

¹ PyDMF: A Python package for variational
² double-ended reaction-path and transition-state
³ optimization

⁴ Shin-ichi Koda  ^{1,2} and Shinji Saito  ^{1,2}

⁵ Institute for Molecular Science, National Institutes of Natural Sciences, Okazaki, 444-8585, Japan ²
⁶ Graduate University for Advanced Studies, SOKENDAI, Okazaki, 444-8585, Japan 
⁷ author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

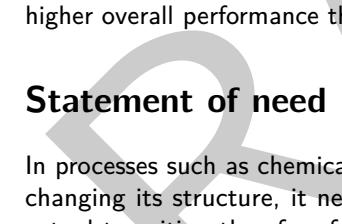
- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: 

Submitted: 04 December 2025

Published: unpublished

License

Authors of papers retain copyright
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](#))

Summary

Identifying accurate reaction paths and transition states is essential for understanding structural changes in molecular systems. PyDMF is a Python package that implements two recent methods for reaction-path optimization: the direct MaxFlux method (Koda & Saito, 2024b), which improves computational efficiency through a variational formalism, and the flat-bottom elastic network model (Koda & Saito, 2024a, 2025), which improves the applicability of reaction-path optimization by generating chemically plausible initial paths. Through integration with the Atomic Simulation Environment, PyDMF can use a wide range of atomistic simulation software to evaluate energies along a path. Benchmark studies demonstrate that PyDMF achieves higher overall performance than existing reaction-path optimization methods.

Statement of need

In processes such as chemical reactions, where a material moves between stable states while changing its structure, it necessarily passes through energetically unstable structures. The actual transition therefore follows a path that minimizes the rise in energy. Understanding such structural transitions requires characterizing the reaction path, particularly the transition state that forms the energy maximum along it. Because transition states are transient, they are difficult to observe experimentally and are usually obtained computationally. As a result, locating transition states with standard atomistic simulation software is a fundamental task for both theoretical and experimental researchers.

Double-ended methods, a major class of approaches for locating reaction paths and transition states, optimize the entire path between two given states. Their main advantage is that only the endpoints must be specified, eliminating the need to guess the unknown transition-state structure. Well-established techniques such as the nudged elastic band (NEB) method (Henkelman et al., 2000; Henkelman & Jónsson, 2000) and the string method (E et al., 2002) are implemented in many atomistic simulation programs and are widely used. However, these methods face two key limitations: they require energy evaluations for many discrete structures along the path, reducing computational efficiency, and they are sensitive to the choice of the initial path, which limits their applicability. Overcoming these issues is crucial for accelerating computational studies.

We recently proposed two methods that substantially alleviate these limitations: the direct MaxFlux method (DMF) (Koda & Saito, 2024b) and the flat-bottom elastic network model (FB-ENM) (Koda & Saito, 2024a, 2025). DMF, based on a variational formalism, locates the region near the transition state using only a small number of energy evaluation points,

41 greatly reducing the computational cost. FB-ENM generates chemically plausible initial paths
42 by enforcing constraints that exclude nonchemical structures, enabling reliable construction of
43 energetically favorable pathways.

44 PyDMF provides Python implementations of both methods ([Koda, 2026](#)). Because reaction-
45 path optimization is a fundamental component of studies across chemistry, physics, and
46 materials science, PyDMF offers an efficient and robust framework that improves the practical
47 accessibility of transition-state searches.

48 State of the field

49 As noted above, existing double-ended optimization methods such as NEB are implemented
50 internally in many atomistic simulation programs. The Atomic Simulation Environment (ASE)
51 ([Larsen et al., 2017](#)), a Python package that provides a unified interface to many atomistic
52 simulation programs, also provides its own implementations of NEB and several of its variants.
53 In addition, the image-dependent pair potential (IDPP) method ([Smidstrup et al., 2014](#)) for
54 generating initial paths is available in software such as ORCA and ASE.

55 From a theoretical standpoint, DMF implemented in PyDMF differs fundamentally from
56 the existing approaches. DMF is based on a variational formulation in which reaction-path
57 optimization is expressed as a well-defined minimization problem with an explicit objective
58 function. In contrast to NEB or string methods, which rely on non-variational schemes,
59 DMF can directly leverage general-purpose nonlinear optimization algorithms. In practice,
60 PyDMF employs the state-of-the-art optimizer IPOPT ([Wächter & Biegler, 2006](#)) via its
61 Python interface cyipopt, enabling efficient and robust optimization without introducing
62 method-specific path-update algorithms.

63 This design choice necessarily introduces a dependency on external optimization libraries.
64 For this reason, PyDMF was developed as a standalone software package rather than as a
65 contribution to an existing framework. Requiring DMF-specific optimizer dependencies, which
66 also require the use of conda, for all users of an integrated framework such as ASE would
67 impose unnecessary constraints on users who do not require DMF functionality. Implementing
68 PyDMF as an independent package therefore provides a practical separation of concerns,
69 allowing advanced variational reaction-path optimization capabilities to be offered exclusively
70 to users who need them.

71 Software design

72 PyDMF is designed around a clear separation between reaction-path optimization algorithms
73 and energy evaluation backends. Its core design principle is to focus on the implementation of
74 reaction-path optimization methods, while delegating energy and force evaluations to existing
75 atomistic simulation software. ASE provides a well-established abstraction layer that enables
76 this separation, and PyDMF inherits this abstraction by directly interfacing with ASE rather
77 than reimplementing backend-specific functionality.

78 Through its integration with ASE, PyDMF can access a broad range of electronic-structure
79 and force-field engines for transition-state searches. Well-established packages such as VASP,
80 Quantum ESPRESSO, CP2K, ORCA, Gaussian, GAMESS, LAMMPS, Amber, and GROMACS
81 can be used as backends without any modification to PyDMF. This design choice preserves
82 backend flexibility while allowing PyDMF to integrate smoothly into diverse atomistic modeling
83 workflows. Usage examples and API documentation are available in the project's GitHub
84 repository ([Koda, 2026](#)).

85 In PyDMF, the implementation focuses on defining the nonlinear optimization problem, while
86 the actual optimization is performed using powerful external libraries. This design choice
87 introduces certain disadvantages, such as a strong reliance on conda-based environments and

88 the resulting difficulty of inclusion within frameworks such as ASE. Nevertheless, this trade-off
89 was made deliberately, prioritizing improved performance and robustness of transition-state
90 searches over minimizing external dependencies.

91 Research impact statement

92 The research impact of PyDMF is supported by both quantitative benchmark results and
93 its integration into user-facing computational workflows. The efficiency and robustness of
94 DMF and FB-ENM relative to existing reaction-path optimization approaches have been
95 demonstrated in benchmark studies reported in their original publications ([Koda & Saito,](#)
96 [2024b, 2024a, 2025](#)). Using a dataset of 121 representative chemical reactions involving typical
97 elements ([Ásgeirsson et al., 2021](#)), DMF reduced the computational cost by approximately
98 70% compared with conventional NEB calculations. FB-ENM was shown to generate more
99 energetically favorable reaction paths than IDPP and to produce chemically plausible initial
100 paths even for complex reactions where IDPP often fails.

101 Beyond these benchmarks, PyDMF has been adopted as a core component in higher-level
102 computational workflows. Notably, it serves as the transition-state search engine in
103 ColabReaction, a web-based application for transition-state searches ([Karasawa et al., 2025](#)).
104 This integration demonstrates that PyDMF is not only a methodological implementation
105 but also a practical and reusable software component that supports accessible and reliable
106 transition-state searches.

107 AI usage disclosure

108 ChatGPT was used during the development of PyDMF and the preparation of this manuscript.
109 For the source code, suggested code snippets were used specifically for implementing parts of
110 the MPI parallelization, as well as for guidance on Python package directory structure and
111 the preparation of `pyproject.toml`. All such code was reviewed and tested by the authors.
112 ChatGPT was also used interactively to assist in drafting the manuscript, the GitHub repository
113 `README`, and the API documentation. In all cases, the final content was written, verified,
114 and approved by the authors.

115 Acknowledgements

116 This work has been supported by JSPS KAKENHI, Grant Number JP22K14652 (S.-i.K.),
117 JP21H04676, and JP23K17361 (S.S.). The software development and benchmarking were
118 performed using Research Center for Computational Science, Okazaki, Japan (Project: 23-
119 IMS-C196, 24-IMS-C193, and 25-IMS-C223).

120 References

- 121 Ásgeirsson, V., Birgisson, B. O., Bjornsson, R., Becker, U., Neese, F., Ripplinger, C., & Jónsson,
122 H. (2021). Nudged elastic band method for molecular reactions using energy-weighted
123 springs combined with eigenvector following. *Journal of Chemical Theory and Computation*,
124 17(8), 4929–4945. <https://doi.org/10.1021/acs.jctc.1c00462>
- 125 E, W., Ren, W., & Vanden-Eijnden, E. (2002). String method for the study of rare events.
126 *Physical Review B*, 66(5), 052301. <https://doi.org/10.1103/physrevb.66.052301>
- 127 Henkelman, G., & Jónsson, H. (2000). Improved tangent estimate in the nudged elastic band
128 method for finding minimum energy paths and saddle points. *The Journal of Chemical
129 Physics*, 113(22), 9978–9985. <https://doi.org/10.1063/1.1323224>

- 130 Henkelman, G., Uberuaga, B. P., & Jónsson, H. (2000). A climbing image nudged elastic band
131 method for finding saddle points and minimum energy paths. *The Journal of Chemical*
132 *Physics*, 113(22), 9901–9904. <https://doi.org/10.1063/1.1329672>
- 133 Karasawa, M., Leow, C. S., Yajima, H., Arai, S., Nishizaki, H., Terada, T., & Sato, H. (2025).
134 ColabReaction: Accelerating transition state searches with machine learning potentials on
135 google colaboratory. *Journal of Chemical Information and Modeling*, 65(21), 11908–11914.
136 <https://doi.org/10.1021/acs.jcim.5c02398>
- 137 Koda, S. (2026). PyDMF: A python package for double-ended reaction path/transition state
138 optimization. In *GitHub repository*. GitHub. <https://github.com/shin1koda/dmf>
- 139 Koda, S., & Saito, S. (2024a). Flat-bottom elastic network model for generating improved
140 plausible reaction paths. *Journal of Chemical Theory and Computation*, 20(16), 7176–7187.
141 <https://doi.org/10.1021/acs.jctc.4c00792>
- 142 Koda, S., & Saito, S. (2024b). Locating transition states by variational reaction path
143 optimization with an energy-derivative-free objective function. *Journal of Chemical Theory*
144 *and Computation*, 20(7), 2798–2811. <https://doi.org/10.1021/acs.jctc.3c01246>
- 145 Koda, S., & Saito, S. (2025). Correlated flat-bottom elastic network model for improved bond
146 rearrangement in reaction paths. *Journal of Chemical Theory and Computation*, 21(7),
147 3513–3522. <https://doi.org/10.1021/acs.jctc.4c01549>
- 148 Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M.,
149 Friis, J., Groves, M. N., Hammer, B., Hargus, C., Hermes, E. D., Jennings, P. C.,
150 Jensen, P. B., Kermode, J., Kitchin, J. R., Kolsbjerg, E. L., Kubal, J., Kaasbjerg, K.,
151 Lysgaard, S., ... Jacobsen, K. W. (2017). The atomic simulation environment—a Python
152 library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27), 273002.
153 <https://doi.org/10.1088/1361-648x/aa680e>
- 154 Smidstrup, S., Pedersen, A., Stokbro, K., & Jónsson, H. (2014). Improved initial guess for
155 minimum energy path calculations. *The Journal of Chemical Physics*, 140(21), 214106.
156 <https://doi.org/10.1063/1.4878664>
- 157 Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter
158 line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*,
159 106(1), 25–57. <https://doi.org/10.1007/s10107-004-0559-y>