



Quantum Dynamics E-CAM modules II

E-CAM Deliverable 3.3

Deliverable Type: Other

Delivered in March, 2018



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	D3.3
Deliverable Nature	Other
Dissemination Level	Public
Contractual Date of Delivery	Project Month 26(1 st December, 2017)
Actual Date of Delivery	30 th March, 2018
Deliverable Description	6 software modules delivered to the E-CAM repository in the area of quantum dynamics based on user requests and their documentation.

Document Control Information

Document	Title:	Quantum Dynamics E-CAM modules II
	ID:	D3.3
	Version:	As of 30 th March, 2018
	Status:	Accepted by WP leader
	Available at:	https://www.e-cam2020.eu/deliverables
Review	Document history:	Internal Project Management Link
Review	Review Status:	Reviewed
Authorship	Written by:	Sara Bonella(EPFL)
	Contributors:	Momir Mališ (EPFL), Alan O’Cais (Juelich Supercomputing Centre), Liang Liang (Maison de la Simulation)
	Reviewed by:	Alan O’Cais (Juelich Supercomputing Centre)
	Approved by:	Sara Bonella (EPFL)

Document Keywords

Keywords:	E-CAM, Quantum Dynamics, Module, Time Correlation Functions, Qubits, Local Control theory, CECAM
-----------	--

30th March, 2018

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.

Copyright notices: This deliverable was co-ordinated by Sara Bonella¹ (EPFL) on behalf of the E-CAM consortium with contributions from Momir Mališ (EPFL), Alan O’Cais (Juelich Supercomputing Centre), Liang Liang (Maison de la Simulation) . This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0>.



¹sara.bonella@epfl.ch

Contents

Executive Summary	1
1 Introduction	2
1.1 Overall scope of the module set	2
1.2 General applications and possible exploitation of the codes	2
2 Modules	4
2.1 LocConQubit	4
2.1.1 Practical application and exploitation of the code	4
2.2 OpenQubit	4
2.2.1 Practical application and exploitation of the code	4
2.3 PaPIM	5
2.3.1 Practical application and exploitation of the code	5
2.3.2 PaPIM code performance analysis	5
2.4 PIM_wd	8
2.4.1 Practical application and exploitation of the code	8
2.5 PIM_qcf	8
2.5.1 Practical application and exploitation of the code	8
2.6 Openmpbeads	8
2.6.1 Practical application and exploitation of the code	8
3 Conclusion and Outlook	10
References	11

List of Figures

1 Strong scalability test for the PaPIM code using the CH ₅ ⁺ system on the Jureca cluster.	6
2 Parallel efficiency test for the PaPIM code using the CH ₅ ⁺ system on the JUQUEEN cluster.	6
3 Timeline view of this application runs on a single node using 4 cores (left) and on two nodes with 32 cores (right). Colors indicate the state of the processes: running (blue), MPI routine (orange and yellow), MPI process not yet created (write) and I/O (brown).	7
4 The breakdown view of the MPI calls during the sampling procedure for the 4 cores test displayed in Figure 3. In region A the input data is read from a file and then distributed among cores. Region B starts the sampling steps in parallel and then calculates the average results. Region C writes data to the output file. Colors indicate the MPI calls in: MPI_Bcast (yellow), MPI_Allredce (light red), MPI_Allgather (grey) and MPI_Barrier (red).	7

Executive Summary

In this report for Deliverable 3.3 of E-CAM, 6 software modules in quantum dynamics are presented. Four modules stem from some of the activities performed during the [ESDW held by E-CAM at University College Dublin in July 2017](#) and originate from input of E-CAM's academic user base. The other two modules were developed following discussions with our industrial partner IBM, in the framework of E-CAM's pilot project on quantum computing.

Following the order of presentation, the 6 modules are named: `LocConQubit`, `OpenQubit`, `PaPIM`, `PIM_wd`, `PIM_qcf`, `Openmpbeads`. They include code for generation of controlled pulses for qubits and for calculation of quantum time correlation functions and their documentation. In this report, a short description is written for each module, followed by a link to the respective Merge-Request on the GitLab service of E-CAM. These merge requests contain detailed information about the code development, testing and documentation of the modules. A performance analysis for `PaPIM`, a package merging the functionality of several modules for quantum dynamics developed in E-CAM and structured to act as a high-performance container for future modules, is also presented. This analysis was performed by the E-CAM software group, in collaboration with the [POP](#) Center of Excellence for Computing Applications.

1 Introduction

Work Package 3 (WP3) of E-CAM provides a means for academic and industrial users of addressing computational questions that involve quantum dynamics, in particular, by developing scalable, open source software with verified quality standards, appropriate documentation and testing. E-CAM's activities in this area can be divided in two, connected, main pillars: the production of modules solicited by our user base (academic and industrial) and the creation of an environment to foster the transition from in house software to high performing and reliable community packages with long-term sustainability. To that end, following on the activities described in deliverable D3.1: Quantum dynamics E-CAM modules I [1], we have:

- Created modules to assist development of superconducting qubits;
- Continued to develop modules for computing quantum time correlation functions;
- Completed first assembly and performed initial scaling tests of PaPIM, E-CAM's overall package for quantum dynamics modules.

The module developments are done on the E-CAM's [GitLab](#) server. This Server allows full control of software development, including, for example, the private hosting of the code.

In order to describe each module, a module documentation repository is created in an [E-CAM GitLab repository](#) which automatically updates documentation for the [E-CAM Quantum Dynamics Module Documentation on Read the Docs](#). To submit a module, a developer can fork the GitLab repository and create his own feature branch in which he documents his latest progress about module development, source code link/patch file, new feature commit, etc. The developer can ask to merge their branch into the E-CAM master branch, which is called a Merge-Request. The E-CAM software manager and developers then run the module through it's acceptance criteria (coding style and structure, source code documentation and module testing). The documentation will only appear as part of the [Work Package 3 Software Modules on ReadTheDocs.org](#) once the Merge-Request is accepted.

In the next section, a short description is provided for each module created, followed by a link to the Merge-Request on the GitLab service of E-CAM. More detailed information about code development, testing and documentation can be found there.

1.1 Overall scope of the module set

Quantum mechanics describes matter at typical atomic length and energy scales. As our ability to engineer materials reaches this scale, simulations of quantum properties acquire significant technological interest. Since exact numerical algorithms for solving this problem scale exponentially with the size of the system, brute force calculations are, in general, out of reach for the foreseeable future. E-CAM currently addresses this challenge via two complementary approaches: The development of software based on approximate methods for solving quantum dynamics, and collaboration with IBM in the area of quantum computing. In the first line of work, we report in particular our efforts on the systematic development and testing of a modular code to compute quantum time-correlation functions. These are the key microscopic quantities that must be calculated to obtain information about the macroscopic properties of the systems (e.g. velocity correlation functions relate to mass transport in the system, current correlations describe charge transport) or to understand, and manipulate towards specific goals, interactions with external probes (e.g. dipole correlations relate to vibrational spectra from which crucial information on the microscopic structure can be inferred, correlation of the initial and final wave function after photo excitation can be used to understand or predict reaction channels in photo chemistry, and methods based on time correlation can also be used to characterise, and suggest procedures to mimic, key natural reactions such as photosynthesis). In the second line of work, based on the [pilot project](#) pursued in collaboration with our industrial partner IBM, we are developing software to automatize part of the development of logical gates in superconducting qubits. The modules developed are used to engineer sequences of electromagnetic pulses to control the characteristics of the qubits, in isolation and (in the longer term) keeping into account the effects of the environment. The development of quantum computers, with their potential for solving exponentially complex problems, has a clear strategic interest for quantum dynamics via the development of a new enabling hardware architecture.

1.2 General applications and possible exploitation of the codes

The modules developed in WP3 represent our on-going effort towards (1) the creation of modular packages with the ambition to become standard community codes and (2) developing simulation tools to assist the development of experimental protocols to engineer qubits.

PaPim modules

PaPIM is a package to study the properties of quantum materials, and in particular time correlation functions, via the so-called mixed quantum classical methods. In these schemes, quantum evolution is approximated by appropriately combining a set of classical trajectories for the system. Several quantum effects, for example the possibility to find atoms in classically forbidden regions (tunnelling), are reproduced at a manageable fraction of the cost of exact solutions. As described also in the E-CAM deliverable D3.1 [1], the E-CAM developments build on an in house code developed in the group of the WP3 leader (S. Bonella, EPF-Lausanne), which has been completely restructured to make it modular, and thus ready to act as a skeleton to include several computational methods in this family, and to achieve very good scalability (see E-CAM deliverable D7.2: E-CAM Software Porting and Benchmarking Data I [2]). This code was identified as a good target for E-CAM in the State of the Art workshop on quantum dynamics (see E-CAM deliverable D8.1: Industrial Engagement I²). Tests for scaling and optimization were discussed at the [E-CAM HPC meeting at the University of Barcelona](#) and performed in collaboration with the [POP](#) Center of Excellence for Computing Applications. PaPIM is currently used in collaborations with several groups: N. English, University College Dublin, to study properties of H₂ molecules in clathrates (materials for capture and storage of hydrogen and CO₂ in energy applications; F. Finocchi, Université Pierre et Marie Curie, to study infrared characterization of molecules, and from it understand the effect that the environment has on their chemical properties, in the atmosphere; R. Vuilleumier, Ecole Normale Supérieure Paris, to characterize hydrogen at extreme pressures in the context of geophysical applications; and D. Marx, University of Bochum, to test new potentials to efficiently characterize the chemical reactivity of small water clusters, again with possible applications on the physics of the atmosphere in reactions related to green house effect. These collaborations were initiated last year and are on-going. In this reporting period, the set of modules enabling full PaPIM calculations has been completed and assembled. Systematic tests of the scaling properties of the code have been performed.

Qubit modules

The modules developed are additions to the open source software [QuTiP](#). Their content and characteristics were tuned to satisfy the needs of our industrial partner, but the outcome of the work is interesting for a broader community and extends significantly the functionality of QuTiP. The modules were tested and transferred to the industrial partner. The pulses constructed in silico via our software developments are currently being tested in experiments conducted at IBM Research Laboratory in Rüschlikon with very encouraging results.

²Sep. 2017, confidential document, only available for members of the consortium (including the Commission Services).

2 Modules

In this section, a short description is written for each module, followed by a link to the Merge-Request on [GitLab service of E-CAM](#), which shows detailed information about code development, testing and documentation.

2.1 LocConQubit

The `LocConQubit` module implements the Local Control Theory,[3, 4] an algorithm for on-the-fly construction of a time-dependent potential that drives the evolution of a Hamiltonian towards one of its eigenstates. The algorithm is applicable to any Hamiltonian that is separable into a time-dependent and into a time-independent part, where the first part is directly incorporated into the algorithm, while the latter defines the basis of system states from which a designated target state is selected. States with vanishing interaction elements cannot be treated with the aforementioned algorithm. The algorithm is fine-tuned by the user with a single parameter in order to assure physical range of the generated time-dependent potential. This free parameter can be time-dependent while certain constrains in pulse generation can be directly incorporated into the algorithm. The module is accompanied with subroutines for pulse frequency analysis, post-processing, fidelity calculation and visualization of pulses and system evolution. The module is written in Python 3 programming language and is an addition to the open source [QuTiP](#) software package. The module uses the OpenMP functionalities available in QuTiP to parallelize the calculation of the pulse fidelity in order to search more efficiently for an optimal control pulse.

2.1.1 Practical application and exploitation of the code

This module enables to construct more efficient control pulses for superconducting transmon qubits coupled to a single tunable coupler whose energy is controlled with an external electromagnetic pulse. By properly modulating the energy of the tunable coupler with an external control pulse, the coupler operates as a quantum logic gate between coupled qubits. To improve gate performance and thus overall performance of quantum computers, pulses are tailored to make gate operations faster while maintaining at the same time the highest possible fidelity. The Local Control Theory was applied to these systems to generate efficient state preparation pulses which transfer populations completely from one qubit state to the other, as well as pulses for the SWAP gates which completely exchange quantum states between two qubits. A set of pulses capable of transferring populations with a full fidelity to designated target states was generated and, by post-processing this set, an optimal set of pulses for experimental implementation was obtained. This set is currently being tested at IBM. In parallel, capabilities as well as limits of the Local Control theory to manipulate such systems have been investigated in detail. Results of this work are going to be published in two scientific papers. The module will serve also as the basis for further implementation of new functionalities (e.g. addition of pre- and post-constrains into the Local Control Theory, extensions for open quantum systems etc.). In addition, the current OpenMP parallelization will be upgraded with a more advance parallelization scheme that will enable more efficient utilization of HPC resources and an easier implementation of parallelized optimization techniques.

Merge Request	Merge-Request of LocConQubit module.
Direct Documentation Link	readme.rst of LocConQubit module.

2.2 OpenQubit

The `OpenQubit` module is a patch to the `LocConQubit` module extending it to simulate the effects of control pulses in more realistic conditions in which the dissipating effects of an environment are modelled. The module incorporates the Lindblad master equation[5, 6] into the system propagator. This can be used directly, in conjunction with the Local Control Theory, to generate a control pulse, or to examine via post-processing the dissipating effects of previously constructed control pulses.

2.2.1 Practical application and exploitation of the code

Dissipating effects compete with target gate's operations and affect fidelity. Taking them into account is crucial for making contact with experiments and develop an effective and accurate simulation protocol to pre-select control pulses. This module enables to model, in particular, decoherence effects (T_1 and T_2) in superconducting transmon qubits, providing a more realistic description of the experiments currently conducted at IBM.

Merge Request	Merge-Request of OpenQubit module.
Direct Documentation Link	readme.rst of OpenQubit module.

2.3 PaPIM

The PaPIM module is a high performance code for calculating time correlation functions of systems in thermal equilibrium. It assembles efficiently the complementary functionalities of independent E-CAM modules, into a Fortran 90/95 MPI parallelized package for calculating system's time dependent observables. The code represents the current optimised assembly of the following modules:

- PIM_wd and PIM_qcf modules (see below) for exact quantum sampling of the Wigner phase space probability distribution function and the corresponding calculation of specific quantum correlation functions, respectively;
- [ClassMC](#) module (described in D3.1 [1]) for Monte Carlo sampling of classical Maxwell-Boltzmann distribution and calculation of corresponding correlation-functions;
- [PotMod](#) module (described in D3.1 [1] and extended in this reporting period), a library for model potentials and interfaces to external codes for potential energy calculations used by the sampling modules. This module, whose development started in the last reporting period, has been enhanced with an interface (still under testing) to couple PaPIM with the [CP2K](#) package for electronic structure calculations;
- [AuxMod](#) module (described in D3.1 [1]) which provides a tailored set of MPI commands used for code parallelization as well as input handling subroutines.

Details of each module can be further found in the corresponding module description on the [E-CAM GitLab](#).

2.3.1 Practical application and exploitation of the code

The code has been extensively used for the calculation of the infrared absorption spectrum of CH_5^+ in the gas phase.[7] This highly flexible molecule is considered a standard benchmark of approximate quantum methods, and has experimental interest, for example, in the context of green chemistry. The calculations performed with PaPIM were used to benchmark both the PIM method for time-correlation functions[8] (with positive results) and, perhaps more importantly for this report, to realize the code performance analysis presented below. The results obtained for CH_5^+ are currently under preparation for publication. In addition, one master thesis at École Polytechnique Fédérale de Lausanne (EPFL) relied on calculations performed using the final version of the code. This thesis included initial investigations of the origin and quantum effects in the peaks of the infrared spectrum of small water cluster systems and a protonated water dimer system. Studies of this system, of high experimental interest in the physics of the atmosphere, are on-going.

Merge Request	Merge-Request of PaPIM module.
Direct Documentation Link	readme.rst of PaPIM module.

2.3.2 PaPIM code performance analysis

The core ingredient for calculating time-correlation functions in PaPIM (both in [ClassMC](#) and in [PIM_wd](#)) is Monte Carlo sampling of independent phase space points to be used as initial conditions for the dynamics. Results of the calculations converge with a, large, number of sampled points so the computational efficiency is highly dependent on the optimization and parallelization of the sampling procedure. In the current version of the code, the sampling procedure was parallelized using the Message Passing Interface (MPI), and a thorough analysis of the code's performance was made.

In fact, three independent performance analyses of the PaPIM code were conducted. Two were carried out by the E-CAM programmers Dr. Liang Liang at [Maison de la Simulation](#) and [IDRIS](#) (Saclay, France) and Dr. Alan O'Cais at [JSC](#) (Jülich, Germany), while the third was conducted by the Performance Optimisation and Productivity team of the [POP Center of Excellence](#). The three conducted performance analysis are referred as PA1, PA2 and PA3, respectively, in the following text. Firstly, a brief digest of the analysis reports is given, followed by the explanation of the observed performance, and strategies on how to eliminate/mitigate the observed performance issues.

In PA1 a strong scaling analysis of the PaPIM code using the [Scalasca](#) analysis tools, as well as the internal PaPIM code calculation time outputs has been performed. The analysis was conducted on the CH_5^+ system and performed on the [JURECA](#) cluster at the [JSC](#). Figure 1 displays the results of the performed strong scaling tests.

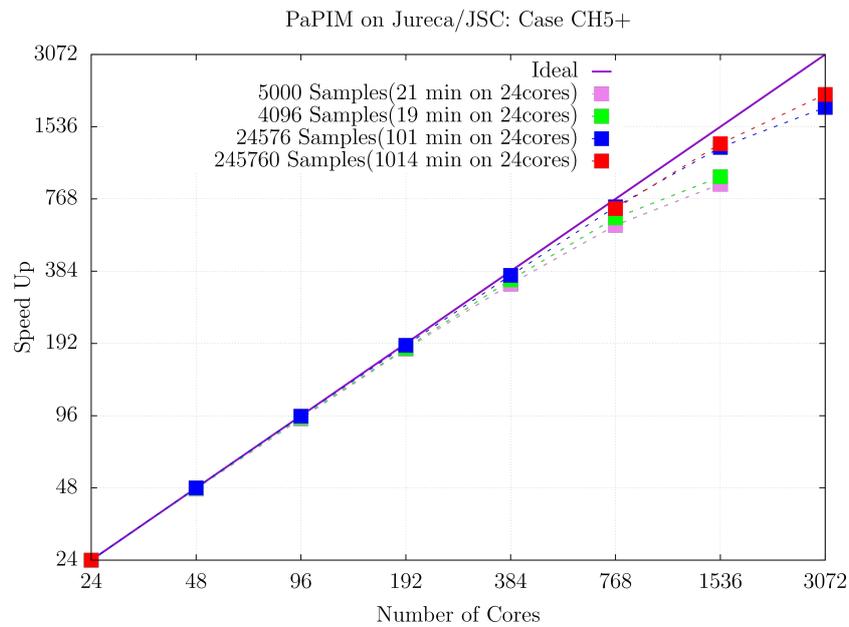


Figure 1: Strong scalability test for the PaPIM code using the CH₅⁺ system on the Jureca cluster.

In PA1 it was observed that the runtimes of MPI subroutines used in the code (namely MPI_INIT and MPI_ALLREDUCE) significantly increase for calculations with a large number of cores (384 or more) and approach the runtimes of the potential energy calculations. An imbalance of potential energy calculation runtimes for each core was also observed in cases when the number of sampling points was approaching the number of cores. The observed effect was reduced by increasing the number of total sampled points.

In PA2 a parallel efficiency test on the JUQUEEN cluster at JSC has been conducted. This analysis indicates that the code scales reasonably well. It also specifies that because the execution times of sampling subroutines during the tests were extremely short (average run time per sample point was only 8 ms), the efficiency seems less than the one that would result in production runs (where the average time per sampled point is of several minutes). This efficiency improves by increasing the number of sample points as clearly display in Figure 2, which is consistent with the observations in PA1. In PA2 the code has been successfully executed on 131,072 processor cores.

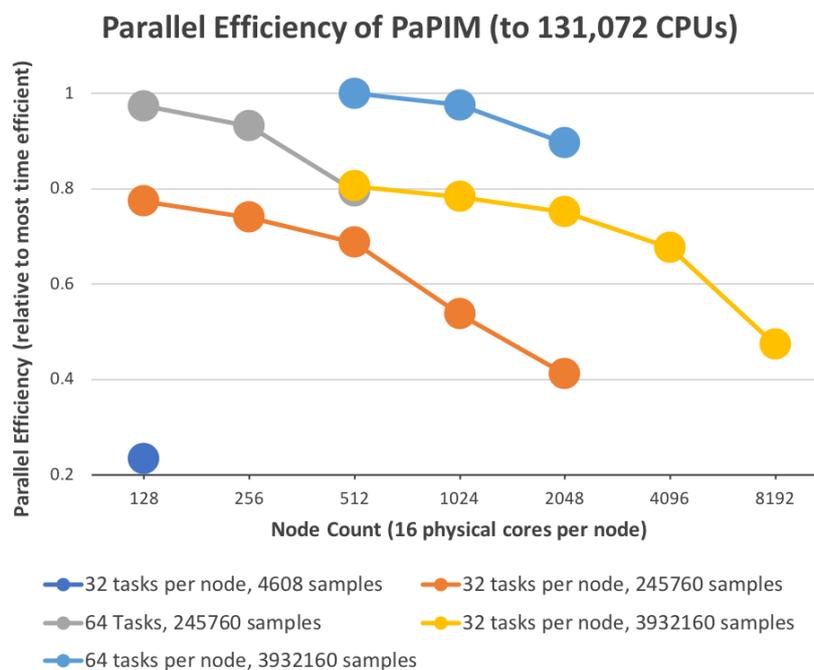


Figure 2: Parallel efficiency test for the PaPIM code using the CH₅⁺ system on the JUQUEEN cluster.

In PA3 a detailed analysis of the code's performance was conducted using the [extrae 3.4.1](#) software. The analysis was made on the OH system, and calculations were performed on the Fionn cluster of the [Irish Centre for High-End](#)

Computing (ICHEC) with 4 and 32 processor cores. The main analysis results obtained with 4 and 32 processor cores are shown in Figure 3.

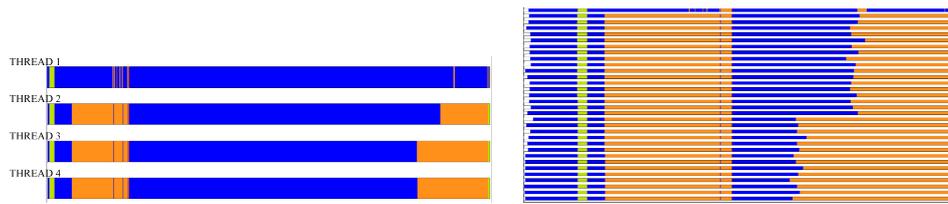


Figure 3: Timeline view of this application runs on a single node using 4 cores (left) and on two nodes with 32 cores (right). Colors indicate the state of the processes: running (blue), MPI routine (orange and yellow), MPI process not yet created (white) and I/O (brown).

Results obtained in PA3 are in agreement with those of PA1 and PA2. In particular, for the OH system, unequal load balance between processor cores was observed as the most severe bottleneck. Figure 4 displays the three distinctive MPI call regions of the code which were designated as A (input data read), B (sampling) and C (output data writing). The load balance decreases for the region A from an already low 33% at 4 cores to just 7% at 32 cores. Region C displays an even worse reduction from 26% to 4%, while the region B shows a smaller decrease from 93% to 73%. The total load balance decreases almost entirely due to the used MPI subroutines (MPI_Bcast, MPI_Allredce and MPI_Allgather) from 80% to 33% when increasing the number of cores from 4 to 32.

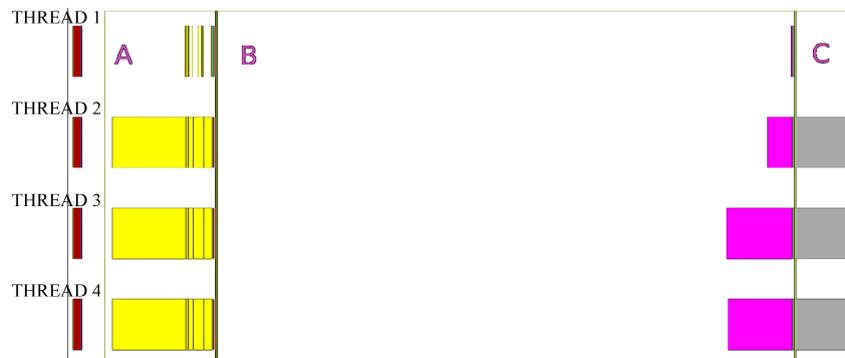


Figure 4: The breakdown view of the MPI calls during the sampling procedure for the 4 cores test displayed in Figure 3. In region A the input data is read from a file and then distributed among cores. Region B starts the sampling steps in parallel and then calculates the average results. Region C writes data to the output file. Colors indicate the MPI calls in: MPI_Bcast (yellow), MPI_Allredce (light red), MPI_Allgather (grey) and MPI_Barrier (red).

All analysis indicate that the currently used MPI subroutines are the main bottleneck for efficient scaling of the PaPIM code. This effect is enhanced and exhibited particularly in the test runs performed here, where the total number of trajectories is small compared to a usual production run, and where also the number of Monte Carlo and the number of auxiliary Monte Carlo steps is kept low. In these cases, the runtime of the sampling algorithm for an individual trajectory is quite short and approaches the runtime of the MPI subroutine for data distribution and acquisition. As observed by the POP team, one of the main reasons for this MPI bottleneck is the use of individual sequence passing of the read input variables at the beginning of the code and their individual sequential gathering at the end of the sampling procedure and during generation of output results. Although variables are arranged in derived data types, they are still currently passed as individual variables between the master core and slave cores. POP suggests using MPI_TYPE_CREATE_STRUCT command for communicating the derived type variables as whole between processor cores instead of separating them and passing individually as in the current implementation. While the variable passing between cores can be addressed, the MPI_INIT command cannot be amended in the same way and will remain the bottleneck, unless a modified communicator scheme is applied (e.g. splitting of MPI communicators). These two suggestions will be implemented into the PaPIM code in the near future. The advanced MPI commands for communicating the derived data types instead of each variable independently will further simplify the code and improve its scalability, while the scheme with the global MPI communicator splitting has already been considered as it will be used, in particular, to enable the external potential energy subroutines to work in parallel inside the already parallelized PaPIM code. It should also be noted that the current MPI_INIT and the few other MPI commands within the code will not cause significant performance reduction in real production runs, because the sampling runtime for the individual phase space point will be by several orders of magnitude larger than any sampling runtime performed in the tests. Therefore, we expect even better scalability in real production runs.

2.4 PIM_wd

The PIM_wd module implements the exact quantum Wigner probability distribution function sampling algorithm of the Phase Integration Method[8], and is the main subroutine for the quantum correlation function calculations in the PaPIM code. Module PIM_wd is written in Fortran 90/95.

2.4.1 Practical application and exploitation of the code

The module was developed as the main component of the PaPIM code (section 2.3), but also as a standalone subroutine that can be easily implemented in other methods (e.g. the whole family of so-called linearized approximations of quantum dynamics) for which phase space sampling of the exact quantum Wigner probability distribution is required. Because the Phase Integration Method samples a set of independent phase space points, independent instances of the PIM_wd module can be run in parallel in order to parallelize the phase space sampling. In the PaPIM package, the parallelization is accomplished using MPI, which has proved to provide good scalability of the code. This module was employed in all calculations and collaborations mentioned in section 1.2.

Merge Request	Merge-Request of PIM_wd module.
Direct Documentation Link	readme.rst of PIM_wd module.

2.5 PIM_qcf

The PIM_qcf module is a library of quantum correlation functions used by the Phase Integration Method for calculation of time-dependent correlation function values. Module PIM_qcf is written in Fortran 90/95 programming language.

2.5.1 Practical application and exploitation of the code

PIM_qcf module accompanies the PIM_wd sampling module in the calculation of time-dependent correlation functions from the sampled Wigner phase space probability distribution. While the latter module provides the initial phase space configuration, from which a trajectory propagating in time is initiated, and the corresponding probability density, the PIM_qcf calculates the corresponding correlation between the observable value determined at the initial sampled point and observable at each trajectory snapshot, giving a time correlation of the corresponding observable in time. Position and velocity auto-correlation functions for dipole operators in point charge approximations are currently implemented into the PIM_qcf library. Both have been extensively used in the calculation of CH₅⁺ system infrared spectrum, as well as other systems studied with the PaPIM code (see section 1.2). They are also intended as examples for user construction of new correlation functions within the PIM approach. More information on the practical use of PIM_qcf module for calculations for CH₅⁺ can be found in section 2.3.1.

Merge Request	Merge-Request of PIM_qcf module.
Direct Documentation Link	readme.rst of PIM_qcf module.

2.6 Openmpbeads

The Openmpbeads module is a functionality extension to the PaPIM (sec. 2.3) and PIM_wd (sec. 2.4) modules that enhances their performance by further parallelizing their sampling routines.

2.6.1 Practical application and exploitation of the code

Sampling of the quantum thermal density of a system of quantum particle is isomorphic to that of a system of classical polymers interacting via appropriate potentials (this is the so-called path integral formalism, first developed by Richard Feynman[9]). In order to accurately capture quantum effects, specifically those originating from tunneling of light particles, or very low temperatures, the polymer chains have to contain many beads. Since the potential energy has to be calculated for each polymer bead in the sampling procedure, polymer sampling subroutines become a bottleneck in these simulations. Openmpbeads is a patch to the PaPIM code which increases the code's performance by parallelizing the evaluation of the potential on the different beads of the polymer chain via OpenMP. In particular, the

module introduces OpenMP parallelization loops for the sampling of the positions of the beads.

Merge Request	Merge-Request of Openmpbeads module.
Direct Documentation Link	readme.rst of Openmpbeads module.

3 Conclusion and Outlook

The report of Deliverable 3.3 of E-CAM describes 6 Software modules in quantum dynamics related to the Extended Software Development Workshop (ESDW) held at University College Dublin in July 2017 (with a [wrap-up event](#) in March 2018) and to the quantum computing pilot project in collaboration with IBM. They include codes and documentations for approximate calculations of quantum time correlation functions, and codes for generation of control pulses in qubit systems. All of these modules have been accepted into the E-CAM software library at [this location](#).

Work for several other new modules was also initiated by the participants to the Dublin ESDW. These include the development of a prototypical interface of PaPIM with packages for electronic structure calculations (in particular CP2K), new sampling and dynamical procedures, and improved parallel implementation of the LocConQubit suite. Modules will also be developed as discussed in [deliverable D3.2: Identification/Selection of E-CAM Quantum Dynamics Codes for Development](#).

References

Acronyms Used

CECAM Centre Européen de Calcul Atomique et Moléculaire

WP Work-package

HPC High Performance Computing

ESDW Extended Software Development Workshop

MPI Message Passing Interface

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/142>
sara.bonella@epfl.ch ... <mailto:sara.bonella@epfl.ch>
<http://creativecommons.org/licenses/by/4.0> ... <http://creativecommons.org/licenses/by/4.0>

Page 1

ESDW held by E-CAM at University College Dublin in July 2017 ... <https://www.cecacm.org/workshop-0-1407.html>
POP ... <https://pop-coe.eu/>

Page 2

GitLab ... <https://gitlab.e-cam2020.eu/>
E-CAM GitLab repository ... <https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/tree/master/Quantum-Dynamics-modules>
E-CAM Quantum Dynamics Module Documentation ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/>
Read the Docs ... <https://readthedocs.org/>
Work Package 3 Software Modules ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/ReadTheDocs.org> ... <http://readthedocs.org>
pilot project ... <https://www.e-cam2020.eu/pilot-project-ibm/>

Page 3

E-CAM HPC meeting at the University of Barcelona ... <https://www.cecacm.org/workshop-1512.html>
POP ... <https://pop-coe.eu/>
QuTiP ... <http://qutip.org/index.html>

Page 4

GitLab service of E-CAM ... <https://gitlab.e-cam2020.eu/>
QuTiP ... <http://qutip.org/index.html>
Merge-Request of LocConQubit module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/13
readme.rst of LocConQubit module. ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/LocConQubit/readme.html>

Page 5

Merge-Request of OpenQubit module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/15
readme.rst of OpenQubit module. ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/OpenQubit/readme.html>
ClassMC ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/ClassMC/readme.html>
PotMod ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/PotMod/readme.html>
CP2K ... <https://www.cp2k.org/>
AuxMod ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/AuxMod/readme.html>
E-CAM GitLab ... <https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/tree/master/Quantum-Dynamics-Modules>

modules

Merge-Request of PaPIM module. ... https://gitlab.e-cam2020.eu:10443/e-cam/Quantum-Dynamics-Modules/merge_requests/20

readme.rst of PaPIM module. ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/PaPIM/readme.html>

Maison de la Simulation ... <http://www.maisondelasimulation.fr/en/index.php?a>

IDRIS ... <http://www.idris.fr>

JSC ... http://www.fz-juelich.de/ias/jsc/EN/Home/home_node.html

POP Center of Excellence ... <https://pop-coe.eu/>

Scalasca ... <http://www.scalasca.org>

JURECA ... http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JURECA/Configuration/Configuration_node.html

JSC ... http://www.fz-juelich.de/ias/jsc/EN/Home/home_node.html

Page 6

JUQUEEN ... http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUQUEEN/Configuration/Configuration_node.html

extrae 3.4.1 ... <https://tools.bsc.es/extrae>

Irish Centre for High-End Computing (ICHEC) ... <https://www.ichec.ie/about/infrastructure/fiomm>

Page 8

Merge-Request of PIM_wd module. ... https://gitlab.e-cam2020.eu:10443/e-cam/Quantum-Dynamics-Modules/merge_requests/25

readme.rst of PIM_wd module. ... http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/PIM_wd/readme.html

Merge-Request of PIM_qcf module. ... https://gitlab.e-cam2020.eu:10443/e-cam/Quantum-Dynamics-Modules/merge_requests/24

readme.rst of PIM_qcf module. ... http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/PIM_qcf/readme.html

Page 9

Merge-Request of Openmpbeads module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/14

readme.rst of Openmpbeads module. ... <http://e-cam.readthedocs.io/en/latest/Quantum-Dynamics-Modules/modules/Openmpbeads/readme.html>

Page 10

wrap-up event ... <https://www.cecarn.org/workshop-0-1407.html>

this location ... <https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/tree/master/Quantum-Dynamics-Modules/modules>

deliverable D3.2 ... <https://www.e-cam2020.eu/deliverables/>

Citations

- [1] S. Bonella, M. Mališ, G. Worth, D. Lauvergnat, and L. Liang, “Quantum dynamics E-CAM modules I,” Sep. 2017. [Online]. Available: <https://doi.org/10.5281/zenodo.1191081>
- [2] A. O’Cais, L. Liang, and J. Castagna, “E-CAM Software Porting and Benchmarking Data I,” Sep. 2017. [Online]. Available: <https://doi.org/10.5281/zenodo.1191428>
- [3] B. F. E. Curchod, T. J. Penfold, U. Rothlisberger, and I. Tavernelli, “Local control theory in trajectory-based nonadiabatic dynamics,” *Phys. Rev. A*, vol. 84, p. 042507, 2011. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.84.042507>
- [4] V. Engel, C. Meier, and D. J. Tannor, *Local Control Theory: Recent Applications to Energy and Particle Transfer Processes in Molecules*. John Wiley & Sons, Inc., 2009, pp. 29–101. [Online]. Available: <http://dx.doi.org/10.1002/9780470431917.ch2>
- [5] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems*. Oxford University Press, 2002. [Online]. Available: <https://books.google.ch/books?id=DkcJPwAACAAJ>
- [6] F. Tabakin, “Model dynamics for quantum computing,” *Annals of Physics*, vol. 383, pp. 33 – 78, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0003491617301252>

-
- [7] O. Asvany, P. K. P. B. Redlich, I. Hegemann, S. Schlemmer, and D. Marx, "Understanding the infrared spectrum of bare CH_5^+ ," *Science*, vol. 309, pp. 1219–1222, 2005. [Online]. Available: <http://science.sciencemag.org/content/309/5738/1219>
- [8] M. Monteferrante, S. Bonella, and G. Ciccotti, "Quantum dynamical structure factor of liquid neon via a quasiclassical symmetrized method," *The Journal of Chemical Physics*, vol. 138, p. 054118, 2013. [Online]. Available: <https://doi.org/10.1063/1.4789760>
- [9] R. P. Feynman and A. R. Hibbs, *Quantum mechanics and path integrals*, ser. International series in pure and applied physics. McGraw-Hill, 1965. [Online]. Available: <https://books.google.ch/books?id=14ApQAAMAAJ>