

Software update

Version 2.0 - cashocs: A Computational, Adjoint-Based Shape Optimization and Optimal Control Software

Sebastian Blauth

Fraunhofer ITWM, Kaiserslautern, Germany

ARTICLE INFO

MSC:

65K05

49M05

49M41

35Q93

49Q10

Keywords:

PDE constrained optimization

Shape optimization

Topology optimization

Space mapping

ABSTRACT

In this paper, we present version 2.0 of cashocs. Our software automates the solution of PDE constrained optimization problems for shape optimization and optimal control. Since its inception, many new features and useful tools have been added to cashocs, making it even more flexible and efficient. The most significant additions are a framework for space mapping, the ability to solve topology optimization problems with a level-set approach, the support for parallelism via MPI, and the ability to handle additional (state) constraints. In this software update, we describe the key additions to cashocs, which is now even better-suited for solving complex PDE constrained optimization problems.

Refers to

Sebastian Blauth, cashocs: A Computational, Adjoint-Based Shape Optimization and Optimal Control Software, SoftwareX, 13 (2021), p. 100646, <https://doi.org/10.1016/j.softx.2020.100646>.

Code metadata

Current code version	v2.0.12
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00390
Code Ocean compute capsule	na
Legal Code License	GNU GPL v3.0 (or later)
Code versioning system used	git
Software code languages, tools, and services used	Python, FEniCS, NumPy, PETSc, MPI, meshio, Gmsh
Compilation requirements, operating environments & dependencies	FEniCS, meshio, Gmsh
If available Link to developer documentation/manual	https://cashocs.readthedocs.io/
Support email for questions	sebastian.blauth@itwm.fraunhofer.de

1. Description of the software-update

This article serves as update to our software cashocs [1] and discusses some of the key additions, changes, and improvements to the software. We describe both new functionalities which have been added to cashocs to extend its capabilities as well as structural improvements which make the software more efficient and easier to apply.

2. New functionalities

Since its inception, cashocs has received many new functionalities, which we present in the following. For a detailed description of all changes and additions to cashocs, we refer to the release notes at https://cashocs.readthedocs.io/en/stable/release_notes/. One of the most practical additions to cashocs are space mapping methods which

DOI of original article: <https://doi.org/10.1016/j.softx.2020.100646>.

E-mail address: sebastian.blauth@itwm.fraunhofer.de.

<https://doi.org/10.1016/j.softx.2023.101577>

Received 16 June 2023; Received in revised form 13 October 2023; Accepted 2 November 2023

Available online 10 November 2023

2352-7110/© 2023 The Author(s). Published by Elsevier B.V. All rights reserved.

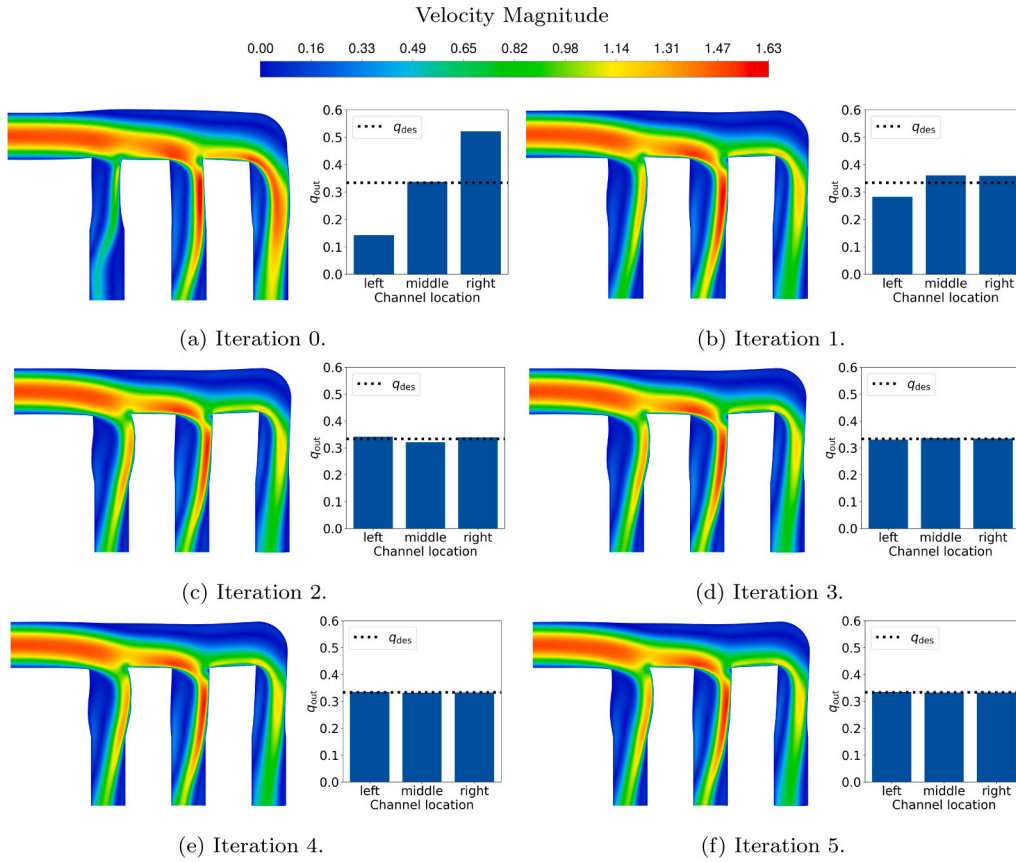


Fig. 1. Example for the space mapping method for shape optimization from [4]. The picture shows the evolution of the geometry, velocity field, and outlet flow rates over the space mapping iterations.

enable the solution of highly complex problems in the following way: The space mapping technique utilizes a model hierarchy consisting of a fine (detailed, complex) and a coarse (approximate, cheap) model. It provides an efficient way to optimize the fine model by successively optimizing and correcting a sequence of coarse model approximations. Particularly, there is no need for a direct optimization of the fine model. This makes the technique particularly interesting for industrial applications, where often commercial solvers without optimization capabilities are used for simulation. For more details on the space mapping technique, we refer the reader to, e.g., [2–4]. Moreover, we note that corresponding tutorials for using the space mapping technique in cashocs can be found at https://cashocs.readthedocs.io/en/stable/user/demos/shape_optimization/demo_space_mapping_semilinear_transmission/ and https://cashocs.readthedocs.io/en/stable/user/demos/shape_optimization/demo_space_mapping_uniform_flow_distribution/.

Whereas space mapping techniques for optimal control problems have already been investigated in the literature (see, e.g., [5–7]), space mapping methods for shape optimization have only been introduced very recently in [4]. Moreover, to the best of our knowledge, cashocs is the only software providing a framework for implementing and solving general space mapping problems in the context of PDE constrained optimization for both shape optimization and optimal control problems.

As a reference, we briefly take a look at a problem from [4], which is solved with the space mapping methods for shape optimization and is depicted in Fig. 1. The aim of the optimization problem is to achieve a uniform flow distribution over three outlet pipes. For the fine model, the Navier–Stokes equations with a Reynolds number of 1000 are used and solved with Ansys Fluent [8], whereas the coarse model is given by the linear Stokes system and is solved on a coarser grid with FEniCS [9–11]. We see that the space mapping

method converges really fast in about five iterations and that the optimized geometry yields a nearly perfect uniform flow distribution. For more details, we refer the reader to [4] and to the corresponding tutorial at https://cashocs.readthedocs.io/en/stable/user/demos/shape_optimization/demo_space_mapping_uniform_flow_distribution/, where the corresponding tutorial can be found.

Another major new feature for cashocs is its support for solving topology optimization problems with a level-set method. Our software provides several optimization algorithms for such problems, including state-of-the-art algorithms for topology optimization [12] as well as novel quasi-Newton methods for topology optimization proposed in [13]. Due to the inherent difficulty of deriving topological derivatives, the latter cannot yet be computed with automatic differentiation methods but have to be supplied by the user. However, the automatic derivation of adjoint systems is already implemented in cashocs. This feature extends the applicability of cashocs to the field of topology optimization with topological sensitivity information. For more details on how to solve topology optimization problems with cashocs, we refer the reader to the corresponding tutorials at https://cashocs.readthedocs.io/en/stable/user/demos/topology_optimization/.

Further, cashocs has received methods for the automatic treatment of additional constraints for optimization problems, such as state or control constraints. In particular we have implemented a quadratic penalty and an augmented Lagrangian method (see, e.g., [14]) for dealing with constraints other than PDEs. These methods are applicable for a wide variety of constraints, including state constraints, which opens up new problem classes for cashocs to solve. We refer the reader to https://cashocs.readthedocs.io/en/stable/user/demos/optimal_control/demo_constraints/, where a demo for using additional constraints with cashocs can be found. An overview over all types of optimization problems that can be solved with cashocs is given in Fig. 2.

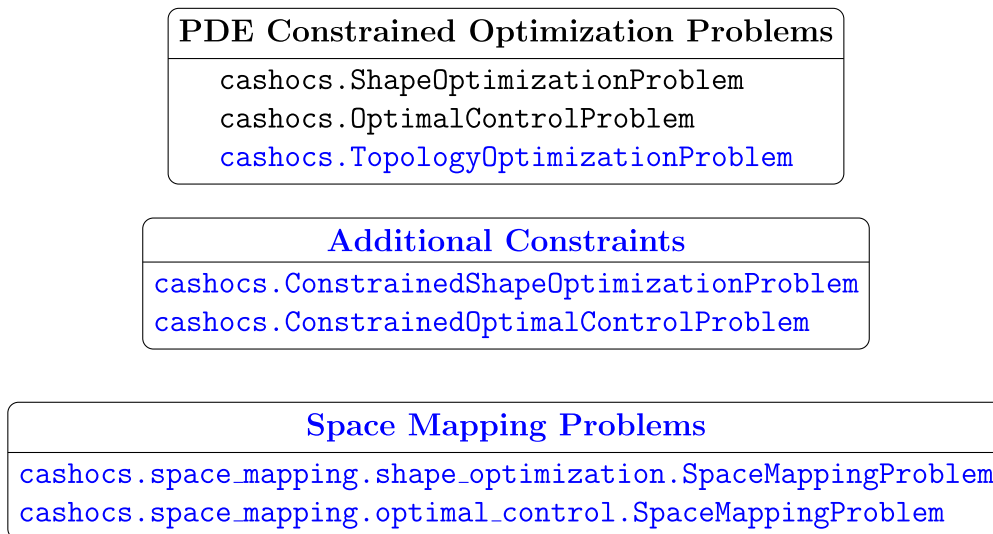


Fig. 2. Overview of the types of optimization problems that can be solved with cashocs. Items colored in blue are new additions to version 2.0.

The new version of cashocs also comes with the ability to automatically scale individual terms of the cost functional based on their magnitude for the initial iteration. This makes it easier for users to weigh different parts of the cost functional according to their needs. Moreover, this also facilitates the solution of multi-criteria optimization problems with scalarization methods, see, e.g., [15,16]. We refer the reader to the corresponding cashocs tutorial at https://cashocs.readthedocs.io/en/stable/user/demos/shape_optimization/demo_scaling/.

With the new version of cashocs, users now have the possibility to define custom scalar products for the optimization, which is particularly useful for shape optimization, where the choice of the scalar product for computing the shape gradient is particularly important. The reader is referred to the cashocs tutorials at https://cashocs.readthedocs.io/en/stable/user/demos/shape_optimization/demo_custom_scalar_product/ for more information. Moreover, the updated version of cashocs also supports computing the shape gradient with the p -Laplace equations, based on the approach presented in [17]. This approach may preserve the mesh quality and can yield better optimized geometries for certain problems. For more details, the reader is referred to the corresponding demo at https://cashocs.readthedocs.io/en/stable/user/demos/shape_optimization/demo_p_laplacian/.

Finally, we have also added more sophisticated line search methods to cashocs, which make use of polynomial models to efficiently compute a suitable stepsize. During the line search, a quadratic or cubic model of the cost functional along the search direction is formed and minimized to obtain suitable stepsizes in fewer iterations compared to a classical Armijo backtracking search (see, e.g., [18]).

3. Structural improvements

In addition to the new features described above, there have also been some major structural improvements to cashocs which we discuss in the following.

The most important structural improvement is the support for parallelism with MPI. In particular, most of the PDE constrained optimization problems that can be solved in serial can now also be solved in parallel without the need for any code modifications. Therefore, cashocs is now able to be run on high-performance-computing systems and, thus, can solve huge optimization problems which cannot be treated in serial. The MPI implementation builds on and leverages the MPI support of FEniCS, so that there are only minimal code changes necessary on the user's side.

Another structural improvement for cashocs is a new remeshing workflow which overcomes the limitations induced by the previous one. The benefit of the new workflow is that remeshing can now be applied also for the space mapping and constraint handling problems, which was not possible beforehand. As remeshing is of particular importance for solving shape optimization problems, this change greatly increases the applicability of cashocs. However, user's must now follow a slightly different, but still straightforward, syntax to solve shape optimization problems with remeshing. The corresponding tutorial can be found at https://cashocs.readthedocs.io/en/stable/user/demos/shape_optimization/demo_remeshing/.

Finally, we note that the remeshing workflow itself is not parallelized with MPI due to limitations of the used software tools Gmsh [19], which supports only OpenMP parallelism, and meshio [20], which does not support parallelism at all. This can potentially create bottlenecks when solving problems where lots of remeshing is required. However, as usually the solution of the state and adjoint systems is the most CPU and memory consuming task, the bottleneck of the remeshing is not too severe for many applications.

4. Conclusion and outlook

With the update to version 2.0, our software cashocs has gained significant new features, such as a space mapping framework, support for parallelism via MPI, and the ability to treat topology optimization problems. These additions make cashocs even more flexible and relevant for solving PDE constrained optimization problems for practical and industrial applications.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

- [1] Blauth S. cashocs: A Computational, Adjoint-Based Shape Optimization and Optimal Control Software. *SoftwareX* 2021;13:100646. <http://dx.doi.org/10.1016/j.softx.2020.100646>.
- [2] Bakr MH, Bandler JW, Madsen K, Søndergaard J. An Introduction to the Space Mapping Technique. *Optim Eng* 2001;2(4):369–84. <http://dx.doi.org/10.1023/A:1016086220943>, Special issue on surrogate modeling and space mapping for engineering optimization (Lyngby, 2000).
- [3] Echeverría D, Hemker PW. Space Mapping and Defect Correction. *Comput Methods Appl Math* 2005;5(2):107–36. <http://dx.doi.org/10.2478/cmam-2005-0006>.
- [4] Blauth S. Space Mapping for PDE Constrained Shape Optimization. *SIAM J Optim* 2023;33(3):1707–33. <http://dx.doi.org/10.1137/22M1515665>.
- [5] Hintermüller M, Vicente LN. Space Mapping for Optimal Control of Partial Differential Equations. *SIAM J Optim* 2005;15(4):1002–25. <http://dx.doi.org/10.1137/S105262340342907X>.
- [6] Marheineke N, Pinnau R. Model hierarchies in space-mapping optimization: Feasibility study for transport processes. *J Comput Methods Sci Eng* 2012;12(1–2):63–74. <http://dx.doi.org/10.3233/JCM-2012-0404>.
- [7] Totzeck C, Pinnau R. Space mapping-based receding horizon control for stochastic interacting particle systems: dogs herding sheep. *J Math Ind* 2020;10. <http://dx.doi.org/10.1186/s13362-020-00077-1>, Paper No. 11, 19.
- [8] Ansys, Inc. *Ansys® Fluent User's Guide, Release 2022 R1*. 2022.
- [9] Alnæs MS, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, et al. The FEniCS Project Version 1.5. *Arch Numer Softw* 2015;3(100). <http://dx.doi.org/10.11588/ans.2015.100.20553>.
- [10] Logg A, Mardal K-A, Wells GN, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer; 2012, <http://dx.doi.org/10.1007/978-3-642-23099-8>.
- [11] Logg A, Wells GN. DOLFIN: Automated finite element computing. *ACM Trans Math Softw* 2010;37(2). <http://dx.doi.org/10.1145/1731022.1731030>, Art. 20, 28.
- [12] Amstutz S, Andrä H. A new algorithm for topology optimization using a level-set method. *J Comput Phys* 2006;216(2):573–88. <http://dx.doi.org/10.1016/j.jcp.2005.12.015>.
- [13] Blauth S, Sturm K. Quasi-Newton methods for topology optimization using a level-set method. *Struct Multidiscip Optim* 2023;66(9):203. <http://dx.doi.org/10.1007/s00158-023-03653-2>.
- [14] Nocedal J, Wright SJ. *Numerical Optimization*. Springer series in operations research and financial engineering, 2nd ed.. Springer, New York; 2006, <http://dx.doi.org/10.1007/978-0-387-40065-5>, xxii+664.
- [15] Ehrgott M. *Multicriteria Optimization*. 2nd ed.. Berlin: Springer-Verlag; 2005, <http://dx.doi.org/10.1007/3-540-27659-9>, xiv+323.
- [16] Blauth S, Baldan M, Osterroth S, Leithäuser C, Apfel U-P, Kleinhaus J, et al. Multi-Criteria Shape Optimization of Flow Fields for Electrochemical Cells. 2023, [arXiv:2309.13958](https://arxiv.org/abs/2309.13958).
- [17] Müller PM, Kühl N, Siebenborn M, Deckelnick K, Hinze M, Rung T. A novel p -harmonic descent approach applied to fluid dynamic shape optimization. *Struct Multidiscip Optim* 2021;64(6):3489–503. <http://dx.doi.org/10.1007/s00158-021-03030-x>, 4342704.
- [18] Kelley CT. *Iterative Methods for Optimization*. *Frontiers in Applied Mathematics*, Vol. 18, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; 1999, <http://dx.doi.org/10.1137/1.9781611970920>, xvi+180.
- [19] Geuzaine C, Remacle J-F. *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*. *Internat J Numer Methods Engrg* 2009;79(11):1309–31. <http://dx.doi.org/10.1002/nme.2579>, 2566786.
- [20] Schlömer N. meshio: Tools for mesh files. <http://dx.doi.org/10.5281/zenodo.1173115>, <https://github.com/nschloe/meshio>.