Plan Overview

A Data Management Plan created using DMPonline

Title: Fundamental Investigation of Hairy Surface Effect for Enhanced Fluid Flow Control in Hydraulic Systems

Creator: Petr Konas

Principal Investigator: Jan Vimmr, Pavel Rudolf

Data Manager: Simona Fialová

Affiliation: Other

Funder: Other

Template: DCC Template

ORCID iD: 0000-0003-3311-4592

ORCID iD: 0000-0003-2622-7898

Project abstract:

Friction reduction in rotating mechanical components is a critical challenge in engineering, particularly in high-speed applications such as bearings, shafts, and seals. Conventional approaches to minimizing friction include polymer brush coatings and surface texturing, both of which enhance lubrication performance and reduce direct asperity contact. Polymer brushes, consisting of polymer chains tethered to surfaces, create a lubricious layer that lowers frictional forces, making them suitable for bearing applications. Similarly, micro-scale surface texturing has been shown to improve lubrication by trapping lubricants and controlling their distribution, thereby reducing friction and wear.

This research aims to bridge critical gaps in the understanding of fluid-filled hairy surfaces by combining experimental testing and computational modeling. Our primary contribution lies in systematically analyzing how hair density, flexibility, and fluid retention influence friction reduction in rotating components. By optimizing these surface properties, we seek to develop advanced lubrication strategies that enhance efficiency and minimize energy losses in highspeed rotating systems. Given the widespread need for improved tribological performance in engineeing applications, our work provides a crucial step toward more effective and sustainable friction-reducing technologies.

ID: 174536

Start date: 01-01-2026

End date: 01-01-2028

Last modified: 28-03-2025

Fundamental Investigation of Hairy Surface Effect for Enhanced Fluid Flow Control in Hydraulic Systems

Data Collection

What data will you collect or create?

- **Data types**
  - Simulation results (numerical values, parameters).
  - Training logs and evaluation metrics.
  - Documentation and metadata for reproducibility.
  - Visualizations (graphs, plots).
  - Experimental data

  Data formats
  - npz, csv, txt, png, jpg, pdf, ps, eps, tex, py, tar.gz, tgz

  Volume of data
  - several units of MB for source files
  - approximately 2 TB for training data (rough estimation)

  Use and Reuse potential

  Chosen formats and software usually enable sharing and long access of data. There is some exception for validation data which is usually realized on closed commercial software. Open Science principles will be followed to ensure the dataset, models, and results are reusable for researchers working on python, optimization, reduced models,...

  Data description
  - Generated from custom simulations using scientific computing frameworks (e.g., TensorFlow, PyTorch, JAX).
    Pre-existing datasets from open repositories when necessary (e.g., CFD datasets, PDE solutions).

  FAIR Principles Compliance
  - **Findable:**

  Metadata will be generated using standard formats (e.g., JSON, YAML). All datasets and code will have Digital Object Identifiers (DOIs) via Zenodo. Repository URLs and versions will be documented.

  - **Accessible:**

  Code and datasets will be openly available on public repositories (see Section 5). Data will be accessible for download without restrictions. Licensing will ensure proper attribution.

  - **Interoperable:**

  Standard formats such as CSV, NPZ and HDF5 will be used for numerical data. Python-based machine learning scripts will follow best practices to ensure cross-platform compatibility. Metadata will adhere to standards for metadata annotation (JSON, YAML).

  - **Reusable:**

  Open-source licenses will apply (Apache 2.0 or MIT for code, Creative Commons for data). The repository will include README files with full documentation. Jupyter Notebooks will provide examples for reproducibility.

  Only open data with no third-party obligations or any confidential agreements will be published.

How will the data be collected or created?

- Data will be generated using common numerical simulations in PyTorch/JAX, TensorFlow, Scikit-Learn,...
- Pre-existing datasets from public sources will be processed and used when relevant.
- Workflows will be documented in GitHub repositories.
- Scripts for data preprocessing, training, and evaluation of models will be provided.

Documentation and Metadata

What documentation and metadata will accompany the data?

A comprehensive README.md file will be included in all repositories (GitHub, Zenodo, Kaggle) and will provide:

Project Overview: A brief description of the research, objectives, and the dataset's purpose.
Dataset Description: Explanation of the files, structure, and contents.
File Formats: List of formats used (e.g., CSV, HDF5, JSON, YAML) and how they should be interpreted.
Data Collection & Processing: Description of how the data was obtained, preprocessed, and transformed.
Usage Instructions: Steps on how to load, analyze, and use the dataset.
Dependencies: List of required software libraries (e.g., PyTorch, NumPy, Pandas).
License Information: Details about the dataset's licensing (CC BY 4.0 for data; MIT/Apache 2.0 for code).
How to Cite the Work: DOI reference (if published on Zenodo).

A structured metadata file in JSON, YAML, or Dublin Core XML format will accompany the dataset, containing the common items e.g.:

Title: "Investigation of the properties and behaviour of magnetically sensitive bodies and fluids for the design of advanced bearing elements"
Description: A summary of the dataset, including its purpose and key features.
Creators: Names, affiliations, and ORCID IDs of the researchers.
Versioning: Version number of the dataset.
Keywords: Relevant terms (e.g., "Physics-Informed Neural Networks", "Neural Operators", "Numerical Simulation", "Machine Learning for PDEs").
Creation Date: When the dataset was generated.
Data Sources: If external data was used, proper attribution will be given.
File Structure: Hierarchical organization of data files.
Variable Definitions: Description of columns and variables in CSV/HDF5 files (e.g., parameters of differential equations, loss metrics).
Licensing: Open Science licensing information.

Ethics and Legal Compliance

How will you manage any ethical issues?

- No personally identifiable information (PII) or sensitive data is included in this research.
- Compliance with Open Science policies to promote transparency and reproducibility.

How will you manage copyright and Intellectual Property Rights (IPR) issues?

Data published under FAIR principle will be considered the following IPRs
Code: MIT or Apache 2.0 License.
Data & Models: Creative Commons Attribution 4.0 (CC BY 4.0).
Documentation: Creative Commons Attribution 4.0 (CC BY 4.0).

Storage and Backup

How will the data be stored and backed up during the research?

- Primary storage will be GitHub Repository (for source code and small datasets).
  Large datasets will be stored on Zenodo, Kaggle and in CESNET repositories (for extra large during computational tasks).
- Google Drive or Microsoft cloud services will be used for temporary storage and backup.
- Regular backups are planned to prevent data loss.

How will you manage access and security?

- Public repositories will be used whenever possible to promote Open Science.
- Sensitive data is not expected in this project just now.
- GitHub repositories will have regular access controls and read/write permissions for collaborators.

Selection and Preservation

Which data are of long-term value and should be retained, shared, and/or preserved?

Raw and Processed Datasets
Raw Data:
Original simulation outputs (numerical results of PDEs, physical field values).
Saved in HDF5, CSV, or NPZ formats for accessibility and interoperability.
Stored in Zenodo, Kaggle, or institutional repositories with proper versioning.
Processed Datasets:

Cleaned, formatted data used for training models.
Includes feature engineering results, normalized datasets, and preprocessed inputs.

What is the long-term preservation plan for the dataset?

- Zenodo will be used for long-term archiving of datasets and research artifacts.
- Regular updates and issue tracking will be maintained in GitHub repositories.
- Repositories and publication DOIs will ensure persistent access.

Data Sharing

How will you share the data?

Source Code: GitHub (https://github.com)
Repository with version control and documentation.
Datasets: Zenodo (https://zenodo.org) - For long-term storage and dataset persistence with DOI.
Kaggle (https://www.kaggle.com) - For publicly sharing datasets with an interactive environment.
Jupyter Notebooks and Documentation: GitHub/GitHub Pages.
Preprints and Publications:
arXiv (https://arxiv.org) - For publishing preprints related to project.

Are any restrictions on data sharing required?

- Code, datasets, models, and documentation will be publicly available on repositories like GitHub, Zenodo, and Kaggle.
- Licensing ensures proper citation but allows unrestricted reuse (e.g., MIT for code, CC BY 4.0 for data).
- No confidential, personal, or sensitive data is involved, so full public sharing is permitted.
- The dataset and models are intended for scientific research and educational purposes only.
- A disclaimer will be included in the documentation, stating that users are responsible for ethical use.
- If any third-party datasets (e.g., pre-existing PDE datasets) are used they must be licensed for open use and cited accordingly. Any restrictions on these external datasets will be clearly stated in the dataset metadata.

Responsibilities and Resources

Who will be responsible for data management?

Simona Fialová will be in role of Data Manager or Data Steward

What resources will you require to deliver your plan?

Storage and computing resources

Resource Type	Details	Purpose
Cloud Storage	Github	Version control, sharing source code, documentation, metadata
	Zenodo	Long-term dataset and model preservation
	Kaggle	Public dataset hosting with built-in tools for exploration
	Google drive/Institutional Cloud	Backup and temporary storage of large files before final transfer
HPC	Institutional HPC cluster or cloud services (e.g., Google Colab, AWS, or local GPU servers)	Running large-scale simulations, training
Local Storage	At least 1-2 TB disk space	Temporary storage of raw and processed datasets before archival

Software and Tools

Tool/Platform	Purpose	Open Source
Programming	Python (PyTorch, TensorFlow, JAX)	Yes
Version control	Git, GitHub	Yes
Dataset management	Pandas, NumPy, HDF5, NetCDF for structured data handling	Yes
Metadata, Documentation	YAML, JSON for structured metadata files	Yes
Notebook Environment	Jupyter Notebooks for reproducibility and tutorials	Yes
Workflow Automation	Node-Red / Snakemake / Makefiles for dataset and model pipeline automation	Yes
Data Archiving	Zenodo API	Yes
Visualization	Matplotlib, Seaborn, Plotly for data visualization	Yes

Funding and budget

Cost Category	Estimated Requirements
Storage	Free (Github, Zenodo, Kagle) or 100EUR/year for institutional cloud storage
HPC/GPU	Free (Institutional Computing) or 5000EUR/year for Public Cloud Computing
Personal training	Optional workshops on FAIR principles, Github principles to keep up2date (1000 EUR for training)
DOI registration	Free