# Davidson Algorithm for Eigenvalue Problems

CH481

# 1 Introduction

In many problems of quantum chemistry, we need to find a few of the lowest eigenvalues and eigenvectors of a very large symmetric (or Hermitian) matrix $\mathbf{A}$, such as a Hamiltonian or Fock matrix. Full diagonalization of $\mathbf{A}$ is computationally expensive, requiring $\mathcal{O}(n^3)$ operations and $\mathcal{O}(n^2)$ memory. When $n$ is in the tens or hundreds of thousands, this becomes impossible. The **Davidson algorithm**, proposed by Ernest Davidson in 1975, is an iterative *subspace method* that efficiently computes a few lowest eigenpairs of $\mathbf{A}$ using only repeated matrix–vector products $\mathbf{Ax}$. It is widely used in quantum chemistry and other large-scale eigenvalue problems.

# 2 The Eigenvalue Problem

We begin with the symmetric eigenvalue equation:

$$\mathbf{Ax} = \lambda \mathbf{x}, \qquad \mathbf{A} \in \mathbb{R}^{n\times n},\ \mathbf{A} = \mathbf{A}^T. \tag{1}$$

We seek the smallest few eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{n_{\text{eig}}}$ and their normalized eigenvectors $\mathbf{x}_i$ satisfying $\mathbf{x}_i^T \mathbf{x}_i = 1$. For large $n$, direct diagonalization is infeasible. The Davidson method constructs a much smaller *subspace* in which approximate eigenpairs (called *Ritz pairs*) are computed.

# 3 Subspace Projection Principle

Let $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ be a set of $m$ orthonormal basis vectors, representing the current search subspace $\mathcal{V}_m$. We project the large eigenvalue problem onto this subspace:

$$\mathbf{T}_m = \mathbf{V}_m^T \mathbf{A} \mathbf{V}_m, \tag{2}$$

where $\mathbf{T}_m$ is an $m \times m$ symmetric matrix called the **projected matrix**. Solving the smaller eigenproblem

$$\mathbf{T}_m \mathbf{s}_j = \theta_j \mathbf{s}_j \tag{3}$$

yields approximate eigenvalues $\theta_j$ and eigenvectors $\mathbf{s}_j$. The approximate eigenvector in the full space is reconstructed as:

$$\mathbf{x}_j = \mathbf{V}_m \mathbf{s}_j, \tag{4}$$

and the pair $(\theta_j, \mathbf{x}_j)$ is called a **Ritz pair**.

# 4 Residuals and Correction Vectors

The accuracy of the Ritz vector $\mathbf{x}_j$ can be tested by computing the residual:

$$\mathbf{r}_j = (\mathbf{A} - \theta_j \mathbf{I})\mathbf{x}_j. \tag{5}$$

If $\mathbf{r}_j = 0$, the approximation is exact. The residual measures how far the approximate eigenvector is from satisfying the true eigenvalue equation.

To improve the approximation, the Davidson method seeks a correction vector $\delta \mathbf{x}_j$ that approximately satisfies:

$$(\mathbf{A} - \theta_j \mathbf{I})\, \delta \mathbf{x}_j = -\mathbf{r}_j. \tag{6}$$

Solving this equation directly is costly, so Davidson proposed a simple diagonal preconditioner:

$$\delta \mathbf{x}_j \approx \frac{\mathbf{r}_j}{\theta_j - \mathrm{diag}(\mathbf{A})}. \tag{7}$$

This expression divides each component of the residual by the difference between $\theta_j$ and the corresponding diagonal element of $\mathbf{A}$. The resulting correction vector $\mathbf{q}_j = \delta \mathbf{x}_j$ is added to the subspace after orthonormalization.

# 5 Algorithm Outline

The Davidson algorithm proceeds as follows:

1. **Initialization:** Choose $k$ orthonormal initial guess vectors $\mathbf{t}_1, \ldots, \mathbf{t}_k$. Set the initial subspace $\mathbf{V}_k = [\mathbf{t}_1, \ldots, \mathbf{t}_k]$. For example, unit vectors as a guess.

2. **Projection:** Form the projected matrix $\mathbf{T}_m = \mathbf{V}_m^T \mathbf{A} \mathbf{V}_m$.

3. **Diagonalization:** Solve $\mathbf{T}_m \mathbf{s}_j = \theta_j \mathbf{s}_j$ for the lowest $n_{\text{eig}}$ eigenvalues.

4. **Ritz Vectors:** Form approximate eigenvectors $\mathbf{x}_j = \mathbf{V}_m \mathbf{s}_j$.

5. **Residuals:** Compute $\mathbf{r}_j = \mathbf{A}\mathbf{x}_j - \theta_j \mathbf{x}_j$.

6. **Check Convergence:** If $\|\mathbf{r}_j\| < \varepsilon$ for all $j$, stop.

7. **Correction:** Compute Davidson corrections:

$$\mathbf{q}_j = \frac{\mathbf{r}_j}{\theta_j - \mathrm{diag}(\mathbf{A})}.$$

   Set elements corresponding to NaN or Inf to zero.

8. **Orthonormalization:** Orthonormalize $\mathbf{q}_j$ against the existing subspace $\mathbf{V}_m$ using the Gram–Schmidt or QR process.

9. **Subspace Expansion:** Append new directions to the subspace:

$$\mathbf{V}_{m+k} = [\mathbf{V}_m, \mathbf{q}_1, \ldots, \mathbf{q}_k].$$

10. **Repeat:** Return to Step 2 with the expanded subspace.

# 6 Pseudocode

**Algorithm 1** Block Davidson Algorithm (for Symmetric Matrices)

**Require:** Matrix $\mathbf{A}$, initial guess vectors $\{\mathbf{t}_j\}_{j=1}^{k}$, desired eigenvalue count $n_{\text{eig}}$, tolerance $\varepsilon$.

Initialize $\mathbf{V}_k = \text{orthonormalize}(\{\mathbf{t}_j\})$
$m \leftarrow k$
$\theta_{\text{old}} \leftarrow 10^6$
**while** not converged **do**
    $\mathbf{T}_m = \mathbf{V}_m^T \mathbf{A} \mathbf{V}_m$
    Solve $\mathbf{T}_m \mathbf{s}_j = \theta_j \mathbf{s}_j$
    Sort $\theta_j$ in ascending order
    **for** $j = 1 \dots n_{\text{eig}}$ **do**
        $\mathbf{x}_j = \mathbf{V}_m \mathbf{s}_j$
        $\mathbf{r}_j = \mathbf{A}\mathbf{x}_j - \theta_j \mathbf{x}_j$
        **if** $\|\mathbf{r}_j\| < \varepsilon$ for all $j$ **then**
            **break**
        **end if**
        $\mathbf{q}_j = \mathbf{r}_j / (\theta_j - \text{diag}(\mathbf{A}))$
        Replace NaN/Inf entries with zero
    **end for**
    Orthonormalize $\{\mathbf{q}_j\}$ against $\mathbf{V}_m$
    Expand subspace: $\mathbf{V}_{m+k} = [\mathbf{V}_m, \mathbf{q}_1, \dots, \mathbf{q}_k]$
    $m \leftarrow m + k$
    $\Delta E = \left\| \boldsymbol{\theta}_{1:n_{\text{eig}}} - \boldsymbol{\theta}_{\text{old}} \right\|$
    **if** $\Delta E < \varepsilon$ and all $\|\mathbf{r}_j\| < \varepsilon$ **then**
        Converged $\rightarrow$ Stop
    **end if**
    $\boldsymbol{\theta}_{\text{old}} \leftarrow \boldsymbol{\theta}_{1:n_{\text{eig}}}$
**end while**
**return** $\{\theta_j, \mathbf{x}_j\}_{j=1}^{n_{\text{eig}}}$

# 7 Program Flow Summary

| Stage | Description |
|---|---|
| Initialization | Choose $k$ guess vectors, orthonormalize them. |
| Projection | Build $\mathbf{T}_m = \mathbf{V}_m^T \mathbf{A} \mathbf{V}_m$. |
| Diagonalization | Solve for Ritz pairs $(\theta_j, \mathbf{s}_j)$. |
| Residuals | Compute $\mathbf{r}_j = (\mathbf{A} - \theta_j \mathbf{I})\mathbf{x}_j$. |
| Preconditioning | Compute $\mathbf{q}_j = \mathbf{r}_j / (\theta_j - \text{diag}(\mathbf{A}))$. |
| Orthonormalization | Orthogonalize new $\mathbf{q}_j$ against $\mathbf{V}_m$. |
| Subspace Expansion | Form new subspace $\mathbf{V}_{m+k}$. |
| Convergence Check | Stop when $\|\mathbf{r}_j\| < \varepsilon$ and $\Delta E < \varepsilon$. |

# 8 Important Notes

- **Orthonormalization:** Always reorthogonalize the subspace. This prevents numerical instabilities due to linear dependence.
- **Preconditioner:** The diagonal preconditioner works well when $\mathbf{A}$ is diagonally dominant. For other matrices, consider incomplete factorizations or block preconditioners.
- **Block Size ($k$):** Ensure $k \geq n_{\text{eig}}$. Using $k$ slightly larger than $n_{\text{eig}}$ often improves convergence for clustered eigenvalues.
- **Safeguards:** Always sanitize correction vectors (remove NaN/Inf), normalize new vectors before adding, and check for near-zero norms (skip nearly zero directions).
- **Convergence Criteria:** Combine residual norm thresholds with changes in Ritz values ($\Delta E$) to avoid false convergence.