# Lanczos Algorithm for Qiskit Dynamics

Mentor – Mirko Amico, IBM

Mentee – R K Rupesh, University of Hyderabad

$$|\psi(t)\rangle = e^{-iHt}\,|\psi(0)\rangle$$

Qiskit

# Why Lanczos

Qiskit

- Local Hamiltonians describing qubit systems are sparse

$$H = \begin{pmatrix} 0 & 0 & \alpha & \beta \\ 0 & d_0 & 0 & 0 \\ \alpha^* & 0 & 0 & 0 \\ \beta^* & 0 & 0 & d_1 \end{pmatrix}$$

- Calculating time evolution requires exponentiating the Hamiltonian

$$|\psi(t)\rangle = e^{-iHt}|\psi_0\rangle$$

- This can be done by diagonalizing $H$

$$e^{-iHt} = S^\dagger e^{-iDt} S$$

# Why Lanczos

- Time evolution requires only sparse matrix – vector multiplication

$$|\psi(t)\rangle = e^{-iHt}|\psi_0\rangle$$

$$= \sum_n \frac{(-it)^n}{n!} H^n |\psi_0\rangle$$

$$= \sum_n \frac{(-it)^n}{n!} |u_n\rangle$$

- $|u_0\rangle = |\psi\rangle \quad |u_1\rangle = H|u_0\rangle \quad |u_2\rangle = H|u_1\rangle \; \ldots.$

# Why Lanczos (Krylov Subspace)

- $K_r = \{|\psi\rangle,\ A|\psi\rangle, A^2\,|\psi\rangle, A^3\,|\psi\rangle \ldots A^{k-1}|\psi\rangle\}$ Is the krylov subspace for a given matrix $A$ and vector $|\psi\rangle$ of order $k$
- One can construct a basis $\{|\phi_i\rangle\}$ for this subspace using Gram-Schmidt

$$\left|\widetilde{\phi}_{k-1}\right\rangle = |u_{k-1}\rangle - \sum_i \langle\phi_i|u_{k-1}\rangle\,|\phi_i\rangle$$

$$|\phi_{k-1}\rangle = \frac{\left|\widetilde{\phi}_{k-1}\right\rangle}{\left\langle\widetilde{\phi}_{k-1}\middle|\widetilde{\phi}_{k-1}\right\rangle}$$

# Why Lanczos (Krylov Subspace)

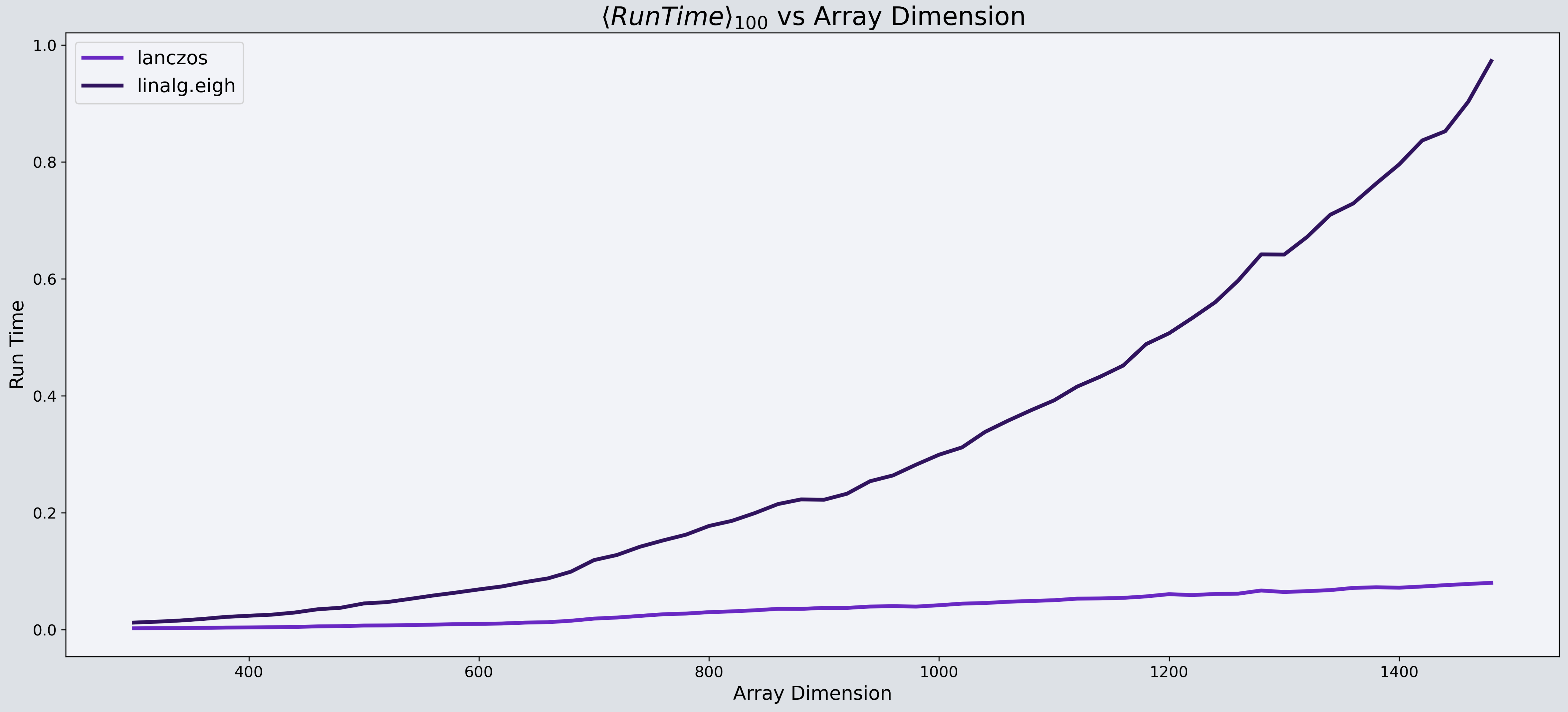
- One can construct an orthogonal matrix $Q_{n,k}$ With $|\phi_i\rangle$ as the columns such that

$$T_{k,k} = Q_{k,n}^{\dagger} H_{n,n} Q_{n,k}$$

- Where $T$ is a Tridiagonal matrix
- Diagonalizing this Tridiagonal matrix is a lot faster since typically, $k \ll n$
- The Eigen-vectors of T is then an approximation of the lowest k Eigen vectors of H
- Therefore, we have $V_n = Q_{n,k} V_k$ where $V\_k$ are the eigenvectors of T

# Lanczos vs NumPy

# (ground state calculation)



# Lanczos Time-evolution

Qiskit

- Once we have the basis vectors and the Tridiagonal projection, we have the equation

$$H_{n,n} = Q_{n,k} T_{k,k} Q^{\dagger}_{k,n}$$

- Thus, the time evolution unitary becomes
- $e^{-iHt} = e^{-iQTQ^{\dagger}t} = Qe^{-iTt}Q^{\dagger} = QVe^{-i\mathbf{diag}(T)t}\, V^{\dagger}Q^{\dagger}$
- If we had chosen the initial vector of the Lanczos iteration to be same as the initial state which we want to evolve, then the rows of $Q$ (other than first) are orthogonal to $|\psi_i\rangle$

# Lanczos Time-evolution

- If we had chosen the initial vector of the Lanczos iteration to be same as the initial state which we want to evolve, then the rows of $Q$ (other than first) are orthogonal to $|\psi_i\rangle$
- $e^{-iHt} = QV_k e^{-i\mathbf{diag}(T)t}\, V_k^\dagger Q^\dagger |\psi_i\rangle$

$$= QV_k e^{-i\mathbf{diag}(T)t}\, V_k^\dagger \delta_{0,k}$$

$$= QV_k e^{-i\mathbf{diag}(T)t}\, V_0^\dagger = QV_k \exp(-iE_T t)|\psi_i\rangle$$

- This increases the accuracy of the simulation since we aren't affected by loss of orthogonality.

# The PR

Qiskit

## Implementing Lanczos algorithm as a new solver method #109

Open **rupeshknn** wants to merge 27 commits into `Qiskit:main` from `rupeshknn:lanczos`

Conversation 1 | Commits 27 | Checks 0 | Files changed 5

**rupeshknn** commented 3 days ago • edited

### Summary

Lanczos algorithm is an approximate diagonalisation method. It is implemented as an LMDE method and is a considerable speedup compared to `scipy.expm`

### Details and comments

This PR adds a new fixed-step solver method `lanczos_diag`. This method only works with hermitian generators and works best in sparse evaluation mode. A follow up with a Jax implementation of the same is in progress.

# Thank You

R K Rupesh

MSc Physics, University of Hyderabad

Gachibowli, India

GitHub: github.com/rupeshknn/lanczos-QD