Supplemental Material to A Real-time Instanton Approach to Quantum Activation

1. Stability Analysis

Using the classical equations of motion we are able to identify steady states of the system and classify them as either saddles or stable points. Our overall goal is to find the switching trajectories which allow the system to escape from the stable points and reach the saddle point, and then calculate the actions of these trajectories with a view to obtain the switching rates. These paths don't exist in the classical case, but they do in the full system including the quantum fields.

In this section we will use the Jacobian matrix to linearise the full equations of motion. This will reveal that: 1. The quantum fields allow for escape trajectories along new unstable eigenvectors leaving the classically stable fixed points. 2. Around the saddle point there is a new stable eigenvector which allows the system to reach the saddle point from the quantum part of the phase space.

This stability analysis will be crucial for setting the boundary conditions for the switching trajectories, as we will see in the next section.

1.1. Equations of Motion

The starting point for our analysis is the full Hamiltonian of the system, which we recall is given by

$$H(x_c, p_c, x_q, p_q) = \left(\delta + \frac{\chi}{2}\left(x_c^2 + p_c^2 - x_q^2 - p_q^2\right)\right)(p_c\, x_q - x_c\, p_q) \\ - \kappa\,(x_c\, x_q + p_c\, p_q) + \kappa\,(x_q^2 + p_q^2) + 2\epsilon\, x_q.$$

The full dynamics are described in terms of the classical and quantum fields by Hamilton's equations:

$$\dot{\mathbf{z}}_c = \frac{\partial H}{\partial \mathbf{z}_q}, \qquad \dot{\mathbf{z}}_q = -\frac{\partial H}{\partial \mathbf{z}_c},$$

where we have defined the field vectors as

$$\mathbf{z}_c = (x_c, p_c), \quad \mathbf{z}_q = (x_q, p_q).$$

For convenience in the next section, we also denote the full state vector by

$$\mathbf{Z} = (x_c, p_c, x_q, p_q).$$

1.2. Linearisation

In order to linearise the equation of motion, we write the state of the system as

$$\mathbf{Z}(t) = \mathbf{Z}_0 + \Delta\mathbf{Z}(t),$$

where

$$\mathbf{Z}_0 = (x_{c0}, p_{c0}, x_{q0}, p_{q0})$$

denotes the coordinates of a fixed point, and $\Delta\mathbf{Z}(t)$ represents a small deviation from that point. Expanding the equations of motion to first order in $\Delta\mathbf{Z}(t)$ yields

$$\frac{d}{dt}\,\Delta\mathbf{Z}(t) = J(\mathbf{Z}_0)\,\Delta\mathbf{Z}(t),$$

where $J(\mathbf{Z}_0)$ is the Jacobian matrix evaluated at $\mathbf{Z}_0$. The Jacobian matrix J, evaluated at a fixed point $\mathbf{Z}_0$, is given by

$$J(\mathbf{Z}_0)_{ij} = \left.\frac{\partial \dot{\mathbf{Z}}_i}{\partial \mathbf{Z}_j}\right|_{\mathbf{Z}=\mathbf{Z}_0}$$

The equations of motion can be written in vector form as

$$\dot{\mathbf{Z}} = (\dot{x}_c, \dot{p}_c, \dot{x}_q, \dot{p}_q) = \left(\frac{\partial H}{\partial x_q}, \frac{\partial H}{\partial p_q}, -\frac{\partial H}{\partial x_c}, -\frac{\partial H}{\partial p_c}\right)$$

and in explicit form, the 4×4 Jacobian matrix is given by:

$$J(\mathbf{Z}_0) = \begin{pmatrix} \frac{\partial^2 H}{\partial x_q \partial x_c} & \frac{\partial^2 H}{\partial x_q \partial p_c} & \frac{\partial^2 H}{\partial x_q^2} & \frac{\partial^2 H}{\partial x_q \partial p_q} \\ \frac{\partial^2 H}{\partial p_q \partial x_c} & \frac{\partial^2 H}{\partial p_q \partial p_c} & \frac{\partial^2 H}{\partial p_q \partial x_q} & \frac{\partial^2 H}{\partial p_q^2} \\ -\frac{\partial^2 H}{\partial x_c^2} & -\frac{\partial^2 H}{\partial x_c \partial p_c} & -\frac{\partial^2 H}{\partial x_c \partial x_q} & -\frac{\partial^2 H}{\partial x_c \partial p_q} \\ -\frac{\partial^2 H}{\partial p_c \partial x_c} & -\frac{\partial^2 H}{\partial p_c^2} & -\frac{\partial^2 H}{\partial p_c \partial x_q} & -\frac{\partial^2 H}{\partial p_c \partial p_q} \end{pmatrix}$$

We can now study the stability of any fixed point by finding the eigenvalues and eigenvectors of this matrix.

1.3. Stability Analysis

Background

The eigenvalues and eigenvectors of the Jacobian $J(\mathbf{Z}_0)$ provide the local stability properties of the fixed point. Specifically, consider the eigenvalue problem

$$J(\mathbf{Z}_0)\,\mathbf{v}_i = \lambda_i\,\mathbf{v}_i.$$

Any small deviation from the fixed point can be expressed as a linear combination of these eigenvectors. In particular, we write

$$\Delta\mathbf{Z}(t) = \sum_{i=1}^{4} c_i\,\mathbf{v}_i\,e^{\lambda_i t},$$

where the coefficients c_i are determined by the initial condition

$$\Delta\mathbf{Z}(0) = \sum_{i=1}^{4} c_i\,\mathbf{v}_i.$$

This expansion illustrates how $\Delta\mathbf{Z}(t)$ evolves over time: each eigenmode evolves as $e^{\lambda t}$, where the real part of λ governs the exponential growth or decay. Specifically, eigenvalues with positive real parts lead to exponential growth of the corresponding perturbations, while those with negative real parts result in exponential decay. Moreover, if an eigenvalue is complex, its imaginary part will introduce oscillatory behavior on top of this exponential trend.

Paired Spectrum

Because the full system is Hamiltonian—with canonical coordinates for the classical and quantum fields—the Jacobian obtained by linearizing the equations of motion is a symplectic matrix, meaning it satisfies:

$$J^T\Omega J = \Omega, \quad \text{where} \quad \Omega = \begin{pmatrix} 0 & I_2 \\ -I_2 & 0 \end{pmatrix}$$

where I_2 is the 2×2 identity matrix. This fact immediately leads to some general expectations regarding its eigenvalues.

For any symplectic matrix, if λ is an eigenvalue, then $-\lambda$ must also be an eigenvalue. Furthmore, any complex eigenvalues also come in conjugate pairs, leading overall to a spectrum that appears in quadruplets: λ, $-\lambda$, λ, and $-\lambda$.

Finally we note that since the Jacobian is not Hermitian, the eigenvectors are not necessarily orthogonal. However, for a symplectic matrix, if $\mathbf{v}$ is a right

eigenvector with eigenvalue λ, then $\Omega\mathbf{v}$ is a left eigenvector with eigenvalue $-\lambda$. This leads to a biorthogonality relation between eigenvectors: if $\mathbf{v}_1$ and $\mathbf{v}_2$ are eigenvectors with eigenvalues λ_1 and λ_2, then $\mathbf{v}_1^T\Omega\mathbf{v}_2 = 0$ unless $\lambda_1 = -\lambda_2$. This structure plays an important role in determining the stable and unstable subspaces around the fixed points.

Stable Points

The classical equations of motion are not Hamiltonian due to the presence of drive and dissipation. When we perform stability analysis around the stable fixed points the eigenvalues of the Jacobian are given either by:

$$\lambda \in \{-\kappa_1, -\kappa_2\}$$

near a saddle-node bifurcation, or by

$$\lambda \in \{-\kappa - i\omega, -\kappa + i\omega\}$$

deeper into the bistable regime. The first case describes a node with different decay rates in the two directions, while the second describes a focus point with a spiralling decaying motion.

When we extend our analysis to include the quantum fields the system becomes Hamiltonian and we find two additional eigenvalues with positive real parts that complete the expected quadruplets. So we find either:

$$\lambda \in \{-\kappa_1, -\kappa_2, \kappa_1, \kappa_2\}$$

near a saddle-node bifurcation, or

$$\lambda \in \{-\kappa - i\omega, -\kappa + i\omega, \kappa - i\omega, \kappa + i\omega\}$$

away from it. Due to the biorthogonality condition above, each eigenvector from the classical sector is paired with an eigenvector that partially lies in the quantum sector. However we note that the quadruplet of eigenvectors is not orthogonal and the new eigenvectors will in general have components in the classical sector as well as the quantum sector.

Saddle Point

Meanwhile, at the saddle point in the classical case the two eigenvalues are given by:

$$\lambda \in \{-\kappa_1, +\kappa_2\}.$$

These describe the incoming and outgoing motion around the saddle point in the classical co-ordinate plane. Again, when we extend our analysis to include the quantum fields these two eigenvalues are paired with two new eigenvalues that describe incoming and outgoing motion along directions in the full four-dimensional phase space:

$$\lambda \in \{-\kappa_1, +\kappa_2, +\kappa_1, -\kappa_2\}.$$

As above, the new eigenvectors obey a biorthogonality condition with the classical eigenvectors and in general will have components in both the classical and quantum sectors.

1.4. Numerical Results

Now that we have an understanding of the eigenvalues and eigenvectors of the Jacobian, we can use this to study the dynamics of the system across parameter space. For this purpose we use the `generate_map.py` and `generate_jacobian_spectrum.py` scripts from [40].

```
from metastable.map.map import FixedPointMap
from metastable.generate_stability_map import generate_stability_map

if __name__ == '__main__':
    # Load the fixed point map
    map = FixedPointMap.load("map.npz")

    # Generate stability map
    map_with_stability = generate_stability_map(map, n_workers=20)

    # Save the updated map
    map_with_stability.save("map-with-stability.npz")
```

The `generate_stability_map` function iterates over the fixed points stored in the `FixedPointMap` object created using `generate_map.py` and computes the eigenvalues and eigenvectors of the Jacobian at each fixed point. The results can then be accessed at the `FixedPointMap.eigenvalues` and `FixedPointMap.eigenvectors` attributes.

Dim Fixed Point

Using a `FixedPointMap` produced at $\chi = -0.1$ and $\delta = 7.8$, we now plot the real and imaginary parts of the λ_0 and λ_1 eigenvalues around the dim fixed point. We also plot the upper and lower bifurcation lines to clearly mark the limits of the bistable regime.

Let's examine the key features of the eigenvalue spectrum:

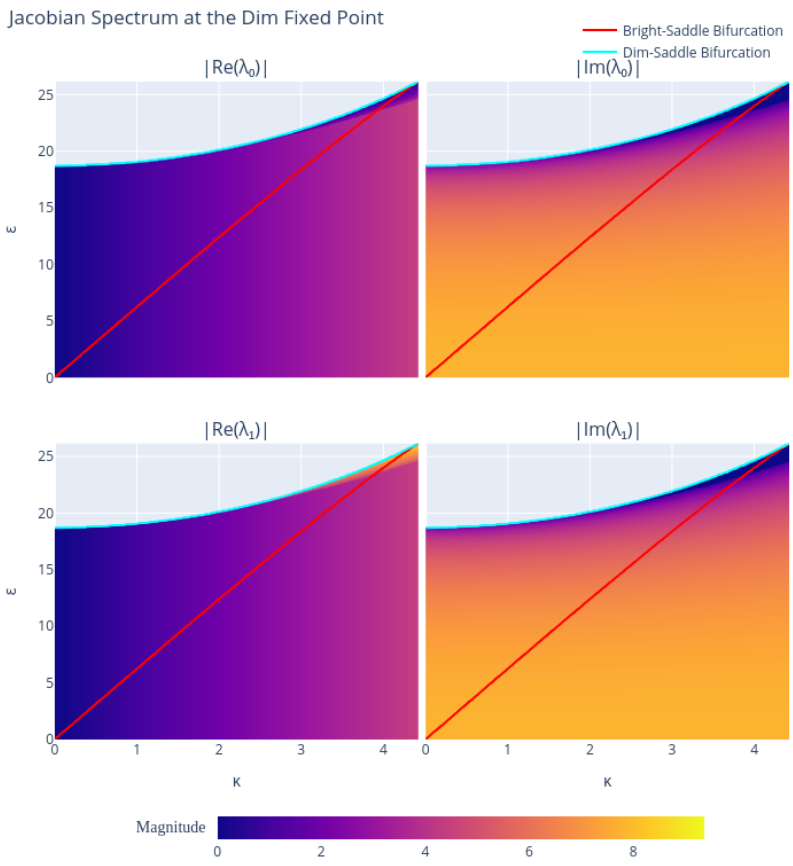


Figure 1: Jacobian spectrum of dim fixed point

1. **Complex Eigenvalues and Oscillatory Motion**: Across most of the parameter space, the eigenvalues form complex conjugate pairs. This indicates oscillatory motion near the fixed points, classifying them as focus points.

2. **Relationship to Decay Rate**: For focus points, the real component's magnitude equals the system decay rate κ. As $\kappa \to 0$, while the real component vanishes, the imaginary component persists. This suggests that oscillatory motion occurs on a much faster timescale than decay or escape processes.

3. **Behavior Near Bifurcation**: Near the dim-saddle bifurcation point, the eigenvalues become purely real, transforming the fixed point into a node. At this transition:

    - The previously equal real parts take on different values.
    - One eigenvalue approaches zero while the other remains finite.
    - The vanishing eigenvalue reveals a soft mode connecting the fixed point to the saddle point This regime corresponds to the one-dimensional Kramers problem discussed in [13].

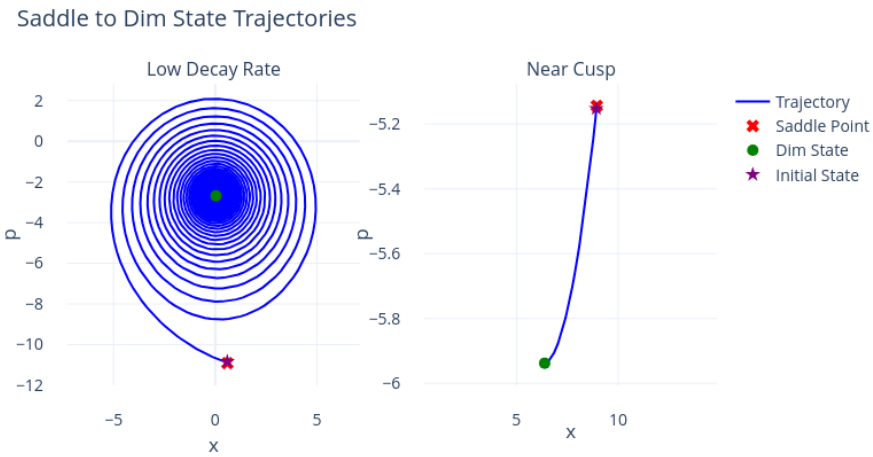
These points can be illustrated in more detail by examining specific trajectories near the fixed points:

In the figure above we plot two classical trajectories of the system as it decays from the saddle point to the dim fixed point. In the top-left panel we choose to examine the system at small decay rate ($\kappa = 0.1$, $\epsilon = 10.0$) deep within the bistable regime. As mentioned above, this leads to a focus point with a decaying spiralling motion. The small value of the decay rate causes the motion to be almost circular.

This will pose a significant challenge for us when we attempt to find escape trajectories. The eigenvectors of the Jacobian which allow escape are part of the same quadruplet as ingoing eigenvectors which control decaying motion. Just as the decaying path has significant spiralling motion with slow relaxation to the fixed point, the escaping trajectories will have significant spiralling motion with slow escape rate. The widely differing timescales and the accumulation of numerical errors will make it challenging to find the exact path going from the fixed point to the saddle point.

In the top-right panel we plot the same trajectory close to the cusp point where the bistable regime closes ($\kappa = 4.3$, $\epsilon = 25.6$). The motion is quite different. We now see that the dim fixed point is a node and the system moves almost linearly from one point to the other. This makes it much easier to find the exact trajectory.

We also include a table of the eigenvalues and parameters, as well as the bottom panel to show where the top panels lie within the bistable regime.



Operating Point	State	λ_0	λ_1
Low Decay Rate κ=0.1, ε=10.0	Dim Saddle	-0.100+7.068j 4.200+0.000j	-0.100-7.068j -4.400+0.000j
Near Cusp κ=4.3, ε=25.6	Dim Saddle	-0.503+0.000j 0.198+0.000j	-8.097+0.000j -8.798+0.000j

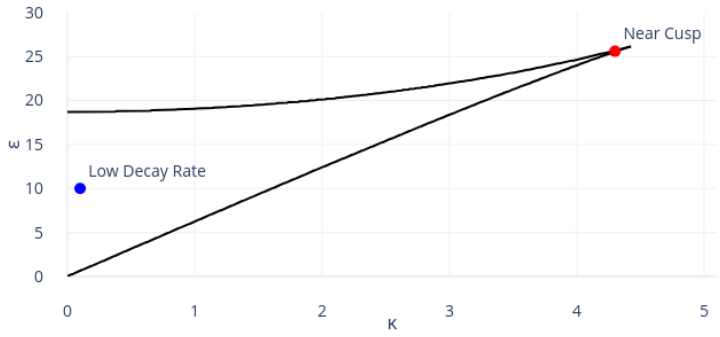


Figure 2: Trajectories and bifurcation

Bright Fixed Point

Next we move on to the bright fixed point. As above we observe complex eigenvalues corresponding to focus points in the majority of the parameter space. At small decay rates the real components of the eigenvalues are close to zero and the motion is almost circular. Just as above we also see that close to the bifurcation point the imaginary components of the eigenvalues vanish and the fixed points become nodes.

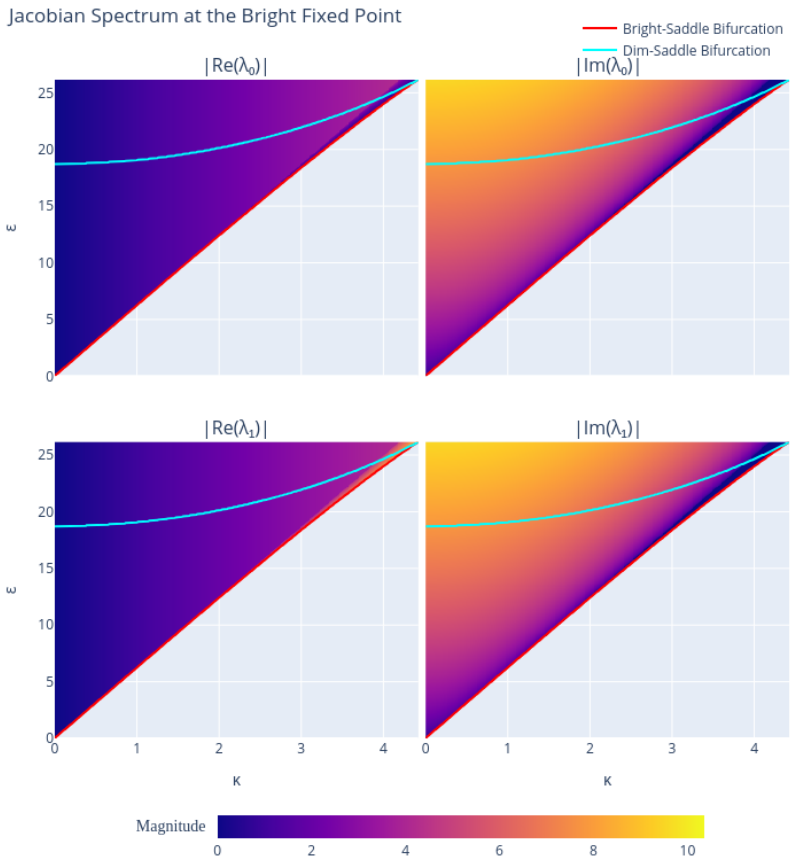


Figure 3: Jacobian spectrum of bright fixed point

Saddle Point

Next, we examine the eigenvalue spectrum around the saddle point. Unlike the stable fixed points, the saddle point has real eigenvalues of opposite signs throughout the bistable regime, reflecting its unstable nature. This is consistent

with our earlier theoretical analysis where we predicted eigenvalues of the form $\lambda \in \{-\kappa_1, +\kappa_2, +\kappa_1, -\kappa_2\}$.

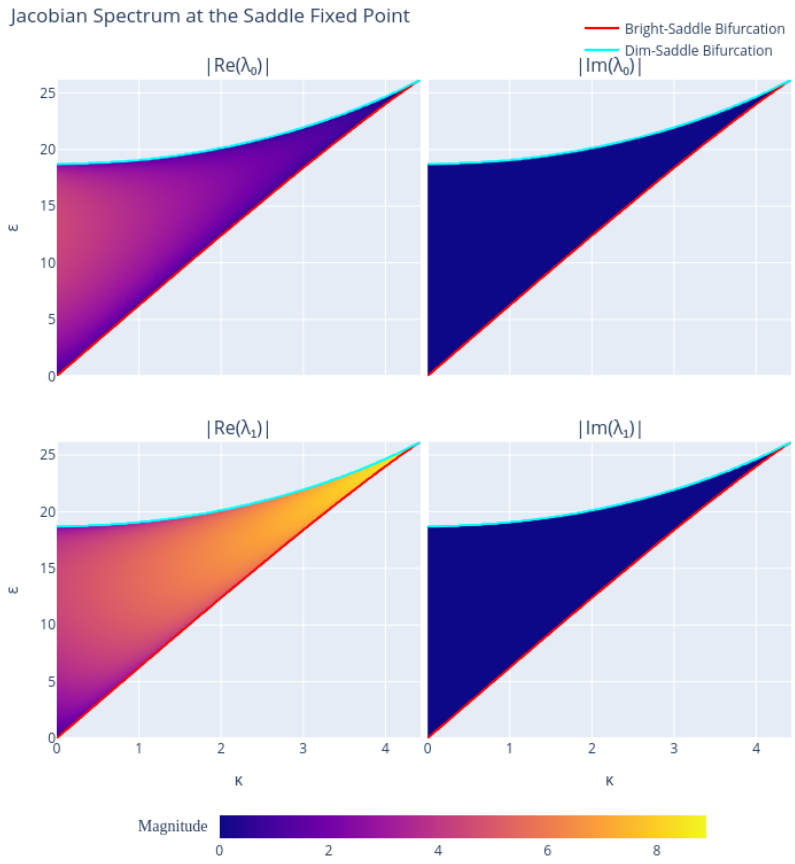


Figure 4: Jacobian spectrum of saddle fixed point

The key features of the saddle point spectrum are:

1. **Pure Real Eigenvalues**: Unlike the stable fixed points which typically show complex conjugate pairs, the saddle point maintains real eigenvalues throughout the parameter space, indicating pure exponential growth or decay without oscillations.

2. **Symmetry in Magnitude**: The eigenvalues appear in pairs of equal magnitude but opposite sign, reflecting the Hamiltonian nature of the full system and ensuring conservation of phase space volume.

3. **Bifurcation Behavior**: Near the bifurcation points, one pair of eigenvalues approaches zero while the other pair remains finite. This corresponds

to the merging of the saddle point with one of the stable fixed points at the bifurcation.

Regimes

We note that the Jacobian spectra of both the dim and bright fixed points indicate that we will struggle to find escape trajectories at small decay rates. In the figure below we plot the ratio of the imaginary to real parts of the eigenvalues for the dim and bright fixed points. As expected, the ratio diverges as the decay rate approaches zero, but this can be counteracted to some extent by staying close to a bifurcation point.

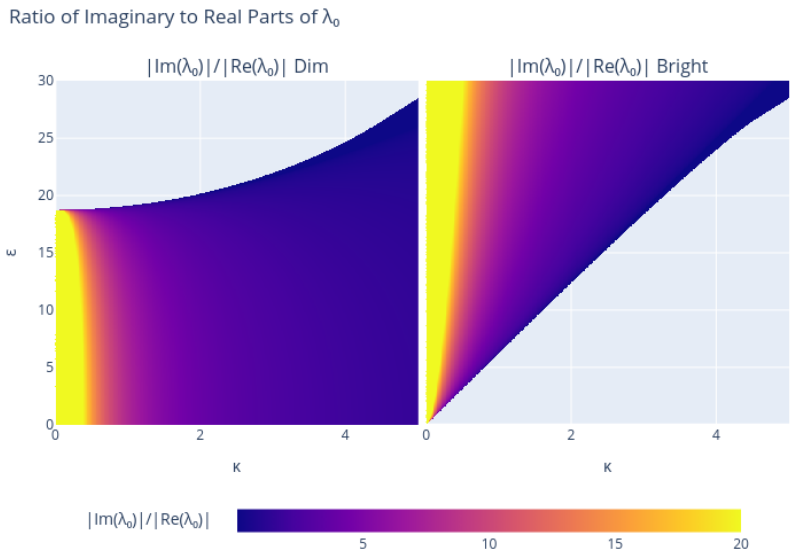


Figure 5: Instability ratios

Finally we plot the ratio of the eigenvalues of the Jacobian at the saddle point to illustrate the onset of the soft mode.

The vanishing of the λ_0 eigenvalue at the bifurcation point is indicative of the vanishing of the force pushing the system from the saddle point to the node as they merge. # 2. Switching Paths

In this section we can now turn our attention to finding the switching paths connecting the stable fixed points to the saddle point. We will use our prevoius stability analysis to set the boundary conditions for the switching paths. We will then use a collocation method implemented in SciPy [41] to solve the resulting boundary value problem and find the switching paths.

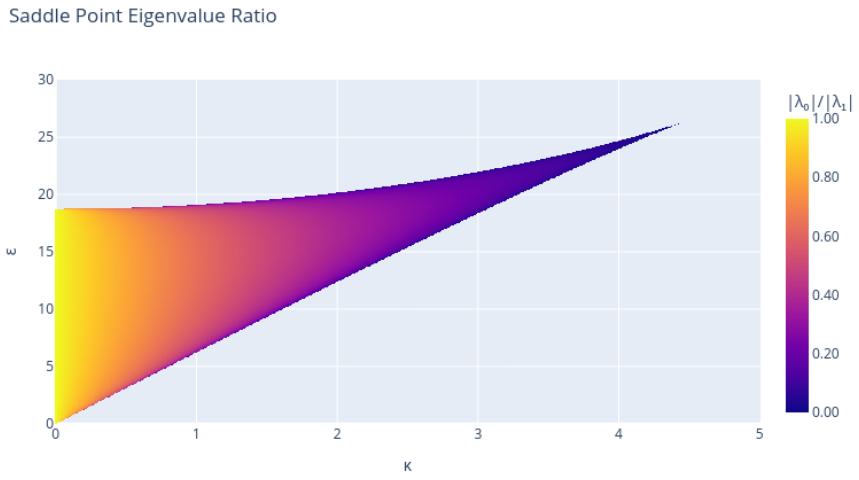


Figure 6: Saddle eigenvalue ratio

2.1. Problem Overview

2.1.1. Path

Our key objective is to find a path $\mathbf{Z}(t)$ that connects a **stable fixed point** $\mathbf{Z}_0$ (node or focus) to a **saddle point** $\mathbf{Z}_s$.

The key idea is to use the eigenvectors of the Jacobian at both points to set the boundary conditions. Small deviations from the fixed points can be expressed in the eigenvector basis of the Jacobian and we control the boundary conditions by specifying the coefficients of the eigenvectors in those deviations.

At each fixed point the Jacobian has a pair of incoming and outgoing eigenvectors. We simply apply the condition that at the start of the path the deviation is purely along the eigenvectors leaving the stable point and at the end of the path the deviation is purely along the eigenvectors arriving at the saddle point.

2.1.2. Action

With these paths we can finally calculate the corresponding actions using:

$$S_{\text{aux}} = -\int dt \Big[\dot{x}_c\, x_q + \dot{p}_c\, p_q - H(x_c, p_c, x_q, p_q)\Big].$$

Since the Hamiltonian is conservative (i.e. time independent) and the initial and final points are in the classical plane where $H = 0$, the action can be simplified to:

$$S_{\text{aux}} = -\int dt \Big[\dot{x}_c \, x_q + \dot{p}_c \, p_q \Big].$$

2.2. Mathematical Framework

2.2.1. At the Stable Fixed Point

Let $\mathbf{Z}_0$ denote the coordinates of the stable fixed point. Small deviations from $\mathbf{Z}_0$ can be expressed in the eigenvector basis of the Jacobian $J(\mathbf{Z}_0)$:

$$\Delta \mathbf{Z}(t) \equiv \mathbf{Z}(t) - \mathbf{Z}_0 = \sum_{i=1}^{4} c_i \, \mathbf{v}_i \, e^{\lambda_i t},$$

where: - $\mathbf{v}_i$ are the eigenvectors, - λ_i are the corresponding eigenvalues, ordered by their real parts with $\text{Re}(\lambda_1), \text{Re}(\lambda_2) < 0$ (stable) and $\text{Re}(\lambda_3), \text{Re}(\lambda_4) > 0$ (unstable), - c_i are coefficients determined by the initial displacement.

2.2.2. At the Saddle Point

- Let $\mathbf{Z}_s$ denote the coordinates of the saddle point.
- Near $\mathbf{Z}_s$, the deviation is similarly expressed in terms of the eigenvectors of the Jacobian $J(\mathbf{Z}_s)$:

$$\Delta \mathbf{Z}(t) \equiv \mathbf{Z}(t) - \mathbf{Z}_s = \sum_{j=1}^{4} d_j \, \mathbf{u}_j \, e^{\mu_j t},$$

where: - $\mathbf{u}_j$ are the eigenvectors at the saddle, - μ_j are the corresponding eigenvalues, ordered by their real parts with $\text{Re}(\mu_1), \text{Re}(\mu_2) < 0$ (stable) and $\text{Re}(\mu_3), \text{Re}(\mu_4) > 0$ (unstable), - d_j are coefficients that characterize the deviation.

2.3. Formulating the Boundary Value Problem

2.3.1. Physical Interpretation of Boundary Conditions

The switching trajectory represents the optimal (least-action) path connecting a metastable fixed point to the saddle point. It is a solution to the equations of motion derived from the auxiliary Hamiltonian. Our stability analysis then provides natural boundary conditions:

Stable Fixed Point $\mathbf{Z}_0$

At $t \to -\infty$ (approaching the stable fixed point) we have:

$$\mathbf{Z}(t) \to \mathbf{Z}_0 \quad \Longrightarrow \quad \Delta\mathbf{Z}(t) \approx \sum_i c_i \, \mathbf{v}_i \, e^{\lambda_i t},$$

Since the trajectory must *depart* within the unstable subspace, the initial deviation $\Delta\mathbf{Z}(t)$ aligns with eigenvectors having $\mathrm{Re}(\lambda_i) > 0$ and we can set $c_0 = 0$ and $c_1 = 0$.

Depending on exactly which initial direction we choose, the system will follow a different path. Our goal is to the path which connects to the saddle point.

Saddle Point $\mathbf{Z}_s$

At $t \to +\infty$ (approaching the saddle point):

$$\mathbf{Z}(t) \to \mathbf{Z}_s \quad \Longrightarrow \quad \Delta\mathbf{Z}(t) \approx \sum_j d_j \, \mathbf{u}_j \, e^{\mu_j t},$$

Since the trajectory must *arrive* within the stable subspace, the final deviation $\Delta\mathbf{Z}(t)$ lies in the subspace of eigenvectors with $\mathrm{Re}(\mu_j) < 0$. Therefore we can set $d_2 = 0$ and $d_3 = 0$. If we have chosen the correct initial trajectory, the system will follow a path that naturally arrives at the saddle point within this subspace.

2.3.2 Projection

To enforce our asymptotic boundary conditions at finite times t_i and t_f, we must project the deviations onto the eigenbasis of the Jacobian at the fixed points. This is achieved by using the left eigenvectors to extract the coefficients in the expansion of the deviation in terms of the right eigenvectors.

Let $\Delta Z(t)$ be the deviation from a fixed point (either Z_0 or Z_s). Suppose the right eigenvectors are arranged in a matrix R and the corresponding left eigenvectors (normalized such that $L_i \cdot R_j = \delta_{ij}$) are arranged in a matrix L. Then the expansion

$$\Delta Z(t) = \sum_i c_i v_i \iff \Delta Z(t) = Rc$$

allows us to extract the coefficients by projecting onto the left eigenvectors:

$$c = L\Delta Z(t).$$

At the stable fixed point, for instance, we only want deviations along the unstable directions. This requirement translates into setting the coefficients corresponding to the stable modes to zero:

$$c_i = L_i \cdot [Z(t) - Z_0] = 0 \text{ for stable modes.}$$

Similarly, at the saddle point, to ensure the trajectory approaches along the stable manifold, the coefficients for the unstable directions must vanish:

$$d_j = L_j \cdot [Z(t) - Z_s] = 0 \text{ for unstable modes.}$$

By enforcing these constraints at t_i and t_f, the finite-time boundary value problem inherits the correct asymptotic behavior, ensuring that the switching trajectory departs and arrives in the proper subspaces.

2.3.3. Numerical Solution

The task of finding the switching paths is now cast as a boundary value problem (BVP). Our next task is apply a numerical method to find a solution. For this we will use SciPy's `solve_bvp` [41] implementation of a collocation method, meaning a numerical technique for solving differential equations by approximating the solution using a set of basis functions, such as cubic splines, and enforcing the differential equations are satisfied to a given tolerance at specific points called collocation points.

Convergence

When applying this solver to our problem there are two key considerations to check convergence of the solution:

1. **Finite Time Domain**: Since we cannot numerically integrate from $t \to -\infty$ to $t \to +\infty$, we must choose finite initial and final times t_i and t_f. These should be chosen such that the system is sufficiently close to the fixed points at the boundaries.
2. **Numerical Parameters**: The key parameter for solution quality is the error tolerance, which caps the allowed residuals at the collocation points [41]. Lower error tolerances can be reached by increasing the number of collocation points, at the cost of increased computational time and memory usage.

Whether or not they are we have converged to a desired solution can be judged by whether or not the action along the path has reached to a stable value. This can be checked by recalculating the paths and actions using different values of $t_f - t_i$, error tolerances and numbers of collocation points and checking if the resulting actions are consistent.

Initial Guess

The BVP solver requires an initial guess for the solution. This can be constructed by linear interpolation between the fixed points. This initial guess is most effective

near the saddle-node bifurcations where the stable and saddle points are closest to each other. For more distant points with more complex paths, this linear guess may not converge to a solution, in which case we can reuse solutions from neighbouring points in the parameter space (numerical continuation).

2.3.4. Implementation Example

Here's an example of how to implement the switching paths calculation in Python using our codebase [40]:

```
from pathlib import Path
from metastable.map.map import FixedPointMap, FixedPointType
from metastable.map.visualisations.bifurcation_lines import plot_bifurcation_diagram
from metastable.paths import (
    get_bistable_kappa_range,
    generate_sweep_index_pairs,
    map_switching_paths
)
from metastable.action.map import map_actions
from metastable.paths.visualization import plot_parameter_sweeps


# Load the fixed point map
map_path = Path("fixed_points/examples/map-with-stability.npz")
fixed_point_map = FixedPointMap.load(map_path)

# Create the bifurcation diagram
fig = plot_bifurcation_diagram(fixed_point_map)

# Choose an epsilon index for the kappa cut
epsilon_idx = 380

# Get the bistable kappa range for this epsilon
kappa_boundaries = get_bistable_kappa_range(fixed_point_map.bistable_region, epsilon_idx)

# Generate kappa cuts
kappa_cuts = generate_sweep_index_pairs(kappa_boundaries, bright_sweep_fraction=0.4, dim_swe

# Get the actual epsilon value from index
epsilon_value = fixed_point_map.epsilon_linspace[epsilon_idx]

# Plot the bistable range
if kappa_boundaries.dim_saddle is not None:
    kappa_start = fixed_point_map.kappa_linspace[kappa_boundaries.dim_saddle.kappa_idx]
    kappa_end = fixed_point_map.kappa_linspace[kappa_boundaries.bright_saddle.kappa_idx]
    fig.add_scatter(
```

```
        x=[kappa_start, kappa_end],
        y=[epsilon_value, epsilon_value],
        mode='markers',
        marker=dict(size=10, color='red'),
        name='Bistable Range ($\kappa$)'
    )

# Plot parameter sweeps
fig = plot_parameter_sweeps(fixed_point_map, kappa_cuts, fig)

output_path = Path("sweep")

# Map switching paths for bright fixed point
path_results_bright = map_switching_paths(
    fixed_point_map,
    kappa_cuts.bright_saddle,
    output_path,
    t_end=10.0,
    endpoint_type=FixedPointType.BRIGHT,
    max_nodes=1000000,
    tol=1e-3
)

# Map switching paths for dim fixed point
path_results_dim = map_switching_paths(
    fixed_point_map,
    kappa_cuts.dim_saddle,
    output_path,
    t_end=10.0,
    endpoint_type=FixedPointType.DIM,
)

# Calculate actions for all switching paths
fixed_point_map = FixedPointMap.load(output_path / "map.npz")
fixed_point_map_with_actions = map_actions(fixed_point_map)
fixed_point_map_with_actions.save(output_path / "map.npz")
```

Key aspects of the implementation:

1. **Parameter Selection**: We first choose a fixed value of ϵ (using index 380) and find the bistable range of κ values.
2. **Visualization Setup**: The code creates a bifurcation diagram and visualizes the parameter sweep regions.
3. **Path Calculation**: Two separate calls to `map_switching_paths` calculate:

- Paths from bright fixed points to saddle points
- Paths from dim fixed points to saddle points

4. **Numerical Parameters**:
   - `t_end=10.0`: Integration time domain (from -10.0 to 10.0)
   - `tol=1e-3`: Error tolerance for the bright-to-saddle paths
   - `max_nodes=1000000`: Maximum number of collocation points for the bright paths
5. **Action Calculation**: After finding the paths, the corresponding actions are calculated using `map_actions`.

This implementation allows us to systematically explore the switching paths across different regions of parameter space and visualize the results together with the bifurcation structure.

2.4. Results

We continue to examine the system studied in the stability analysis. We begin by finding the switching paths and actions as a function of κ at fixed $\epsilon/\delta = 2.44$. The results are shown in the interactive visualization below, which consists of two panels:

1. **Upper Panel (Bifurcation Diagram)**: Shows the bifurcation structure in the $(\kappa/\delta, \epsilon/\delta)$ plane. The red and blue lines represent the unstable-bright and unstable-dim bifurcation boundaries respectively. The horizontal dashed black line indicates our chosen $\epsilon/\delta = 2.44$ cut, and the grey shaded region indicates the bistable region where switching paths exist.
2. **Lower Panel (Action Values)**: Displays the calculated actions for the switching paths:
   - Red line: Keldysh action $R_{b\to u}$ for the bright-to-unstable transition
   - Blue line: Keldysh action $R_{d\to u}$ for the dim-to-unstable transition
   - Purple dash-dot line: Kramers (analytical) prediction for $R_{b\to u}$
   - Green dash-dot line: Kramers (analytical) prediction for $R_{d\to u}$

The actions are measured relative to the scaled Planck cosntant $\lambda = \chi/\delta$. The close agreement between the numerically computed Keldysh actions and the analytical Kramers predictions [13, 14] validates our numerical approach close to the bifurcation points. Deeper into the bistable regime we see that the two methods diverge indicating the breakdown of the 1-D approximation on which the Kramers approach is based.

The switching paths were computed using boundary conditions that ensure proper alignment with the stable and unstable manifolds at each fixed point, with thresholds set to 10^{-3} for both stable and saddle points. The numerical integration was performed over a finite time domain $(t_f - t_i)\delta = 78.0$, which proved sufficient for convergence of the action values.

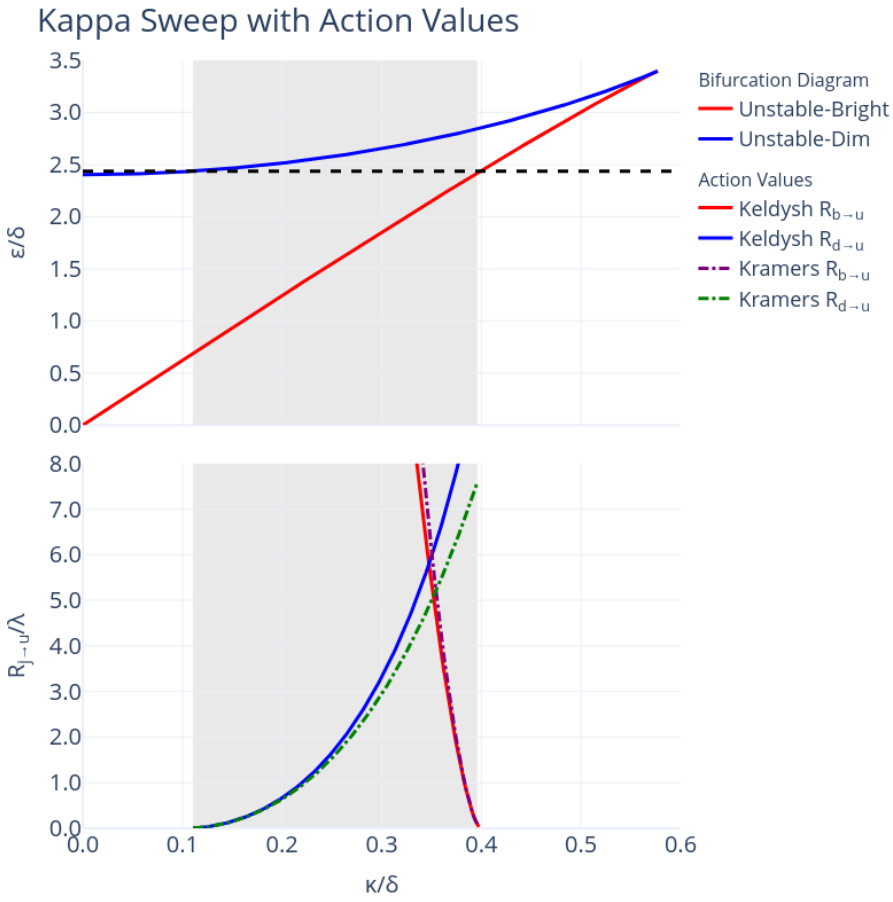


Figure 7: Kappa sweep with actions

Then as a function of ϵ at fixed $\kappa/\delta = 0.240$. The results are shown in the interactive visualization below, which consists of two panels:

1. **Upper Panel (Bifurcation Diagram)**: Shows the bifurcation structure in the $(\kappa/\delta, \epsilon/\delta)$ plane. The red and blue lines represent the unstable-bright and unstable-dim bifurcation boundaries respectively. The vertical dashed black line indicates our chosen $\kappa/\delta = 0.240$ cut, and the grey shaded region indicates the bistable region where switching paths exist.

2. **Lower Panel (Action Values)**: Displays the calculated actions for the switching paths:

    - Red line: Keldysh action $R_{b\to u}$ for the bright-to-unstable transition
    - Blue line: Keldysh action $R_{d\to u}$ for the dim-to-unstable transition
    - Purple dash-dot line: Kramers (analytical) prediction for $R_{b\to u}$
    - Green dash-dot line: Kramers (analytical) prediction for $R_{d\to u}$

As above, the agreement between the numerically computed Keldysh actions and the analytical Kramers predictions helps to validate our numerical approach.

The switching paths were computed using boundary conditions that ensure proper alignment with the stable and unstable manifolds at each fixed point, with thresholds set to 10^{-2} for both stable and saddle points. The numerical integration was performed over a finite time domain $(t_f - t_i)\delta = 85.8$, which proved sufficient for convergence of the action values close to the bifurcation points. However in the middle of the bistable regime convergence was not achieved so actions are not plotted here.

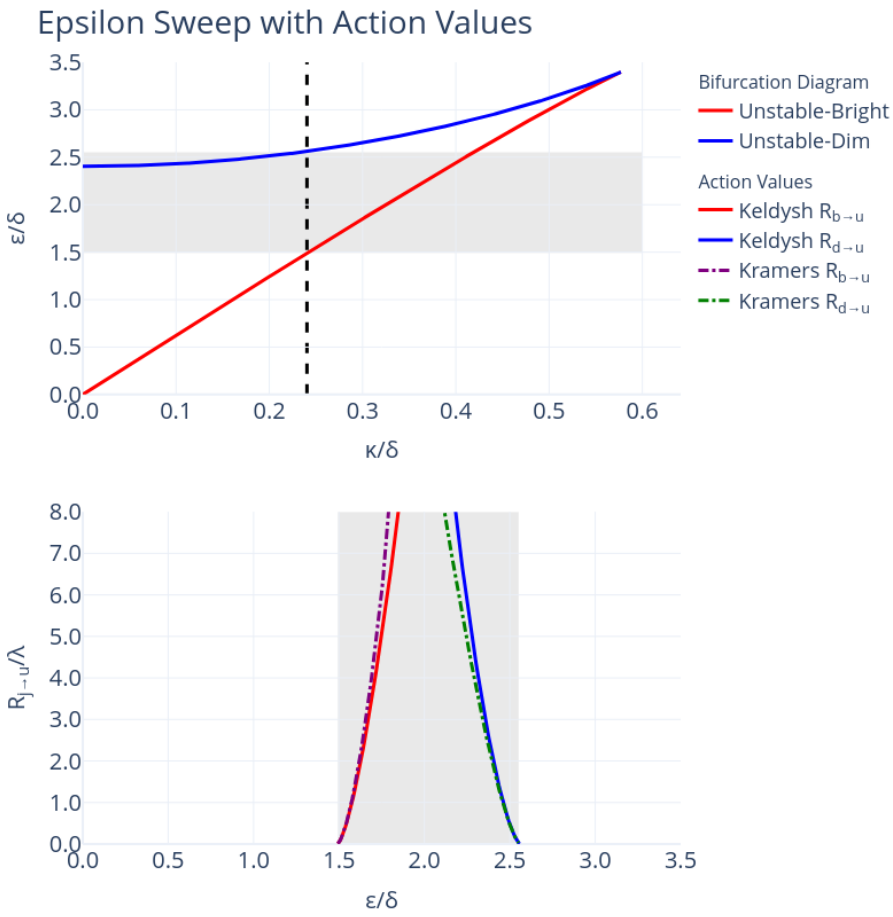


Figure 8: Epsilon sweep with actions