure. The basic step can then be optimized by a simple process known as "bump chasing".

Rayleigh Quotient

- We now consider how the eigenvalues and eigenvectors of an approximation to a matrix are related to the actual eigenpairs. This is crucial for application to large sparse matrices where direct methods such as the QR algorithm are too expensive to contemplate.
- Consider the **Rayleigh quotient** $\rho(u,A) = (u,Au)/(u,u)$ which has the spectral representation

$$\rho(u,A) = \frac{(u,Au)}{(u,u)} = \sum_j \lambda_j \frac{|(z_j,u)|^2}{\sum_k |(z_k,u)|^2}.$$

- This is a weighted mean of eigenvalues, so it satisfies the inequalities

$$\lambda_i \leq \min_{\substack{u\in\mathrm{span}(z_i,\ldots,z_{i+j})\\ u\neq 0}} \rho(u,A) = \min_{\substack{u\in\mathrm{span}(z_i,\ldots,z_{i+j})\\ \|u\|=1}} (u,Au) \leq \lambda_{i+j}$$

where we now have ordered the eigenvalues such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$.

Ritz Pairs

- The extrema of the Rayleigh quotient may be found by varying the components $u_i = (z_i, u)$ of u subject to the constraint that $\|u\| = 1$. Introducing a Lagrange multiplier μ we find $(A - \mu)u = 0$, whose solutions are the eigenpairs $\mu = \lambda_j$ with $u = z_j$ for $j = 1, \ldots, N$.
- If we restrict u to an m-dimensional subspace $S_m \subseteq \mathbb{R}^N$ then the extrema of the Rayleigh quotient are the **Ritz pairs** $\mu = \theta_j$ with $u = y_j$ for $j = 1, \ldots, m$.
- The Ritz pairs are the eigenvalues and eigenvectors of the restriction H of A to the subspace, $H = Q^\dagger A Q$, that is $Hs_j = \theta_j s_j$. Here Q is an $N \times m$ matrix with orthonormal columns ($Q^\dagger Q = 1$) and $QQ^\dagger$ is an orthogonal projector onto S_m.
- Note that the vectors $y_j = Qs_j \in \mathbb{R}^N$ are not eigenvectors of A, although $Q^\dagger(A - \theta_j)y_j = 0$, since the **residual** $R = AQ - QH \neq 0$ in general.
- For large matrices our strategy is to compute the Ritz pairs of A for suitably chosen subspaces S_m; so we are interested in how well these may approximate the actual eigenvalues of A itself.

MinMax and MaxMin Bounds

- Let $S_j \subseteq \mathbb{R}^N$ be an arbitrary subspace of dimension j, and $C_{j-1} \subseteq \mathbb{R}^N$ another arbitrary subspace of codimension $j-1$, i.e., a subspace of dimension $N-j+1$ (if $N < \infty$).
- There must be a non-zero vector $v \in S_j \cap C_{j-1}$. v depends upon S_j and C_{j-1}, of course. Hence $\min_{u \in S_j} \rho(u,A) \leq \rho(v,A) \leq \max_{u \in C_{j-1}} \rho(u,A)$.
- Taking the maximum over all S_j and the minimum over all C_{j-1} we obtain $\max_{S_j \subseteq \mathbb{R}^N} \min_{u \in S_j} \rho(u,A) \leq \min_{C_{j-1} \subseteq \mathbb{R}^N} \max_{u \in C_{j-1}} \rho(u,A)$.
- For the particular subspace $S_j = \mathrm{span}(z_1, \ldots, z_j)$ we know that the minimum of the Rayleigh quotient is λ_j, and likewise for $C_{j-1} = \mathrm{span}(z_j, \ldots, z_N)$ its maximum is also λ_j, so

$$\lambda_j \leq \max_{S_j \subseteq \mathbb{R}^N} \min_{u \in S_j} \rho(u,A) \leq \min_{C_{j-1} \subseteq \mathbb{R}^N} \max_{u \in C_{j-1}} \rho(u,A) \leq \lambda_j.$$

- We have thus established the **MaxMin** and **MinMax** bounds

$$\max_{S_j \subseteq \mathbb{R}^N} \min_{u \in S_j} \rho(u,A) = \lambda_j = \min_{C_{j-1} \subseteq \mathbb{R}^N} \max_{u \in C_{j-1}} \rho(u,A).$$

Spectral Stability

- From these bounds we see that for any subspace S_m the Ritz values satisfy $\lambda_j \geq \theta_j \geq \lambda_{j+N-m}$ for $j = 1, \ldots, m$.
  - For the first bound note that $\theta_j = \max_{S_j \subseteq S_m} \min_{u \in S_j} \rho(u, A)$.
  - For the second replace A by $-A$ to get $\lambda_{N-j'} \geq \theta_{m-j'}$, and set $j' = m - j$.
- More precisely, one can show that there are m of A's eigenvalues $\lambda_{j'}$ such that $|\lambda_{j'} - \theta_j| \leq \|R\|$ for $j = 1, \ldots, m$.
- Similarly, we can also show that the eigenvalues are stable against small or low-rank perturbations of the matrix A; this is crucial because
  - The spectrum of a good discrete approximation to a continuous operator is close to the spectrum of the underlying operator itself.
  - We can often extract reliable estimates for the spectrum of A from the spectrum of the restriction of A to relatively small subspaces, in which we can use dense matrix methods such as the QR algorithm discussed before.
- Such nice behaviour does not always hold for the eigenvectors.
  - This should not be suprising: consider small perturbations of a matrix with a degenerate eigenvalue; different perturbations can lift the degeneracy in completely different directions.
  - On the other hand, the eigenspaces corresponding to well-separated eigenvalues are stable.

Krylov Spaces

- We want to consider a sequence of subspaces such that the restriction of A to them converges to A.
  - Here we are assuming that there is some underlying "continuum" operator in an infinite dimensional Hilbert space, and that we can use the topology on this space to define what we mean by convergence.
- In practice we do not have an explicit matrix representation of the large (sparse) matrix A, but we merely have some functional "black box" representation that allows us to apply it to a vector in $\mathbb{R}^N$.
- Almost the only spaces we can construct from this are the **Krylov spaces** $\mathcal{K}_n(A,u) = \text{span}(u, Au, A^2u, \ldots, A^{n-1}u)$ where u is some more-or-less arbitrary starting vector.
- The only simple generalization are **block Krylov spaces** where we start from more than one vector.

Arnoldi Method

- The vectors $\{A^j u\}$ do not form an orthonormal basis for the Krylov space.
- Furthermore, the corresponding unit vectors $A^j u / \|A^j u\|$ converge to the largest eigenvector of A, as they are just successive iterates of the power method. They therefore provide a particularly *bad* choice of basis for numerical computations.
- It is natural to construct a good orthonormal basis by deflation and normalization,

$$q_1 = u/\|u\|, \qquad u_{j+1} = Aq_j - \sum_{k=1}^{j} q_k (q_k, Aq_j), \qquad q_{j+1} = \frac{u_{j+1}}{\|u_{j+1}\|};$$

  in other words the Gram–Schmidt procedure. This is called the **Arnoldi method**. We see immediately that $(q_{j+1}, Aq_j) = (q_{j+1}, u_{j+1}) = \|u_{j+1}\|$.
- The $n \times n$ matrix Q whose columns are $Qe_j = q_j$ therefore furnishes an orthogonal projector onto $\mathcal{K}_n(A, u)$.

Arnoldi Method

- The restriction of A to the Krylov space is Hessenberg:

$$H = Q^\dagger A Q = \begin{pmatrix} H_{1,1} & H_{1,2} & & H_{1,n-2} & H_{1,n-1} & H_{1,n} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,n-2} & H_{2,n-1} & H_{2,n} \\ 0 & H_{3,2} & & H_{3,n-2} & H_{3,n-1} & H_{3,n} \\ \vdots & & \ddots & & \vdots & \\ 0 & 0 & & H_{n-1,n-2} & H_{n-1,n-1} & H_{n-1,n} \\ 0 & 0 & \cdots & 0 & H_{n,n-1} & H_{n,n} \end{pmatrix}.$$

- We can diagonalize this matrix using the QR algorithm to obtain $\Theta = Y^\dagger H Y$, where Θ is the diagonal matrix of Ritz values, $\Theta_{ij} = \theta_j \delta_{ij}$, and Y the $n \times n$ unitary (orthogonal) matrix whose columns are the corresponding Ritz vectors, $y_j = Y e_j$.
- We may hope that some of the Ritz values approximate the eigenvalues of A, $\theta_j \approx \lambda_{j'}$, and that some of the Ritz vectors approximate its eigenvectors, $QYe_j = Qy_j \approx z_{j'}$, if the residual is small, since $A(QY) = (QH + R)Y = (QY)\Theta + \mathcal{O}(\|R\|)$.

Lánczos Algorithm

- We are interested the special case of the Arnoldi method for a Hermitian matrix A, which means that the matrix H is also Hermitian $H^\dagger = (Q^\dagger A Q)^\dagger = Q^\dagger A^\dagger Q = H$. A matrix which is both Hessenberg and Hermitian is **tridiagonal**:

$$H = Q^\dagger A Q = \begin{pmatrix} \alpha_1 & \beta_1 & 0 & & 0 & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 & 0 & 0 \\ 0 & \beta_2 & \alpha_3 & & 0 & 0 & 0 \\ 0 & 0 & \beta_3 & & 0 & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & \alpha_{n-2} & \beta_{n-2} & 0 \\ 0 & 0 & 0 & \cdots & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & 0 & 0 & & 0 & \beta_{n-1} & \alpha_n \end{pmatrix}$$

where $\beta_j = \|u_{j+1}\| = (q_{j+1}, Aq_j)$ and $\alpha_i = (q_i, Aq_i)$ are real.

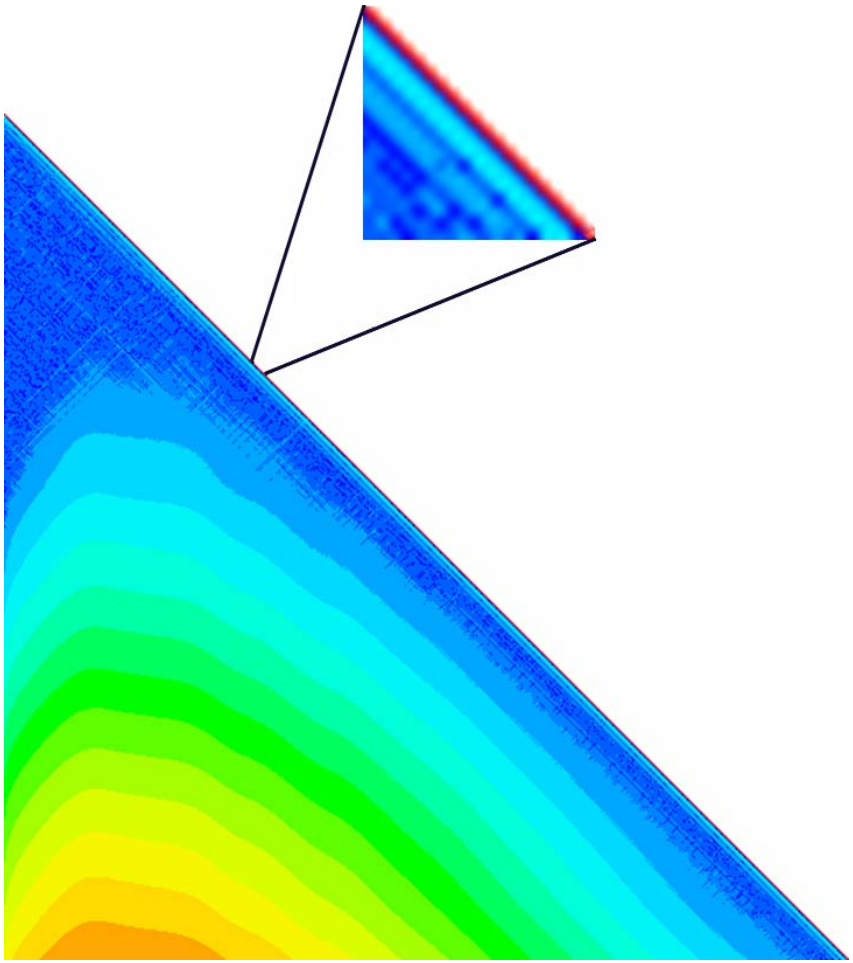
Lánczos Algorithm

- We thus have a three-term recurrence relation

$$Aq_j = \beta_j q_{j+1} + \alpha_j q_j + \beta_{j-1} q_{j-1}.$$

  This defines the **Lánczos algorithm**.
- This greatly simplifies the computation; not only is it particularly easy to diagonalize a tridiagonal matrix using the QR algorithm, but also means that Aq_j is automatically (implicitly) orthogonal to all q_i except for q_{i-1}, q_i, and q_{i+1}.
- Unfortunately, floating-point arithmetic does not respect implicit orthogonality.

Loss of Implicit Orthogonality



- The figure shows the lower triangle of the matrix $Q^\dagger Q$ where the absolute values of the elements are colour-coded from blue (zero) to red (one) on a logarithmic scale.
- This is not very significant when we use the Lánczos scheme to invert a matrix, as in the **conjugate gradient** or **SYMMLQ** algorithms, but here it leads to the well-known disease that the same eigenvectors are found multiple times.
- The goal of the **LANSO** algorithm is to alleviate this problem.

Selective Orthogonalization

- We will deem a Ritz vector y_j to be "good" if it lies within the Krylov space, that is if $(Qy_j, q_{n+k}) = (Qy_j, Qe_{n+k}) = (y_j, e_{n+k}) \approx 0$ for $k = 1, 2, \ldots$. Eigenvalues that are not good will be called "bad".
- Paige showed that the loss of implicit orthogonality occurs primarily in the direction of the good Ritz vectors. This is not suprising:
  - If q_{n+1} and q_{n+2} are orthogonal to an eigenvector z of A then all future Lánczos vectors will also be orthogonal to z, since

    $$(z, q_{n+k}) = 0 \Rightarrow (z, Aq_{n+k}) = (Az, q_{n+k}) = \lambda(z, q_{n+k}) = 0 \Rightarrow (z, q_{n+k+1}) = 0.$$

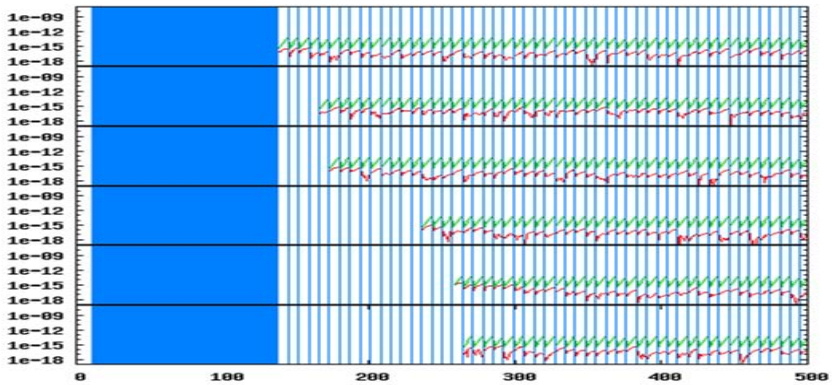
  - Therefore, any rounding errors that appear in the computation of q_{n+k} with a component in the direction z will not be suppressed by orthogonalization to the previous two Lánczos vectors.
  - Moreover, the this component will grow as $(\lambda/\lambda')^k$ where λ' is the largest "bad" eigenvalue.
- It therefore suffices to explicitly orthogonalize the current Lánczos vectors q_n and q_{n+1} with respect to good eigenvectors sufficiently frequently. This is much cheaper than explicitly orthogonalizing with respect to all the previous Lánczos vectors at each step as in the Arnoldi method.

LANSO

- The question is how often do we need to carry out this orthogonalization?
- As rounding errors are of order[5] ε it seems reasonable to choose to do so when the loss of orthogonality has accumulated to be of $\mathcal{O}(\sqrt{\varepsilon})$.
- We therefore choose to orthogonalize $q_{n'}$ and $q_{n'+1}$ with respect to a good eigenvector z when $(z, q_{n'}) > \sqrt{\varepsilon}$.
- In their **LANSO** algorithm Parlett & Scott introduce two bounds:
  - The τ bound, $\tau_{ij} \geq |(z_i, q_j)|$, that is used to trigger orthogonalisation with respect to z_i. This bound is cheaply computed by a three-term scalar recurrence.
  - The κ bound, $\kappa \geq \|Q^\dagger Q - 1\|$, that is used to trigger a search for new good eigenvectors by running the QR algorithm. This is computed by a more complicated scalar recurrence.

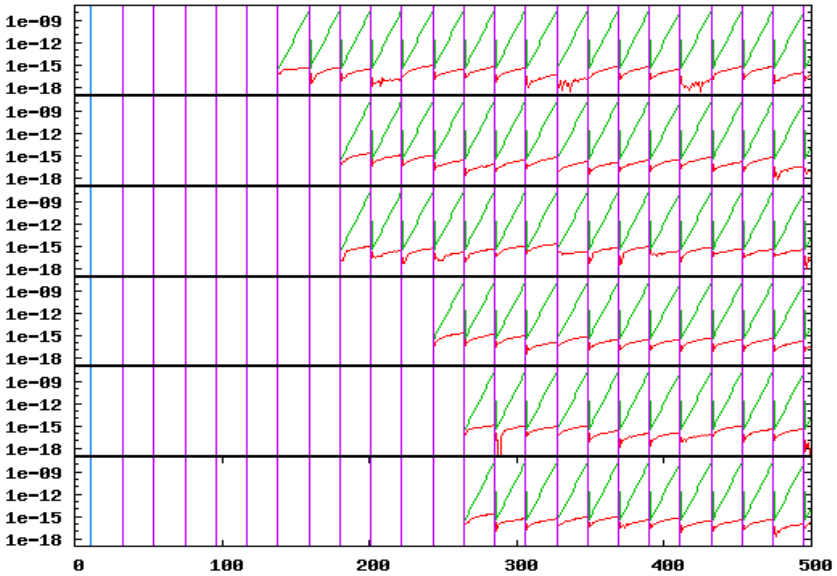
[5] ε is the smallest number such that $1 \oplus \varepsilon \neq 1$ in floating-point arithmetic, it is approximately 10^{-7} for single precision and 10^{-14} for double precision.

LANSO Bounds



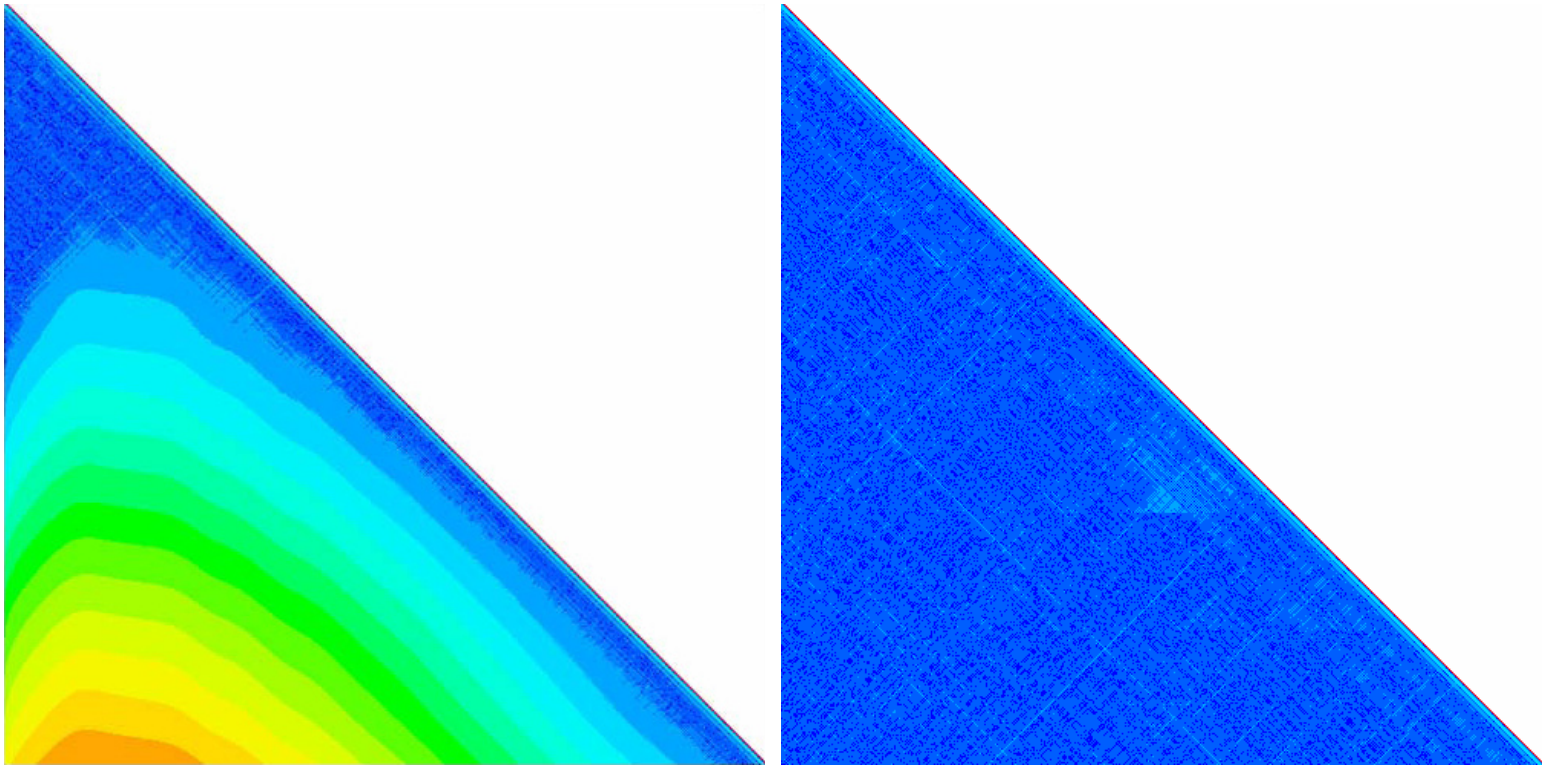
- The figure shows these bounds as a function of the number of Lánczos steps j (dimension of the Krylov subspace).
- The vertical blue lines show when the κ bound triggers a search for new eigenvectors followed by an orthogonalisation with respect to the good ones found.
- The green lines show the value of the τ bounds for six good eigenvectors, and the red lines the actual values for $|(z_i, q_j)|$.
- We see that while the κ bound works it is very loose, and so the τ bounds never get a chance to trigger a selective orthogonalization with respect to their eigenvectors.

LANSO Bounds



- Instead of using the κ bound we decided to look at the τ bound for the "least bad eigenvector", namely the one most likely to converge next.
- The figure shows that the QR algorithm is run much less often. Selective orthogonalization is performed after each such "pause", but not more frequently.
- The κ trigger still occurs at the start to trigger the first QR pause.

Recovery of Orthogonality



Left: Lower triangle of the matrix $Q^\dagger Q$ for Lanczos algorithm with the absolute values of the elements are colour-coded from blue (zero) to red (one) on a logarithmic scale.

Right: The same quantity for Lanczos algorithm with selective orthogonalization (LANSO).

Degenerate Eigenspaces and Restarting

- In exact arithmetic only one eigenvector will be found for each distinct eigenvalue: if an eigenvalue is degenerate then the this vector will be the projection of the initial vector onto its eigenspace.
- In floating-point arithmetic rounding errors will eventually cause the other eigenvectors to appear; this will take longer in higher-precision arithmetic. This is a case where using floating-point arithmetic is an advantage.
- Such degenerate eigenvectors can also be found by restarting the Lánczos algorithm with a new initial vector and deflating with respect to the previously known good eigenvectors. This can be repeated until no more degenerate eigenvectors are found. Presumably a block version of the algorithm could be used too, but the choice of block size is not obvious if the maximum degeneracy is not known *a priori*.
- A cluster of nearby eigenvalues behaves just like a degenerate subspace until sufficient accuracy to resolve the eigenvalues has been attained.

Polynomials and Krylov Spaces

- Let us now study how rapidly we may expect eigenpairs to be computed to a given accuracy. We shall follow the method introduced by Kaniel, and corrected and extended by Paige and Saad.
- Consider the angle $\angle(z_j, u)$ between an eigenvector z_j and an arbitary vector $u \in \mathcal{K}_n(A, v)$.
- We shall assume that the intial vector is not orthogonal to the eigenvector, $(z_j, v) \neq 0$: this would be very unlikely for a randomly chosen v. Indeed, we expect than on average $(z_j, v) \approx 1/\sqrt{N}$ if v is a random unit vector.
- As u is in the Krylov space it can be written as $u = p(A)v$ where p is a polynomial with $\deg p \leq n - 1$. This relationship between matrix polynomials and Krylov vectors is central to the analysis.

Angle Between Eigenvector and Krylov Space

- We may expand u in the eigenbasis of A as

$$u = p(A)\sum_i z_i(z_i, v) = \sum_i p(\lambda_i) z_i(z_i, v).$$

- Taking the ratio of the norm of the component of u perpendicular to z_j to the component parallel to it we see that

$$|\tan\angle(z_j, u)|^2 = \frac{\|u - z_j(z_j, u)\|^2}{\|z_j(z_j, u)\|^2} = \frac{\sum_{i\neq j} |p(\lambda_i)(z_i, v)|^2}{|p(\lambda_j)(z_j, v)|^2}.$$

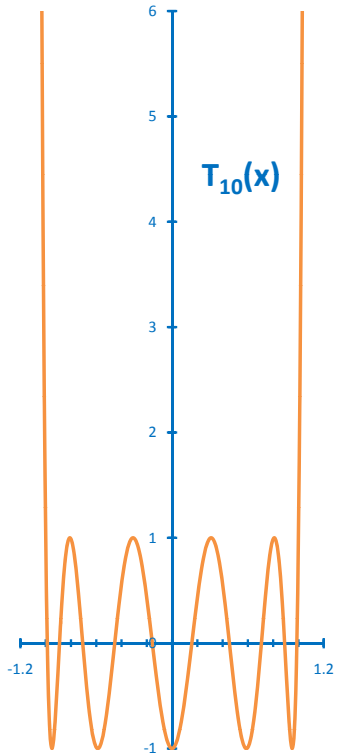
- The angle $\angle_{j,n}$ between z_j and the nearest vector in $\mathcal{K}_n(A, v)$ is therefore given by

$$|\tan\angle_{j,n}|^2 = \min_{u\in\mathcal{K}_n(A,v)} |\tan\angle(z_j, u)|^2 = \min_p \frac{\sum_{i\neq j} |p(\lambda_i)(z_i, v)|^2}{|p(\lambda_j)(z_j, v)|^2}.$$

- We can find a good upper bound by a judicious choice of polynomial p; we want to make $|p(\lambda_i)|$ as small as possible for all $i \neq j$ while making $|p(\lambda_j)|$ as large as possible.

Chebyshev Polynomials

- A very useful class of polynomials for this purpose are **Chebyshev polynomials** of the first kind, which may be defined as
$$T_n(x) = \cos(n \cos^{-1} x).$$
- It is immediately obvious that $|T_n(x)| \leq 1$ for $|x| \leq 1$.
- At first sight it is perhaps suprising that it is a polynomial in x, but application of the binomial theorem shows that
$$T_n(x) = \sum_{j=0}^{\lfloor n/2 \rfloor} \sum_{\ell=0}^{j} \binom{2n}{j} \binom{j}{\ell} (-1)^{j+\ell} x^{n+2(\ell-j)},$$
so it is indeed a polynomial of degree n.
- Moreover it is also easy to see that
$$T_n(x) = \tfrac{1}{2} \left[e^{n \ln\left(x+\sqrt{x^2-1}\right)} + e^{n \ln\left(x-\sqrt{x^2-1}\right)} \right]$$
.
- We thus deduce that $T_n(x) \geq \tfrac{1}{2} e^{n \ln\left(x+\sqrt{x^2-1}\right)}$ for $x \geq 1$.



Bounds on Convergence of the Largest Eigenvalue

- To bound the angle between the largest eigenvector and $\mathcal{K}_n(A, v)$ we can choose the polynomial $p(x) = T_{n-1}\big(\gamma_1(x)\big)$, where

$$\gamma_i(x) = 2\frac{x - \lambda_N}{\lambda_{i+1} - \lambda_N} - 1 = 1 + 2\frac{x - \lambda_{i+1}}{\lambda_{i+1} - \lambda_N},$$

  which maps the interval $[\lambda_{i+1}, \lambda_N] \mapsto [-1, 1]$.

- This choice gives the bound

$$\begin{aligned} |\tan\angle_{1,n}| &\leq \frac{1}{T_{n-1}\big(\gamma_1(\lambda_1)\big)}\sqrt{\frac{\sum_{i\neq 1}|(z_i, v)|^2}{|(z_1, v)|^2}} \\ &\leq \frac{|\tan\angle_{1,1}|}{T_{n-1}\big(\gamma_1(\lambda_1)\big)} \leq 2|\tan\angle_{1,1}|e^{-(n-1)\left(2\mu_1+\mathcal{O}(\mu_1^2)\right)} \end{aligned}$$

  where $\gamma_i(\lambda_i) = 1 + 2\mu_i^2$ with $\mu_i = \sqrt{\frac{\lambda_i - \lambda_{i+1}}{\lambda_{i+1} - \lambda_N}}$ and $\ln\Big(\gamma_i(\lambda_1) + \sqrt{\gamma_i(\lambda_i)^2 - 1}\Big) = 2\mu_i + O(\mu_i^2)$.

Bounds on Convergence of Other Eigenvalue

- For smaller eigenvalues (λ_j for $j > 1$), we can choose the polynomial

$$p(x) = T_{n-j}\big(\gamma_j(x)\big) \prod_{i=1}^{j-1} \frac{\lambda_i - x}{\lambda_i - \lambda_N}.$$

- This ensures that $p(\lambda_i) = 0$ for $i < j$ and $|p(\lambda_i)| \leq 1$ for $i > j$ while $p(\lambda_j) = T_{n-j}(1 + 2\mu_j^2)/K_j$ grows exponentially in n, the number of Lánczos steps.
- Unfortunately the constant $K_j = \prod_{i=1}^{j-1} \frac{\lambda_i - \lambda_N}{\lambda_i - \lambda_j}$ grows exponentially with j, as is particularly large for densely spaced eigenvalues.
- This leads to the bound

$$|\tan \angle_{j,n}| \leq \frac{|K_j \tan \angle_{j,1}|}{T_{n-j}\big(\gamma_j(\lambda_j)\big)} \leq 2|\tan \angle_{j,1}| K_j e^{-(n-j)\left(2\mu_j + \mathcal{O}(\mu_1^2)\right)}.$$

- We can tinker with the polynomial, but nevertheless it seems hopeless to get a useful bound for the eigenvalues far into the interior of the spectrum, which are exactly the ones that we are interested in.

Bounds on Eigenvalue and Eigenvectors

- It is easy to show from the MinMax and MaxMin bounds that

$$0 \leq \lambda_j - \theta_j \leq \rho(u, \lambda_j - A)$$

  for any vector $u \in \mathcal{K}_n(A, v)$ that is orthogonal to the $j-1$ largest Ritz vectors, $(y_i, u) = 0$ for $i < j$.
- This Rayleigh quotient may be bounded[6] in terms of the angle $\angle_{j,n}$ to give

$$0 \leq \lambda_j - \theta_j \leq (\lambda_j - \lambda_N) \left(\frac{K_j \tan \angle_{j,1}}{T_{n-j}\left(\gamma_j(\lambda_j)\right)} \right)^2 .$$

- There is an analogous formula for the error in the Ritz vector $\|z_j - y_j\|$, but as discussed before this must be weaker for nearly degenerate eigenspaces.

[6] For details see chapter 12 of Beresford N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, ISBN 0-89871-402-8.

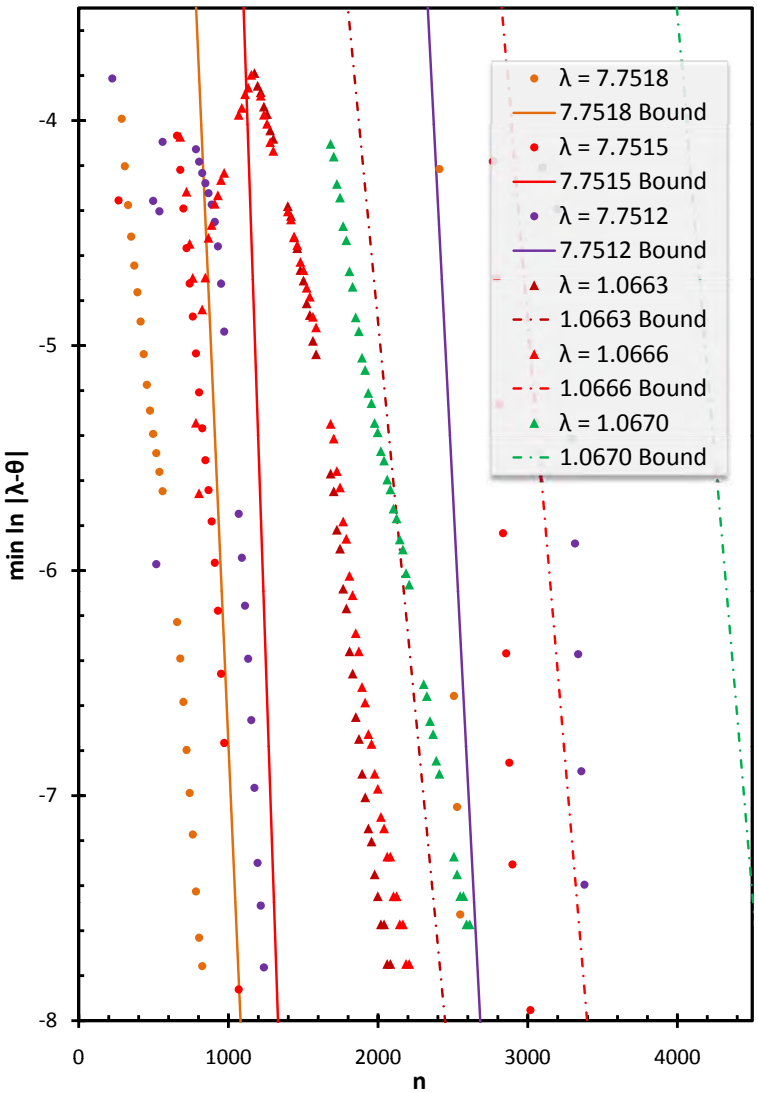
Bounds on Convergence of Interior Eigenvalues

- Nevertheless, notice that the "interior" eigenvalues of A near Λ are the largest eigenvalues of $A' = -(A-\Lambda)^2$.
- Furthermore the Krylov space for A is contained within that for A', albeit with twice the size but the same number of applications of A: $\mathcal{K}_n\left(-(A-\Lambda)^2, v\right) \subseteq \mathcal{K}_{2n}(A, v)$. The Kaniel–Paige–Saad bounds for $\mathcal{K}_n\left(-(A-\Lambda)^2, v\right)$ must therefore also hold for the latter.
- The eigenvalues of A' are $\lambda'_1 > \lambda'_2 > \cdots > \lambda'_N$, with $\lambda'_i = -(\lambda_{\pi_i} - \Lambda)^2$ where the permutation π of the indices is required because the non-linear transformation is not order preserving.
- Since the matrix has been squared the "scaled gap" becomes

$$\left|\frac{\lambda'_i - \lambda'_{i+1}}{\lambda'_{i+1} - \lambda'_N}\right| = \left|\frac{(\lambda_{\pi_i} - \Lambda)^2 - (\lambda_{\pi_{i+1}} - \Lambda)^2}{(\lambda_{\pi_{i+1}} - \Lambda)^2 - (\lambda_{\pi_N} - \Lambda)^2}\right| = \left|\frac{(\lambda_{\pi_i} - \lambda_{\pi_{i+1}})(\lambda_{\pi_i} + \lambda_{\pi_{i+1}} - 2\Lambda)}{(\lambda_{\pi_i} - \lambda_{\pi_N})(\lambda_{\pi_{i+1}} + \lambda_{\pi_N} - 2\Lambda)}\right|.$$

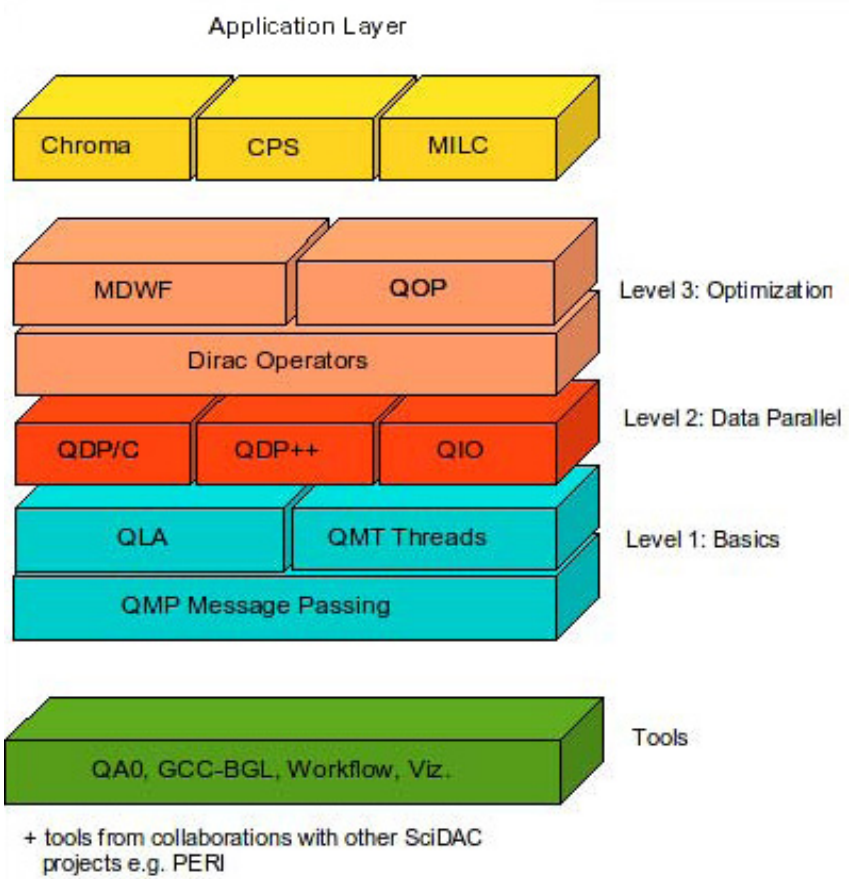
- This quantity is maximized by taking Λ as far away from the average of λ_{π_i} and $\lambda_{\pi_{i+1}}$ as possible, but we cannot make it too big without reordering the eigenvalues and changing the permutation π.
- If λ_i is near a big gap in the spectrum then we can choose Λ near the middle of the gap to get a good bound.

Numerical Results



- The figure shows the convergence of the three largest and three smallest positive eigenvalues of our typical Hermitian Wilson–Dirac operator with $N = 7,962,624$.
- The spectrum is more-or-less symmetric about zero, with a large gap around zero.
- The solid lines show the Kaniel–Paige–Saad bounds for the large Ritz values of the same colour.
- The dashed lines show the bounds for the interior Ritz values obtained using the bounds for $-A^2$, i.e., $\Lambda = 0$.
- Note the appearance of degenerate (or nearly degenerate) eigenvectors due to rounding errors.

Implementation



- After prototyping variants of the LANSO algorithm in **Maple** we implemented it in the Chroma software system, which allows it to run efficiently on a variety of massively parallel architectures.
- Our implementation uses the same interface as the current Chroma eigensolver, which uses a "Ritz functional" method.
- We carried out our tests using the UK's HECToR CRAY XT4, XT6, and XE6 systems (`www.hector.ac.uk`).

Parallelization and Pipelining

- The "large" linear algebra operations and application of the Hermitian Wilson–Dirac operator are carried out in parallel, with each processor working on a small local block of the space-time lattice.
- The "small" linear algebra operations, namely the QR algorithm executed at each pause, is not (yet) parallel; we make use of the efficient implementation of LAPACK available on the CRAY system for this.
- The most time consuming part of the code is the parallel construction of the "large" eigenvectors from the "small" Ritz vectors after each pause, as this requires running through the all the Lánczos vectors q_j.
- There are some interesting architecture-dependent trade-offs to be investigated here. Depending on the amount of memory available and the memory bandwith we can choose between
  1. Storing the Lánczos vectors in main memory (DRAM);
  2. Storing the Lánczos vector in secondary storage (disk or Flash RAM);
  3. Recomputing the Lánczos vectors at each pause. This minimizes off-chip data transfer, and is "embarassingly parallel" up to a few global sum operations (for inner products and norms).

Conclusions

- Our implementation of LANSO is significantly faster than the Ritz method currently implemented in Chroma, which is widely used.
- The Ritz method finds the eigenpairs of A^2 and then has to separate the positive and negative eigenvalues of A that become nearly degenerate upon squaring.
- We have shown that the bounds for convegence of the physically interesing small-magnitude eigenvalues of A in the Lánczos algorithm are essentially the same as for A^2, and that these bounds are comparable with those for the large-magnitude eigenvalues because there is a large void in the spectrum of A around zero.
- Moreover, by using the Krylov space for A rather than A^2 the positive and negative eigenvalues are automatically separated.

Conclusions

- We have modified the original LANSO procedure by ignoring the κ bound (which tests for loss of orthogonality), as it is too pessimistic for our large matrix.
- In its place we use the τ bound for the "least bad eigenvector"; this appears to work very satisfactorily.
- We only look for Ritz values in a specified interval when running the QR algorithm at pauses, so we do not have to compute eigenpairs we are not interested in.
  - This means we expect to lose orthogonality w.r.t. good Ritz eigenvectors outside this interval, but we do not care as we only generate spurious duplicate eigenvectors for eigenvalues we are ignoring.
  - This makes the κ bound useless.
- We follow the original LANSO suggestion of restarting the algorithm (but deflating w.r.t. previously found good eigenvectors) until we have found all degenerate eigenvectors in the spectral interval of interest with correct multiplicities.