- Basic idea of Krylov space methods:
  Find (exponentially good in n) approximation of E_0, $|E_0\rangle$ (and possibly a few excited states) by projecting H to an $(n+1)\times(n+1)$ matrix within $K_{n+1}(H, |\psi_i\rangle)$.
- Crucial ingredient: Approximate <u>invariance</u> of $K_{n+1}(H, |\psi_i\rangle)$ for sufficiently large n.

Interlude: Invariant subspaces

- $S \subset \mathcal{H}$ is an invariant subspace of H, if $|x\rangle \in S \Rightarrow H|x\rangle \in S$
- If $Q = (|q_0\rangle, \dots, |q_n\rangle)$ with $Q^\dagger Q = \mathbb{1}_{(n+1)}$, then

orthonormal basis of S

$T = Q^\dagger H Q$ is the projection of H onto the invariant subspace S, and we have $\boxed{HQ = QT}$

$(n+1)\times(n+1)$

$\rightsquigarrow$ From eigenvectors $\vec{y}_\lambda$ of T with $T\vec{y}_\lambda = E_\lambda \vec{y}_\lambda$, we find that $QT\vec{y}_\lambda = E_\lambda Q\vec{y}_\lambda \iff HQ\vec{y}_\lambda = E_\lambda Q\vec{y}_\lambda$,

i.e. $Q\vec{y}_\lambda$ is an eigenstate of H with the <u>same</u> eigenvalue E_λ.

$$\left(|E_\lambda\rangle = Q\vec{y}_\lambda \right)$$

- Relation to Krylov methods: $K_n(H, |\psi_i\rangle)$ is to exponentially good precision in n an invariant subspace containing the ground state, since $H^n|\psi_i\rangle$ converges to extremal eigenstates.

- Simple and successful method: Lanczos algorithm
- Start with a random initial state $|q_0\rangle := |\Psi_i\rangle$ (normalized) and iteratively apply H as follows:

- $$H|q_0\rangle = \underset{\langle q_0|H|q_0\rangle}{a_0}|q_0\rangle + b_1|q_1\rangle$$

$$H|q_0\rangle = \sum_{e=0}^{D \,(\dim \mathcal{H})} |e\rangle\langle e|H|q_0\rangle \underset{|q_0\rangle =: |0\rangle}{=} \overbrace{\langle q_0|H|q_0\rangle}^{a_0}|q_0\rangle + \underbrace{\sum_{e\neq 0}^{D} |e\rangle\langle e|H|q_0\rangle}_{=: b_1|q_1\rangle}$$

$\rightsquigarrow \langle q_0|q_1\rangle = 0$ by construction.

$b_1 \in \mathbb{R}$ is a phase convention on $|q_1\rangle$, and $a_0 \in \mathbb{R}$ due to $H = H^\dagger$.

- $H|q_1\rangle = b_1'|q_0\rangle + a_1|q_1\rangle + b_2|q_2\rangle$, where

$$\langle q_0|H|q_1\rangle = b_1' \overset{(b_1\in\mathbb{R})}{=} \langle q_1|H|q_0\rangle = b_1$$

$\langle q_0 | q_1 \rangle = \langle q_0 | q_2 \rangle = 0$

- $H |q_i\rangle = b_i |q_{i-1}\rangle + a_i |q_i\rangle + b_{i+1} |q_{i+1}\rangle$

  with $\langle q_j | q_{i+1} \rangle = 0 \quad \forall_{j < i+1}$.

↳ This generates an orthonormal basis $|q_0\rangle, \ldots, |q_n\rangle$ of the $(n+1)$-th Krylov space in n steps, and with $Q = (|q_0\rangle, \ldots, |q_n\rangle)$, $T = Q^\dagger H Q$ is of the tridiagonal form:

$$T = \begin{pmatrix} a_0 & b_1 & & & \\ b_1 & a_1 & b_2 & & \\ & b_2 & a_2 & b_3 & \\ & & \ddots & \ddots & b_n \\ & & & b_n & a_n \end{pmatrix}$$

Why is $\langle q_j | q_{i+1} \rangle = 0 \ \forall_{j<i+1}$ true?

$$\langle q_0 | H | q_2 \rangle = \left(\langle q_0 | a_0 + \langle q_1 | b_1 \right) | q_2 \rangle = 0$$

$$\langle q_0 | a_0 + \langle q_1 | b_1$$

$$\left[\langle q_0 | q_2 \rangle = \langle q_1 | q_2 \rangle = 0 \right]$$

$$=$$

$$= \langle q_0 | H | q_2 \rangle = \langle q_0 | \left(b_1 | q_1 \rangle + a_2 | q_2 \rangle + b_3 | q_3 \rangle \right) =$$

$$= \langle q_0 | q_3 \rangle b_3 = 0$$

$$\langle q_0 | q_1 \rangle = \langle q_0 | q_2 \rangle = 0$$

$\Rightarrow$ For any $b_3 \neq 0$, we have $\langle q_0 | q_3 \rangle = 0$ ✓

Continue this argument by induction in i (here i=2).

$\rightsquigarrow$ Find the lowest eigenvalue $\mathcal{E}_0$ of T with eigenvector $\bar{y}_0$

$\rightsquigarrow$ For large n $\;\mathcal{E}_0 \to E_0$ and $Q\bar{y}_0 \to |E_0\rangle$

[up to numerical instabilities]

- Lanczos pseudocode (with $D = \dim \mathcal{H} = 2^N$)

- Initialize D-vectors p and q, and n-vectors α, β,

  random   0-vector $n \sim 100$

  double temp, int $j = 0$

- Fix $\varepsilon > 0$ as a cutoff ($\sim 10^{-13}$) and set $\beta[0] = 1.0$;

- Fix maxit ~ 100;

while $(|\beta[j]| > \varepsilon \wedge j < \text{maxit} - 1)$ {

```
if (j ≠ 0) {
        int i = 0;
        for(i=0, ..., dim-1) {   [% tidy up q and p for next step]
            temp = p[i];
            p[i] = q[i]/β[j]
            q[i] = - β[j].temp ;
        }
}                           [% matrix-vector mult. (see B.1 for implementation)]
q = q + H.p
                                 [% scalar product of dim-vectors]
j++;
α[j-1] = (p, q) ;
q = q - α[j-1]p;
β[j] = sqrt((q,q)) ;
}
```

$\rightsquigarrow$ tridiagonal (symmetric) matrix T stored in

$\beta[1:j-1]$, $\alpha[0:j-1]$.

$\{b_1, \dots, b_{j-1}\}$ $\{a_0, \dots, a_{j-1}\}$