Tau Matrix Method for Berry Phase Calculation

May 13, 2025

1 Introduction

This document provides a detailed explanation of the tau matrix method implemented in the Arrowhead project for computing Berry phases in quantum systems. The tau matrix method is a numerical approach for calculating the Berry connection and Berry phase by evaluating the evolution of eigenstates along a closed path in parameter space.

2 Theoretical Background

2.1 Arrowhead Hamiltonian

The system is described by a 4×4 arrowhead Hamiltonian matrix with the following structure:

$$H(\boldsymbol{R}_\theta) = \begin{pmatrix} \hbar\omega + \sum_i V_x(\boldsymbol{R}_i) & t_{01} & t_{02} & t_{03} \\ t_{01} & V_e(\boldsymbol{R}_0) & 0 & 0 \\ t_{02} & 0 & V_e(\boldsymbol{R}_1) & 0 \\ t_{03} & 0 & 0 & V_e(\boldsymbol{R}_2) \end{pmatrix} \qquad (1)$$

where:

- $\hbar\omega$ is the energy quantum of the system

- $V_x(\boldsymbol{R}_i) = a_{Vx}|\boldsymbol{R}_i|^2$ is the potential energy function

- $V_e(\boldsymbol{R}_i) = \sum_j V_x(\boldsymbol{R}_j) + V_a(\boldsymbol{R}_i) - V_x(\boldsymbol{R}_i)$ is the effective potential

- $V_a(\boldsymbol{R}_i) = a_{Va}(|\boldsymbol{R}_i - x_{\text{shift}}|^2 + c)$ is the additional potential

- t_{0i} are the transition dipole moments between states

The parameter vector $\boldsymbol{R}_\theta$ traces a perfect circle orthogonal to the $x = y = z$ line in 3D space, parameterized by angle θ.

2.2 Berry Connection and Berry Phase

The Berry connection $\boldsymbol{A}_n(\boldsymbol{R})$ for the n-th eigenstate is defined as:

$$\boldsymbol{A}_n(\boldsymbol{R}) = i\langle n(\boldsymbol{R})|\nabla_{\boldsymbol{R}}|n(\boldsymbol{R})\rangle \tag{2}$$

The Berry phase γ_n is the integral of the Berry connection around a closed loop C in parameter space:

$$\gamma_n = \oint_C \boldsymbol{A}_n(\boldsymbol{R}) \cdot d\boldsymbol{R} \tag{3}$$

3 The Tau Matrix Method

3.1 Definition of the Tau Matrix

The tau matrix method is a numerical approach for computing the Berry connection and Berry phase. The key element is the tau matrix, whose elements are defined as:

$$\tau_{nm}(\theta) = \langle \psi_m(\theta)|\frac{\partial}{\partial\theta}|\psi_n(\theta)\rangle \tag{4}$$

where $|\psi_n(\theta)\rangle$ is the n-th eigenstate of the Hamiltonian $H(\boldsymbol{R}_\theta)$.

3.2 Numerical Implementation

In practice, the derivative with respect to θ is approximated using finite differences:

$$\tau_{nm}(\theta_i) \approx \langle \psi_m(\theta_i)|\frac{|\psi_n(\theta_{i+1})\rangle - |\psi_n(\theta_{i-1})\rangle}{2\Delta\theta}\rangle \tag{5}$$

where $\Delta\theta$ is the step size in the discretization of the path.

The diagonal elements $\tau_{nn}(\theta)$ correspond to the Berry connection in the θ basis:

$$A_n(\theta) = \tau_{nn}(\theta) \tag{6}$$

The Berry phase is then obtained by integrating the Berry connection around the closed loop:

$$\gamma_n = \oint \tau_{nn}(\theta)d\theta \approx \sum_i \tau_{nn}(\theta_i)\Delta\theta_i \tag{7}$$

3.3 Off-Diagonal Elements

The off-diagonal elements $\tau_{nm}(\theta)$ for $n \neq m$ provide information about the coupling between different eigenstates. These elements can be used to analyze transitions between states and to identify regions in parameter space where eigenstates are strongly coupled.

4 Implementation Details

4.1 Eigenvector Gauge Fixing

In numerical calculations, the eigenvectors returned by diagonalization routines may have arbitrary phases. To ensure consistency in the calculation of the Berry connection, the phases of the eigenvectors are fixed using the following procedure:

$$|\psi_n(\theta_{i+1})\rangle \to |\psi_n(\theta_{i+1})\rangle \cdot \frac{\langle\psi_n(\theta_i)|\psi_n(\theta_{i+1})\rangle}{|\langle\psi_n(\theta_i)|\psi_n(\theta_{i+1})\rangle|} \tag{8}$$

This ensures that the overlap between consecutive eigenstates is positive real, minimizing the artificial phase jumps.

4.2 Handling Boundary Conditions

Special care is taken to handle the boundary conditions at the beginning and end of the parameter path. For a closed loop, the eigenvectors at $\theta = 0$ and $\theta = 2\pi$ should be identical up to a phase factor. The implementation uses modular arithmetic to ensure proper wrapping around the loop:

```
if i == 0:
    psi_prev = eigvectors_all[N - 1, :, n]  # Vector at theta_max
    psi_next = eigvectors_all[1, :, n]
    delta_theta_for_grad = 2 * (theta_vals[1] - theta_vals[0])
elif i == N - 1:
    psi_prev = eigvectors_all[N - 2, :, n]
    psi_next = eigvectors_all[0, :, n]  # Vector at theta_0
    delta_theta_for_grad = 2 * (theta_vals[1] - theta_vals[0])
else:
    psi_prev = eigvectors_all[i - 1, :, n]
    psi_next = eigvectors_all[i + 1, :, n]
    delta_theta_for_grad = theta_vals[i + 1] - theta_vals[i - 1]
```

Listing 1: Handling boundary conditions in tau matrix calculation

4.3 Integration Methods

The implementation offers two methods for integrating the Berry connection to obtain the Berry phase:

1. Simple Riemann sum:

$$\gamma_n \approx \sum_i \tau_{nn}(\theta_i)\Delta\theta_i \tag{9}$$

2. Trapezoidal rule:

$$\gamma_n \approx \sum_i \frac{\tau_{nn}(\theta_i) + \tau_{nn}(\theta_{i-1})}{2}\Delta\theta_i \tag{10}$$

The trapezoidal rule generally provides more accurate results, especially for coarser discretizations of the parameter path.

5 Alternative Methods

5.1 Wilson Loop Method

In addition to the tau matrix method, the implementation also includes the Wilson loop method for calculating Berry phases. This method computes the Berry phase directly from the overlaps between consecutive eigenstates:

$$\gamma_n = -\mathrm{Im}\left[\ln \prod_i \langle \psi_n(\theta_i)|\psi_n(\theta_{i+1})\rangle\right] \tag{11}$$

```python
def compute_berry_phase_overlap(eigvectors_all):
    """
    Compute Berry phases \$\gamma_n\$ for each eigenstate n along a
     closed path in R-space,
    using the overlap (Wilson loop) method.

    Parameters:
    - eigvectors_all: ndarray of shape (N, M, M), eigenvectors at
    each \$\theta\$

    Returns:
    - berry_phases: ndarray of shape (M,), Berry phase for each
    eigenstate in radians
    """
    N, M, _ = eigvectors_all.shape
    berry_phases = np.zeros(M)

    for n in range(M):
        phase_sum = 0.0
        for i in range(N):
            psi_i = eigvectors_all[i, :, n]
            psi_next = eigvectors_all[(i + 1) % N, :, n]

            # Normalize (defensive)
            psi_i /= np.linalg.norm(psi_i)
            psi_next /= np.linalg.norm(psi_next)

            # Overlap gives phase evolution
            overlap = np.vdot(psi_i, psi_next)
            phase_diff = np.angle(overlap)
            phase_diff = np.unwrap(np.array([phase_diff]))  #
    Ensure phase is continuous

            # Correct for sign flips if necessary
            if np.vdot(psi_i, psi_next) < 0:
                phase_diff += np.pi

            phase_sum += phase_diff[0]

        berry_phases[n] = phase_sum
```

```
38        return berry_phases
```

Listing 2: Implementation of the Wilson loop method for Berry phase calculation

This approach avoids the explicit calculation of the Berry connection and can be more numerically stable in some cases. Also we got the Berry phases from the tau matrix method and the Wilson loop method, as:

$$\gamma^{\text{2-level}} = \begin{pmatrix} 0 & 2\pi \\ 2\pi & 0 \end{pmatrix} \tag{12}$$

An example run of the Wilson loop method shows the following results:

```
1 Tau diagonal: [-1. -1. -1. ...  1.  1.  1.]
2 Tau diagonal imaginary part: [0. 0. 0. ... 0. 0. 0.]
3 Tau diagonal real part: [-1. -1. -1. ...  1.  1.  1.]
4 Theta values: (50000,)
5 Eigenvectors: (50000, 4, 4)
6 Berry phases: [0.         6.28318531 6.28318531 0.          ]
```

Listing 3: Example output from the Wilson loop method calculation

This output confirms the theoretical prediction of the 2π (approximately 6.28) phase for the relevant eigenstates, while the others show zero phase. The large number of discretization points (50,000) ensures high numerical accuracy. This is great for the two level case, but for the three level case we need to use the tau matrix method, as the Wilson loop method is not able to give the correct Berry phases for cases, such as between the other states.

5.2 Direct Calculation of Berry Curvature

The implementation also includes methods for directly calculating the Berry curvature, which is the curl of the Berry connection. For a 1D parameter space, this simplifies to the derivative of the Berry connection with respect to the parameter:

$$\Omega_n(\theta) = \frac{d}{d\theta} A_n(\theta) \tag{13}$$

The Berry curvature provides information about the local geometric properties of the eigenstate manifold and can be used to identify topological features such as Dirac points or band inversions.

6 Numerical Results and Analysis

6.1 Berry Phase Matrix Structure

The numerical results show that the Berry phase matrix γ_{nm} has a specific structure with the following features:

- The diagonal elements γ_{nn} are typically small (order 10^{-3} to 10^{-4}), indicating minimal geometric phase accumulation for individual states.

- Large off-diagonal elements (order 10^0) appear in specific positions, particularly between states 1-2, 2-1, 1-3, and 3-1.

- The values close to $\pm 2\pi$ in the [1,2] and [2,1] positions indicate complete phase winding, which is a signature of non-trivial topology.

6.2   Two-Level Approximation Validity

The structure of the Berry phase matrix strongly supports the use of a two-level approximation (TLA) for this system:

- The large off-diagonal elements (relative to the diagonal) suggest strong coupling between specific state pairs.

- The 2π phase difference between states 1 and 2 indicates a complete phase winding, characteristic of a two-level system.

- The small values of other off-diagonal elements justify neglecting their contributions in the TLA.

6.3   Physical Interpretation

The observed Berry phase structure has important physical implications:

- The system exhibits non-trivial geometric phase accumulation primarily in the subspace spanned by states 1 and 2.

- The antisymmetric nature of the off-diagonal elements ($\gamma_{12} \approx -\gamma_{21}$) is consistent with the expected behavior of a quantum system with time-reversal symmetry.

- The 2π phase difference indicates that the system's parameter space trajectory encloses a topological singularity.

7   Visualization and Analysis Tools

The implementation includes several tools for visualizing and analyzing the results:

7.1   Tau Matrix Evolution

The function `plot_matrix_elements` visualizes the evolution of specific matrix elements of both τ and γ as functions of the parameter θ:

```
1  def plot_matrix_elements(tau, gamma, theta_vals, output_dir):
2      """
3      Plot the evolution of specific matrix elements (01, 12, 13) for
        both tau and gamma matrices.
4      """
5      plt.figure(figsize=(12, 8))
6
7      # Elements to plot
8      elements = [(0, 1), (1, 2), (1, 3)]
9
10     # Plot real and imaginary parts of tau
11     plt.subplot(2, 1, 1)
12     for i, j in elements:
13         plt.plot(theta_vals, np.real(tau[i, j, :]),
14                  label=f'Re($\tau_{i+1}{j+1}$)', linestyle='-')
15         plt.plot(theta_vals, np.imag(tau[i, j, :]),
16                  label=f'Im($\tau_{i+1}{j+1}$)', linestyle='--')
17     plt.xlabel('$\theta$')
18     plt.ylabel('$\tau$')
19     plt.title('Evolution of $\tau$ matrix elements')
20     plt.legend()
21     plt.grid(True)
```

Listing 4: Plotting tau matrix elements

7.2 Eigenvalue and Eigenvector Analysis

The implementation includes classes for analyzing the eigenvalues and eigenvectors of the Hamiltonian:

- `Eigenvalues`: Computes and visualizes the eigenvalues as functions of the parameter θ.

- `Eigenvectors`: Handles the computation, gauge fixing, and visualization of the eigenvectors.

7.3 Verification Tools

The implementation includes tools for verifying the correctness of the calculations:

- `verify_circle_properties`: Verifies that the parameter path forms a perfect circle orthogonal to the $x = y = z$ line.

- Eigenvalue equation verification: Compares $H|\psi_n\rangle$ with $E_n|\psi_n\rangle$ to verify the diagonalization.

8 Conclusion

The tau matrix method provides a powerful and flexible approach for calculating Berry phases and analyzing the geometric properties of quantum systems. The

implementation in the Arrowhead project includes a comprehensive set of tools for computing, visualizing, and analyzing the results.

The numerical results show that the system exhibits non-trivial geometric phases, with strong coupling between specific pairs of states. The structure of the Berry phase matrix supports the use of a two-level approximation and indicates the presence of topological features in the parameter space.

9   References

1. M. V. Berry, *Quantal phase factors accompanying adiabatic changes*, Proc. R. Soc. Lond. A 392, 45-57 (1984)

2. D. J. Thouless, *Topological Quantum Numbers in Nonrelativistic Physics*, World Scientific (1998)

3. B. Simon, *Holonomy, the Quantum Adiabatic Theorem, and Berry's Phase*, Phys. Rev. Lett. 51, 2167 (1983)

4. J. E. Avron, R. Seiler, L. G. Yaffe, *Adiabatic Theorems and Applications to the Quantum Hall Effect*, Commun. Math. Phys. 110, 33-49 (1987)