Classical MD E-CAM Modules V

E-CAM Deliverable 1.6
Deliverable Type: Report
Delivered in March 2021

E-CAM
The European Centre of Excellence for
Software, Training and Consultancy
in Simulation and Modelling

Funded by the European Union under grant agreement 676531

Project and Deliverable Information

Project Title	E-CAM: An e-infrastructure for software, training and discussion in simulation and modelling
Project Ref.	Grant Agreement 676531
Project Website	https://www.e-cam2020.eu
EC Project Officer	Juan Pelegrín
Deliverable ID	D1.6
Deliverable Nature	Report
Dissemination Level	Public
Contractual Date of Delivery	Project Month 62(30th November, 2020)
Actual Date of Delivery	31st March, 2021
Description of Deliverable	Final Software modules delivered to the E-CAM repository in the area of Classical Molecular Dynamics responding to requests of users, and their documentation.

Document Control Information

Document	Title:	Classical MD E-CAM Modules V
	ID:	D1.6
	Version:	As of 31st March, 2021
	Status:	Accepted by WP leader
	Available at:	https://www.e-cam2020.eu/deliverables
	Document history:	Internal Project Management Link
Review	Review Status:	Reviewed
Authorship	Written by:	David W.H. Swenson (École Normale Supérieure de Lyon)
	Contributors:	Pascal Carrivain (École Normale Supérieure de Lyon), Donal Mac Kernan (University College Dublin), Sander Roet (Norwegian University of Science and Technology), Andreas Singraber (University of Vienna), Ana Catarina Mendonça (EPFL)
	Reviewed by:	Christoph Dellago (University of Vienna)
	Approved by:	Christoph Dellago (University of Vienna)

Document Keywords

Keywords:	E-CAM, CECAM, Molecular Dynamics, Rare Events, Path Sampling, OpenPathSampling, neural network potentials

31st March, 2021

***Disclaimer**:This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements.*

[1] dwhs@hyperblazer.net

Contents

Executive Summary

In this report for Deliverable 1.6 of E-CAM, 9 software modules in classical dynamics are presented. These modules represent improvements and new features in response to the needs of users.

These modules represent a wide range of topics in classical molecular dynamics, including developments in software for rare event sampling, for developing neural network potentials, for modeling polymer dynamics, for the optimized design and in-silico validation of fusion proteins especially sensors, and for general analysis of MD trajectories. Modules have been developed that interact with OpenPathSampling, n2p2, ODE, and LAMMPS, as well as a new, independent package called Dask-Traj. Many of these modules are designed to leverage the performance and scalability of other tools. In particular, a number of these module interface with or extend LAMMPS, which has excellent scalability. Additionally, there has been a focus in these module on high throughput (task-based) computing as a paradigm to enhance scalability, in particular using the Python package Dask.

The 9 modules presented here are:

1. SimStore: OPS new storage subsystem (core)
2. SimStore: Storable functions
3. dask-traj
4. minDist2segments_KKT_for_SRP
5. velocities_resolve_EVC_for_LAMMPS
6. n2p2 - Improved link to HPC MD software
7. n2p2 - Polynomial Symmetry Functions
8. lammps_pyinterfaceExt
9. PIUtils

Each module is thoroughly tested, includes in-code documentation as well as external documentation, frequently in the form of Jupyter notebook examples.

Section 1 of this report gives a brief description of E-CAM modules and the role of this deliverable in the broader goals of E-CAM Work Package 1 (WP1). Section 2 provides background material on the contexts shared between multiple modules: rare events and applications path sampling, simulation setup tools, and neural network potentials. In section 3, we describe each of the modules and provide links to their documentation. Section 4 describes performance aspects of these modules, and section 6 summarizes the deliverable and describes the outlook for future development of modules within WP1, including the increasing importance of transverse actions across simulation communities and work packages.

As this deliverable is the last one of its series, a section on the overall impact of the results achieved within the Workpackage WP1 Classical Molecular Dynamics was also included in section 5.

1 Introduction

Notwithstanding the exponential increase in computing power over the last few decades and the development of efficient molecular dynamics algorithms, many processes are still beyond the reach of simulation. Challenges that remain are frequently related to the difficulties of long-time sampling. Many problems, such as nucleation and protein folding, require long waiting times before seeing the transition of interest occur. The time scale problem can also manifest because the underlying dynamics are computationally expensive, as when *ab initio* dynamics are required to describe reactive behavior. Addressing such time scale problems and developing scientific software able to overcome them has been one of the central goals of Work Package 1 (WP1) of the E-CAM-Project. It does so by providing academic and industrial users the means to address such questions using open source software with verified quality standards, appropriate documentation and testing, disseminated and in part generated through state of the art workshops; industry scoping workshops; Extended Software Development Workshop (ESDW) events; and industry pilot projects. E-CAM software is produced primarily through E-CAM pilot industry projects funded directly by E-CAM, and through the ESDW's. ESDW's typically of 1–2 weeks duration are a unique approach to combine software development with training, i.e. "training by doing", and of engaging with the wider simulation community.

In E-CAM Deliverable D1.1 [1], we gave an overview of existing software for rare events and future needs, and highlighted areas where E-CAM could make useful contributions. That included the development of modules for rare events methods, particularly in the context of the OpenPathSampling (OPS) package. More recently, we have revisited the needs of the community through the State of the Art Workshops. In addition to a continued interest in path sampling, those workshops emphasized the growing importance of artificial neural networks. The use of neural networks to approximate *ab initio* force calculations has been repeatedly highlighted. These considerations, as well as the explicitly expressed interests of ESDW participants and the needs arising from new collaborations with experimental researchers, have guided the selection of the modules listed here. In this way, these modules are responding to the requests of users.

1.1 This Deliverable in the context of E-CAM

This report covers the fifth and final group of nine modules delivered as part of E-CAM WP1. As described in the grant agreement, they are "in the area of classical molecular dynamics responding to requests of users, and their documentation."

Six of the modules delivered here were developed by E-CAM PDRAs. An additional two were developed in collaboration with a PDRA, and the final module was developed as part of an E-CAM ESDW.

Selection of modules for the ESDW was driven by the combination of feasibility and relevance to the goals of the Deliverable and of the project, as laid out in previous reports. These software development tasks were also used as part of a practical introduction to advanced programming techniques and hardware environments for the participants of the ESDWs. Therefore, they have high value for the training component of E-CAM, as well as their intrinsic software value.

2 Background and Themes

Classical molecular dynamics can be used to study a broad range of chemical topics, with applications ranging from medicine to materials development. In this section we will provide a brief overview of the scientific topics that are relevant to multiple modules, as well as a discussion of several software design themes that have emerged as central to these modules.

2.1 Scientific background

2.1.1 Rare events and path sampling

In many simulations, we come across the challenge of bridging timescales. The desire for high resolution in space (and therefore time) is inherently in conflict with the desire to study long-time dynamics. To study molecular dynamics with atomistic detail, we must use timesteps on the order of a femtosecond. However, many problems in biological chemistry, materials science, and other fields involve events that only spontaneously occur after a millisecond or longer (for example, biomolecular conformational changes, or nucleation processes). That means that we would need around 10^{12} time steps to see a single millisecond-scale event. This is the problem of "rare events" in theoretical and computational chemistry.

While modern supercomputers are beginning to make it possible to obtain trajectories long enough to observe some of these processes (such as millisecond dynamics of a protein [2]), even then, we may only find one example of a given transition. To fully characterize a transition (with proper statistics), we need many examples. This is where path sampling comes in. Path sampling approaches obtain many trajectories using a Markov chain Monte Carlo approach: An existing trajectory is perturbed (usually using a variant of the "shooting" move), and the resulting trial trajectory is accepted or rejected according to conditions that preserve the distribution of the path ensemble. As such, path sampling is Monte Carlo in the space of paths (trajectories). Conceptually, this enhances the sampling of transitions by focusing on the transition region instead of the stable states. In direct MD, trajectories spend much more time in stable states than in the transition region (exponential population differences for linear free energy differences); path sampling skips over that time in the stable states.

OpenPathSampling is a software package for trajectory-based rare events methods, such as transition path sampling. Two modules delivered here were involved in the development of a new storage subsystem (SimStore) for OPS. These modules improve the performance of OPS storage, and also prepare OPS for parallelization with Dask, enabling parallel path sampling approaches.

2.1.2 Neural network potentials

Many systems in computational physics and chemistry can be successfully studied with empirical force fields at the atomistic level. In the context of these "molecular mechanics" models, atoms are treated as particles without internal structure and their interactions are defined via rather simple expressions deduced from physical/chemical intuition. Usually a small number of free parameters is enough to tune the potential to reproduce experimental properties with good agreement. However, there are systems for which a satisfying description within this framework is not possible. Take as an example the formation and breaking of covalent bonds. This is the territory of *ab initio* methods which use quantum mechanics to accurately model the behavior of the system. Unfortunately the additional level of detail comes at a cost. Even in small systems *ab initio* methods are usually many orders of magnitude slower than empirical force fields. Moreover, the computational cost increases unfavorably with the number of atoms which makes it impractical to perform large simulations.

With rising influence of machine learning algorithms in science and technology a new category of interatomic potentials has emerged. Machine learning potentials (MLPs) aim at bridging the gap between *ab initio* methods and empirical force fields. In contrast to the latter, MLPs are not bound by a predetermined fixed functional form of the interaction but rather build on the flexibility of an underlying machine learning model, such as artificial neural networks. These are known for their capability to reproduce any complicated function, which in this case is the desired potential energy surface, but rely on a separate training stage before they are ready for use. During this phase the MLP "learns" from a large data set how energies and forces depend on atomic positions. The reference energy landscape is typically computed from expensive *ab initio* methods. Once the training is completed the MLP can accurately predict energies and forces for new (unseen during training) atomic configurations at a fraction of the cost of the reference method. Hence, with MLPs times scales become accessible in molecular dynamics simulations close to those of empirical potentials while maintaining the *ab initio* level of accuracy.

Today MLPs exist in various forms and combine different atomic environment descriptors as inputs for all kinds of machine learning models. A very successful variant is the high-dimensional neural network potential (HDNNP)[3]. The software n2p2 implements the method, provides tools for training and supplies an interface to the popular molec-

ular dynamics package LAMMPS. In this deliverable we present two modules related to MLPs: n2p2 - Improved link to HPC MD software and n2p2 - Polynomial Symmetry Functions.

2.1.3 Polymer dynamics

To study the long term memory of the initial conformation of a highly entangled polymer we need to preserve the topology. It means that two bonds cannot cross. It is of great importance for the study of post-mitotic chromosome unfolding. Preservation of topology is also used in the framework of "Dissipative-Particle-Dynamics" in particular for the study of rheological properties.

To resolve the excluded volume constraints one could use a soft or hard potential between the two points (each point belong to one of the two overlapping bonds) associated to the minimal distance. We use Karush-Kuhn-Tucker conditions to compute it (mathematical optimization field). Soft potential resolves the excluded volume constraints over many time-steps of molecular dynamics while hard potential resolves it in one time-step.

Here, we propose to change the relative velocity between overlapped bonds to resolve the excluded volume constraint in one time-step of molecular dynamics. The idea is to reverse the relative velocity (along the normal to the contact) to prevent further overlapping and make the two bonds bounce on each other surface. We can split the computation based on a collision graph and solve each graph independently (parallelism). The method does not need potential parameters choice.

This velocity-based method comes from physics engine field and we adapted for polymer simulation.

We developed one module (minDist2segments_KKT_for_LAMMPS) to implement correct computation of minimal distance between two bonds for LAMMPS. We developed another module (velocities_resolve_EVC_for_LAMMPS) that is a new fix (for LAMMPS) to resolve excluded volume constraints based on relative velocity. It takes advantage of the underlying parallelism.

2.1.4 Alchemical free energy and particle insertion

While alchemical methods to estimate free energy quantities such as chemical potentials have been around for several decades, they have been limited to small changes in the molecules present in systems. In particular while they have been very useful in estimating the effect of the insertion of small molecules and ligands into dense environments they are still not capable of efficiently allowing the insertion of large molecules including proteins and peptides. This restriction means for example that to estimate the binding affinity between say an antibody and an antigen, or between the moieties of spilt protein, a rare-event biased sampling methods (e.g. umbrella sampling or metadynamics) is needed together with suitable collective variables able to probe the often complex path of binding and disassociation. Also while the key interest are the binding and disassociated states, a large number of intermediary configurations also must be sampled. In the course of the current E-CAM CoE developed a novel perturbative method that can circumvent such costly calculations by allowing much larger molecules to be inserted or deleted. Here we report on two modules which we have developed to implement the method. The first, PIUtils, is a python based module that effectively performs the complex book-keeping entailed during insertion and deletion a wide number of systems, including interfacing with LAMMPS, and python codes such as pymbar coming from the Chodera and Shirts groups. In developing PIUtils, we had to improve the LAMMPS python interface. The second module lammps_pyinterfaceExt does just that.

2.1.5 Analysis tools for molecular dynamics simulations

More and more, analyzing molecular dynamics requires techniques associated with “big data.” Instead of single trajectories, we may analyze ensembles of trajectories. Individual trajectories or the result of the analysis can be too large to easily fit in memory. This leads to a need for analysis tools that can both scale up to larger systems and can run out-of-core. One of the modules presented here, dask-traj, is designed for this purpose.

2.2 Software design themes

2.2.1 Leverage existing high-performance codes

There exist many highly performant open source MD codes. One of the approaches that E-CAM has used is to extend and improve those codes, allowing us to implement methods that can immediately scale up, thanks to the underlying parallelism. This is true of several codes that E-CAM has made major contributions to, including OpenPathSampling and n2p2. In this deliverable, there has been a particular focus on exploiting the scalability of LAMMPS. The following modules leverage the scalability of LAMMPS:

- minDist2segments_KKT_for_SRP

- velocities_resolve_EVC_for_LAMMPS
- n2p2 - Improved link to HPC MD software
- lammps_pyinterfaceExt
- PIUtils

2.2.2 High throughput computing as a parallelization scheme

High throughput computing, where problems are described as a set of loosely coupled tasks which can then be parallelized, is a powerful way of performing parallelization. This approach can wrap around existing parallelization schemes, enabling additional layers around MPI, just as MPI is often used as an additional layer around OpenMP. This could help codes make efficient use of exascale computing resources. We have largely focused on Dask as a Python library for task-based computing. Dask provides the advantages both of being able to scale to more computational resources, as well as allowing out-of-core computation for use cases that are bound by memory, not by computation.

The modules that were designed with high throughput computing are dask-traj, which uses Dask to analyze molecular dynamics trajectories, and the two modules of SimStore, the new storage subsystem for OpenPathSampling. The previous storage subsystem was incompatible with Dask, and the modules presented here have enabled an experimental branch of OPS with simulations parallelized by Dask.

3 Modules

This Deliverable is comprised of 9 modules, which include 2 focused on the new storage subsystem of OpenPathSampling, two based on the n2p2 library for neural network potentials, two based on the particle insertion methods that have been developed in the context of E-CAM, and one which is an external package developed as the result of an E-CAM ESDW. Of these, 1 was purely the result of an ESDW, 6 were contributions from PDRAs, and 2 resulted from collaborations with E-CAM PDRAs.

Material in this section is largely drawn from the detailed module documentation files hosted at the Classical MD section of the E-CAM Library. Links are provided for more information about each module. Modules that have been completed and accepted into the E-CAM library also have a link to the specific module page in the E-CAM library documentation website. Further details about the code contributed and the development process can be found through those links. In addition, each module consists of at least one example of showing how to use it, linked in the "Examples" section of the linked module documentation.

3.1 SimStore: OPS new storage subsystem (core)

This module provides the core of SimStore, a new storage subsystem for OpenPathSampling, which is more flexible and has better performance than the older storage system. This module is also necessary to serialize data for transfer over a network, as is needed for parallelization across multiple nodes.

3.1.1 Module description

OpenPathSampling has the following needs from a storage subsystem:

1. In addition to data objects (data created by the simulation), the simulation objects (i.e., the details of how the simulation was run) should be stored. This helps track provenance and enhances reproducibility.
2. All objects should have universally unique identifiers (UUIDs). References in data objects to the UUIDs of the simulation objects that generated them are important for provenance and reproducibility.
3. Because users may add new simulation objects, storing simulation objects should be very general, and should place minimal burden on the users who create new simulations objects.
4. The results from functions that are calculated during a simulation should be stored in such a way that they can be retrieved again, instead of recalculating them.
5. Path sampling generates large quantities of data. Because of this, and because analysis is frequently done in layers (e.g., one can perform the TIS rate analysis without reloading any coordinate information), it should be possible to reload some information without reloading the entire object.
6. Storage of simulation objects should be (nearly) human readable. While there are some exceptions to human readability, it is important that, for the most part, simulation parameters can be read and interpreted if a user wishes to load OPS data in, e.g, another programming language.

This set of requirements was met by the previous storage subsystem, `netcdfplus`. However, `netcdfplus` was written in a recursive style, which means that every load from disk was a separate request. This made `netcdfplus` very slow. Additionally, the storable function results (#4 above) were written in a way that was not compatible with parallelization. Finally, the base storage class of `netcdfplus` inherits from `netcdf4`, meaning that it was tied to a single backend.

With this module, we introduce SimStore, which is being added as an experimental module in OpenPathSampling, with the intent of replacing `netcdfplus` in OPS 2.0. SimStore will have all the same features, with better performance, more flexibility for users and developers, and a design that is prepared for parallelization. Until version 2.0, both storage subsystems will coexist in the OPS library.

This module, in particular, provides the core storage capabilities (#1, #3, and #6 above) and the proxy-based lazy loading (#5). The UUIDs (#2) are still provided by `netcdfplus`. A future module will address the problem of storing function results (#4). Importantly, the API for flexible storage of general simulation objects (#3) remains the same as in `netcdfplus`, facilitating the transition to SimStore.

Some of the specific functionality covered includes:

- **SQL backend**: The first backend for SimStore is SQL, defaulting to sqlite3. However, in principle, it should be nearly trivial to use a MySQL or PostgreSQL instance instead, which would be suitable for parallel usage.

- **Schema-based storage**: The description of data objects, which does not vary much for a given application, is provided by a human-readable schema, including specification of what objects should be loaded as lazy proxies. This makes it easy for users or developers to see and understand the overall data model.
- **Dynamic registration of new tables**: Some aspects of the data model do not vary for a certain application (e.g., in OpenPathSampling, trajectories are always lists of snapshots). However, some aspects do depend on the specific use case (e.g., in OpenPathSampling, the size of the coordinates array depends on the specific molecular system being studied). SimStore allows dynamic registration of tables in order to create new tables of the correct size for, e.g., snapshots coordinates.
- **Extensible JSON-based simulation object serialization**: In order to create a (mostly) human readable description of simulation objects, SimStore (like `netcdfplus`) uses JSON. However, some simulation objects from outside packages (e.g., instances of `simtk.quantity`, which pair a value with a unit, and are used in OpenMM) require custom serialization. In `netcdfplus`, that custom serialization was inside a `netcdfplus` function, and extension by the user required editing the `netcdfplus` code. SimStore uses a simple registration protocol so that new custom JSON serialization can be provided by the user without digging into the internals.

This module only deals with the generic and reusable aspects of SimStore. Integration of SimStore and OpenPathSampling will be the subject of a future module.

3.1.2 Motivation and exploitation

This module was motivated primarily by the need for a new storage subsystem that is suitable for parallel simulations. In addition, this new storage subsystem is more easily maintained and extended than the previous version.

This module will be the basis of the storage to be used in OpenPathSampling 2.0 and later, and it is already in use by methods developers at the Norwegian University of Science and Technology.

Additional Details

Direct Documentation Link	readme.rst of SimStore: OPS new storage subsystem (core).
Merge Request	Merge Request of SimStore: OPS new storage subsystem (core).

3.2 SimStore: Storable functions

This module adds "storable functions" to SimStore, the new storage subsystem for OpenPathSampling. Storable functions cache the results of previous calculations to disk. This new implementation will support future parallelization approaches.

3.2.1 Module description

Trajectory-based methods to study rare events, such as transition path sampling (TPS), frequently require calculation of some collective variables during the simulation. In some cases, these collective variables can be relatively expensive to calculate, and may be calculated hundreds of thousands of times during simulation.

For some types of simulations, such as the one-way shooting variable in TPS, parts of trajectories can be reused, making it advantageous to store the results of collective variables in memory. Furthermore, those same collective variables are frequently used in analysis, making it advantageous to store the results to disk.

This module introduces the parts of SimStore that manage that storage. This includes the `StorableFunction` class itself, which wraps around a user-defined function and handles caching results in memory, and looking up results cached to disk. The user-defined function must take a data object (such as a snapshot or a trajectory), which has a unique universal identifier (UUID), and must return the same value every time it operates on the same input (i.e., it must be a "pure" function).

A `StorableFunction` can be used in different modes: in `'analysis'` mode, it first searches the memory cache, then the disk storage, then finally evaluates the internal function. In `'production'` mode, it first searches the memory cache, then evaluates the function. Finally, in `'no-caching'` mode, it always evaluates the internal function.

One of the challenges in designing the new storable function infrastructure was to ensure that it would be compatible with parallelization. This module includes functionality so that the memory caches from different remote workers can be returned with the other results, and combined into a master memory cache of the process that also stores results to disk.

3.2.2 Motivation and exploitation

As with the other module on the OPS storage subsystem, this module was strongly motivated by the necessity of replacing the existing approaches in OPS with an approach that will work in parallel.

This module will be the basis of the collective variables to be used in OpenPathSampling 2.0 and later, and it is already in use by methods developers at the Norwegian University of Science and Technology.

Additional Details

Direct Documentation Link	readme.rst of SimStore: Storable functions.
Merge Request	Merge Request of SimStore: Storable functions.

3.3 dask-traj

For analysis of MD simulations MDTraj is a fast and commonly used analysis. However MDTraj has limitations, such as the requirement that the whole trajectory and result of the computation fits into memory. This module rewrites part of MDTraj to work with Dask in order to achieve out-of-memory computations, and combined with dask-distributed results in possible out-of-machine parallelization, essential for HPCs and a (surprising) speed-up even on a single machine.

3.3.1 Module description

Using MDTraj is a fast and easy way to analyze MD trajectories. However, MDTraj has a couple limitations:

- The whole trajectory needs to fit into memory, or gathering results becomes inconvenient
- The result of the computation also needs to fit into memory
- All processes need access to all the memory, preventing out-of-machine parallelization, and HPC scaling

Dask-traj solves all 3 limitations by rewriting the MDTraj functions to work with dask.arrays. This is done for both the trajectory and the computation functions. As dask.arrays know how to spill to disk, this lifts the requirement to fit into memory on both.

Together with dask-distributed it also allows the computation to be executed in a distributed way, which allows scaling out of a single machine. In preliminary tests this approach even leads to a speedup on a single machine, which is surprising as MDTraj is already a parallel code.

The splitting of everything in Dask-traj is done in the time-axis of the MD trajectory and as a lot of analysis is embarrassingly parallel, this leads to nice non-communicating compute graphs.

3.3.2 Motivation and exploitation

This module was contributed at an ESDW. It was inspired by the Dask-array and Dask-dataframe tools that allow Dask to quickly replace workflows based on NumPy and pandas, repectively.

Additional Details

Direct Documentation Link	readme.rst of dask-traj.
Merge Request	Merge Request of dask-traj.

3.4 minDist2segments_KKT_for_SRP

The minDist2segments_KKT_for_SRP module returns the minimal distance between two line segments. It uses the Karush-Kuhn-Tucker conditions (KKT) for the minimization of distance under constraints. The module implements the previous function for the SRP fix in LAMMPS. Indeed, the SRP function to compute the minimal distance does not always give the correct solution.

3.4.1 Module description

To study the long term memory of the initial conformation of a highly entangled polymer we need to preserve the topology. That means that two polymer bonds cannot cross. It is of great importance for the study of post-mitotic chromosome unfolding. Minimal distance between two bonds can be used in Dissipative-Particle-Dynamics to prevent bond crossings too. To resolve the excluded volume constraints one could use a repulsive potential between the two points associated to the minimal distance. We propose a new option in the computation of the minimal distance in the SRP fix for LAMMPS. Indeed, SRP fix computes the minimal distance between two infinite lines and reset the

solution to occur along the interior of the bond. This method is not always accurate. The KKT conditions allows to solve the problem of minimal distance such finite segment length constraint holds.

3.4.2 Motivation and exploitation

This module is used by the ongoing work "velocities_resolve_EVC" module implementation for LAMMPS. It is part of E-CAM post-doc pilot project. We would use it to avoid topology violation in an entangled polymer system.

Additional Details

Direct Documentation Link	readme.rst of minDist2segments_KKT_for_SRP.
Merge Request	Merge Request of minDist2segments_KKT_for_SRP.

3.5 velocities_resolve_EVC_for_LAMMPS

The **velocities_resolve_EVC_for_LAMMPS** is a module that resolves the excluded volume constraint with a velocity formulation (no potential applied between two bonds). It is an implementation for LAMMPS of an already existing module. The **velocities_resolve_EVC_for_LAMMPS** uses the module minDist2segments_KKT_for_SRP (you can find on the minDist2segments_KKT_for_SRP module) to find the minimal distance between two bonds.

3.5.1 Module description

To study the long term memory of the initial conformation of a highly entangled polymer we need to preserve the topology. It means that two bonds cannot cross. It is of great importance for the study of post-mitotic chromosome unfolding. To resolve the excluded volume constraints one could use a soft or hard potential between the two points associated to the minimal distance. Here, we propose to change the relative velocity between overlapped bonds to resolve the excluded volume constraint in one time-step of molecular dynamics. We propose to implement this functionality as a new fix for LAMMPS.

3.5.2 Motivation and exploitation

This module is a key part of the “Implementation of contact joint to resolve excluded volume constraints” pilot project.

Additional Details

Direct Documentation Link	readme.rst of velocities_resolve_EVC_for_LAMMPS.
Merge Request	Merge Request of velocities_resolve_EVC_for_LAMMPS.

3.6 n2p2 - Improved link to HPC MD software

This module documents efforts to improve the interaction of n2p2 with existing HPC software, in particular the molecular dynamics (MD) software package LAMMPS.

3.6.1 Module description

Although n2p2 was already shipped with source files for patching LAMMPS before, the build process required manual intervention of users. To avoid this in future versions of LAMMPS a pull request was created to include the n2p2/LAMMPS interface by default as a user package. In order to conform with LAMMPS contribution guidelines multiple issues were resolved, triggering these changes/additions to LAMMPS and n2p2:

- Modify the traditional build process (via makefiles) to include n2p2
- Modify the CMake build process to search and include n2p2
- Create additional documentation about the build settings
- Create a suitable example which can be shipped with LAMMPS
- Adapt documentation of the LAMMPS "`pair_style nnp`" command
- Change n2p2 to conform with LAMMPS "`bigbig`" settings
- Change the source files "`pair_nnp.(cpp/h)`" to conform with the LAMMPS coding style

Furthermore, the n2p2 build system was adapted to allow for multiple interfaces to other software packages, with an option to select only those of interest to the user. As a first application, the user contributed CabanaMD interface was integrated in the new build process. CabanaMD is an ECP proxy application which makes use of the Kokkos performance portability library and n2p2 to port neural network potentials in MD simulations to GPUs and other HPC hardware.

3.6.2 Motivation and exploitation

The integration of neural network potentials directly in LAMMPS via a user package with linkage to n2p2 greatly enhances the visibility and user experience. Users may retrieve information about the neural network potential method and its use directly on the LAMMPS documentation page. Modifying the n2p2 build process to allow for multiple interfacing software simplifies the development of CabanaMD. This contribution of n2p2 users can be viewed as a precursor of a Kokkos implementation of NNPs in LAMMPS. Ultimately, such an addition to n2p2/LAMMPS would be of great value for the community as it would allow for running molecular dynamics simulation with NNPs on GPUs.

Additional Details

Direct Documentation Link	readme.rst of n2p2 - Improved link to HPC MD software
Merge Request	Merge Request of n2p2 - Improved link to HPC MD software

3.7 n2p2 - Polynomial Symmetry Functions

This module introduces a new set of atomic environment descriptors for high-dimensional neural network potentials (HDNNPs) in n2p2. Polynomial symmetry functions are designed to mimic closely the behavior of traditional Behler-Parrinello symmetry functions but with a significantly reduced computational cost.

3.7.1 Module description

The symmetry functions proposed in the original work of Behler and Parrinello contain expressions of the form $\exp(-\eta r_{ij}^2) f_c(r_{ij})$ in the innermost loop over all neighbors of atoms. Often the cutoff function $f_c(r)$ is chosen to be a cosine or hyperbolic tangent. Considering the computational cost of these transcendental functions an alternative formulation of symmetry functions based on polynomials like

$$((15-6x)x-10)x^3+1 \tag{1}$$

has been proposed recently. Here, cheap polynomials are combined to form compact functions in the radial and angular domain which mimic the behavior of Behler-Parrinello type symmetry functions at a significantly reduced execution time. The benefits, benchmarks and many example applications presented in the above preprint will also be published soon in a peer-reviewed journal.

This module's changes to the n2p2 code comprise of new classes for different types of polynomial symmetry functions (PSFs), some helper classes and a redesign of the symmetry function caching mechanism.

3.7.2 Motivation and exploitation

The redesign of atom-centered symmetry functions resulted in a significant reduction of their computational cost. When using n2p2 together with LAMMPS the performance gain further enhances the accessible time scales of parallel molecular dynamics simulations. Furthermore, due to their simple and cost-effective functional form polynomial symmetry functions are well-suited to serve as a basis for future high-performance implementations, in particular on GPUs.

Additional Details

Direct Documentation Link	readme.rst of n2p2 - Polynomial Symmetry Functions
Merge Request	Merge Request of n2p2 - Polynomial Symmetry Functions

3.8 lammps_pyinterfaceExt

The module contains patch files for the Stable release 29 October 2020 version of LAMMPS, to enable accessing simulation force-filed parameters from python.

3.8.1 Module description

When performing alchemical free energy calculations, it is necessary to change the attributes of various particles in a simulations — Atom properties such as charge and mass or force field properties such as ϵ or σ of Lennard-Jones potentials.

The LAMMPS library along with its Python interface provides the ability to directly access and change many such attributes in a running simulation. However, for pair potentials while it is possible to change their parameters it is currently not possible to read the existing parameters from a running simulation. This is required for the alchemical free energy calculations using the Particle Insertion approach as described in the Particle Insertion Core module, where the scaling of the forcefield attribute depends on the existing attributes.

This module addresses this limitation by extending the LAMMPS library and the Python interface to add a function allowing read access to pair-potential parameters. It also adds an extract method to the the pair_*.cpp files associated with some commonly used pair-potential in LAMMPS as described in fix adapt.

3.8.2 Motivation and exploitation

This module was developed in order to facilitate alchemical free energy calculations using the Particle Insertion approach implemented in a previously reported module, Particle Insertion: Core.

Additional Details

Direct Documentation Link	readme.rst of lammps_pyinterfaceExt.
Merge Request	Merge Request of lammps_pyinterfaceExt.

3.9 PIUtils

This module ports the already existing modules PICore and PIhydration from the LAMMPS scripting language to Python. It allows to apply the method described in those modules to a larger variety of systems. Additionally it also generalizes the method, thereby allowing use of different forcefields.

3.9.1 Module description

This module performs particle insertion/deletion of any type of particle in dilute or dense conditions in a variety of thermodynamic ensembles via a novel perturbative approach using LAMMPS and PyLammps, a python interface to LAMMPS. This will be extended to other MD engines such as GROMACS at a later stage.

The main advantage of this type of alchemical free energy calculation is that it does not use soft-core potential as many of the approaches to date do. As such, there are less alchemical pathways to compute as the electrostatic and VdW interactions can be switched along with all other types of interactions. This results in being able to compute the free free energy differences faster with less simulation time. The other main advatange is that due to the mathematical form of rescaling used, the singularity of insertion can be avoided.

3.9.2 Motivation and exploitation

This type of alchemical insertion and deletion is useful in a whole host of situations, where one would like to compute the free energy changes associated with adding or removing particle/molecule from a complex. Common applications would include:

- Computing the binding energy of ligands to proteins
- Computing the binding energy of protein-protein complexes
- Computing the free energy change associated with increasing or decreasing solvent (hydration/dehydration)
- Computing the free energy change associated with mixing solvents

Additional Details

Direct Documentation Link	readme.rst of PIUtils.
Merge Request	Merge Request of PIUtils.

4 Performance Considerations

As discussed in Sec. 2.2, two major software design themes in WP1 relate to performance. First, we leverage the performance of existing software whenever possible. Second, we have found that high throughput computing provides a flexible approach to parallelization that is well-suited to both simulation and analysis. Both of these approaches have been used by these modules

4.1 Leveraging existing performance

For OpenPathSampling, a package that has received major contributions from E-CAM modules, a fundamental design principle was to count on the performance aspects of underlying molecular dynamics engines. By interfacing with engines such as OpenMM and Gromacs, the most demanding parts of path sampling simulations are performed with the scalability of those codes. This is also true of n2p2, which acts as a plug-in for LAMMPS. n2p2 provides a neural network potential, but the actual dynamics are performed efficiently with LAMMPS.

The modules delivered here have a particular emphasis on leveraging the scalability of LAMMPS. This focus on LAMMPS has been part of a general movement toward LAMMPS in E-CAM, which also resulted in E-CAM developing tutorial materials about optimizing LAMMPS performance in HPC environments.

The modules minDist2segments_KKT_for_SRP and minDist2segments_KKT_for_SRP are traditional LAMMPS contributions: these are C++ implementations of functionality used during the underlying dynamics. The PIUtils module uses the Python interface to LAMMPS to extend the functionality of LAMMPS. The modules lammps_pyinterfaceExt and n2p2 - Improved link to HPC MD software are more about making it easier to LAMMPS to communicate with other packages. In lammps_pyinterfaceExt, some internals of LAMMPS are exposed to users of its Python interface. In n2p2 - Improved link to HPC MD software, modifications are made in preparation to bring n2p2 in as a core element of LAMMPS. Together, these modules represent a wide range of approaches with the common theme that they immediately benefit from the performance inherent in LAMMPS.

4.2 High throughput computing

High throughput computing offers the possibility of adding an additional layer of parallelization. Just as MPI can be combined with OpenMP by using OpenMP on inner loops and MPI on an outer loop, task based parallelization can wrap multiple MPI-parallelized tasks that can run simultaneously. This approach may help codes scale to the exascale. Task-based approaches also have benefits for small-scale computing by enabling out-of-core computation. This is essential for analysis of the large datasets that can be generated with classical MD.

The modules on the OpenPathSampling storage subsystem are also designed with task-based parallelization in mind. The existing OPS storage subsystem was incompatible with Dask, and the kinds of tasks that OPS performs, such as replica exchange transition interface sampling, are ideal opportunities to wrap already-parallelized molecular dynamics as tasks that can be run simultaneously. Though not provided as a module here, there in now an experimental branch of OPS that can use Dask for parallelization. That branch builds on contributions of the two OPS-focus modules in this deliverable.

Another module that uses Dask is dask-traj, a library inspired by MDTraj. Dask-traj uses Dask's parallel arrays under the hood, and reimplements some of MDTraj's functionality on top of that. In addition to allowing parallelization of the reading and analysis of large data sets, this also enables out-of-core analysis: allowing users to use their laptops to analyze trajectories that otherwise wouldn't fit in memory.

5 Overall impact of the results achieved within the Work-package

5.1 Overview of the results achieved so far

E-CAM's WP1 "Classical Molecular Dynamics" provide a means for academics and industrialists to address computational questions that involve classical MD calculations. In particular, the WP focuses on the development of software for path-based sampling and analysis to study rare events.

So far within this WP, **61** software modules have been developed (against a target of 43) which were certified according to the E-CAM guidelines [4], and other 11 that are work in progress or are under review to get certification. The modules developed consist of:

- **Modules for studying the thermodynamics and kinetics of rare events**. Under the framework of an E-CAM pilot project on binding kinetics in collaboration with the software vendor SME BiKi Technologies, we have investigated the binding/unbinding of a selective reversible inhibitor for protein GSK3B using path sampling methods (see sec. 2.1.1 of this deliverable to understand rare events and path sampling). This inhibitor has been proposed as a potential drug for Alzheimer's Disease. In addition to the specific results from this project, the software developed has a broader applicability than just protein-ligand binding, and for instance, the module Contact Maps applies to other binding processes, such as protein-protein aggregation or DNA-protein binding, as well as to many questions about large scale conformational changes in biomolecules, such as protein folding. The module Contact concurrences extends the capabilities of the contact maps module and represents the development of new analysis techniques for ligand-protein binding, as well as the code to implement those techniques.
- **Modules that contribute to the development of OpenPathSampling (OPS)**. OPS [5],[6] is a software package to perform path sampling simulations and other trajectory-based approaches to study rare events, and is one of the key application codes of WP1. Much of the development of OPS has been supported by E-CAM through resources and at ESDWs. The methods implemented in OPS can be used to study many kinds of problems, including drug binding and unbinding, self-assembly processes, conformational changes in biomolecules, and chemical reactions. During the lifetime of the project we have
  - **Improved the performance and usability of OPS**
  - **provided support for OPS to interface with other codes in the MD community** (e.g. Integrating LAMMPS with OpenPathSampling, Gromacs engine in OpenPathSampling, and PLUMED Wrapper for OpenPathSampling)
  - **added more trajectory-based rare event methods to OPS**
  - **maintained and improved the quality of the OPS code base**
  - **integrated OPS with E-CAM's High Throughput Computing (HTC) library jobqueue_features**. OPS was expanded to integrate jobqueue-features library, which can manage thousands or millions of tasks in parallel. To leverage E-CAM's HTC library though, which is based on Dask, OPS needed to be serialised, a work that represented a considerable effort for E-CAM (see SimStore: OPS New Storage Subsystem). Due to it, OPS can now almost seamlessly transition from use on a personal laptop to some of the largest HPC sites in Europe! A white paper describing this work is in ref. [7]. This work was done in collaboration with a PRACE team in Poland.
  - **promoted the use of OPS for studying systems of highly importance, such as the SARS-CoV-2 main protease**. The integration of OPS and the HTC library described in the previous point resulted in an unprecedented parallelised committor simulation capability. The committor analysis is used to generate initial trajectories for the transition path sampling approach. In specific, it is being used to study the SARS-CoV-2 main protease. An initial analysis of the stable states, based on a long trajectory provided by D.E. Shaw Research suggests that a loop region of the protein may act as a gate to the active site. This conformational change may regulate the accessibility of the active site of the main protease, and a better understanding of its mechanism could aid drug design.

  The full portfolio of modules that are based on OPS and were developed in the context of E-CAM can be found here.
- **Modules that are tools to optimize a protein based biosensor**. An E-CAM transverse action is the development of a protein based sensor[2] with applications in medical diagnostics, scientific visualisation and therapeutics. At the heart of the sensor is a novel protein based molecular switch which allows extremely sensitive real time

[2] EP3265812A2, 2018-01-10 and WO2018047110A1, 2018-03-15

measurement of molecular targets to be made, and to turn on or off protein functions and other processes accordingly. For a description of the sensor, see here.

At the heart of the sensor is advanced simulation using massively parallel computation, rare-event methods and genetic engineering. More recently the sensor caught a great deal of attention due to its potential application in detecting the presence of SARS-CoV-2. Work in this direction is currently being developed, in straight collaboration with industry (see the story here). Experimental teams will provide a proof of concept for the COVID-19 sensors designed entirely in silico using software developed in the context of E-CAM.

Besides the specific results from this work, this whole exercise shows the transformation of a beautiful idea born via simulation into a commercial opportunity and a potential start-up creation. We have been reporting the successes of this work through a news blog on our website: From idea to market.

- **Modules that study polymer dynamics** . In the context of a pilot project on the Implementation of contact joint to resolve excluded volume constraints we have developed a new approach (adapted from physics engine to the present specific usage) to resolve the excluded volume constraint problem. The modules originating from this work have been reported here in sections 3.4 and 3.5 and can be used to study mitotic chromosome unfolding and the rheological properties of polymers. More specifically, the work is being used in a genome wide simulation of the fruit fly (publication in progress). The pilot project webpage provides the complete list of modules developed in the context of this work.
- **Modules that further expand the Neural Network Potential Package n2p2**. The n2p2 package contains software that allows users to use an existing neural network potential parameterization to predict energies and forces (with standalone tools but also in conjunction with the MD software LAMMPS). In addition, it allows to train new neural network potentials with the provided training tools. E-CAM has been supporting n2p2 developments in the context of a pilot project on the Implementation of neural network potentials for coarse-grained models.

  n2p2 can run on HPC through its interface with LAMMPS. Additionally, an integration of n2p2 to the official LAMMPS repository has been recently proposed (see here), which can greatly increase the availability and potential exploitation of n2p2. Furthermore, the n2p2 build system was adapted to allow for multiple interfaces to other software packages, with an option to select only those of interest to the user. As a first application, the user contributed CabanaMD package was integrated in the new build process. CabanaMD is a proxy application for MD which makes use of the Kokkos performance portability library and n2p2 to port neural network potentials in MD simulations to GPUs and other HPC hardware.

  The list of modules developed in the context of this work is available here.

The full portfolio of modules developed under WP1 is accessible from the software library for WP1 at https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/.

21% of the total number of modules developed under WP1 come from external contributions to the project (i.e. participants to our ESDWs and collaborations which are not directly supported by the project), and the remaining are from our Postdoctoral Research Associate (PDRA)s.

5.2 Dissemination and exploitation

11 scientific publications originated from the project in the area of Classical MD, and at least two are work in progress.

1. Improved Description of Atomic Environments Using Low-Cost Polynomial Functions with Compact Support, Bircher, M. P., Singraber, A. and Dellago, C. *arXiv*:2010.14414 [cond-mat, physics:physics] **2020**, submitted also to peer-reviewed journal. https://arxiv.org/abs/2010.14414
2. Unfolding the prospects of computational (bio)materials modelling, G. J. Agur Sevink et al., *J. Chem. Phys.* **2020**, 153, 100901.
   DOI: https://doi.org/10.1063/5.0019773. Open access version here.
3. Reliable Computational Prediction of the Supramolecular Ordering of Complex Molecules under Electrochemical Conditions, Benedikt Hartl et al., J. Chem. Theory Comput. **2020**, 16, 8, 5227–5243.
   DOI: https://doi.org/10.1021/acs.jctc.9b01251.
4. Atomistic insight into the kinetic pathways for Watson-Crick to Hoogsteen transitions in DNA, Jocelyne Vreede et al., *Nucleic Acids Research* **2019**, Vol. 47, No. 21, 11069–11076.
   DOI: https://doi.org/10.1093/nar/gkz837.

5. OpenPathSampling: A Python Framework for Path Sampling Simulations. 1. Basics David W. H. Swenson et al, *J. Chem. Theory Comput.* **2019**, 15, 813-836.
DOI: https://doi.org/10.1021/acs.jctc.8b00626.

6. OpenPathSampling: A Python Framework for Path Sampling Simulations. 2. Building and Customizing Path Ensembles and Sample Schemes, David W. H. Swenson et al, *J. Chem. Theory Comput.* **2019**, 15, 837-856.
DOI: https://doi.org/10.1021/acs.jctc.8b00627.

7. The asymmetric Wigner bilayer, Moritz Antlanger et al., *J. Chem. Phys.* **2018**, 149, 244904.
DOI: https://doi.org/10.1063/1.5053651. Open access version here.

8. Unimolecular FRET sensors: Simple linker designs and properties, Shourjya Sanyal et al., *Nano Communication Networks* **2018**, 18, 44–50.
DOI: 10.1016/j.nancom.2018.10.003. Open access version here.

9. The opposing effects of isotropic and anisotropic attraction on association kinetics of proteins and colloids, Arthur C. Newton at al, *J. Chem. Phys.* **2017**, 147, 155101.
DOI: http://doi.org/10.1063/1.5006485. Open access version here.

10. Benchmarking a Fast Proton Titration Scheme in Implicit Solvent for Biomolecular Simulations, Fernando Luís Barroso da Silva and Donal MacKernan, *J. Chem. Theory Comput.* **2017**, 13, 2915-2929.
DOI: https://doi.org/10.1021/acs.jctc.6b01114. Open access version here.

11. Rich Polymorphic Behavior of Wigner Bilayers, Moritz Antlanger et al., *Phys. Rev. Lett.* **2016**, 117, 118002.
DOI: https://doi.org/10.1103/PhysRevLett.117.118002.

12. Equilibrium structures of anisometric, quadrupolar particles confined to a monolayer, Thomas Heinemann et al., *J. Chem. Phys.* **2016**, 144, 074504.
DOI: https://doi.org/10.1063/1.4941585. Open access version here.

The software produced within this WP was disseminated via the articles above, but also at conferences (e.g. FOODSIM, PASC) and workshops organized by the members of the WP (see next sections); via the project website (e.g. in success stories, in the modules of the month category, on the newsletter, etc.) - for an overview of the news items on our website that are associated to this WP see here, and through six deliverables produced during the project lifetime and that are listed here (Number 1 deliverables).

We would like to note that the activities within this WP fostered several scientific collaborations. In particular, the work on the development of OPS has been used in Bachelor, Master and PhD thesis. The 4th paper on this list is an example of such collaboration. Collaborations with the University of Amsterdam (P. Bolhuis) and the Chodera lab at MSKCC in the US (John Chodera) are to be highlighted too when mentioning the developments on OPS, rare events and path sampling.

The modules developed for the excluded volume constraints problem, are being used in collaborations at the ENS Lyon and at the University of Grenoble Alpes. Furthermore, these modules will be used by a funded Agence National de la Recherche (ANR) project (CRYOCHROM).

5.3 Industrial impact

Industry connection within this WP happened through two pilot project in collaboration with industry:

- Binding Kinetics, in collaboration with BiKi Technologies (see point 1 from section 5.1)
- Food and Pharmaceutical Proteins, in collaboration with APC. This work is in direct connection to the development of a molecular bionsensor (see point 3 of sec.5.1).

Furthermore, we have held six workshops in WP1 that dealt with topics of interest for industry:

1. State-of-the-Art Workshop in Reaction Coordinates from Molecular Trajectories, 29 August - 2 September 2016, Lorentz Centre, Leiden, The Netherlands. See the workshop report.
2. State-of-the-Art Workshop in Large Scale activated event simulations, 1 - 3 October 2018, CECAM-AT, Austria. See the workshop report.
3. Scoping workshop: Building the bridge between theories and software: SME as a boost for technology transfer in industrial simulative pipelines, 23 - 25 May 2018, Fondazione Instituto Italiano di Tecnologia (IIT), Genoa, Italy. See the workshop report.

4. Scoping Workshop: Solubility prediction, 14 - 15 May 2018, CECAM-FR-RA, Ecole Normale Supérieure de Lyon, France. See the workshop report.
5. Scoping workshop on Electrochemical energy storage: Theory meets industry, 12 - 14 June 2019, CECAM-FR-MOSER, France. See the workshop report.
6. Simulation of open systems in Chemistry, Pharma, Food Science and Immuno-diagnostics: Rare-event methods at constant chemical potentials including constant pH, 25 February, 2 March, 23 March and 25 March 2021. Workshop page.

Three of these events were organized in collaboration with industry and industrial attendance at these events was very satisfactory.

5.4 Training

During the lifetime of the project we organized five ESDWs in the area of Classical MD:

- Inverse Molecular Design & Inference: building a Molecular Foundry, 4 - 8 2019, CECAM-IRL, Ireland. Workshop page.
- Topics in Classical MD, 3 - 12 April 2019, CECAM-FR-RA, ENS Lyon, France. Workshop report.
- Intelligent High Throughput Computing for Scientific Applications, 16 - 20 July 2018, CECAM-IT-SIMUL, Politécnico de Torino, Turin, Italy. Workshop report.
- Classical Molecular Dynamics, 14 - 25 August 2017, CECAM-NL, Lorentz Centre, Leiden, The Netherlands. Workshop report.
- Trajectory Sampling, 114 - 25 November 2016, CECAM-AT, Traunkirchen, Austria. Workshop report.

Additionally, we held a joint PRACE/E-CAM Tutorial on Machine Learning and Simulations on the 10-13 March 2020. And in January and February 2021 we organized a series of three webinars on High Throughput Computing with Dask and E-CAM's HTC library Jobqueue-Features. Key lectures are available here.

In total, 128 people were trained at the events organized within WP1.

5.5 Societal impact

The societal benefits arising from the developments in WP1 are twofold. On the fundamental level, the tools will enable new discoveries and developments, for instance in material science and molecular medicine. This will be made possible by the scientific community using the sampling and analysis tools that are being developed.

On an economic level, industry can benefit from software containing efficient and easy to use simulation and analysis tools to extract observables for applications including but not limited to:

- Drug design
- Diagnostics
- Food/dairy industry
- Genomics
- Materials science

Society can benefit from the methods developed and ultimately the software, which can make drug development cheaper, improve diagnostics &medicine, food quality, and help in the development of high-tech materials.

Furthermore, the success of E-CAM in the development of software tools for rare event simulations can have an important societal impact. Many processes of importance in the fields of materials science and drug design are determined by rare events. So a detailed molecular understanding of new materials or drugs relies on our ability to study rare events in large-scale computer simulations. The simulation algorithms and tools developed in E-CAM are an important step in this direction.

6 Outlook

The report of Deliverable 1.5 of E-CAM describes 9 software modules of WP1 in classical molecular dynamics. As described in the grant agreement, they are "in the area of classical molecular dynamics responding to requests of users." These modules cover several domains within the area of classical molecular dynamics. The specific modules were primarily selected based on direct user requests, or based on the needs of collaborations with industrial or academic research partners, or, for ESDW modules, based on the interests of the ESDW participants. In this way, these modules respond to the requests of users.

The modules delivered here include 6 by PDRAs, 2 from a collaboration with a PDRA, and 1 from an ESDW. One of these modules is a new package for molecular dynamics analysis. Two modules deal with OpenPathSampling, and another two deal with n2p2. Finally, two are part of a particle insertion approach being developed by an E-CAM PDRA with the intent of meeting the needs of E-CAM's industrial partners.

The modules in this and previous deliverables have dealt with a wide range of themes within classical molecular dynamics, including rare events, neural network potential energy surfaces, polymer modeling, and alchemical free energy and particle insertion. They have been contributed by PDRAs and by ESDW participants. They have been driven by the needs of pilot projects and industrial partnerships, as well as by the requests of users. Many of the modules that have been developed have a strong focus on obtaining excellent scaling, either by leveraging the scalability of existing codes or by using modern techniques, such as high throughput computing, to scale up.

In this process, E-CAM WP1 has developed and contributed to the development of many open source software package, including many contributions that have not been formalized as modules. Contact Map Explorer was initially developed as part of an E-CAM pilot project, and the vast majority of its development came through E-CAM. At this point, nearly 50% of the current code in OpenPathSampling has been developed as part of E-CAM. A significant part of the recent extensions and maintenance of n2p2 has come through E-CAM's support with over 14000 new lines of code amounting in total to 35% of the current source file content.

In addition, E-CAM WP1 PDRAs have contributed to many other software packages, both with modules and with code that hasn't been formalized as a module. PRDAs have made significant contributions to domain-specific tools such as MDTraj and OpenMMTools. Additionally, PDRAs have made small contributions to more general open source projects, including projects with millions of monthly downloads, such as ujson and tqdm.

Although this is the last deliverable from E-CAM WP1, there are 16 other modules that have been completed that have not been included in any deliverable. In addition, work continues on at least 9 modules that have not yet been completed. E-CAM has made and will continue to make significant contributions to software for classical molecular dynamics.

References

Acronyms Used

CECAM Centre Européen de Calcul Atomique et Moléculaire
ESDW Extended Software Development Workshop
WP Work Package
MD molecular dynamics
OPS OpenPathSampling
PDRA Postdoctoral Research Associate

URLs referenced

Page ii

https://www.e-cam2020.eu ... https://www.e-cam2020.eu
https://www.e-cam2020.eu/deliverables ... https://www.e-cam2020.eu/deliverables
Internal Project Management Link ... https://redmine.e-cam2020.eu/issues/165
dwhs@hyperblazer.net ... mailto:dwhs@hyperblazer.net
http://creativecommons.org/licenses/by/4.0 ... http://creativecommons.org/licenses/by/4.0

Page 1

Jupyter notebook ... http://jupyter.org

Page 2

E-CAM Deliverable D1.1 ... https://dx.doi.org/10.5281/zenodo.841694

Page 3

millisecond dynamics of a protein ... http://pubs.acs.org/doi/abs/10.1021/acs.jpcb.6b02024

Page 4

LAMMPS ... https://lammps.sandia.gov/

Page 6

Classical MD section of the E-CAM Library ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/index.html#readme-classical-md

Page 7

readme.rst of SimStore: OPS new storage subsystem (core). ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/OpenPathSampling/ops_new_storage/readme.html
Merge Request of SimStore: OPS new storage subsystem (core). ... https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/-/merge_requests/63

Page 8

readme.rst of SimStore: Storable functions. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/OpenPathSampling/ops_new_storage_2/readme.html
Merge Request of SimStore: Storable functions. ... https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/-/merge_requests/252
MDTraj ... http://mdtraj.org/
Dask ... https://dask.org/
dask-distributed ... https://distributed.dask.org/en/latest/
MDTraj ... http://mdtraj.org/
dask.arrays ... https://docs.dask.org/en/latest/array.html
dask-distributed ... https://distributed.dask.org/en/latest/
readme.rst of dask-traj. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/dask_traj/readme.html
Merge Request of dask-traj. ... https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/-/merge_requests/50
(KKT) ... https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions
SRP fix in LAMMPS ... https://lammps.sandia.gov/doc/pair_srp.html
SRP fix ... https://lammps.sandia.gov/doc/pair_srp.html

Page 9

LAMMPS ... https://lammps.sandia.gov
E-CAM post-doc pilot project ... https://www.e-cam2020.eu/contact-joint-to-resolve-volume-constraints/
readme.rst of minDist2segments_KKT_for_SRP. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/minDist2segments_KKT_for_SRP/readme.html

Merge Request of minDist2segments_KKT_for_SRP. . . . https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/-/merge_requests/246
readme.rst of velocities_resolve_EVC_for_LAMMPS. . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/velocities_resolve_EVC_for_LAMMPS/readme.html
Merge Request of velocities_resolve_EVC_for_LAMMPS. . . . https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/-/merge_requests/247
pull request . . . https://github.com/lammps/lammps/pull/2626

Page 10

CabanaMD . . . https://github.com/ECP-copa/CabanaMD
ECP proxy application . . . https://proxyapps.exascaleproject.org/
readme.rst of n2p2 - Improved link to HPC MD software . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/n2p2/n2p2_improved_link_hpc/readme.html
Merge Request of n2p2 - Improved link to HPC MD software . . . https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/-/merge_requests/248
proposed recently . . . https://arxiv.org/abs/2010.14414
readme.rst of n2p2 - Polynomial Symmetry Functions . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/n2p2/n2p2_polynomial_symfuncs/readme.html
Merge Request of n2p2 - Polynomial Symmetry Functions . . . https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/-/merge_requests/281
Stable release 29 October 2020 . . . https://github.com/lammps/lammps/releases/tag/stable_29Oct2020

Page 11

fix adapt . . . https://lammps.sandia.gov/doc/fix_adapt_fep.html
Particle Insertion: Core . . . https://gitlab.e-cam2020.eu/mackernan/particle_insertion
readme.rst of lammps_pyinterfaceExt. . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/PI/lammps_pyinterfaceExt/readme.html
Merge Request of lammps_pyinterfaceExt. . . . https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/-/merge_requests/282
PICore . . . https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/-/tree/master/Classical-MD-Modules/modules/PIcore
PIhydration . . . https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/-/tree/master/Classical-MD-Modules/modules/PIhydration
LAMMPS . . . https://lammps.sandia.gov/
PyLammps . . . https://lammps.sandia.gov/doc/Python_pylammps.html
the singularity of insertion . . . https://doi.org/10.1080/00268979300102371
readme.rst of PIUtils. . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/PI/PIutils/README.html
Merge Request of PIUtils. . . . https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/-/merge_requests/266

Page 13

E-CAM pilot project on binding kinetics . . . https://www.e-cam2020.eu/pilot-project-biki/
Contact Maps . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/contact_maps/readme.html
Contact concurrences . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/contact_concurrences/readme.html
Integrating LAMMPS with OpenPathSampling . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/lammps_ops/readme.html
Gromacs engine in OpenPathSampling . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/OpenPathSampling/ops_gromacs_engine/readme.html
PLUMED Wrapper for OpenPathSampling . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/OpenPathSampling/ops_plumed_wrapper/readme.html
jobqueue-features library . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/HTC/decorators/readme.html
SimStore: OPS New Storage Subsystem . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/OpenPathSampling/ops_new_storage/readme.html
D.E. Shaw Research . . . https://www.deshawresearch.com/downloads/download_trajectory_sarscov2.cgi/
here . . . https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/index.html#openpathsampling
EP3265812A2, 2018-01-10 . . . https://worldwide.espacenet.com/patent/search?q=pn%3DEP3265812A2
WO2018047110A1, 2018-03-15 . . . https://worldwide.espacenet.com/patent/search?q=pn%3DWO2018047110A1

Page 14

here ... https://www.e-cam2020.eu/from-rational-design-of-molecular-biosensors/
here ... https://www.e-cam2020.eu/proof-of-concept-metamorphosis/
From idea to market ... https://www.e-cam2020.eu/from-idea-to-market/
Implementation of contact joint to resolve excluded volume constraints ... https://www.e-cam2020.eu/contact-joint-to-resolve-volume-constraints/
pilot project webpage ... https://www.e-cam2020.eu/contact-joint-to-resolve-volume-constraints/
Implementation of neural network potentials for coarse-grained models ... https://www.e-cam2020.eu/implementation-of-neural-network-potentials-for-coarse-grained-models/
here ... https://github.com/lammps/lammps/pull/2626
CabanaMD ... https://github.com/ECP-copa/CabanaMD
Kokkos ... https://github.com/kokkos/kokkos
here ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/index.html#n2p2
https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/ ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/index.html
here ... https://zenodo.org/record/4046053#.YENS9JNKi3J

Page 15

here ... https://arxiv.org/abs/1808.07345
here ... https://zenodo.org/record/1483844#.YENVeZNKi3K
here ... https://zenodo.org/record/1043242#.YENVo5NKi3I
here ... https://zenodo.org/record/814188#.YENWXZNKi3I
here ... https://arxiv.org/abs/1512.01162
here ... https://www.e-cam2020.eu/tag/classical-md/
here ... https://www.e-cam2020.eu/deliverables/
workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2017/02/SAW-WP1.pdf
workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2019/03/ECAM-State-of-the-Art-Workshop-Large-scale-activated-event-simulations.pdf
workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2019/05/E-CAM-SCOW-IIT-Report_web.pdf

Page 16

workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2018/09/SCOW_Solubility_prediction_REPORT.pdf
workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2020/05/SCOW-EES-Report.pdf
Workshop page ... https://www.cecam.org/workshop-details/10
Workshop page ... https://www.cecam.org/workshop-details/81
Workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2019/06/ESDW19_ClassicalMD_REPORT.pdf
Workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2019/07/ESDW9_Intelligent_HTC_REPORT.pdf
Workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2018/06/ESDW6_Classical_MD_REPORT.pdf
Workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2018/06/Trajectory_sampling_REPORT.pdf
PRACE/E-CAM Tutorial on Machine Learning and Simulations ... https://www.cecam.org/workshop-details/1
High Throughput Computing with Dask ... https://www.cecam.org/workshop-details/1022
here ... https://www.youtube.com/playlist?list=PLmhmpa4C4MzZ2_AUSg7Wod62uVwZdw4Rl

---

[1] Christoph Dellago, David Swenson, Donal MacKernan, Ralf Everaers, and Jony Castanga. Identification/selection of E-CAM MD codes for development, November 2016. URL https://doi.org/10.5281/zenodo.841694.

[2] Kresten Lindorff-Larsen, Paul Maragakis, Stefano Piana, and David E. Shaw. Picosecond to millisecond structural dynamics in human ubiquitin. *J. Phys. Chem. B*, 120(33):8313, 2016.

[3] Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98(14):146401, April 2007. doi: 10.1103/PhysRevLett.98.146401.

[4] Alan Ó Cais. Esdw technical software guidelines i, March 2016. URL https://doi.org/10.5281/zenodo.841735.

[5] David W. H. Swenson, Jan-Hendrik Prinz, Frank Noe, John D. Chodera, and Peter G. Bolhuis. OpenPathSampling: A Python framework for path sampling simulations. 1. Basics. *Journal of Chemical Theory and Computation*, 15 (2):813–836, 2019. doi: 10.1021/acs.jctc.8b00626. URL https://doi.org/10.1021/acs.jctc.8b00626. PMID: 30336030.

[6] David W. H. Swenson, Jan-Hendrik Prinz, Frank Noe, John D. Chodera, and Peter G. Bolhuis. OpenPathSampling: A Python framework for path sampling simulations. 2. Building and customizing path ensembles and sample schemes. *Journal of Chemical Theory and Computation*, 15(2):837–856, 2019. doi: 10.1021/acs.jctc.8b00627. URL https://doi.org/10.1021/acs.jctc.8b00627.

[7] Milosz Bialczak, Alan O'Cais, Mariusz Uchronski, and Adam Wlodarczyk. Intelligent htc for committor analysis, November 2020. URL https://doi.org/10.5281/zenodo.4382017.