

Supercomputing Frontiers and Innovations

2023, Vol. 10, No. 3

Scope

- Future generation supercomputer architectures
- Exascale computing
- Parallel programming models, interfaces, languages, libraries, and tools
- Supercomputer applications and algorithms
- Novel approaches to computing targeted to solve intractable problems
- Convergence of high performance computing, machine learning and big data technologies
- Distributed operating systems and virtualization for highly scalable computing
- Management, administration, and monitoring of supercomputer systems
- Mass storage systems, protocols, and allocation
- Power consumption minimization for supercomputing systems
- Resilience, reliability, and fault tolerance for future generation highly parallel computing systems
- Scientific visualization in supercomputing environments
- Education in high performance computing and computational science

Editorial Board

Editors-in-Chief

- **Jack Dongarra**, University of Tennessee, Knoxville, USA
- **Vladimir Voevodin**, Moscow State University, Russia

Editorial Director

- **Leonid Sokolinsky**, South Ural State University, Chelyabinsk, Russia

Associate Editors

- **Pete Beckman**, Argonne National Laboratory, USA
- **Arndt Bode**, Leibniz Supercomputing Centre, Germany
- **Boris Chetverushkin**, Keldysh Institute of Applied Mathematics, RAS, Russia
- **Alok Choudhary**, Northwestern University, Evanston, USA
- **Alexei Khokhlov**, Moscow State University, Russia
- **Thomas Lippert**, Jülich Supercomputing Center, Germany

- **Satoshi Matsuoka**, Tokyo Institute of Technology, Japan
- **Mark Parsons**, EPCC, United Kingdom
- **Thomas Sterling**, CREST, Indiana University, USA
- **Mateo Valero**, Barcelona Supercomputing Center, Spain

Subject Area Editors

- **Artur Andrzejak**, Heidelberg University, Germany
- **Rosa M. Badia**, Barcelona Supercomputing Center, Spain
- **Franck Cappello**, Argonne National Laboratory, USA
- **Barbara Chapman**, University of Houston, USA
- **Yuefan Deng**, Stony Brook University, USA
- **Ian Foster**, Argonne National Laboratory and University of Chicago, USA
- **Geoffrey Fox**, Indiana University, USA
- **William Gropp**, University of Illinois at Urbana-Champaign, USA
- **Erik Hagersten**, Uppsala University, Sweden
- **Michael Heroux**, Sandia National Laboratories, USA
- **Torsten Hoefler**, Swiss Federal Institute of Technology, Switzerland
- **Yutaka Ishikawa**, AICS RIKEN, Japan
- **David Keyes**, King Abdullah University of Science and Technology, Saudi Arabia
- **William Kramer**, University of Illinois at Urbana-Champaign, USA
- **Jesus Labarta**, Barcelona Supercomputing Center, Spain
- **Alexey Lastovetsky**, University College Dublin, Ireland
- **Yutong Lu**, National University of Defense Technology, China
- **Bob Lucas**, University of Southern California, USA
- **Thomas Ludwig**, German Climate Computing Center, Germany
- **Daniel Mallmann**, Jülich Supercomputing Centre, Germany
- **Bernd Mohr**, Jülich Supercomputing Centre, Germany
- **Onur Mutlu**, Carnegie Mellon University, USA
- **Wolfgang Nagel**, TU Dresden ZIH, Germany
- **Alexander Nemukhin**, Moscow State University, Russia
- **Edward Seidel**, National Center for Supercomputing Applications, USA
- **John Shalf**, Lawrence Berkeley National Laboratory, USA
- **Rick Stevens**, Argonne National Laboratory, USA
- **Vladimir Sulimov**, Moscow State University, Russia
- **William Tang**, Princeton University, USA
- **Michela Taufer**, University of Delaware, USA
- **Andrei Tchernykh**, CICESE Research Center, Mexico
- **Alexander Tikhonravov**, Moscow State University, Russia
- **Eugene Tyrtshnikov**, Institute of Numerical Mathematics, RAS, Russia
- **Roman Wyrzykowski**, Czestochowa University of Technology, Poland
- **Mikhail Yakobovskiy**, Keldysh Institute of Applied Mathematics, RAS, Russia

Technical Editors

- **Andrey Goglachev**, South Ural State University, Chelyabinsk, Russia
- **Yana Kraeva**, South Ural State University, Chelyabinsk, Russia
- **Dmitry Nikitenko**, Moscow State University, Moscow, Russia
- **Mikhail Zymbler**, South Ural State University, Chelyabinsk, Russia


Contents

The Roofline Analysis of Special Relativistic Hydrodynamics Coarray Fortran Code I.M. Kulikov, I.G. Chernykh, D.A. Karavaev, V.G. Prigarin, A.F. Sapetina, I.S. Ulyanichev, O.R. Zavyalov	4
MHD-PIC Supercomputer Simulation of Plasma Injection into Open Magnetic Trap T.V. Liseykina, G.I. Dudnikova, V.A. Vshivkov, M.A. Boronina, I.G. Chernykh, I.S. Chernoshtanov, K.V. Vshivkov	11
CPU vs RAM in the Issue of <i>ab initio</i> Simulations of Doped Hafnium Oxide for RRAM and FRAM T.V. Perevalov, D.R. Islamov	18
Recurrent Monitoring of Supercomputer Noise Vad.V. Voevodin, D.A. Nikitenko	27
The Parallel Performance of SLNE Atmosphere–Ocean–Sea Ice Coupled Model R.Yu. Fadeev	36
Quantum-Chemical Study of Gas-Phase 5/6/5 Tricyclic Tetrazine Derivatives V.M. Volokhov, V.V. Parakhin, E.S. Amosova, D.B. Lempert, T.S. Zyubina	61
MOUSE2: Molecular Ordering Utilities for Simulations, Edition 2 M.K. Glagolev, A.A. Glagoleva, V.V. Vasilevskaya	73
Digital Twins in Large-Scale Scientific Infrastructure Projects D.V. Kosyakov, M.A. Marchenko	88



This issue is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

The Roofline Analysis of Special Relativistic Hydrodynamics Coarray Fortran Code

*Igor M. Kulikov*¹ , *Igor G. Chernykh*¹, *Dmitry A. Karavaev*¹,
*Vladimir G. Prigarin*¹, *Anna F. Sapetina*¹, *Ivan S. Ulyanichev*¹,
*Oleg R. Zavyalov*¹

© The Authors 2023. This paper is published with open access at SuperFri.org

Our previous papers are dedicated to the development of the first code for computational astrophysics using Coarray Fortran technology. The main result of the study of the developed code is the achievement of weak scalability at the level of MPI implementations, which allows to fully concentrate on using Coarray Fortran for developing new program codes. Coarray Fortran is based on the MPI directives, and helps software developer to create simple code without Send/Receive or synchronization commands. At the same time, such scalability can be achieved due to the weak implementation of the sequential part of the program code, which is characterized by frequent cache misses, inefficient memory usage and poor overall performance. In this article, we propose a method for analyzing program code performance using the roofline analysis. We used Intel Advisor software package from Intel oneAPI toolkit. High performance and efficient work with the memory of both individual key procedures and the code as a whole are demonstrated.

Keywords: HPC analysis, Coarray Fortran, high performance computing, roofline analysis.

Introduction

Many astrophysical phenomena, such as relativistic jets in active galactic nuclei [1], are characterized by relativistic velocities. Relativistic jets play an essential role in several important astrophysical processes, such as star formation, galactic binaries interaction, microquasars, active galaxies, and quasars physics. The main tool for studying relativistic jets is the mathematical modeling using high-performance computing systems [2]. We can find a lot of astrophysical papers dedicated to new astrophysical codes. But most of codes cannot be used for numerical simulation of real big problems, because they are not suitable to run on high-performance computing clusters. There are different strategies to parallelize computational code. It depends on mathematical and numerical models. In our case, because of the hydrodynamical approach, one of the best ways to maximize the performance is to optimize efficiency on each node using OpenMP and deep vectorization. Then we need to optimize data exchange between computational nodes of a cluster. In our research, we are using Coarray Fortran, which is based on MPI technology but is more reliable for software development. The most important advantage of the MPI 3.0 standard is the effective implementation of one-way communications, allowing to move to one of the most promising parallel programming models PGAS (Partitioned Global Address Space). Computational experiments on continuum mechanics models [3–5] have shown that Coarray Fortran code has the same scalability as MPI code. At the same time, the complexity of code development is noticeably reduced, which leads to the development of new libraries [7] and language extensions [8]. We should also note that the multidimensionality of the decomposition of calculations does not affect the performance of the code [9, 10]. Based on the Coarray Fortran technology, we have developed a new code for the numerical solution of equations of special relativistic hydrodynamics [6]. In this paper, we analyze the code's performance using Intel Advisor software from the Intel OneAPI toolkit.

¹Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russian Federation

In the next section, we briefly describe the structure of the developed code. The third section is devoted to the analysis of software implementation using The Roofline Analysis. The Roofline Analysis is very important for optimizing the code's performance by vectorization of loops inside a program. The fourth section will present the simulation results. The fifth section formulates the conclusion.

1. Special Relativistic Hydrodynamics Coarray Fortran Code

The mathematical apparatus and parallel implementation are described in detail in [6]. We will focus on the structure of the code presented in Fig. 1. The figure shows an enlarged block

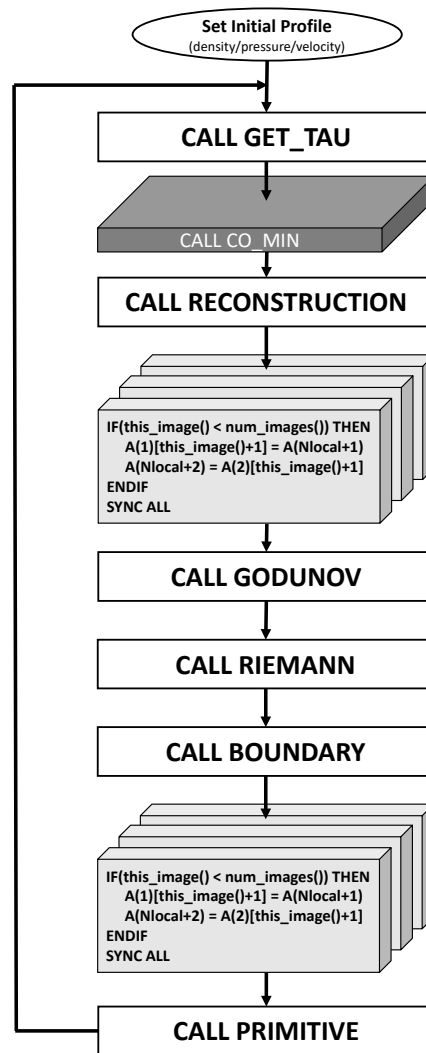


Figure 1. Code structure

diagram of the software implementation of the main computing cycle. The **GET_TAU** function calculates the time step in each subdomain. Then we use the Coarray Fortran reducing function **CO_MIN** to calculate the global minimum time step. At the next step, the **RECONSTRUCTION** function is called for a piecewise parabolic representation of physical variables (here we do not provide the entire list of functions). Its implementation requires an exchange of

overlap areas using Coarray Fortran. At the next stage, the **GODUNOV** function is called to implement the numerical method, which uses the **RIEMANN** and **BOUNDARY** functions. The latter function also requires the exchange of overlap areas. Then the physical variables are restored in the **PRIMITIVE** function.

2. Roofline Analysis

In our research, we used the Roofline model [11] to provide performance estimates of our astrophysical code running on a compute node based on two Intel Xeon Scalable 6248R processors with 24 cores each. Each node has 192 GB DDR4 RAM. For our tests, we did not use any data output for maximum performance. Roofline analysis was made by Intel Advisor [12] software from Intel oneAPI [13] software package. This software calculates performance values for each function of our code as well as the most compute-bound function. We can also compare performance values with peak performance values (scalar peak, single precision peak performance values, double precision vector peak, double precision FMA peak) of processors.

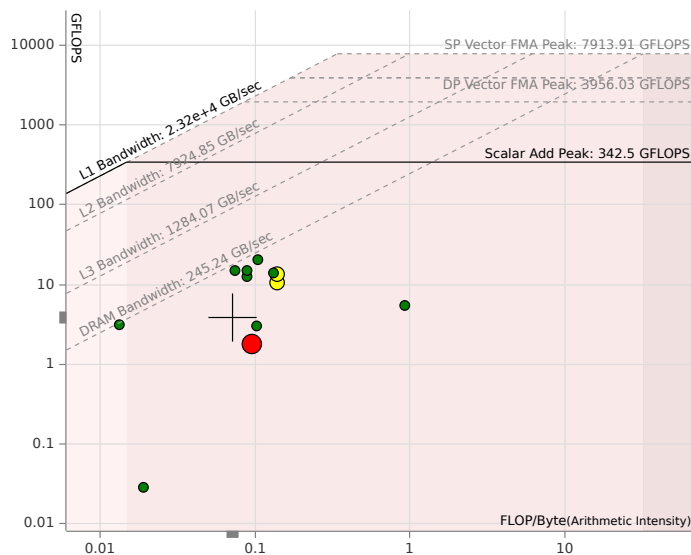


Figure 2. Roofline analysis results. Autovectorization by compiler turned off. Computational node: 2x Intel Xeon 6248R, 192 GB DDR4 RAM

This analysis can help to improve the performance of each function in developed code because Intel Advisor shows loops that can be autovectorized and loops whose structure should be improved for autovectorization. Why do we need to think about loop vectorization every second during the development of high-performance computing software? The peak performance of modern x86 processors from Intel or AMD as well as modern ARM processors is based on the CPU vector units. We can see that the scalar add peak performance for our compute node is about 342 GFLOPS and the dual precision vector FMA peak is about 3956 GFLOPS. The difference is about 10 times. Each core of the Intel Xeon Scalable Gold/Platinum processor has AVX512 registers and FMA mathematics instructions. These instructions can multiply and add eight double precision values at one CPU cycle. It means that we should help the compiler build vectorized loops for maximum performance. In this section, we will show the importance of auto-vectorization for code performance.

Figure 2 shows the results of roofline analysis for our astrophysical code without any compiler vectorization optimizations. The red dot is the the performance of the building parabola function

of the PPML method. Green and yellow dots are the performance of the other functions such as Riemann solver or part of a Godunov scheme. The total performance of the code is equal to 3.79 GFLOPS. The performance of the function with the longest calculation time (PPML method realization) is equal to 1.8 GFLOPS. Also, we can see that all functions in this optimization case are limited by the DRAM memory bandwidth.

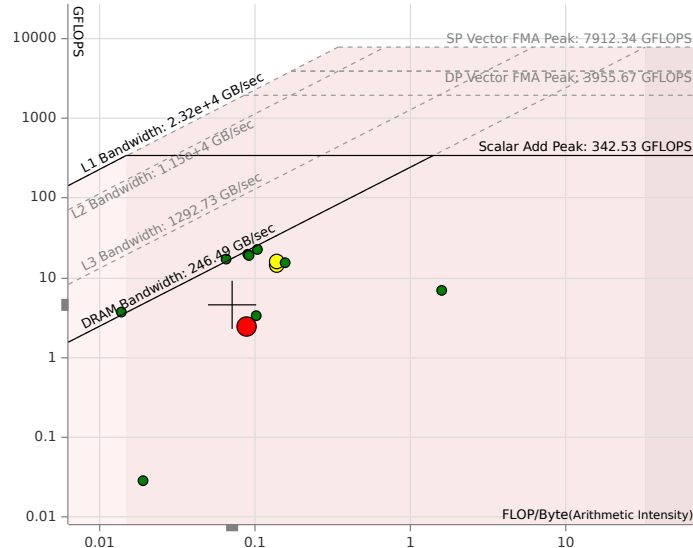


Figure 3. Roofline analysis results. AVX2 autovectorization by compiler. Computational node: 2x Intel Xeon 6248R, 192 GB DDR4 RAM

Figure 3 shows the results of roofline analysis for our code with AVX2 auto-vectorization optimizations made by Intel’s compiler. The red dot is the the performance of the building parabola function of the PPML method. Green and yellow dots are the performance of the other functions such as Riemann solver or part of a Godunov scheme. The total performance of the code is equal to 4.61 GFLOPS. The performance of the function with the longest calculation time is equal to 2.5 GFLOPS. This is the same function as shown in Fig. 2. We can see that the Riemann solver bandwidth in this optimization case became a little bit better than the DRAM memory bandwidth. AVX2 optimized code works with the math-related intrinsic functions which can use 256-bit double precision vectors containing 4 doubles. AVX2 provides instructions that fuse multiplication and addition by using FMA intrinsics which also works with 256-bit double precision vectors. These vector instructions are suitable for most Intel and AMD processors. But if you have the latest Intel Xeon Scalable processors you should use AVX512 vectorization for better performance. The next figure will show the advantages of AVX512 vector registers.

Figure 4 shows the results of roofline analysis with AVX512 auto-vectorization optimizations made by Intel’s compiler. The red dot is the the performance of the building parabola function of the PPML method. Green and yellow dots are the performance of the other functions such as Riemann solver or part of a Godunov scheme. The total performance of the code is equal to 7.73 GFLOPS. The performance of the function with the longest calculation time is equal to 3.66 GFLOPS. This is also the same function as shown in Figs. 2–4. We can see that the Riemann solver bandwidth in this optimization case is about 435 GB/sec with a performance of about 28.4 GFLOPS. AVX512 optimized code works with the math-related intrinsic functions which can use 512-bit double precision vectors containing 8 doubles. AVX512 provides instructions that fuse multiplication and addition by using FMA intrinsics which also works with 512-bit

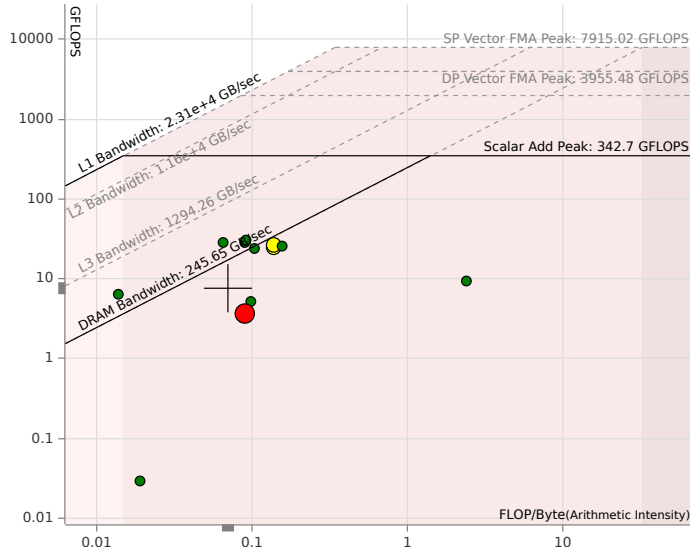


Figure 4. Roofline analysis results. AVX512 autovectorization by compiler. Computational node: 2x Intel Xeon 6248R, 192 GB DDR4 RAM

double precision vectors. The average estimated speed-up of vectorized code compared to the scalar version is equal to 5.5 times. And this is not the best result. Intel Advisor suggests some optimizations that can help build faster code. These optimizations are to add data padding, vectorize serialized functions, and convert some functions to Fortran SIMD-enabled functions.

For collecting the performance data, we use the same technique as in [14]. We did not change the source of our astrophysical code during tests. We only change the target architecture by adding `-ax` compiler option with a set of processor’s instructions which can be used for target code.

3. Numerical Modeling

The formulation of the problem of the relativistic jet evolution can be found in [2]. Figure 5 shows the results of modeling the evolution of the galactic jet. From the simulation results it is clear that a shock wave moves forward, the speed of waves propagation corresponds to the speed of light. Behind the shock front, there is a shell that separates the shock front and the hot region where the maximum temperature is reached. The internal part of the flow has a cocoon and is limited by the contact surface. On the outer side of the cocoon, closer to the base, currents of the reverse flow type propagate, which in turn interact with the jet flow. The characteristic development time of Kelvin–Helmholtz-type instability at the base of the jet is 6000 years, which corresponds to the results of the computational experiment.

Conclusions

Last decade, our group developed codes for numerical simulation of different astrophysical problems. We developed different codes based on the CUDA toolkit, MPI, and OpenMP technologies as well as C++ or Fortran languages. We had some C++ implementations based on AVX512 intrinsics, and this code version has the best performance on Intel Xeon Scalable processors. But this code cannot be used on AMD processors till AMD starts to produce their AVX512-based CPUs. Our latest codes are based on the Coarray Fortran extension, which was started as an extension of Fortran 95/2003 for parallel processing. At this time, Coarray Fortran-

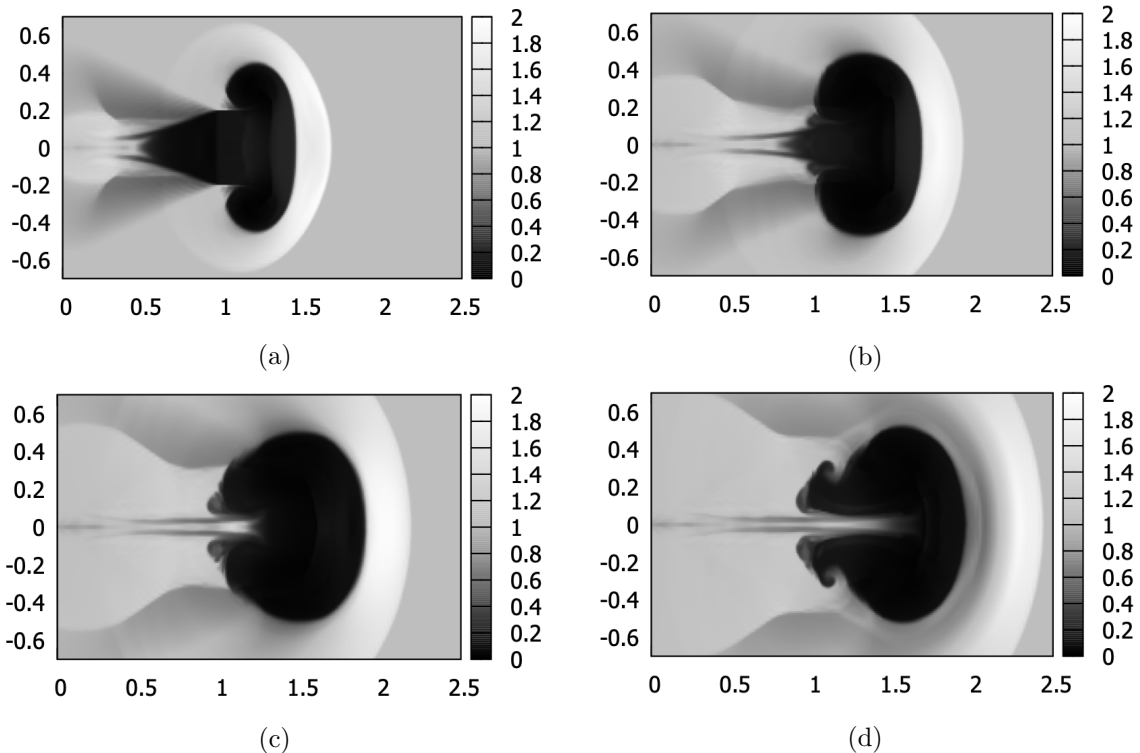


Figure 5. Gas density in the equatorial plane is 10^{-2} cm^{-3} at time points: 3000 years (a), 4500 years (b), 6000 years (c), 7500 years (d)

based codes have the same performance as the MPI codes. However, the creation of the program is much easier than that of the MPI code. Modern Fortran compilers understand this extension and can optimize codes. In this paper, we focused on the auto-vectorization results of the Intel Fortran compiler. We achieved two times the performance speed-up of our astrophysical code only by the compiler options. It is possible to speed up our code more in future with some recommendations from the Intel Advisor toolkit which was used for performance evaluation. The simulation results for the evolution of relativistic jet are in good accordance with the results from our earlier codes based on C++ MPI/OpenMP technologies. In our future works, we will continue to optimize our code, possibly for Advanced Matrix Extensions instruction set.

Acknowledgements

This work was supported by the Russian Science Foundation (project No. 23-11-00014) <https://rscf.ru/project/23-11-00014/>.







This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Sotomayor, P., Romero G.: Nonthermal radiation from the central region of super-accreting active galactic nuclei. *Astronomy & Astrophysics* 664, A178 (2022). <https://doi.org/10.1051/0004-6361/202243682>
2. Kulikov, I.: A new code for the numerical simulation of relativistic flows on supercomputers

- by means of a low-dissipation scheme. *Computer Physics Communications* 257, 107532 (2020). <https://doi.org/10.1016/j.cpc.2020.107532>
3. Reshetova, G., Cheverda, V., Khachkova, T.: A comparison of MPI/OpenMP and Coarray Fortran for digital rock physics application. In: *Parallel Computing Technologies. PaCT 2019. Lecture Notes in Computer Science*, vol. 11657, pp. 232–244 Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25636-4_19
 4. Reshetova, G., Cheverda, V., Khachkova, T.: Numerical experiments with digital twins of core samples for estimating effective elastic parameters. In: *Supercomputing. RuSC-Days 2019. Communications in Computer and Information Science*, vol. 1129, pp. 290–301. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36592-9_24
 5. Reshetova, G., Cheverda, V., Koinov, V.: Comparative efficiency analysis of MPI blocking and non-blocking communications with Coarray Fortran. In: *Supercomputing. RuSC-Days 2021. Communications in Computer and Information Science*, vol. 1510, pp. 322–336. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-92864-3_25
 6. Kulikov, I., Chernykh, I., Karavaev, D., *et al.*: A new parallel code based on a simple piecewise parabolic method for numerical modeling of colliding flows in relativistic hydrodynamics. *Mathematics* 10(11), 1865 (2022). <https://doi.org/10.3390/math10111865>
 7. Wang, Y., Li, Z.: GridFOR: a domain specific language for parallel grid-based applications. *International Journal of Parallel Programming* 44, 427–448 (2016). <https://doi.org/10.1007/s10766-014-0348-z>
 8. Kataev, N., Kolganov, A.: The experience of using DVM and SAPFOR systems in semi automatic parallelization of an application for 3D modeling in geophysics. *The Journal of Supercomputing* 75, 7833–7843 (2019). <https://doi.org/10.1007/s11227-018-2551-y>
 9. Shterenlikht, A., Cebamanos, L.: MPI vs Fortran coarrays beyond 100k cores: 3D cellular automata. *Parallel Computing* 84, 37–49 (2019). <https://doi.org/10.1016/j.parco.2019.03.002>
 10. Guo, P., Wu, J.: One-sided communication in Coarray Fortran: performance tests on TH-1A. In: *Algorithms and Architectures for Parallel Processing. ICA3PP 2018. Lecture Notes in Computer Science*, vol. 11337, pp. 21–33. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05063-4_3
 11. The Roofline Model. https://en.wikipedia.org/wiki/Roofline_model (2023), accessed: 2023-10-25
 12. Intel Advisor tutorial. <https://www.intel.com/content/www/us/en/docs/advisor/tutorial-roofline/2021-1/run-a-roofline-analysis.html> (2021), accessed: 2023-10-25
 13. Intel oneAPI overview. <https://www.intel.com/content/www/us/en/developer/tools/oneapi/overview.html> (2023), accessed: 2023-10-25
 14. Chernykh, I., Vorobyov, E., Elbakyan, V., Kulikov, I.: The impact of compiler level optimization on the performance of iterative Poisson solver for numerical modeling of protostellar disks. In: *Supercomputing. RuSCDays 2021. Communications in Computer and Information Science*, vol. 1510, pp. 415–426. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92864-3_32

MHD-PIC Supercomputer Simulation of Plasma Injection into Open Magnetic Trap

Tatyana V. Liseykina¹ , Galina I. Dudnikova¹ , Vitaly A. Vshivkov¹ ,
Marina A. Boronina¹ , Igor G. Chernykh¹ , Ivan S. Chernoshtanov²,
Konstantin V. Vshivkov¹ 

© The Authors 2023. This paper is published with open access at SuperFri.org

This paper presents a two-dimensional hybrid Magneto-Hydro-Dynamical-Particle-in-Cell numerical model to study the interaction of a beam of plasma, injected into an axisymmetric magnetic trap, with the background trap plasma. We apply a kinetic description for the positively charged ions, treat electrons as a massless charge neutralizing fluid and assume that the direct coupling between ions and electrons is due to the anomalous scattering on the fluctuations of electromagnetic fields only. The model adequately describes nonlinear nonstationary evolution of the plasma and of the magnetic field and allows to follow this evolution for large simulation times in a wide range of the initial magnetic field and plasma parameters. We show in particular, that the continuous injection of plasma beam leads to the displacement of the magnetic field and to the formation and growth of extended region with low amplitude of the field. The numerical results demonstrate the accumulation and capture of the plasma in the magnetic cavity region.

Keywords: particle-in-cell method, hybrid simulations, axisymmetric magnetic trap, diamagnetic “bubble”.

Introduction

Open (open-ended) magnetic trap is one possible solution to the problem of plasma confinement and heating in laboratory experiments [11]. The idea of such a facility was proposed in the 50^s by G.I. Budker [3] and R.F. Post [10] independently, and it was tested for the first time then. It uses the gradient of the magnetic field to trap particles, whereby the confinement of the particle is due to the adiabatic invariance of its magnetic moment, which takes place when the Larmor radius of the particle is small compared to the scale of change of the magnetic field. Although open traps can be operated under steady-state conditions and are attractive from the engineering point of view, they have an important drawback, namely the relatively high losses of plasma along the magnetic field lines. Several proposals of improved traps, that are largely free from this drawback, have been made in the last decades. The diamagnetic confinement is one of those. The main idea behind diamagnetic confinement is to suppress longitudinal losses from an axisymmetric open trap by creating a configuration with an extremely high plasma pressure equal to the magnetic field pressure. Diamagnetic confinement was proposed and theoretically justified in [1], where the magneto-hydro-dynamical (MHD) approximation was used to describe the equilibrium of a plasma with $\beta = 1$, with β being the ratio of the plasma pressure to the magnetic pressure. In particular, it has been shown that a monotonous increase of the plasma pressure in an open trap leads to the formation of a plasma filled region with a displaced magnetic field, the so-called diamagnetic “bubble”. Further increase of plasma pressure results in an increase of the “bubble” radius. The plasma confinement time grows proportionally to the “bubble” radius and can significantly exceed the confinement time in a vacuum magnetic field. Furthermore, the diamagnetic confinement regime with intense off-axis injection of atomic beams at an angle to the magnetic field may additionally suppress losses of the background

¹Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russian Federation

²Budker Institute of Nuclear Physics SB RAS, Novosibirsk, Russian Federation

plasma, raise the temperature of electrons and the lifetime of fast ions arising from the trapping of injected atomic beams by the background plasma [5]. In this paper we present a hybrid MHD-Particle-in-Cell (PIC) numerical model to study the formation of the diamagnetic mode in an axisymmetric open magnetic plasma trap with continuous injection of a plasma beam. The paper is organized as follows. In Section 1, we describe the main aspects of our implementation of the hybrid model, including parallelization strategy³. In Section 2, we present typical results from MHD-PIC simulations of the interaction of a plasma beam injected into a cylindrical trap with the background trap plasma.

1. Numerical Algorithms and Methods

In our hybrid model, we use a kinetic description for the positively charged ions and treat electrons as a *massless* charge neutralizing fluid. This approach is justified because the relevant time and space scales are determined by the ions. For simplicity in the description of our model we consider the case of hydrogen plasma. The evolution of the ion distribution function $f(\vec{r}, \vec{v}, t)$ is governed by the Vlasov equation

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla f + \frac{1}{m_i} \left[q_i \left(\vec{E} + \frac{1}{c} (\vec{v} \times \vec{B}) \right) - \vec{R} \right] \frac{\partial f}{\partial \vec{v}} = 0, \quad (1)$$

while the motion of the electron fluid follows

$$n_e m_e \left(\frac{\partial \vec{V}_e}{\partial t} + (\vec{V}_e \cdot \nabla) \vec{V}_e \right) \equiv 0 = q_e n_e \left(\vec{E} + \frac{1}{c} (\vec{V}_e \times \vec{B}) \right) - \nabla p_e + n_e \vec{R}. \quad (2)$$

Here m_i, m_e and $q_i, q_e = -q_i$ are the ion and electron mass and charge, \vec{E} and \vec{B} are the electric and magnetic fields, n_e is number density of the electrons. We take into account the resistive coupling between electrons and ions via $\vec{R} = \nu m_e (\vec{V}_i - \vec{V}_e)$, with ν being the electron-ion collision frequency which does not depend on the plasma and magnetic field parameters⁴, $\vec{V}_e, \vec{V}_i = \frac{\int \vec{v} f d\vec{v}}{\int f d\vec{v}}$ denote the electron and ion mean velocities. Finally, we assume that the electron pressure p_e is scalar, and that the *quasineutrality* condition $n_e = n_i = \int f d\vec{v}$ is fulfilled. Since we consider only low-frequency processes and do not take into account the displacement current, the total current density $\vec{J} = q_i n_i (\vec{V}_i - \vec{V}_e)$ is obtained from Amperes law and the magnetic field evolves according to Faradays law

$$\nabla \times \vec{B} = \frac{4\pi}{c} \vec{J}, \quad \frac{1}{c} \frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E}. \quad (3)$$

The electric field is calculated from the electron momentum equation (2)

$$\vec{E} = - \left(\frac{1}{c} (\vec{V}_e \times \vec{B}) \right) + \frac{\nabla p_e}{q_e n_e} - \frac{\vec{R}}{q_e}. \quad (4)$$

Finally, the equation for the electron temperature reads

$$n_e \left(\frac{\partial T_e}{\partial t} + (\vec{V}_e \cdot \nabla) T_e \right) + (\gamma - 1) p_e \nabla \cdot \vec{V}_e = (\gamma - 1) [Q_e - \nabla \cdot (\kappa \nabla T_e)], \quad (5)$$

³The very detailed overview of the hierarchy of hybrid models together with the underlying equations and assumptions as well as possible extensions can be found in [9] and references therein.

⁴Note that we consider the collisionless plasmas and assume that the direct coupling between ions and electrons is due to the anomalous scattering on the fluctuations of electromagnetic fields only.

where $Q_e = \frac{J^2}{\sigma}$ is the heat generated in electrons, $\kappa \nabla T_e$ is the electronic heat flux, with κ being the thermal conductivity, $\sigma = \frac{e^2 n_e}{m_e \nu}$ – electric conductivity and $\gamma = \frac{5}{3}$ – the adiabatic index of an atomic gas. In our simulations lengths are given in units of c/ω_{pi} , with $\omega_{pi} = \sqrt{4\pi n_0 e^2/m_i}$ being the ion plasma frequency, and time in units of $1/\omega_{iH}$, where $\omega_{iH} = eB_0/(m_i c)$ is the ion cyclotron frequency with B_0 being the initial amplitude of the magnetic field in the center of the trap and n_0 is the initial background plasma density.

Our code employs state-of-the-art, widely used numerical algorithms and methods. In particular, the particle-in-cell (PIC) method, see e.g. [8], is used to solve the Vlasov equation for the *ion* components of the background plasma and of the injected beam⁵. The first-order weighting is applied to interpolate the fields to the ion positions and to obtain ion current and charge density from the positions of the ion-particles relative to the grid points. The Boris [2] pusher is used to propagate ion positions and velocities. The equations of the evolution of the electromagnetic fields and of the electron temperature are discretized using 2nd-order regular finite difference stencils. More details about the implemented numerical schemes can be found in [7, 12].

1.1. Initialization

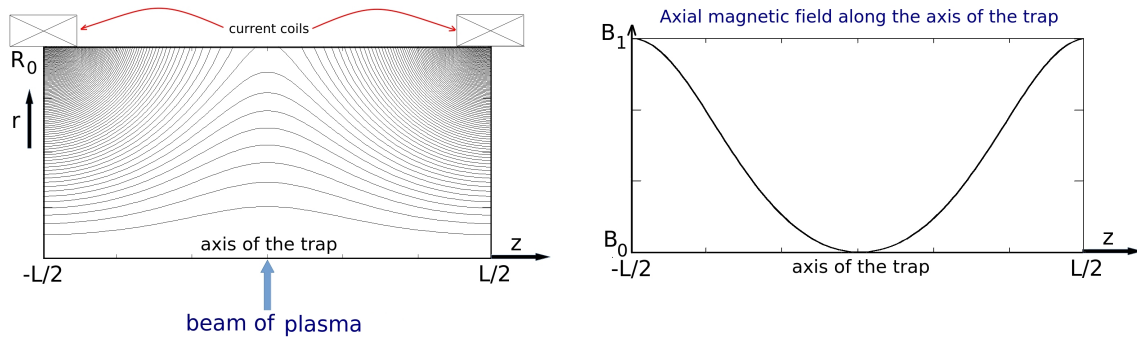
At $t = 0$ the cold uniform hydrogen plasma is located inside a cylindrical chamber of radius R_0 and length L with a magnetic field generated by a pair of identical current coils, located at both ends of the chamber. We assume that the current flows through the coils strictly in the azimuthal direction φ , and that its amplitude does not depend on the azimuthal angle. In this case at $t = 0$ the whole system has an axial symmetry. A beam of neutral hydrogen plasma with finite temperature is injected with constant velocity $|\vec{v}|$ into the chamber at the point $r = 0, z = 0$. The initial angular distribution of the velocity of the beam particles is randomized. In this paper we consider the reduced 2D cylindrical problem, with z -axis directed along the symmetry axis of the trap, i.e. we assume that the field configuration and the evolution of the plasma do not depend on the azimuthal angle through the whole process. In cylindrical geometry the simulation box is a rectangle $[-L/2, L/2] \times [0, R_0]$, Fig. 1a. The distribution of the initial magnetic field along the trap axis is shown in Fig. 1b. The amplitude of the field in the center of the trap at the point $r = 0, z = 0$ is equal to B_0 , the fields B_1 at the points $r = 0, z = \pm L/2$ are higher⁶.

1.2. Parallelization

The computational effort in PIC simulations scales with the number of simulation particles. For large numerical boxes, especially in two-dimensional (2D) or three-dimensional (3D) simulations, this number can be as high as $10^9 \div 10^{12}$. This makes the large simulations unfeasible on a desktop computer. In order to use the parallel computing capabilities of modern supercomputers, an efficient parallelization of the numerical code is required. Our algorithms are local and allow parallelization via domain decomposition. In particular, we divide the simulation domain

⁵The PIC approach is a mean to solve the Vlasov equations for the plasma distribution functions using pseudo particles of the same charge-to-mass ratio as the real particles in the plasma. Here we use the PIC method for the ion plasma component only.

⁶The ratio B_1/B_0 between the maximal and minimal values of the magnetic field on the axis of the trap is called the mirror ratio of the trap [4]. The basis of the mirror effect is the adiabatic invariance of the particle's magnetic moment.



(a) Background magnetic field is created by two identical current coils, located at both ends of the chamber. Black lines are the magnetic field lines (b) The nonuniform magnetic field of a simple pair of current coils forms two magnetic mirrors between which a plasma can be trapped [4]

Figure 1. Simulation setup and the distribution of the magnetic field along the trap axis

into subdomains in z -direction. Each subdomain is assigned to a *group* of processor cores and the particles belonging to the subdomain are distributed among the cores of the group. The maximal number of groups is defined by the grid size and should not exceed $N_z/4$, where N_z is the number of grid cells in z -direction. At the initial stage, the background particles and the particles of the injected beams are distributed evenly between the cores of their group. While the particle positions and velocities can be updated on each subdomain independently, communication between neighboring domains is necessary. The particles crossing the domain boundaries are marked during the update of the positions, collected and then sent (all within a single MPI SEND call) to the respective neighboring domains. Additional communication routines are invoked after depositing the charge and current density on the grid to obtain the values at the boundaries. The same is done when calculating electromagnetic fields and electron temperature.

1.3. Load Balancing

From a computational point of view, numerical routines, handling particle-grid connection, are the most time-consuming. For this reason, the parallelization is only efficient if the number of particles on each core is kept similar during simulation. The large load imbalance, i.e. when some processes contain a much bigger number of simulation particles, will result in a longer duration of the particle propagation step for those processes and the other will be waiting. In this way, valuable computational resources are wasted. To ensure load balancing in our algorithm, we compute the average number of particles N_k in one core per every k^{th} group every few thousand time steps, increase on the basis of N_k the number of cores in a group in dense regions (with larger N_k) to balance the number of particles per core, and redistribute the particles within the new group. More details about the particular implementation can be found in [7].

2. Simulation Results

In this Section, we present typical results from MHD-PIC simulations of the interaction of a proton beam injected into a cylindrical trap with the background trap plasma. The simulations were performed with a temporal resolution of $\Delta t = 10^{-5}\omega_{iH}^{-1}$ and cell size of $\Delta z \times \Delta r = [0.05 \times 0.05](c/\omega_{pi})^2$. The size of the simulation domain was $L \times R_0 = [12 \times 4](c/\omega_{pi})^2$ with $\sim 2 \times 10^5$ background particles. The injection of a plasma beam normal to the z -axis takes place at a rate

of 10^3 particles per time unit. The background magnetic field in the trap is created by a pair of current coils of radius $4.3 c/\omega_{pi}$, placed at $z = \pm L/2 = \pm 6 c/\omega_{pi}$. The amplitude of the current in the coils is such that the mirror ratio of the trap $R_m = \frac{B_1}{B_0} = \frac{B_z|_{r=0, z=\pm L/2}}{B_z|_{r=0, z=0}} = 2$, see Fig. 1b). Note, that on the axis of the trap the magnetic field has only z -component, B_z . All numerical parameters have been checked for convergence. As for the physical parameters, they are chosen to be close to the parameters of laboratory experiments on the CAT installation at the Budker Institute of Nuclear Physics of SB RAS (BINP SB RAS, Novosibirsk, Russia) [6]. Namely, we consider an open magnetic trap with the length $L = 60$ cm and the radius $R_0 = 15$ cm. The number density of the background plasma $n_0 = 4 \cdot 10^{13}$ cm $^{-3}$, the magnetic field strength in the center of the trap $B_0 = 2$ kG. The simulation was performed on 30 groups of cores of the computer facility of the Siberian Supercomputer Center of SB RAS (SSCC SB RAS, Novosibirsk, Russia).

Figure 2 shows the snapshots of the distribution of the magnetic pressure inside the trap at different times. The injection of plasma beam leads to the displacement of the magnetic field and to the formation and growth of the magnetic cavity, i.e. of the region with low amplitude of the field. The values of the magnetic field pressure inside the cavity are very small, $< 5\%$ of the initial value, and at the *sharp* boundaries of the cavity the amplitude of the magnetic field is in turn high. In order to investigate the temporal evolution of the cavity we plot in Fig. 3a the distribution of the amplitude of the magnetic field $|B(r, t)|_{z=0}$ in the central plane $z = 0$. The magnetic field in Fig. 3a is measured in units of B_0 . The size of the cavity in the radial direction grows over time, but this growth has an oscillatory character. The simulations with different mirror ratios R_m have shown that the period of the oscillations is not constant but depends monotonously on the increasing radius of the cavity. Further investigation is needed to confirm this hypothesis and find the exact dependence of the oscillation period on the cavity radius if any, or to deny it. The decreasing rate of growth of the cavity radius with time implies that it is possible to reach a quasistationary regime in which the transverse size of the cavity will remain almost constant over time. Snapshots of the distribution of the injected ions displayed in Fig. 3b show the accumulation and capture of the plasma in the magnetic cavity region.

Conclusion

In this paper, we present a MHD-PIC hybrid numerical model to study the interaction of a beam of plasma injected into an axisymmetric magnetic trap with the background trap plasma. The background magnetic field is created by two identical current coils, located at both ends of the trap chamber. We assume that the field configuration and the evolution of the plasma do not depend on the azimuthal angle through the whole process and consider the reduced 2D cylindrical problem, with z -axis directed along the symmetry axis of the trap. We apply a kinetic description for the positively charged ions and treat electrons as a massless charge neutralizing fluid. The parallel numerical algorithm is based on the domain and particle decomposition. The model adequately describes nonlinear nonstationary evolution of the plasma and of the magnetic field and allows to follow this evolution for large simulation times in a wide range of the initial magnetic field and plasma parameters, as applied to the conditions of laboratory experiments at the BINP SB RAS. Our simulations show that the continuous injection of plasma beam into the trap leads to the formation of an extended region with the displaced magnetic field, the magnetic cavity. The value of the field pressure inside this cavity is less than 5% of the initial value. The space-temporal evolution of the cavity evidences that the growth of its size in the radial direction

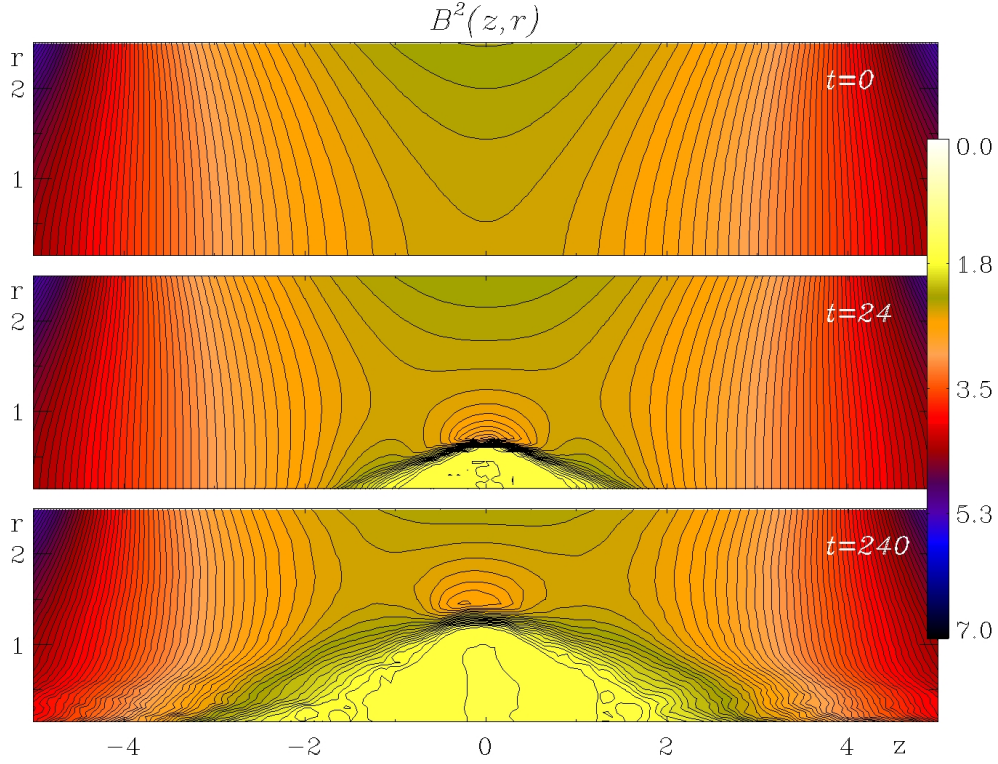
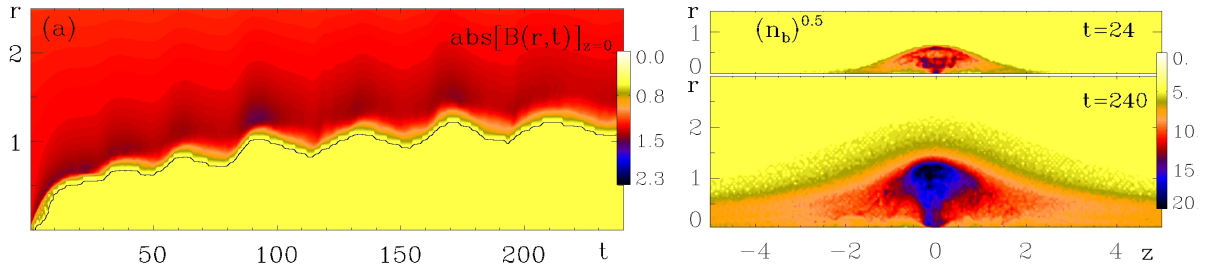


Figure 2. Snapshots of the magnetic field pressure at $t = 0$, $t = 24 \omega_{iH}^{-1}$ and $t = 240 \omega_{iH}^{-1}$ shows the formation and growth of the magnetic cavity. Lengths are measured in units of c/ω_{pi}



(a) The space-temporal distribution of the $|\vec{B}(r, t)|_{z=0}$. (b) Snapshots of the injected ions, $\sqrt{n_b}$, in The thick black line displays the level $|\vec{B}(r, t)|_{z=0} = 0.05$ space at two consecutive moments of time

Figure 3. Distribution of the magnetic field and of the injected ions

has an oscillatory character. Moreover, the simulations with different mirror ratios suggest that the period of the oscillations may monotonously depend on the increasing radius of the cavity. In addition, the decreasing growth rate of the cavity radius with time implies that it is possible to reach a quasistationary regime in which the transverse size of the cavity will remain constant or further change only insignificantly.

Acknowledgements

The research was supported by Russian Science Foundation grant No. 19-71-20026. The authors acknowledge the Siberian Supercomputer Center of the Siberian Branch of the Russian Academy of Sciences for providing the computational resources for the simulations.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Beklemishev, A.D.: Diamagnetic bubble equilibria in linear traps. *Physics of Plasmas* 23(8), 082506 (2016). <https://doi.org/10.1063/1.4960129>
2. Boris, J.P.: Relativistic plasma simulation-optimization of a hybrid code. *Proceeding of Fourth Conference on Numerical Simulations of Plasmas* pp. 3–67 (1970).
3. Budker, G.: *Plasma Physics and the Problem of Controlled Thermonuclear Reactions*. Pergamon Press, New York (1959).
4. Chen, F.F.: Introduction. In: *Introduction to Plasma Physics and Controlled Fusion*, pp. 1–18. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-22309-4_1
5. Chernoshtanov, I.S.: Collisionless particle dynamics in diamagnetic trap. *Plasma Physics Reports* 48(2), 79–90 (2022). <https://doi.org/10.1134/S1063780X22020052>
6. Davydenko, V.I., Deichuli, P.P., Ivanov, A.A., Murakhtin, S.V.: Neutral beam injection system for the cat experiment. *Plasma and Fusion Research* 14, 2402024 (2019). <https://doi.org/10.1585/pfr.14.2402024>
7. Efimova, A., Boronina, M., Vshivkov, K., Dudnikova, G.: Supercomputer simulation of plasma flow in the diamagnetic mode of open magnetic systems. In: Sokolinsky, L., Zymbler, M. (eds.) *Parallel Computational Technologies. Communications in Computer and Information Science*, vol. 1868, pp. 299–310. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38864-4_21
8. Hockney, R., Eastwood, J.: *Computer simulation using particles*. CRC Press (2021).
9. Lipatov, A.S.: *The Hybrid Multiscale Simulation Technology*. Springer-Verlag Berlin, Heidelberg (2002). <https://doi.org/10.1007/978-3-662-05012-5>
10. Post, R.: Summary of UCRL pyrotron programme. *Journal of Nuclear Energy* (1954) 7(3-4), 282 (1958).
11. Ryutov, D.D.: Open-ended traps. *Soviet Physics Uspekhi* 31(4), 300 (1988).
12. Vshivkov, K., Voropaeva, E., Efimova, A.: A new scheme for integrating the equations of particle motions in the particle-in-cell method for plasma physics. *Computational Technologies* 28(2), 27–41 (2023).

CPU vs RAM in the Issue of *ab initio* Simulations of Doped Hafnium Oxide for RRAM and FRAM

Timofey V. Perevalov¹ , Damir R. Islamov^{1,2} 

© The Authors 2023. This paper is published with open access at SuperFri.org

Atomic and electronic structure of doped HfO₂ is studied using first principle simulations. The 96- and 324-atom supercell are used to simulate impurity density in the range of 2–6.3 mol.% that is used in real electronic memory devices. The optimal spatial configurations of impurity atoms with an oxygen vacancy are found. It is shown that there are no defect levels in the band gap doped HfO₂ with the optimal structures. The electronic structure of additional neutral oxygen vacancy in HfO₂ is equivalent to that of neutral oxygen vacancy in pure HfO₂. An increase in the size of a supercell predictably leads to an increase in the need for computing resources. At the same time, the need for RAM is growing faster than for CPU power. Doping HfO₂ with Al/La/Y with concentration of up to 6.2 mol.% has negligible effect on the electronic structure of neutral oxygen vacancies.

Keywords: supercomputer, high performance computing, paradigm of structural calculations, parallelism, quantum chemistry, memristor.

Introduction

Promising candidates for the role of universal memory, which combines the advantages of Random Access Memory (RAM), Hard Drive Disks and Flash memory, are resistive (RRAM) and ferroelectric (FRAM) memories based on hafnium oxide (HfO₂) [16, 22]. In a HfO₂-based RRAM, the resistive switching between states of different resistance is carried out when exposed to an external electric field due to the formation/breaking of a conductive filament. In the HfO₂-based FRAM, the information storage is supported by the dielectric polarization in the ferroelectric layer. The switching of polarization is carried out through the application of an external electric field. The performance of such a device is ensured by the ferroelectric phase stabilization in HfO₂ films. It is known that doping HfO₂ with various metals, such as Al, La and Y, leads to the increased performance of RRAM and FRAM cells: reduced forming voltage, increased memory window and increased number of reprogramming cycles [2, 9, 10, 18, 21, 23]. The mechanisms by which the dopant influences the characteristics of RRAM and FRAM have not been established yet. This problem can be solved using first principle simulations within the density functional theory (DFT). However, to do this, first of all, it is necessary to establish the atomic structure of doped HfO₂. Despite a fairly large number of studies on this issue, they are all limited to the use of model structures, the correctness of which has not been proven yet [3, 6, 13–15, 20, 24–26].

The complexity of the task is determined by two factors. First of all, the use of HfO₂ supercells with the replacement of two Hf atoms by impurity atoms and an oxygen vacancy necessary for the charge compensation of the system, necessitates the search for the most probable (energetically favorable) spatial position of three defects in a supercell. It should be clarified that this particular HfO₂ doped with Al, La, Y model structure is the most popular and justified, since the valency of the considered impurities is one lower than that of hafnium. This requires considering a huge number of nonequivalent defect configurations. Thus, for the 96-atomic supercell of the

¹Rzhanov Institute of Semiconductor Physics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russian Federation

²Novosibirsk State University, Novosibirsk, Russian Federation

monoclinic phase (m-) HfO_2 (with 32 metal atoms and two types of oxygen vacancies), which is mostly used in calculations, obtained by the $2 \times 2 \times 2$ translation of a primitive 12-atomic cell, it is necessary to calculate $C_{32}^2 \times 2 = 992$ configurations. In the vast majority of existing studies, the authors limit themselves to considering the defect configurations in which a pair of impurity atoms is in close proximity to oxygen vacancy.

The second difficulty is that to simulate HfO_2 with an impurity concentration of about 2 mol.%, corresponding to the best characteristics of real RRAM and FRAM elements based on doped HfO_2 , it is necessary to use supercells of 324 atoms (obtained by a symmetric translation of a $3 \times 3 \times 3$ primitive cell). The use of a 96-atom supercell corresponds to the simulation of an overestimated impurity concentration of 6.25 mol.%. The correctness of the results obtained for 324-atom supercells was not verified due to the need to use a lot of computing resources. It is important to note that despite the absence of serious difficulties in calculating of this scale systems for standard DFT, to correctly reproduce the electronic structure and, in particular, the position of defect levels in the band gap, it is necessary to use a significantly more resource-intensive DFT with hybrid exchange-correlation functionals. This, in turn, requires significantly more computational resources.

Thus, the purpose of this work is to study the atomic and electronic structure of HfO_2 doped with Al, La and Y at low concentrations. The study includes, firstly, determining the optimal atomic structures of HfO_2 doped with Al, La and Y from the point of view of energy efficiency; secondly, studying the influence of the supercell size on the resulting optimal structure and, thirdly, simulations the electronic structure of additional oxygen vacancies in the found structures. Additionally, the problem was formulated as studying the influence of the supercell size on the reproducibility of the calculation results of the atomic and electronic structure of defect complexes.

The article is organized as follows. Section 1 is devoted to structures under study and calculation methods of electronic structure of structure defects in the studied electronic systems. In Section 2 we discuss obtained results. Subsection 2.1 is devoted to the description of the atomic and electronic structures of HfO_2 supercells with mutual arrangement of the impurity atoms and oxygen vacancies. In Subsection 2.2 the required computing resources for the simulations of 96- and 324-atom HfO_2 supercells are discussed and compared. Conclusion summarizes the study and points directions for further work.

1. Methods

The simulations were carried out within the DFT in the approximation of a periodic 3D supercell, with a plane-wave basis and optimized norm-conserving Vanderbilt pseudopotentials [7, 8] using the Quantum ESPRESSO (QE) software package [4, 5]. Two types of exchange-correlation functionals were used: PBEsol to calculate structural relaxation and B3LYP to calculate the electronic spectrum of optimal $\text{HfO}_2\text{:X}$ structures (X in one of Al, Y or La). The simulation was carried out for the m- HfO_2 ($P2_1/c$) phase. This phase is the most stable and closest in physical properties to amorphous HfO_2 , and is also observed in real HfO_2 films used in RRAM and FRAM elements. The optimal structures of $\text{HfO}_2\text{:X}$ were found by calculating all possible nonequivalent configurations of the arrangement of oxygen vacancy and a pair of impurity atoms at the Hf substitution position in 96-atom supercells from which configurations with the minimum total energy of the system were selected. For 324-atom supercells, the search for optimal structures was carried out by finding the optimal position of the first impurity atom

at a fixed position of the oxygen vacancy, and then of the second one. The oxygen vacancy, in this case, is a structural element of $\text{HfO}_2:\text{Al}/\text{Y}/\text{La}$ providing a charge compensation for the impurity, and, for convenience, is further referred to as a structural vacancy (V_{O}). The calculations used the plane waves cutoff energy 80 Ry, the Fock exchange operator cutoff energy 100 Ry, the k -point grid $2 \times 2 \times 2$, the Fock operator point grid $1 \times 1 \times 1$, the exact exchange fraction for B3LYP 0.175 (which provides the m- HfO_2 bandgap value of 5.8 eV) and the total energy convergence threshold 10^{-4} Ry. In the found $\text{HfO}_2:\text{X}$ structures, the optimal position of the additional oxygen vacancy with the minimum formation energy (hereinafter denoted V'_{O}) was found by calculating and analyzing all possible positions in the supercell. The spatial distributions of atoms and defects in $\text{HfO}_2:\text{X}$ supercells were visualized using the XCrySDen program [11, 12]. The used computing resources and memory distributions were extracted from QE reports.

The energy of V'_{O} formation (E_{form}) was calculated using the formula:

$$E_{\text{form}} = E(V'_{\text{O}}) - E_{\text{p}} + \mu(\text{O}), \quad (1)$$

where $E(V'_{\text{O}})$ is the energy of the $\text{HfO}_2:\text{X}$ supercell with neutral V'_{O} ; E_{p} is the energy of the ‘perfect’ supercell without V'_{O} ; $\mu(\text{O})$ is the chemical potential on an oxygen atom O. For the convenience of comparing the results with literature data, the $\mu(\text{O})$ value was taken equal to half the total energy of the O_2 molecule in the triplet state, which corresponds to the oxygen-enriched limit.

2. Results and Discussions

2.1. Atomic and Electronic Structures

After calculating all possible spatial configurations of the Al/La/Y atom pair position in supercells of 96- and 324-atoms with oxygen vacancy V_{O} , optimal structures with the lowest total energy were found (Fig. 1). The spread of the total energy of supercells with different defect configurations is about 3 eV, while the structure closest in energy differs from the optimal one by about 0.4 eV. It was established that the features of the atomic structure obtained for 96 and 324-atomic supercells coincide. It is obvious that further decrease in the impurity concentration due to an increase in the supercell size will not lead to changes in the optimal mutual arrangement of the impurity atoms and V_{O} . Thus, the use of a 96-atom supercell is sufficient to reproduce the main features of the relative arrangement of impurity atoms in m- HfO_2 .

In all structures, V_{O} is 3-coordinated. In $\text{HfO}_2:\text{La}$ and $\text{HfO}_2:\text{Y}$, the impurity atoms are spaced from each other at approximately 6 Å, with one of the impurity atoms located close to V_{O} at a distance of $r \approx 2$ Å, and the second – at $r \approx 4.1$ Å from V_{O} . For La/Y near V_{O} the coordination number is 6, for the second La/Y it is 7. It is noteworthy that, the total energy of $\text{HfO}_2:\text{La}$ and $\text{HfO}_2:\text{Y}$ supercells with an optimal structure is lower (by more than 0.4 eV), compared with the total energy of non-optimal structures, that were used in calculations in the works of various authors previously [6, 13–15, 20, 25]. The optimal structure of $\text{HfO}_2:\text{Al}$, on the contrary, meets this assumption: V_{O} is located between the Al atoms at $r \approx 2.3$ Å from each Al, while the Al atoms are relatively close to each other ($r \approx 4.5$ Å). It is probably energetically unfavorable for La and Y atoms to be too close to each other due to their large ionic radius compared to the Hf one: $R_{\text{Hf}} = 0.85$ Å, $R_{\text{La}} = 1.17$ Å, $R_{\text{Y}} = 1.06$ Å [19]. In $\text{HfO}_2:\text{Al}$, a pair of Al atoms might be close to each other due to a small Al ionic radius ($R_{\text{Al}} = 0.68$ Å).

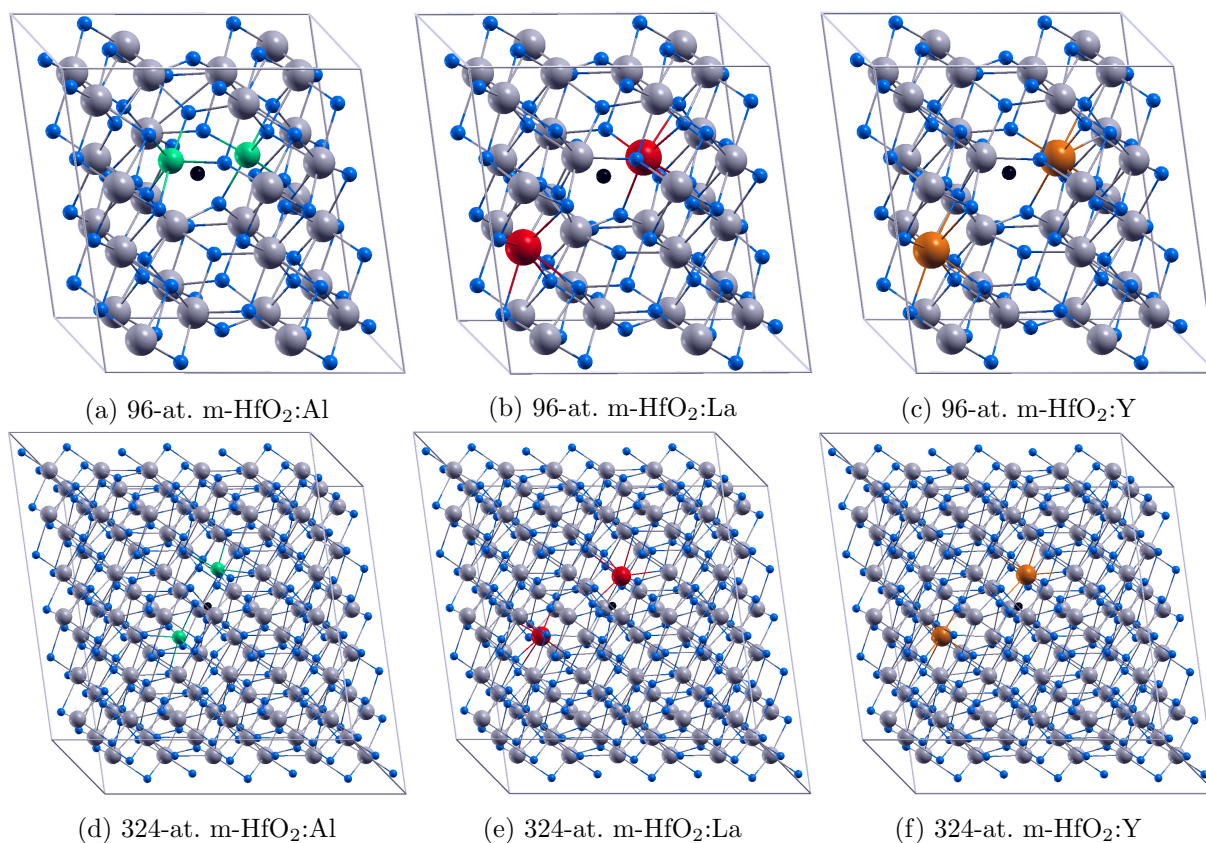


Figure 1. Supercells of 96- and 324-atoms of the optimal m-HfO₂:Al, m-HfO₂:La and m-HfO₂:Y structures. Gray color balls are Hf, blue ones are O, green ones are Al, red ones are La and black ones represent O vacancies (removed O atoms)

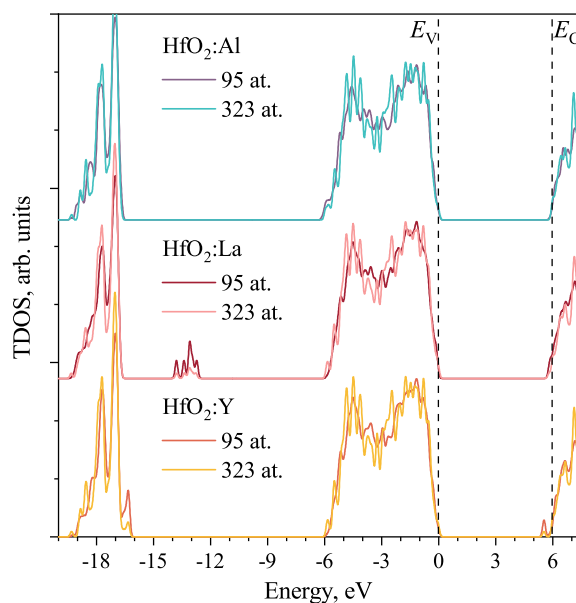


Figure 2. TDOS spectra calculated for the optimal HfO₂:Al, HfO₂:La and HfO₂:Y structures. Zero energy corresponds to the valence band top E_V

The total density of states (TDOS) spectra calculated for HfO₂ doped with Al, La and Y with concentrations of 6.25 mol.% and 2 mol.% show that the band gap of all oxides is empty

(Fig. 2). This result is consistent with the experimental data, according to which doping HfO_2 with lanthanum does not change the bandgap E_g [17]. In contrast, the TDOS spectra for non-optimal structures have an empty level with a depth of about 1 eV [13, 15]. In doped HfO_2 , as well as in pure HfO_2 , $E_g = 5.85$ eV, which is close to the experimental value $E_g = 5.7$ eV [1]. In the case of doping with La, a subband with an energy of about 14 eV below the valence band top E_V is formed predominantly by La5*p* orbitals, which can be seen in the TDOS spectrum.

It was established that the optimal position of an additional oxygen vacancy V'_O in 95- and 323-atomic supercells of $\text{HfO}_2:\text{Al}$, $\text{HfO}_2:\text{La}$ and $\text{HfO}_2:\text{Y}$, with the minimal E_{form} , is near one of the impurity atoms. As a result, in $\text{HfO}_2:\text{La}$ and $\text{HfO}_2:\text{Y}$, both impurity atoms become 6-coordinated. For $\text{HfO}_2:\text{La}$ and $\text{HfO}_2:\text{Y}$, the $V_O-V'_O$ distance is 5.50 Å, and for $\text{HfO}_2:\text{Al}$ it is 3.13 Å. The E_{form} of V'_O weakly depends on the value of the considered defect density and is approximately 0.2 eV less than the E_{form} of the vacancy in undoped HfO_2 . Thus, doping HfO_2 with Al/La/Y facilitates the generation of new oxygen vacancies in the oxide. Recently, the opposite results have been obtained, however, when the non-optimal structures of doped HfO_2 were simulated [20, 25].

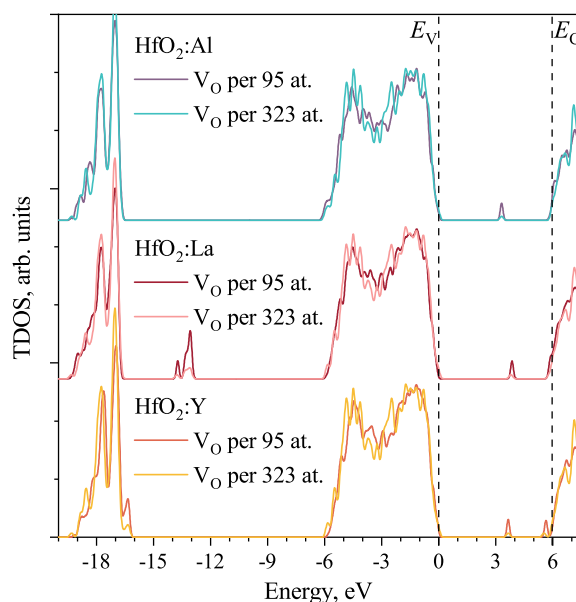


Figure 3. TDOS spectra calculated for HfO_2 , $\text{HfO}_2:\text{Al}$ and $\text{HfO}_2:\text{La}$ structures with additional V'_O . Zero energy corresponds to the valence band top E_V

In doped HfO_2 , a neutral V'_O forms a level filled with two electrons just above the middle of the bandgap, as shown in Fig. 3. The position of this level, firstly, weakly depends on the type of dopant (any of Al, La or Y); secondly, it does not depend on the impurity concentration, and, thirdly, it is close to that for a neutral oxygen vacancy in undoped HfO_2 . Thus, one can conclude that doping HfO_2 with Al/La/Y with a concentration of up to 6.2 mol.% has a negligible effect on the electronic structure of neutral oxygen vacancies.

2.2. Required Computing Resources

The required computing resources for the simulations of 96- and 324-atom HfO_2 supercells are given in Tab. 1. As the cell size increased, the complexity of the problem increased about 3.4 times. At the same time, CPU time increased about 6–7 times, while the memory require-

ments increased more significantly, about 8 times. The memory is mainly used to store data from exact-exchange (EXX) integrals within hybrid functional, wave-functions and β -functions of non-local pseudopotentials. The needs for EXX and β -function data increased 7 times and that is close to the total growth of using RAM resources. Note that the data volume for reducing matrices to the diagonal shape has increased more than 13 times.

Table 1. Resource consumption per calculation

	96 at.	324 at.	324/96 = 3.375
CPU, core×hour	~120	~720	~6–7
RAM, GB	~50–60	~450–470	~7–9
Wavefunctions, GB	~9, 6	~18, 4	~2
EXX, GB	~35	~253	~7
Structure factor, GB	~0.04	~0.08	~2
Local pseudopotentials	~0	~0	–
Nonlocal pseudopotentials (beta functions), GB	~2.4	~16	~7
Nonlocal pseudopotentials (Q functions), GB	~0.5	~0.5	~1
Charge density and potentials, GB	~0.2	~0.2	~1
Charge density in initialization, GB	~0.05	~0.05	~1
Grid vectors, GB	~0.05	~0.1	~2
Iterative diagonalization (matrices), GB	~0.03	~0.4	~13
Iterative diagonalization (scalar products), GB	~0.4	~5	~13
Iterative diagonalization (charge density), GB	~1.2	~7.5	~6
Wavefunctions in initialization, GB	~2	~13	~7

Conclusion

This work is devoted to the thorough study on the atomic and electronic structure of HfO₂ doped with aluminum, lanthanum and yttrium. The simulation was carried out for two impurity concentrations, covering the actual range of doping density of films in real electronic devices. Two types of oxygen vacancies are considered, namely, the oxygen vacancy involved in compensating the impurity charge (V_O) and the additional oxygen vacancy (V'_O). It was established that, in the optimal structure of the doped oxide, La and Y atoms tend to distance themselves from each other at about 6 Å. Just one of the La/Y atom is located near V_O , while both Al atoms are located near one common V_O and the distance between Al is about 4 Å. It was established that the features of the atomic structure obtained for 96 and 324-atomic supercells coincide. The data obtained for HfO₂:La and HfO₂:Y arouse doubts as for all the results previously published on this topic. It was found that there are no defect levels in the bandgap of HfO₂:Al, HfO₂:La and HfO₂:Y with the optimal structures. The formation of additional neutral vacancy V'_O in HfO₂ near Al, Y or La atoms is facilitated, compared to the formation of V_O in undoped HfO₂. The electronic structure of V'_O in HfO₂:Al, HfO₂:La and HfO₂:Y is equivalent to that of neutral V_O in pure HfO₂.

Increasing the size of a supercell leads to an increase in the need for computing resources. At the same time, the need for RAM is growing faster than for the CPU power. Since doping HfO₂

with Al/La/Y with concentration of up to 6.2 mol.% has a negligible effect on the electronic structure of neutral oxygen vacancies, 96-atomic supercells exhibit all features of m-HfO₂.

Acknowledgements

This work was supported by the Russian Science Foundation, grant No. 22-22-00634. The simulation was performed at the ISP SB RAS cluster.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Balog, M., Schieber, M., Michman, M., Patai, S.: Chemical vapor deposition and characterization of HfO₂ films from organo-hafnium compounds. *Thin Solid Films* 41, 247–259 (1977). [https://doi.org/10.1016/0040-6090\(77\)90312-1](https://doi.org/10.1016/0040-6090(77)90312-1)
2. Dai, Y., Zhao, Y., Wang, J., *et al.*: First principle simulations on the effects of oxygen vacancy in HfO₂-based RRAM. *AIP Advances* 5, 017133 (2015). <https://doi.org/10.1063/1.4906792>
3. Gao, B., Zhang, H.W., Yu, S., *et al.*: Oxide-based RRAM: Uniformity improvement using a new material-oriented methodology. In: 2009 Symposium on VLSI Technology, pp. 30–31 (2009).
4. Giannozzi, P., Andreussi, O., Brumme, T., *et al.*: Advanced capabilities for materials modelling with Quantum ESPRESSO. *Journal of Physics: Condensed Matter* 29(46), 465901 (2017). <https://doi.org/10.1088/1361-648x/aa8f79>
5. Giannozzi, P., Baroni, S., Bonini, N., *et al.*: QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter* 21(39), 395502 (2009). <https://doi.org/10.1088/0953-8984/21/39/395502>
6. Gu, C., Ang, D.S.: Impact of lanthanum on positive-bias temperature instability – insight from first-principles simulation. *ECS Transactions* 53(3), 193–204 (2013). <https://doi.org/10.1149/05303.0193ecst>
7. Hamann, D.R.: Optimized norm-conserving vanderbilt pseudopotentials. *Physical Review B* 88, 085117 (2013). <https://doi.org/10.1103/physrevb.88.085117>
8. Hamann, D.R.: Erratum: Optimized norm-conserving vanderbilt pseudopotentials [phys. rev. b 88, 085117 (2013)]. *Physical Review B* 95, 239906 (2017). <https://doi.org/10.1103/physrevb.95.239906>
9. He, R., Wu, H., Liu, S., *et al.*: Ferroelectric structural transition in hafnium oxide induced by charged oxygen vacancies. *Physical Review B* 104, L180102 (2021). <https://doi.org/10.1103/physrevb.104.1180102>

10. Islamov, D.R., Perevalov, T.V.: Effect of oxygen vacancies on the ferroelectric $\text{Hf}_{0.5}\text{Zr}_{0.5}\text{O}_2$ stabilization: DFT simulation. *Microelectronic Engineering* 216, 111041 (2019). <https://doi.org/10.1016/j.mee.2019.111041>
11. Kokalj, A.: XCrySDen – a new program for displaying crystalline structures and electron densities. *Journal of Molecular Graphics & Modelling* 17(3-4), 176–179 (1999). [https://doi.org/10.1016/s1093-3263\(99\)00028-5](https://doi.org/10.1016/s1093-3263(99)00028-5)
12. Kokalj, A.: Computer graphics and graphical user interfaces as tools in simulations of matter at the atomic scale. *Computational Materials Science* 28, 155–168 (2003). [https://doi.org/10.1016/s0927-0256\(03\)00104-6](https://doi.org/10.1016/s0927-0256(03)00104-6)
13. Leitsmann, R., Plänitz, P., Nadimi, E., Ötting, R.: Oxygen related defects and the reliability of high- k dielectric films in FETs. In: 2013 International Semiconductor Conference Dresden - Grenoble (ISCDG). pp. 1–4. IEEE (2013). <https://doi.org/10.1109/iscdg.2013.6656327>
14. Materlik, R., Künneth, C., Falkowski, M., *et al.*: Al-, Y-, and La-doping effects favoring intrinsic and field induced ferroelectricity in HfO_2 : A first principles study. *Journal of Applied Physics* 123, 164101 (2018). <https://doi.org/10.1063/1.5021746>
15. Nadimi, E., Ötting, R., Plänitz, P., *et al.*: Interaction of oxygen vacancies and lanthanum in Hf-based high- k dielectrics: an *ab initio* investigation. *Journal of Physics: Condensed Matter* 23(36), 365502 (2011). <https://doi.org/10.1088/0953-8984/23/36/365502>
16. Park, M.H., Lee, Y.H., Mikolajick, T., *et al.*: Review and perspective on ferroelectric HfO_2 -based thin films for memory applications. *MRS Communications* 8, 795–808 (2018). <https://doi.org/10.1557/mrc.2018.175>
17. Perevalov, T.V., Prosvirin, I.P., Suprun, E.A., *et al.*: The atomic and electronic structure of $\text{Hf}_{0.5}\text{Zr}_{0.5}\text{O}_2$ and $\text{Hf}_{0.5}\text{Zr}_{0.5}\text{O}_2$:La films. *Journal of Science: Advanced Materials and Devices* 6(4), 595–600 (2021). <https://doi.org/10.1016/j.jsamd.2021.08.001>
18. Pešić, M., Fengler, F.P.G., Larcher, L., *et al.*: Physical mechanisms behind the field-cycling behavior of HfO_2 -based ferroelectric capacitors. *Advanced Functional Materials* 26, 4601–4612 (may 2016). <https://doi.org/10.1002/adfm.201600590>
19. Shannon, R.D.: Revised effective ionic radii and systematic studies of interatomic distances in halides and chalcogenides. *Acta Crystallographica Section A* 32(5), 751–767 (1976). <https://doi.org/10.1107/s0567739476001551>
20. Umezawa, N., Shiraishi, K., Sugino, S., *et al.*: Suppression of oxygen vacancy formation in Hf-based high- k dielectrics by lanthanum incorporation. *Applied Physics Letters* 91, 132904 (2007). <https://doi.org/10.1063/1.2789392>
21. Voronkovskii, V.A., Aliev, V.S., Gerasimova, A.K., Islamov, D.R.: Influence of HfO_x composition on hafnium oxide-based memristor electrical characteristics. *Materials Research Express* 5(1), 016402 (2018). <https://doi.org/10.1088/2053-1591/aaa099>
22. Zahoor, F., Zulkifli, T.Z.A., Khanday, F.A.: Resistive random access memory (RRAM): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage,

- modeling, and applications. *Nanoscale Research Letters* 15(1), 1–26 (2022). <https://doi.org/10.1186/s11671-020-03299-9>
23. Zalyalov, T.M., Islamov, D.R.: The influence of the dopant concentration on the ferroelectric properties and the trap density in $\text{Hf}_{0.5}\text{Zr}_{0.5}\text{O}_2\text{:La}$ films during endurance cycling. In: 2022 IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM). pp. 48–51. IEEE (2022). <https://doi.org/10.1109/edm55285.2022.9855130>
24. Zhang, H., Gao, B., Yu, S., *et al.*: Effects of ionic doping on the behaviors of oxygen vacancies in HfO_2 and ZrO_2 : A first principles study. In: 2009 International Conference on Simulation of Semiconductor Processes and Devices. pp. 1–4. IEEE (2009). <https://doi.org/10.1109/sispad.2009.5290225>
25. Zhao, L., Liu, J., Zhao, Y.: Systematic studies of the effects of group-III dopants (La, Y, Al, and Gd) in $\text{Hf}_{0.5}\text{Zr}_{0.5}\text{O}_2$ ferroelectrics by *ab initio* simulations. *Applied Physics Letters* 119, 172903 (2021). <https://doi.org/10.1063/5.0066169>
26. Zhou, Y., Zhang, Y.K., Yang, Q., *et al.*: The effects of oxygen vacancies on ferroelectric phase transition of HfO_2 -based thin film from first-principle. *Computational Materials Science* 167, 143–150 (2019). <https://doi.org/10.1016/j.commatsci.2019.05.041>

Recurrent Monitoring of Supercomputer Noise

Vadim V. Voevodin¹ , Dmitry A. Nikitenko¹ 

© The Authors 2023. This paper is published with open access at SuperFri.org

The presence of noise in supercomputers has long been known, but its scale and impact on the behavior of user applications are nevertheless considerably unclear. Therefore, we decided to develop an approach to determining the noise level on an ongoing basis, which makes it possible to assess the global impact of noise over a long time period. This paper describes a method for recurrent monitoring the noise level and analyzing collected statistics on a real modern supercomputer, and also presents the implementation and evaluation of this method on the Lomonosov-2 supercomputer. The usage of the proposed approach in practice made it possible to identify previously unknown issues and peculiarities, like detection of a faulty compute node, presence of nodes that tend to be more susceptible to noise or the global nature of the noise, which leads to the appearance of noise at multiple nodes simultaneously. This method can as well be ported to other similar computing systems without significant changes.

Keywords: supercomputing, high-performance computing, monitoring, noise, noise measurement, noise level.

Introduction

Modern supercomputers are very complex objects that consist of tens, hundreds of thousands, or even millions of components. They also include a large number of compute nodes; for example, the most powerful supercomputer in the world called Frontier [4] (#1 from the Top500 [6] list for November 2023) has more than 9000 nodes, and the Lomonosov-2 supercomputer [25] (#6 in the Top50 [3] list for March 2023) has about 1700 nodes [2]. To ensure their correct, consistent and efficient operation within a supercomputer, various system software is required (resource manager, monitoring system, distributed file system, operating system (OS), etc.). In addition, many different user applications are run simultaneously on a supercomputer, and in some systems several applications can be launched in parallel on a single node. All this leads to the fact that almost all modern supercomputers suffer from the so-called “noise”. Hereinafter, we will define “noise” as the influence of the software and hardware environment, which leads to a change (most often a slowdown) in the execution time or other properties of applications running on a supercomputer.

The causes of noise can vary: changes in hardware (for example, the occurrence of ECC errors), contention for shared resources with other user applications (for example, for a shared communication network or file system), changes in operating conditions (for example, underclocking the processor due to overheating). One of the most common causes of noise is the operating system.

Although the presence of noise in supercomputers has long been known, the extent of its impact on the behavior of user applications is often unknown, especially given that this impact can differ dramatically on different computing systems. Therefore, we decided to develop an approach to determining noise level on a supercomputer, not as a one-time study, but on an ongoing basis.

The main contribution of this work is the development of a method for recurrent monitoring the noise level and analyzing the collected statistics on a real modern supercomputer, which

¹Lomonosov Moscow State University, Moscow, Russian Federation

allows evaluating the long-term impact of noise in practice. We have implemented and evaluated this method on the Lomonosov-2 supercomputer, but it can be ported to other similar computing systems without significant changes.

The rest of the paper is organized as follows. Section 1 shows existing work aimed at studying noise level on supercomputers. Section 2 is devoted to the description of the proposed approach for continuous monitoring of noise level and its implementation on the Lomonosov-2 supercomputer. Section 3 shows the results of evaluating the approach in practice. In conclusion, the main results of this work are briefly described.

1. Related Work

There is quite a lot of research devoted to the study of noise that occurs in high-performance systems. For example, in the papers [7, 8, 13, 14, 16, 18] the influence of noise on the behavior of supercomputer applications is studied. Other articles (e.g. [9, 10, 12, 17, 19, 24]) explore what causes noise and how one can tackle it. Most of these works are devoted to the study of one specific type of noise: OS noise, which is the most common topic of study in existing research. We can also highlight works [11, 16], which propose models and simulators to predict the scalability of applications, taking noise into consideration. However, these works are mainly aimed at a one-time study of noise and not on studying the noise level on a supercomputer on an ongoing basis.

The work [15] should be mentioned separately, which presents the open-source software tool called Netgauge. This tool allows conveniently measuring OS noise on a machine, mainly on a single server or compute node. Let us briefly describe the operation of Netgauge as it is used in this work. Netgauge runs a larger number of simple identical iterations and calculates what percentage of iterations take significantly longer to execute than the reference iteration. The execution time of a reference iteration is calculated at startup, for which it runs a small number of iterations and determines the minimum time execution value (which must remain minimal for at least 100 iterations in a row). By default, a “noisy” iteration is considered to be an iteration which execution time is at least 9 times higher than the reference one, but this parameter can be changed if desired.

2. Monitoring Noise Level

2.1. Proposed Approach

As the result of our studies under the ExtraNoise project [20], it is proposed to recurrently measure the noise level on a supercomputer in a following way.

On a supercomputer, after each user job completes, a script is run in Slurm epilogue that updates noise level information on the nodes used to run the job itself. Independently for each of these nodes, the script performs the following:

- Checks when the noise level on this node was last measured. To do this, each node has its own empty file, the modification time of which is changed (by the `touch` command) with each new measurement of the noise level.
- If less than a day has passed since the last measurement of the noise level at this node (the threshold can be easily changed), then nothing else is done.

- If more than a day has passed, the script launches the Netgauge software to determine the noise level on this node. The result of Netgauge is saved in a separate file for each node.

Netgauge runs using MPI on all available logical cores (if HyperThreading is enabled on the node, then the number of logical cores is 2 times the number of physical ones). The binding of MPI processes to cores is also used. This allows obtaining accurate and stable results when assessing noise level.

Also, the running time of Netgauge is selected in such a way that the epilogue script runs for no more than 1 minute on average. To do this, a heuristically selected value of a parameter that indicates the number of “noisy” iterations that Netgauge needs to determine before completing its work was specified. Note that the running time of Netgauge itself is not explicitly limited, since it works until it collects the required number of “noisy” iterations, which is non-deterministic and therefore can take varying amounts of time. However, taking into account the experiments carried out to measure the running time, it was decided that there is no need to further reduce the selected value for the number of “noisy” iterations to detect, especially since in this case the stability of the collected noise level decreases.

2.2. Implementation of Proposed Approach

The described approach was implemented on a petaflop-scale supercomputer Lomonosov-2.

The first step was to determine that the proposed solution would not notably slow down the execution of user job flow. Figure 1 shows the running time of 29 thousand script launches (data was collected from November 15 to December 5 2023 and sorted by increasing operating time). Let us remind that each launch is performed on one node and no more than once a day. It can be seen that in the absolute majority of cases (99.88%) the operating time did not exceed 60 seconds, and the maximum was 131 seconds. In 2/3 of the cases, the script worked instantly, since a new recalculation of the noise level was not required, because less than a day has passed since the last measurement.

Taking into account the information above, we can say that the impact of this solution on the execution of user applications is insignificant. We also note that the script for measuring the noise level is never launched while user applications are running, but only after they are completed. Therefore, the operation of the script only affects by the fact that it slightly delays the launch time of new jobs.

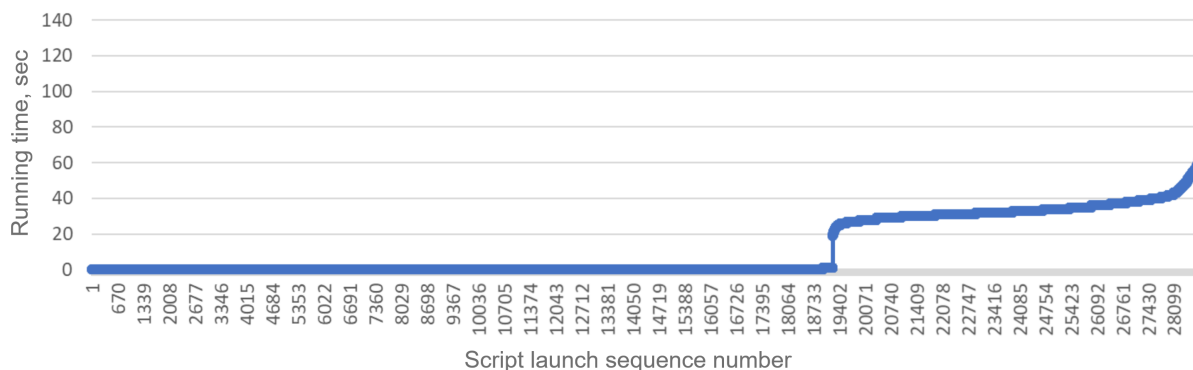


Figure 1. Distribution of running time of the script for noise level measurement

The following conclusions can be drawn from this graph:

- There is one node, the leftmost one on the graph, where the average noise level is significantly higher than the others and overall is very high – 93% (while the next one is 44%). The noise level at this node was collected 4 times during selected time period, i.e. this behavior is not a random anomaly, but is observed regularly. This node will further be considered in more detail.
- The number of noise level measurements at different nodes can vary significantly – from 1 to 6 measurements per node. In general, this is expected, since the frequency of measurements is influenced by many factors:
 - Execution time for user jobs: the longer each job takes to complete (assuming it runs for more than a day), the less often the measurement is performed.
 - Sometimes nodes require repair or reconfiguration, and in these cases they are made unavailable for running user jobs. During this period, noise level assessment is not performed on them.
 - The frequency is also affected by how often the node is idle waiting for jobs. Typically, a node is idle because more nodes are needed to run the next job in the queue than are currently available, so it has to wait for other nodes to become available. However, note that on Lomonosov-2 there are usually very few nodes being idle, due to the constant high load of the supercomputer and the usage of scheduling algorithms in Slurm such as Backfill [1].
 - There are 7 different partitions on the supercomputer, and some of them are small and specialized, where access is given to a limited number of users to carry out special calculations. The load on these partitions is therefore unstable, and during the time period examined, several nodes from these partitions were rarely occupied by user jobs.
- Most often, 2–4 noise level measurements were performed at the nodes during the period under consideration, which corresponds to the expected values.
- 77% of nodes have an average noise level of less than 1%, and such a low noise level can generally be neglected.
- 15% of nodes have an average noise level of more than 10%, and this is quite noticeable. This will also be further discussed.

Let us now consider in more detail the most “noisy” node, which was discussed in the first item. First, let us study the chronology of noise level measurements on it during the considered time period December 1 through December 7 (Tab. 1).

Table 1. The noise level on the node under consideration

Date	Dec 01	Dec 02	Dec 04	Dec 05
Noise, %	92.72	92.76	92.70	93.86

During the considered 7 days, noise was measured 4 times. It can be seen that the noise level at this node is always significant: at least 92% of Netgauge iterations signaled the presence of noise, which is a very high value. In order to further understand the causes of such noise, the running processes on the node were studied. In particular, a typical example of the `top` command output is shown in Fig. 3. Here you can see that the behavior of the node indicates some abnormal situation, since processes that normally almost do not load the processors (`top`,

rcu_sched, irqbalance, as well as the DiMMon monitoring system [23] used on Lomonosov-2) in this case occupy a noticeable part of the processor time. And this behavior generally persists over time; in particular, restarting the monitoring system and rebooting the node did not change the overall situation.

Thanks to this information received, the supercomputer administrators paid attention to this node and began to study its behavior. The root causes are not yet completely clear, but the node turned out to be faulty and requires repair.

VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
164516	2732	1616	R	20.7	0.0	0:03.66	top
8874212	24008	3680	S	15.3	0.0	236:36.54	dimmon
0	0	0	S	5.7	0.0	3322:40	rcu_sched
21680	1392	988	S	5.7	0.0	3056:10	irqbalance
0	0	0	S	2.3	0.0	1435:12	migration/0
166644	5832	4456	S	1.3	0.0	0:01.34	sshd
191128	4108	2624	S	0.8	0.0	776:26.73	systemd

Figure 3. Output of top command on a “noisy” node

Finally, we will study in more detail the nature of the resulting noise level. Let us consider the same time period, but we will analyze not the average values for all nodes (as in Fig. 2), but the last received values for each node, sorted by the time the result was received (Fig. 4). Each value on the X axis corresponds to the last value of the noise level at a certain node. It can be seen that most often high noise levels are grouped by time, i.e. most likely some global processes, which lead to an increase of noise on several nodes at once, are taking place on the supercomputer. This may be due to the peculiarities of the system software used across the entire supercomputer (such as a resource manager or monitoring system) or the shared resources (such as a distributed file system). We are currently studying the reasons for the occurrence of such grouped noise.

You can also see that in this case there are only 4% of nodes with a noise level of more than 10%, which is noticeably lower than 15%, as was the case in Fig. 2. Further, the noise level is most often very low (less than a couple of percent), in other cases it is almost always quite high (40% and higher). This is also confirmed when considering the chronology of noise level measurements for individual nodes (similar to table 1).

Thus, the following conclusions can be drawn:

1. the noise level can be unstable and vary greatly over time;
2. on a compute node, there is normally almost no noise, but it occasionally becomes significant, and there are practically no intermediate cases (i.e. no average noise level values);
3. the high noise level is clearly grouped by time, i.e. most likely the reason lies in some global processes occurring on the supercomputer as a whole, which simultaneously affects many compute nodes;
4. most nodes are not affected (less than 1/4 of nodes were notably influenced during the considered time period).

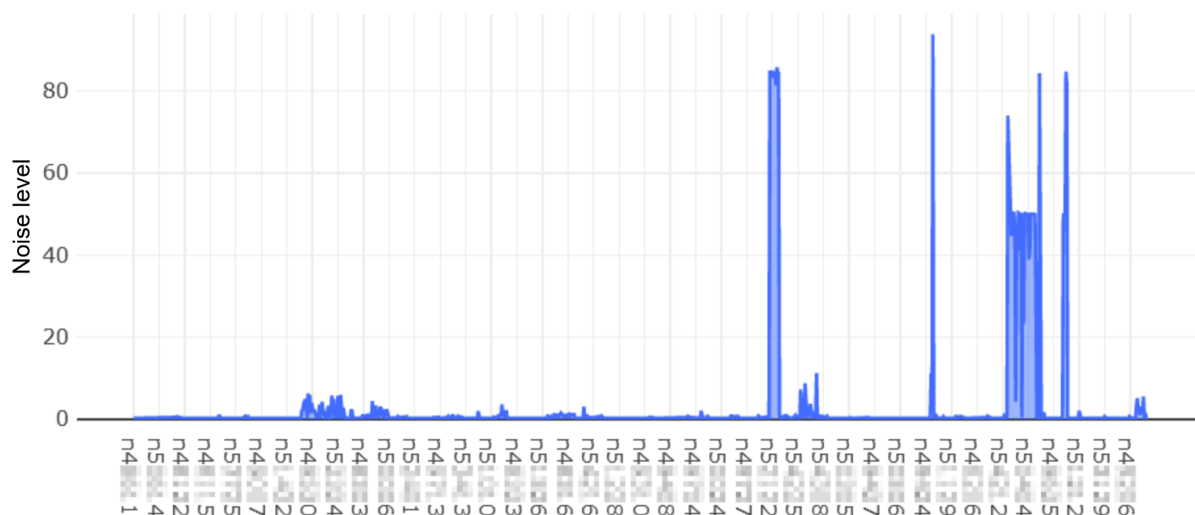


Figure 4. Last value of noise level at different nodes, sorted by the time it was received

Conclusions

This paper describes an approach to recurrent monitoring of OS noise on a supercomputer, which allows assessing the noise level on compute nodes as well as analyzing the dynamics of its change over time and the scale of its influence as a whole.

This approach was applied on the Lomonosov-2 supercomputer. Based on the evaluation results, it was found that the noise level is most often negligible, but on some nodes it can become significant. Further, it was found that noise most often occurs simultaneously on several nodes, which suggests that the cause of the noise is not localized within a node, but is global at the level of the supercomputer as a whole. Also, using the proposed approach, a node was discovered where the noise level is constantly very high, due to its malfunction.

Acknowledgements

The reported study was funded by RFBR and DFG, project number 21-57-12011. The research is carried out using the equipment of shared research facilities of HPC computing resources at Lomonosov Moscow State University.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Backfill scheduling, https://slurm.schedmd.com/sched_config.html#backfill
2. Characteristics of Lomonosov-2 supercomputer, <https://parallel.ru/cluster/lomonosov2.html>
3. Current rating of the 50 most powerful supercomputers in CIS, <http://top50.supercomputers.ru/?page=rating>

4. Frontier supercomputer debuts as world's fastest, breaking exascale barrier, <https://www.ornl.gov/news/frontier-supercomputer-debuts-worlds-fastest-breaking-exascale-barrier>
5. Redash homepage, <https://redash.io/>
6. TOP500 list, <https://top500.org/lists/top500/>
7. Afzal, A., Hager, G., Wellein, G.: Propagation and decay of injected one-off delays on clusters: a case study. In: 2019 IEEE International Conference on Cluster Computing (CLUSTER). pp. 1–10. IEEE (2019). <https://doi.org/10.1109/CLUSTER.2019.8890995>
8. Agarwal, S., Garg, R., Vishnoi, N.K.: The impact of noise on the scaling of collectives: A theoretical approach. In: High Performance Computing – HiPC 2005. HiPC 2005. Lecture Notes in Computer Science, vol. 3769, pp. 280–289. Springer (2005). https://doi.org/10.1007/11602569_31
9. Akkan, H., Lang, M., Liebrock, L.: Understanding and isolating the noise in the Linux kernel. The International Journal of High Performance Computing Applications 27(2), 136–146 (2013). <https://doi.org/10.1177/1094342013477892>
10. De, P., Kothari, R., Mann, V.: Identifying sources of operating system jitter through fine-grained kernel instrumentation. In: Proceedings of the 2007 IEEE International Conference on Cluster Computing, ICCO. pp. 331–340. IEEE (2007). <https://doi.org/10.1109/CLUSTR.2007.4629247>
11. De, P., Mann, V.: jitSim: A simulator for predicting scalability of parallel applications in presence of OS jitter. In: Euro-Par 2010 - Parallel Processing, 16th International Euro-Par Conference, Proceedings, Part I. Lecture Notes in Computer Science, vol. 6271, pp. 117–130. Springer (2010). https://doi.org/10.1007/978-3-642-15277-1_12
12. De, P., Mann, V., Mittal, U.: Handling OS jitter on multicore multithreaded systems. In: Proceedings of the 2009 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2009. pp. 1–12. IEEE (2009). <https://doi.org/10.1109/IPDPS.2009.5161046>
13. Ferreira, K.B., Bridges, P., Brightwell, R.: Characterizing application sensitivity to OS interference using kernel-level noise injection. In: Proceedings of the ACM/IEEE Conference on High Performance Computing, SC 2008. pp. 1–12. IEEE/ACM (2008). <https://doi.org/10.1109/SC.2008.5219920>
14. Garg, R., De, P.: Impact of noise on scaling of collectives: An empirical evaluation. In: High Performance Computing - HiPC 2006, 13th International Conference, Proceedings. Lecture Notes in Computer Science, vol. 4297, pp. 460–471. Springer (2006). https://doi.org/10.1007/11945918_45
15. Hoefler, T., Mehlan, T., Lumsdaine, A., Rehm, W.: Netgauge: A network performance measurement framework. In: High Performance Computing and Communications, Third International Conference, HPCC 2007, Proceedings. Lecture Notes in Computer Science, vol. 4782, pp. 659–671. Springer (2007). https://doi.org/10.1007/978-3-540-75444-2_62

16. Hoefler, T., Schneider, T., Lumsdaine, A.: Characterizing the influence of system noise on large-scale applications by simulation. In: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC'10. pp. 1–11. IEEE (2010). <https://doi.org/10.1109/SC.2010.12>
17. Jones, T.: Linux kernel co-scheduling for bulk synchronous parallel applications. In: Proceedings of the 1st International Workshop on Runtime and Operating Systems for Supercomputers. pp. 57–64. ACM (2011). <https://doi.org/10.1145/1988796.1988805>
18. Khudoleeva, A., Stefanov, K., Voevodin, V.: Evaluating the Impact of MPI Network Sharing on HPC Applications. In: Parallel Computational Technologies. PCT 2023. Communications in Computer and Information Science, vol. 1868, pp. 3–18. Springer (2023). https://doi.org/10.1007/978-3-031-38864-4_1
19. Mondragon, O.H., Bridges, P.G., Levy, S., Ferreira, K.B., Widener, P.: Understanding performance interference in next-generation HPC systems. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2016. pp. 384–395. IEEE (2016). <https://doi.org/10.1109/SC.2016.32>
20. Nikitenko, D., Mohr, B., Wolf, F., *et al.*: Influence of Noisy Environments on Behavior of HPC Applications. Lobachevskii Journal of Mathematics 42(7), 1560–1570 (2021). <https://doi.org/10.1134/S1995080221070192>
21. Nikitenko, D., Antonov, A., Shvets, P., *et al.*: JobDigest – Detailed System Monitoring-Based Supercomputer Application Behavior Analysis. In: Supercomputing. Third Russian Supercomputing Days, RuSCDays 2017, Moscow, Russia, September 25-26, 2017, Revised Selected Papers. Communications in Computer and Information Science, vol. 793, pp. 516–529. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71255-0_42
22. Shvets, P., Voevodin, V., Nikitenko, D.: Approach to Workload Analysis of Large HPC Centers. In: Parallel Computational Technologies. PCT 2020. Communications in Computer and Information Science, vol. 1263, pp. 16–30. Springer (2020). https://doi.org/10.1007/978-3-030-55326-5_2
23. Stefanov, K., Voevodin, V., Zhumatiy, S., Voevodin, V.: Dynamically Reconfigurable Distributed Modular Monitoring System for Supercomputers (DiMMon). Procedia Computer Science 66, 625–634 (2015). <https://doi.org/10.1016/j.procs.2015.11.071>
24. Tsafirir, D., Etsion, Y., Feitelson, D.G., Kirkpatrick, S.: System noise, OS clock ticks, and fine-grained parallel applications. In: Proceedings of the 19th Annual International Conference on Supercomputing. pp. 303–312. ACM (2005). <https://doi.org/10.1145/1088149.1088190>
25. Voevodin, V., Antonov, A., Nikitenko, D., *et al.*: Supercomputer Lomonosov-2: Large scale, deep monitoring and fine analytics for the user community. Supercomputing Frontiers and Innovations 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>

The Parallel Performance of SLNE Atmosphere–Ocean–Sea Ice Coupled Model

Rostislav Yu. Fadeev^{1,2,3,4} 

© The Author 2023. This paper is published with open access at SuperFri.org

The paper presents the first version of SLNE coupled model. SL and NE here are the first two letters from SLAV (Semi-Lagrangian, based on Absolute Vorticity equation) atmospheric model and NEMO (Nucleus for European Modelling of the Ocean) ocean model that have been coupled using OASIS3-MCT software. SLAV uses $0.9^\circ \times 0.72^\circ$ regular lat-lon grid with 96 vertical levels. NEMO incorporates SI3 sea ice model. Both of them use the same ORCA025 tripolar grid. Flux adjustments to correct inconsistencies at the interface between coupled atmosphere–ocean models have not been applied in SLNE. The model design and coupling particularities are described here in detail. A series of numerical experiments with SLNE model were performed to measure its parallel performance. We also investigated the scalability of SLNE model and its components in terms of simulation speed. Based on these results, an optimum configurations of SLNE were identified. It was found that the coupled model showed scaling efficiency of about 85% on 4000 computational cores of Cray XC40-LC in comparison to the SLNE configuration running on 224 cores. Simulations with lead times ranging from a few days to several years showed that there are no significant systematic errors in the coupled model.

Keywords: numerical weather prediction, coupled model, parallel performance, NEMO ocean model, SLAV model, OASIS3-MCT coupler.

Introduction

Modern coupled models for simulating climate change were preceded by their earlier counterparts [12]. The history of the development of such models begins with conceptual models [5, 50], followed by mathematical models of energy balance [4, 53] and radiative transfer [7], as well as simple analog models [19, 47]. Smagorinsky was probably the first researcher to realize the importance of atmosphere–ocean coupling for climate modelling. Under his leadership, the first coupled atmosphere–ocean general circulation model was created [41] at the Geophysical Fluid Dynamics Laboratory (GFDL). Concurrently with GFDL, work on development of the atmospheric general circulation model was carried out at the University of California, Los Angeles (UCLA) Department of Meteorology (known as UCLA model series), and at the US National Center for Atmospheric Research (NCAR) a few years later [33].

Nikita Moiseev’s model [45] developed at the Computer Centre of the Academy of Sciences is apparently the first internationally recognized Soviet coupled model. More recent research by the Alexandrov-led team has focused on studies of global changes in the biosphere, including economic, social and demographic aspects [1]. One of the first models of global atmospheric circulation was developed by Gury Marchuk, who used numerical modelling of atmospheric processes for numerical weather forecasting. From 1973 Marchuk created “mathematic calculations of atmospheric-ocean dynamics” [9] that was later tested on the supercomputer.

Nowadays, software packages based on coupled models are widely used to assess climate change and study the Earth’s climate in the past [13]. Coupled models are also used for the long-range weather prediction in leading World Meteorological Organization (WMO) meteorological

¹Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow, Russian Federation

²Hydrometeorological Research Center of Russian Federation, Moscow, Russian Federation

³Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences, Moscow, Russian Federation

⁴Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

centers: Met Office Global Seasonal forecasting system (GloSea) [8, 37] in the UK Meteorological Service, Meteo-France seasonal forecasting system 8 (SYSTEM8) [25] in France, Climate Forecast System Version 2 (CFSv2) [51] in the USA, the Canadian Seasonal to Interannual Prediction System Version 2 (CanSIPSv2) [36] in Canada, Japan Meteorological Agency/Meteorological Research Institute Coupled Prediction System Version 3 (JMA/MRI-CPS3) [28] in Japan. The typical horizontal mesh resolution of these models ranges from 25 to 80 km (in atmospheric component).

The development and implementation of coupled models for global and regional medium-range weather forecasting (with lead times up to 10 days) is mainly driven by the idea that by resolving ocean circulation, coastal ocean features and air–sea interaction a more accurate representation of atmosphere, ocean and sea ice dynamics can be achieved. Another motivation for using the coupled models is the idea of seamless prediction [29], whereby a single model family can be used for prediction across a range of timescales. Canada, UK and European Centre for Medium-Range Weather Forecasts (ECMWF) are currently using coupled models for medium-range weather forecasting [61]. The coupled model is planned for implementation for medium-range weather prediction in the USA in 2025. In [56] a statistically significant improvement is shown in the accuracy of medium-range weather prediction by the coupled The Global Environmental Multiscale Model (GEM) [21] and Global Ice Ocean Prediction System (GIOPS) that based on Nucleus for European Modelling of the Ocean (NEMO) model in comparison with the operational forecast model of the Meteorological Center of Canada. The [40] discusses the results on the accuracy of tropical cyclone intensity and trajectory predictions based on the coupled atmosphere–ocean model of ECMWF. Note that Coupled Ocean–Atmosphere Mesoscale Prediction System-Tropical Cyclones (COAMPS-TC) [10] model is used for predicting the trajectory and intensity of tropical cyclones in the Atlantic Ocean and the Eastern Pacific Ocean in the USA. A coupled model [62] incorporates ENDGame-based (Even Newer Dynamics for the General Atmospheric Modelling of the Environment) [76] Met Office Unified Model (MetUM) of the UK Meteorological Service coupled with the the NEMO ocean model. This model is developed using Ocean-Atmosphere-Sea-Ice-Soil (OASIS) [68] software to study and forecast typhoons in Southeast Asia (Maritime continent). The spatial resolution of this model is about 4.5 km.

In Russia, coupled models are also being developed. Marchuk Institute of Numerical Mathematics Climate Model (INMCM) [71, 72] participates in the Climate Model Intercomparison Project (CMIP), the objective of which is to better understand past, present and future Earth climate changes in a multi-model context [75]. INMCM is also used in the long-range forecasting system developed in the Hydrometeorological Research Center of Russian Federation (Hydrometcenter of Russia). Team from Institute of Computational Mathematics and Mathematical Geophysics Siberian Branch of the Russian Academy of Sciences (SB RAS) develop Planet Simulator of Institute of Computational Mathematics and Mathematical Geophysics (PlaSim-ICMMG) [49]. This is intermediate complexity climate model, that includes own developed ocean model [22, 23], Portable University Model of the Atmosphere (PUMA) [18] and the Los Alamos sea ice model CICE (Community Ice Code) [30]. Since proprietary software SibCIOM Coupling Module (SCM) [24] is used to couple these models, there is another configuration of SB RAS climate model that incorporates INMCM’s atmospheric model instead of PUMA: Siberian coupled ice-ocean model (INMCM-SibCIOM). A spectral atmospheric model developed at A.I. Voyeykov Main Geophysical Observatory (MGO) is currently used in Hydrometcenter

of Russia as one of the components for probabilistic long-range forecast of weather anomalies. Based on this model, coupled model is also being developed at MGO [42, 43]. It is expected that once the coupled model is finalized, it will be applied for long-range forecasting. Another coupled model for long-range weather prediction under development is SLAV-INMIO-CICE model [14]. This model combines SLAV atmospheric model [63] that developed at Marchuk Institute of Numerical Mathematics RAS (INM RAS) and Hydrometcenter of Russia, INMIO (Institute of Numerical Mathematics and Shirshov Institute of Oceanology RAS) ocean model [31], and a CICE sea ice model. The coupling is performed using the own developed Coupling Modeling Framework (CMF) [32].

Inspired by the idea of seamless prediction, a new coupled model SLNE was developed. SLNE is the acronym of the models to be coupled: SLAV and NEMO. Note that NEMO includes sea ice SI3 model. Both SLAV and NEMO models are applied over a wide range of time scales: medium-range weather forecasting, sub-seasonal (with lead times from 2 to 6 weeks) and long-range ensemble prediction. Initial conditions for these models are provided by the software developed at Hydrometcenter of Russia: 3dvar meteorological data assimilation system [67] is used for SLAV and oceanographic data assimilation system [58] is applied for NEMO. OASIS3-MCT [68] software is used to couple SLAV and NEMO. Spherical Coordinate Remapping and Interpolation (SCRIP) [48, 55] library for remapping the exchanged data. SLNE coupled model has been developed for medium-range weather forecasting and long-range forecasting of weather anomalies.

The coupled models are computationally demanding. This is not only due to the sum of the costs of the individual model components, but also to additional costs of the coupler, mapping procedures and load imbalances of the components. This paper focuses on SLNE computational efficiency and parallel scaling analysis in order to extend the model usability, improve its performance and to focus future research on evaluating simulation accuracy. The paper is organized as follows. Section 1 presents the coupled model and its components. Section 2 describes the parallel structure of SLNE, the results of the parallel scalability study and coupled model optimal configurations. The results of numerical simulations are given in Section 3. Finally, we summarize the conclusions of the paper.

1. Overview of SLNE Coupled Model

1.1. SLAV Atmospheric Model

Atmospheric model SLAV [64] was developed at Marchuk Institute of Numerical Mathematics RAS (INM RAS) and Hydrometcenter of Russia. SLAV20 and SLAV10 are used for the medium-range numerical weather prediction with lead times up to 10 days in Hydrometcenter of Russia [65]. SLAV10 is released in 2023. The features of this model are the high horizontal resolution (about 10 km over the Northern hemisphere compared to 20 km in SLAV20), a more detailed description of the lower troposphere, and improved parameterizations of subgrid-scale processes. A coarser resolution version of SLAV10, named SLAV072L96 is applied for ensemble medium-range weather prediction since 2022. This model is also submitted for operational testing in 2022 for long-range prediction of weather anomalies at time scales of up to 4 month.

SLNE coupled model is based on the most recent version of SLAV072L96 [66] which differs from SLAV2008 in many aspects. The most notable difference between these models is the enhanced spatial resolution and the increased number of vertical levels from 28 to 96. The top

boundary of this model is located at 0.03 hPa. In horizontal, SLAV072L96 uses regular lat-lon grid with the 0.72° grid step in latitude and 0.9° resolution in longitude.

Together with an increase of the horizontal and vertical resolution of SLAV model, the used parameterizations of the subgrid-scale processes have been significantly improved with respect to SLAV2008. The key changes include the methods of describing the radiation transfer in the Earth atmosphere (CLIRAD SW [6, 6] and RRTMG LW [44] packages are now used), the atmospheric boundary layer and land surface model, which was supplemented by a multilayer soil model [70] developed at INM RAS and Research Computer Center of the Lomonosov Moscow State University (RCC MSU). A detailed description of all improvements is the subject of a special paper [66].

It should be noted that a substantial amount of time in the development of the new SLAV072L96 model version was devoted to model tuning in order to match the behavior of all parameterizations and the dynamical core [15]. Due to this study, the monthly and annual averaged characteristics of the model atmosphere (including surface heat fluxes) became close to the ERA5 reanalysis [27]. The annual surface energy budget is less than 1 W/m^2 . The zonal wind speed bias at all heights, including the near surface layer, was significantly reduced in SLAV072L96 due to recent study [17] on improving the deep convection and wind gustiness parameterization.

The vertical grid structure in SLAV072L96 is set to provide increased resolution in the lower troposphere and in the stratosphere. Non-uniform vertical grid spacing allows to describe explicitly the atmospheric processes in the stratosphere. It is shown in [52] that SLAV072L96 successfully reproduces the mean zonal wind speed and temperature distribution in winter and summer seasons in comparison to the ERA reanalysis [27]. The period and amplitude of the quasi-biennial equatorial wind oscillation are also close to the ERA5. Explicit simulation of stratospheric dynamics is important for several reasons. First, the [3] shows that the effect of El Niño–Southern Oscillation on the eddy-driven jet during spring and early summer occurs via the stratosphere. Second, tropical variability associated with the Madden–Julian oscillation (MJO) has been shown to affect the circulation in the extratropical stratosphere during boreal winter [54] and can lead to extended predictability at the surface.

1.2. NEMO-SI3 Ocean-Sea Ice Model Configuration

The coupled SLNE model uses NEMO version 4.0.4 [38] with SI3 sea ice model [69]. NEMO and SI3 use the same ORCA025 grid with a horizontal resolution of about $1/4^\circ$. The ORCA family is a series of global orthogonal curvilinear ocean meshes that are generated using semi-analytical method [39]. This is tripolar grid that has no singularity point inside the computational domain because two north poles of the mesh are introduced and placed on lands. ORCA025 computational grid has 1442 nodes along the first horizontal dimension and 1021 nodes along the second dimension. The ocean model also has 75 vertical levels. Data on temperature and salinity of the river runoff are read from a file. Unfortunately, river runoff temperature is not consistent with the temperature of the lower atmosphere. Coupled model does not include iceberg floats and wave model, but internal wave-driven mixing parameterization is used in NEMO.

SI3 simulates both ice dynamics (two-dimensional continuum elastic-viscous-plastic formulation is used) and thermodynamics via one-dimensional approximation along the vertical coordinate. The number of ice categories in SI3 is 5, the number of ice layers is 2 and there is one snow layer over the sea ice.

The resolution and configuration of NEMO and SI3 are the same as those used at Hydrometcenter of Russia (see [57, 58] for details). NEMO and SI3 share the same Message Passing Interface (MPI) communicator.

1.3. Coupled Model Design

SLAV072L96 and NEMO-SI3 models are coupled using OASIS3-MCT software, that performs parallel exchange of coupling data between SLAV and NEMO. OASIS3-MCT uses the MPI library for direct parallel communications between components of the coupled model. OASIS3-MCT is able to gather and scatter the arrays of coupling data. Since the computational meshes of SLAV and NEMO differ, OASIS3-MCT provides service for run-time remapping of the exchanged data using pre-computed interpolation weights and addresses. The current OASIS3-MCT software internally uses the Model Coupling Toolkit (MCT) [34], that implements parallel remapping as a parallel matrix-vector multiplication. OASIS3-MCT supports coupling of 2D logically-rectangular arrays. 3-dimensional arrays expressed on unstructured grids are also supported by the software using a one dimension degeneration of the data structures.

OASIS3-MCT is a library that is compiled and linked to the coupled model components. At runtime, all components as well as OASIS3-MCT are launched together. The MPI_COMM_WORLD communicator is split using OASIS3-MCT Application Programming Interface (API) into “private” communicators for SLAV and for NEMO as part of the initialization procedure of the coupled model. All of the MPI data transfer within the components is then done via these “private” communicators whereas exchange of data between SLAV and NEMO is done via OASIS3-MCT API. It means that components make OASIS3-MCT API calls to send or to receive data from within the component code directly. The OASIS3-MCT API routines call are located in a part of the program known as the component interface.

OASIS3-MCT has to know information about component computational grid structure, data decomposition and coupling fields identifiers to perform remapping at run-time. Transfer of this information to the OASIS3-MCT occurs at the stage of component initialization. Information on the method and frequency (coupling period) of data exchange between components is specified in the OASIS3-MCT configuration file.

To interact with the rest of the coupled system, SLAV interface for OASIS3-MCT API has been developed. It includes the following stages:

- Preliminary initialization of SLAV as a component of the coupled model, including partition definition, time step and computational grid declaration. The partition definition implies that all the MPI processes of the component have to describe in a global index space the local partitioning of computational grids onto which the coupling data is expressed. OASIS3-MCT supports several partition types: serial, apple (each partition is a segment of the global domain, described by its global offset and its local size), box, orange (each partition is an ensemble of segments describing by its global offset and its local extent in the global domain) and points. In SLNE the box partition approach is used. It means that each partition is a rectangular region of the global domain.
- Component initialization: memory allocation, reading initial condition, preliminary computations, etc.
- Definition phase: coupling data arrays declaration.
- Preliminary data exchanges phase. SLAV model component receiving boundary condition from NEMO model has to wait for the coupling data (sea surface temperature, sea ice

temperature and concentration) before it can perform its own calculations. OASIS3-MCT supports only regular (each coupling period of time) data exchange. Therefore, the preliminary data exchange and appropriate pre-processing of coupling data procedures were implemented into the main part of the code of SLAV and NEMO.

- Main time loop. In the component time step loop, each MPI process additionally performs its part of the coupling data sending and receiving via OASIS3-MCT API (internally uses MPI non-blocking request). The sending (receiving) is performed if the actual time corresponds to a time at which it should be activated. It is controlled by the OASIS3-MCT, to which the component passes its actual time.
- Termination.

In total, six to eight OASIS3-MCT API routines have to be called by each component to get the local MPI communicator, to declare the components id and grid partitioning, define, send and receive the coupling data and, finally, close the MPI context at the components runtime end. These actions are performed in the aforementioned component interface that was developed for the SLAV model. On the NEMO side, the coupling interface was created by the community and customized for use in SLNE. Within SLAV model framework, a number of numerical methods were implemented to compute the lower atmosphere properties and surface fluxes passed to NEMO and SI3.

The time step in SLAV atmospheric model is 1440 s, while in NEMO ocean model 720 s. time step is used. It can be seen that the NEMO time step is half of SLAV's time step. Therefore, the coupling time period used in SLNE is a multiple of SLAV time step. Atmosphere and ocean models exchange two-dimensional data arrays only. Mapping of coupling data is performed using bilinear interpolation from SCRIP [48, 55] library.

SLAV and NEMO models run in parallel as different binaries. Execution on Cray XC-40 (see Sec. 2.1) is performed using the following command:

```
# run script
aprun -n 960 -d 1 -N 32 --mpmd-env OMP_NUM_THREADS=1 ./nemo : \
      -n 48 -d 4 -N 8 --mpmd-env OMP_NUM_THREADS=4 ./slav > slne.out
```

The peculiarity of the program design of SLAV model is the use of MPI-based one-dimensional domain decomposition in latitude supplemented by OpenMP loop parallelization in longitude. Computation of parameterization of subgrid-scale processes are the most resource-demanding part of SLAV model. Parameterizations are calculated independently for each vertical column. This means that the application of OpenMP for SLAV is important, while NEMO does not support OpenMP. Therefore, the coupled model was assembled as two independent executable files, corresponding to the atmosphere and ocean-sea ice models and executed each with its own environment.

1.4. Physical Basis

In coupled mode, SLAV model performs computation for coupling period of time. It calculates the surface fluxes and accumulates precipitation. These data are then passed to the NEMO which uses them as the upper boundary condition while computing ocean dynamics. After coupling period of time updated ocean data is sent to SLAV model to be used as the lower boundary condition. The coupling period defines the moments of time, at which the models are synchronized.

The data coupling between SLAV and NEMO models is organized as follows. The atmospheric model sends 10 surface fields to the ocean model, including two heat fluxes (downward solar radiation and sum of the net surface thermal radiation flux, latent and sensible heat fluxes), wind stress (two components), amount of precipitation during coupling period (kg/m^2 , separately for liquid and solid phases), evaporation flux from the surface ($\text{kg}/\text{s m}^2$). Note that solar radiation, non-solar radiation and evaporation fluxes are computed and sent for water and ice-covered surfaces individually. All the heat fluxes are expressed in W/m^2 . Surface heat fluxes are used to calculate the heat budget while wind stress is used as a direct momentum flux for NEMO to calculate the water motion. Evaporation and precipitation fluxes are used by the ocean model to compute fresh water budget and adjust the salinity of the uppermost ocean layer. Flux adjustments to correct inconsistencies at the interface between coupled atmosphere–ocean models have not been applied in SLNE.

NEMO sends seven data arrays to SLAV: sea surface temperature, ice concentration, sea ice surface temperature, sea ice top layer temperature, ice thickness, sea ice snow cover thickness and sea ice effective conductivity. Sea ice top layer temperature, thickness and effective conductivity are used in SLAV model to calculate the heat flux between the low atmosphere and sea ice. The evolution of sea ice surface temperature T_s is computed in SLAV using the following equation:

$$\frac{\partial T_s}{\partial t} = C_{tc}F_{atm} + C_{ice}(T_{tl} - T_s), \quad (1)$$

where C_{tc} is the thermic coefficient ($\text{Km}^{-2}\text{J}^{-1}$), F_{atm} is the surface net heat flux (W/m^2), T_{tl} is the sea ice top layer temperature, C_{ice} is the sea ice effective conductivity. The first and second terms in the right hand side of (1) are the heat fluxes into the atmosphere and sea ice, respectively. For simplicity, in the Equation (1) we omit the snow related variables and terms. In SLAV model, Equation (1) is incorporated into the vertical diffusion scheme of planetary boundary layer (PBL) parameterization.

The turbulent flux F_ψ of a prognostic variable ψ in vertical direction is calculated in SLAV using the following equation:

$$F_\psi = -K\nabla^2\psi, \quad (2)$$

where K is the diffusion coefficient. On the surface, the Equation (2) takes the form:

$$F_s = -\rho C_{dh}U_L(\psi_L - \psi_s), \quad (3)$$

where ρ is the air density, C_{dh} is the surface exchange coefficient, U_L and ψ_L are the wind speed and prognostic variable value at the lower model level. The surface latent and sensible heat fluxes are calculated in SLAV by equation (3) using the surface temperature T_s . Therefore, over land and sea ice, Equations (2) and (3) are solved together with Equation (1) using an implicit time scheme. Coupling within SLAV of locally one-dimensional models of the atmosphere and sea ice were performed using a numerical algorithm developed to describe the atmosphere–glacier interaction over land [16]. Over the water surface, Equation (1) is redundant because the water temperature is computed in the ocean model.

The coupling of the atmosphere and sea ice models was one of the most time-consuming stages of SLNE model development. The relatively small heat capacity of the upper layer of sea ice and large variability of heat fluxes on its surface in time can lead to the numerical instability expressed in typical two time step oscillations of the sea ice surface temperature. Special attention in the developed model was paid to the consistency of the evolution of the

characteristics of the sea ice surface and the lower levels of the SLAV model in the case when the atmospheric model cell is not completely occupied by sea ice. To describe this case, we will use the mosaic approach [46] implemented into SLAV model earlier.

2. Parallel Structure of SLNE

2.1. HPC System and Tools

The previously developed Parallel Profiler (ParProf) [59] software was used to analyze the parallel structure and scalability of the coupled model. This software allows to measure the average computation time of the program code fragment of interest. This software module was used in the SLAV's interface file, as well as in SLAV and NEMO main code. In the first case, the results of the measurements allowed us to investigate the characteristics of the coupled model and its parallel efficiency. In the latter case, the obtained results allowed us to study the performance of the coupled model components individually.

A massively parallel supercomputer XC40-LC installed at the Main Computer Center of Federal Service for Hydrometeorology and Environmental Monitoring was used to study the parallel structure of the coupled model. Cray XC40-LC consists of 976 compute nodes interconnected via the Cray's proprietary Aries network. Each node has 128 Gb of RAM and two Intel Xeon E5-2697v4 processors with 18 CPU cores and 45 Mb of Intel Smart Cache per processor. The total number of computational cores is 35136. It is important to note that the user is able to use only 32 cores per node since 4 cores per node are reserved to maintain the distributed Lustre file system.

The Cray XC40-LC job scheduling system delegates the entire node to the user to perform computations. This means that the number of cores used by the coupled model is a multiple of 32. This feature was taken into account in the study. The number of delegated computational cores to models was set to a multiple of 32. The amount of computational resources available for the experiments with the coupled model was 157 nodes or 5024 cores.

SLAV, NEMO and SI3 models are implemented using Fortran. To compile the program code of these models, Intel Compiler version 19.1.3.304 was used. A study of model performance depending on compiler options was not performed, because it might produce variations in floating point results. Compiler options that control optimization of atmospheric model code were the same as in SLAV-based long-range prediction system at Hydrometcenter of Russia. Compiler options for ocean and sea ice model were the same as in NEMO-based oceanographic data assimilation system of Hydrometcenter of Russia.

Note that ParProf software output was independently verified using the average step information of each model of SLNE. The performance of the coupled model was also verified on the computational systems installed at Marchuk Institute of Numerical Mathematics RAS and Keldysh Institute of Applied Mathematics RAS.

2.2. Numerical Experiments Methodology

A number of numerical experiments were performed to study the parallel structure of the coupled model. All of them were organized in the same way. The start date of the coupled model computation corresponded to November 11, 2021. Initial data was prepared in advance using 3dvar meteorological data assimilation system [67] and oceanographic data assimilation

system [58]. The assimilation system for the ocean and sea ice is based on the relaxation procedure for ice concentration in SI3 model which is applied along with the 3DVAR analysis for temperature and salinity of sea water in NEMO model. This approach guarantees consistent initial states of both ocean and sea ice in coupled model. Unfortunately, uncoupled assimilation technique lead to inconsistencies in the initial states of the atmosphere and ocean. The coupled model was integrated for 10 days for all numerical experiments to study the parallel performance of the model. This lead time corresponds to 600 time steps of SLAV model and 1200 time steps of NEMO. In the following sections it is shown that the variance of the computation time of one time step in the experiments is relatively small. This means that 10 days is sufficient to evaluate the performance of the coupled model.

Measurements of the computation time of one step of SLAV and NEMO models began after the first data exchange between models was finalized. The first exchange of data between models synchronizes them. We use this approach due to the significantly longer NEMO initialization time in comparison to SLAV. Control points, diagnostic information and output data were not recorded in the experiments. No additional operations (transformations, accumulation, etc.) within OASIS3-MCT were performed on the exchanged data.

As noted in Sec. 2.1, each model was allocated a multiple of 32 computational cores. SLAV and NEMO used different communicators, while NEMO and SI3 shared a common MPI communicator. A feature of SLAV model is its use of a one-dimensional MPI domain decomposition in latitude. For longitude, in addition to MPI, OpenMP is used. Therefore, the best performance is achieved when the number of MPI processes is as close as possible to a divisor of the number of nodes in latitude. In SLAV072L96 the number of grid nodes in latitude is 251 (including pole points). The number of OpenMP threads should be a divisor of the number of nodes of the computational grid by longitude, which equal to 400. Calculated in this way optimal SLAV configurations are presented in Tab. 1.

Table 1. SLAV optimal parallel configurations and their identifiers

Configuration id	MPI proc.	OMP threads	Used cores
32: 8x4	8	4	32
64: 16x4	16	4	64
128: 48x4	32	4	128
192: 48x4	48	4	192
256: 64x4	64	4	256
384: 96x4	96	4	384
512: 128x4	128	4	512

The first column in the Tab. 1 corresponds to SLAV configuration identifier, the second to the number of MPI processes, the third to the number of OpenMP threads and the fourth to the total number of computational cores used.

Note that the number of OpenMP threads equal to four is the best option. Several experiments have been performed for SLAV individually and within the coupled model, where the size of the MPI communicator was increased (decreased) and the number of OpenMP threads was decreased (increased) by the same amount. All experiments showed either close or worse

performance of the atmospheric model compared to the version of SLAV with 4 threads of OpenMP.

NEMO uses two-dimensional MPI decomposition of the computational domain and data. According to the results of experiments (not presented in this paper), it was found that the performance of NEMO and SI3 models degrades in case of excessive stretching of data along one of the coordinates. Optimal parallel configurations of the used NEMO version correspond to those where the computational domain is divided into approximately equal squares. Even then, NEMO model has many parallel configurations compared to SLAV model.

2.3. SLNE Configurations

In numerical experiments SLNE configurations presented in Tab. 2 were used. The first column of this table lists the model configuration identifier, which will be used later in the paper. The second and third columns give the coupling periods (in hours). The fourth column presents information about the time interval between computing the radiative transfer model – the most resource-demanding parameterization in SLAV.

In the first o1a1 SLNE configuration data exchange is performed every 1440 s (each SLAV’s time step). For computational efficiency reasons o1a1 model configuration is not considered as a principal version of SLNE model: computing the radiative transfer model at each step of the model increases its cost by 30–35% compared to o2a2. So, performance of this configuration has not been studied. o2a2 corresponds to the data exchange between the models every second SLAV’s time steps (every fourth NEMO’s time steps). The third o3a3 configuration corresponds to the exchange between models every third SLAV’s time step or, which is the same, sixth NEMO’s time step.

Since the rate of change of ocean surface characteristics is much smaller than in the lower atmosphere, in the third o6a3 model configuration (fourth row) data are sent from SLAV to ocean and sea ice models every third SLAV’s time step (as in o3a3 configuration), while NEMO sends data to the atmospheric model every 6th SLAV’s time step (every 12th NEMO time step). The reduced number of data transfer and associated data interpolations offers the hope of decreasing computation time. However, the SLAV and SI3 do not exchange data directly. Therefore, a disadvantage of this approach is that the sea ice surface properties are constant in the atmospheric model for six SLAV time steps. Since the thermal conductivity and heat capacity of sea ice are significantly different from those of sea water, this can lead to reduced accuracy of the heat fluxes computation procedure in the lower atmosphere above sea ice.

Table 2. SLNE configurations: coupling periods of the components

Configuration id	Coupling periods, hour (min)		SLAV’s radiative transfer model call, hour
	NEMO to SLAV	SLAV to NEMO	
o1a1	0.4 (24)	0.4 (24)	0.4
o2a2	0.8 (48)	0.8 (48)	0.8
o3a3	1.2 (72)	1.2 (72)	1.2
o6a3	2.4 (144)	1.2 (72)	1.2

Data exchange between SLAV and NEMO with a frequency of about one hour is typical for coupled models, where the horizontal resolution of the atmospheric model is about 1° and that of the ocean model is about $1/4^\circ$.

In SLAV072L96, the computation of the radiation fluxes in the atmosphere (shortwave and longwave spectrum) occurs every third time step of the model. At the intermediate time moments, the radiation fluxes are extrapolated in time taking into account the evolution of the zenith angle. This approach is motivated by the cost of the radiative transfer parameterization that is comparable to all other calculations performed to compute one time step. Data exchange between models and radiative transfer computation performed with the same period of time to load balance between SLAV and NEMO.

2.4. Scalability of SLNE

Three series of experiments were performed to study the parallel efficiency of the coupled model. Figure 1a–Figure 1c illustrate the parallel scalability of SLNE on Cray XC40-LC. In these figures the performance (in terms of simulated years per day) as a function of the number of computational cores used by SLNE is shown. The dots indicate different parallel configurations of the model, the solid line corresponds to linear scaling and the dashed curve corresponds to 80% performance compared to the optimal configuration with the minimal number of computational cores (the definition of SLNE optimal configuration is given in the following section). The dots of the same color differ in the number of computational cores allocated for NEMO with a fixed number of cores being used by SLAV. Figure 1a corresponds to o2a3 SLNE configuration, Fig. 1b – o3a3 and Fig. 1c – o6a3.

In Fig. 1a–Fig. 1c one can see that SLNE model scales non-monotonically for the each SLAV configuration. This is due to two reasons. First, NEMO and SI3 models scale non-monotonically with increasing communicator size. The best performance of these models is achieved when the computational area is partitioned into sub-domains which shape is close to square. In other words, the number of cells in the sub-domain along each of the two horizontal coordinates are close to each other. However, the number of computing cores for NEMO and SI3 is allocated in multiples of 32. Therefore, if the computational domain is partitioned into 48×48 boxes, the next “well-partitioned computational domain” configuration with a larger number of computational cores, contains 50×48 rectangles. The increase in the first multiplier occurs up to the configuration with the number of rectangles equal to 64×48 . After that, the second multiplier starts to increase.

The second explanation for the non-monotonic scaling of SLNE with fixed number of computational cores allocated for SLAV is the heterogeneity of the computational system. To illustrate this assertion, together with measurements of the computation time of SLAV time step, we measured the computation time of time step of this model without taking into account the time spent on parallel exchanges with the ocean model, preparation of relevant data, and other coupling overheads. The computation time of SLAV’s time step is illustrated in Fig. 1d. The colors of the dots here correspond to the colors in Fig. 1b. It can be seen that one SLAV’s time step average computation time is almost constant with increasing number of computational cores allocated for NEMO. Small deviations around this value can be interpreted as inhomogeneity of the computational system. The dispersion of these deviations increases with the increasing of the number of computational cores allocated for SLAV.

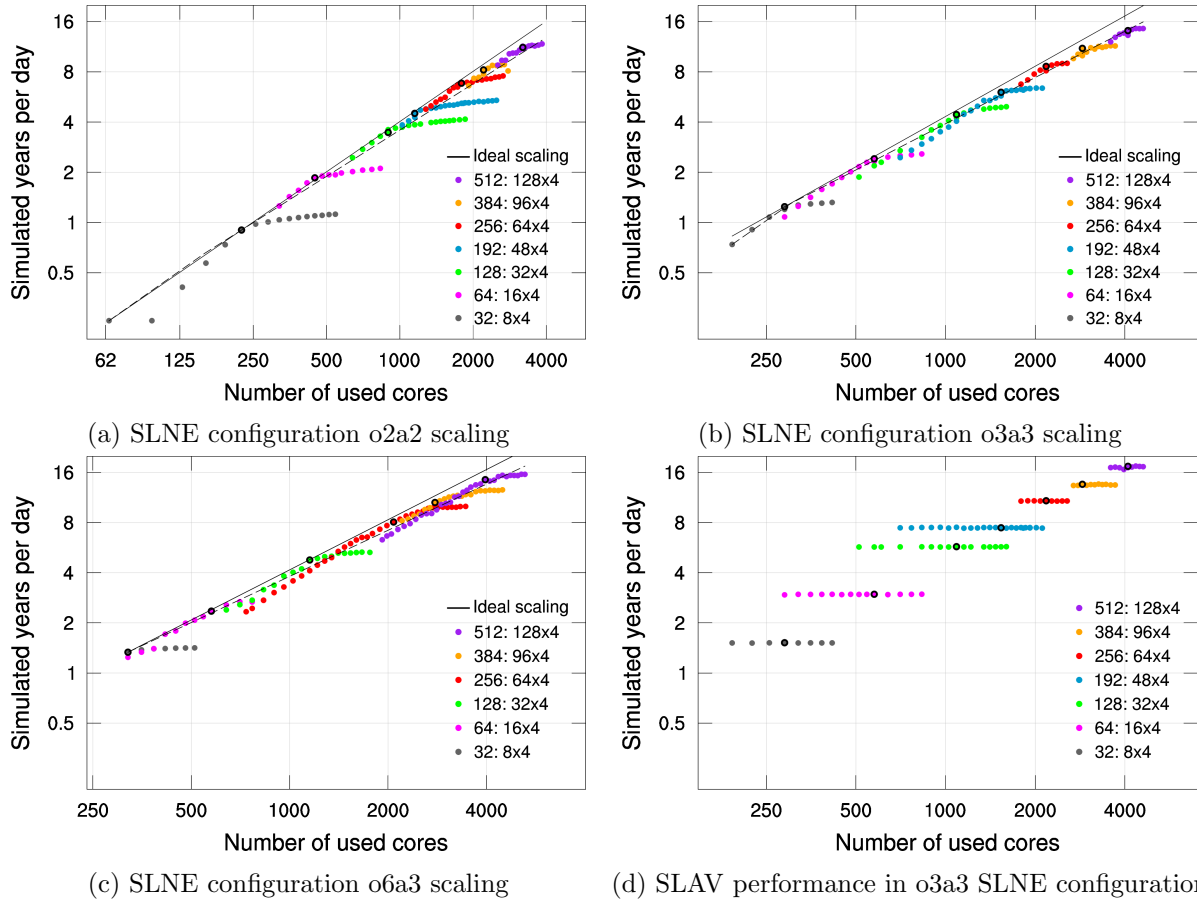


Figure 1. SLNE (a–c) and SLAV (d) computational performance with respect to the number of computational cores allocated for SLNE

The results presented in Fig. 1d allow us to filter out spurious optimal configurations of SLNE model, where the good performance is due to the heterogeneity of the computational system. The optimal configurations will be discussed in the next section.

2.5. SLNE Optimal Configurations

Data exchanges between SLNE components means synchronization of SLAV and NEMO at certain moments of time. If one of the components performs calculations too fast with respect to another model, it will soon be waiting for the data to be exchanged. Waiting for the data means load imbalance between components and a waste of computational resources. The performance optimization of a coupled model relies on the allocation of an optimum number of computational resources to each component. It can be achieved by balancing the load of SLAV and NEMO between the available computing resources.

The optimal configurations of the coupled model are summarized in Tab. 3. The first column in Tab. 3 is the SLNE configuration id; the second column is the number of computational cores used by the coupled model; the third column is the parallel performance of the model (in terms of the number of simulated years per day); the fourth column is the number of computational cores allocated for NEMO; the fifth column represents the way the NEMO computational domain is partitioned; the sixth column is the number of computational cores allocated for SLAV; the seventh column of the table contains the ratio of the computational cores allocated for NEMO and SLAV.

Table 3. SLNE optimal parallel configurations and performance

SLNE id	SLNE comutational cores	Years per day	NEMO comutational cores	NEMO domain decomposition	SLAV comutational cores	NEMO/SLAV comutational cores ratio
o2a2	224	0.9	192	16x12	32	6
	448	1.85	384	24x16	64	6
	896	3.47	768	32x24	128	6
	1152	4.27	960	32x30	192	5
	1728	6.33	1472	46x32	256	5.75
	2208	7.75	1824	48x38	384	4.75
	3200	11.2	2688	56x48	512	5.25
o3a3	288	1.24	256	16x16	32	8
	576	2.35	512	32x16	64	8
	1088	4.44	960	32x30	128	7.5
	1536	6.03	1344	42x32	192	7
	2176	8.63	1920	48x40	256	7.5
	2880	11.06	2496	52x48	384	6.5
	4096	14.13	3584	64x56	512	7
o6a3	320	1.33	288	18x16	32	9
	576	2.41	512	32x16	64	8
	1152	4.77	1024	32x32	128	8
	2080	8.07	1824	48x38	256	7.1
	2784	10.57	2400	50x48	384	6.25
	3968	14.5	3456	64x54	512	6.75

The results presented in Tab. 3 show that the intensity of exchanges directly affects the performance of the model. For example, the o2a2 configuration performance is about 7.75 years per day on 2208 computational cores allocated for the model. In contrast, the o3a3 configuration performance is 8.63 years per day on 2176 computational cores. o3a3 and o6a3 configurations are not significantly different. This can be seen in the Fig. 2, which illustrates the parallel performance of the model on a logarithmic scale. Linear scalability is shown by the solid line. o2a2 configuration parallel efficiency is about 83% on 3000 computational cores with respect to the optimal configuration with the minimal number of computational cores. For the o3a3 and o6a3 configurations SLNE parallel efficiency is about 90% and 88%, respectively. Note, that in our experiments we were limited to 5000 computational cores.

For all model configurations presented in Tab. 3, we can observe a decrease in the last column value. This means that the atmospheric model scales worse than the ocean model. As computational cores allocated for SLNE increase, the atmospheric model requires more resources to have the same performance as NEMO. Possible reasons for this dependence may be the use of one-dimensional domain MPI decomposition in SLAV and the significantly higher resolution of NEMO.

In Fig. 1a–Fig. 1c, it can be seen that the scalability of the coupled model has a similar pattern for different configurations of the atmospheric model. With increasing number of NEMO computational cores, SLNE exhibits a so-called super-linear speed-up, followed by a low speed-up. Each of these segments is an illustration of the imbalance of the coupled model components.

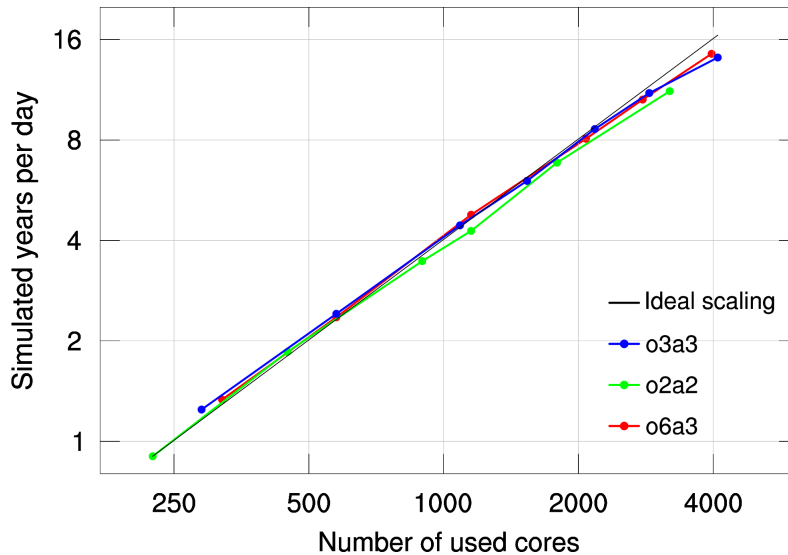


Figure 2. SLNE scaling: optimal configurations only

In these figures, the optimal SLNE configurations for each SLAV configuration are highlighted by the black border of the dot. At optimal configurations both components arrive at about the same speed and costs. At the super-linear speed-up segment the fastest model is SLAV. While at a low speed-up segment NEMO has to wait for the exchanged data before it can perform its own computations. In optimal o2a2 SLNE configurations, the number of computational cores granted for NEMO is 5 to 6 times larger than the number of computational cores allocated for SLAV. In the case of less frequent exchanges between atmosphere and ocean models, this ratio increases and ranges from 7 to 8 times depending on the overall size of the MPI communicator. Therefore, the optimal performance of the coupled model is achieved around configuration where the ocean and sea ice models do not wait for the exchanged data.

Figure 3–Figure 4 illustrate the optimality of SLNE configurations presented in Tab. 3. Figure 3 illustrates the dependence of the relative cost of atmosphere model program code fragment computation (Fig. 3a) and ocean model (Fig. 3b) depending on the number of SLNE computational cores (caption under the column). The number of computational cores allocated for SLAV in these figures is 128 (128 : 48x4 SLAV configuration). Bar segments of different colors correspond to different fragments of program code. These fragments cover the entire time loop body of the corresponding model. The absolute computation time of one time step is given above the bar at the top of the figures. The proportion (given in %) of the computation time of a software fragment relative to the one time step computational time is given when it exceeds 5%. Both Fig. 3a–Fig. 3b correspond to the o2a2 SLNE configuration.

The following fragments of SLAV program code are highlighted with colors in Fig. 3a: **diag** (blue bar) corresponds to the program fragment responsible for diagnostic data calculation; **get** (yellow bar) – receive data from NEMO; **put** (blue bar) – send data to NEMO; **prep2send** (green bar) – computation of SLAV data to be sent to NEMO; **step** (red bar) – calculation of one time step (dynamical core and parameterizations of sub-grid scale processes).

Figure 3b corresponds to NEMO model program code fragments: **sbc_snd** (purple bar) – send data to SLAV; **diag_other** (sand colored bar) – diagnostics procedures, tracer computation; **adv** (dark blue bar) – advection, lateral mixing and vertical diffusion; **dyn** (cyan bar) – dynamic core; **vertdyn** (brown bar) – vertical and lateral physics; **thdyn** (crimson bar) – thermodynamics; **sbc_2** (light blue bar) – add runoffs to fresh water fluxes and control the

freshwater budget, update stochastic parameters; **sbc_ice** (yellow bar) – SI3 ice model; **sbc_rcv** (blue bar) – compute ocean surface boundary condition using data received from SLAV; **sbc** (green bar) – forcing field computation; **sbc_1** (red bar) corresponds to dynamic update and tracer data computation at open boundaries.

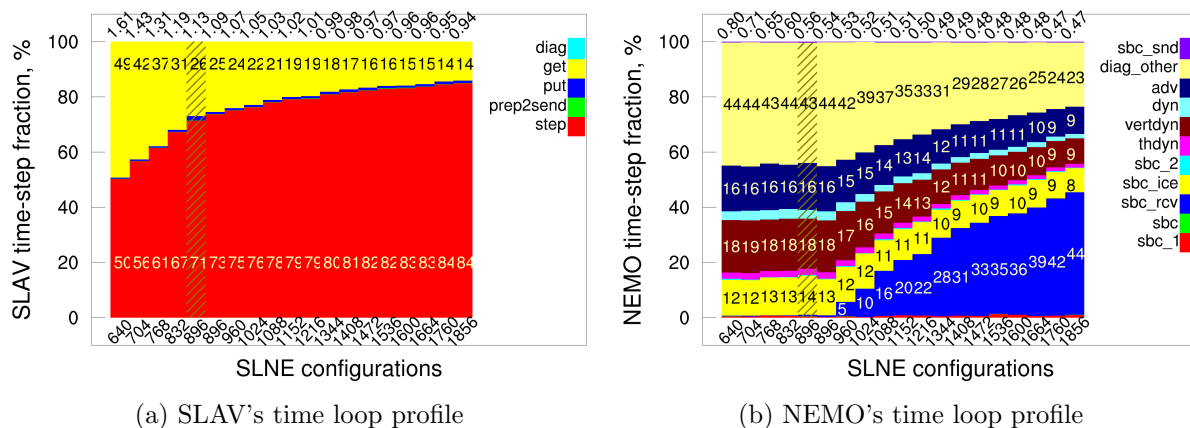


Figure 3. o2a2 SLNE configuration: time step fraction of SLAV (a) and NEMO (b) models with respect to the number of SLNE computational cores (captions under the columns); SLAV model runs on 128 cores

In Fig. 3a, it can be seen that as the number of NEMO computing cores increases, the waiting time of SLAV decreases. In Fig. 3b, the time fraction of the program code fragment **sbc_rcv** responsible for receiving data from SLAV is negligible for the small number of computational cores allocated for NEMO. However, as the number of these computational cores increases, the ocean model starts to compute faster and the fraction of this fragment starts to increase rapidly. In comparison to SLAV, ocean and sea ice model require significantly more computational resources for computation. Therefore, NEMO should not wait for data from SLAV. In contrast, if the atmospheric model spends part of its computational time waiting for data from NEMO, it is not essential because SLAV uses a relatively small fraction of the computational resources (from 11% to 20%). Therefore, the optimal configuration of the coupled model SLNE is achieved when NEMO computes slightly slower in comparison to SLAV.

Another feature of the results presented in Fig. 3a–Fig. 3b is the following. The procedure performing data transfer from SLAV to NEMO requires less time than data exchange from NEMO to SLAV. In the o2a2 configuration, the time fraction of the NEMO **sbc_rcv** procedure increases from about zero to 40%. At the same time, the number of computational cores granted to NEMO doubles. The number of SLAV cores is 128. The time fraction of SLAV **get** function decreases from 25% to 14%. Similar dependence can be seen in Fig. 4, where the o3a3 configuration is presented. Figures 4a and 4b differ in the number of computational cores allocated for SLAV: in Fig. 4a, it corresponds to 128 computational cores, and in Fig. 4b to 512.

2.6. SLNE Performance vs Number of Exchanged Arrays

This section presents the results of a performance study of the optimal o3a3 SLNE configuration as a function of the number of two dimensional arrays exchanged between the atmosphere and ocean models. The numerical experiments were organized as follows. For each optimal SLNE configuration, three additional experiments were performed. In the first experiment, the number of data arrays sent from SLAV to NEMO and from NEMO to SLAV was increased by 10. In

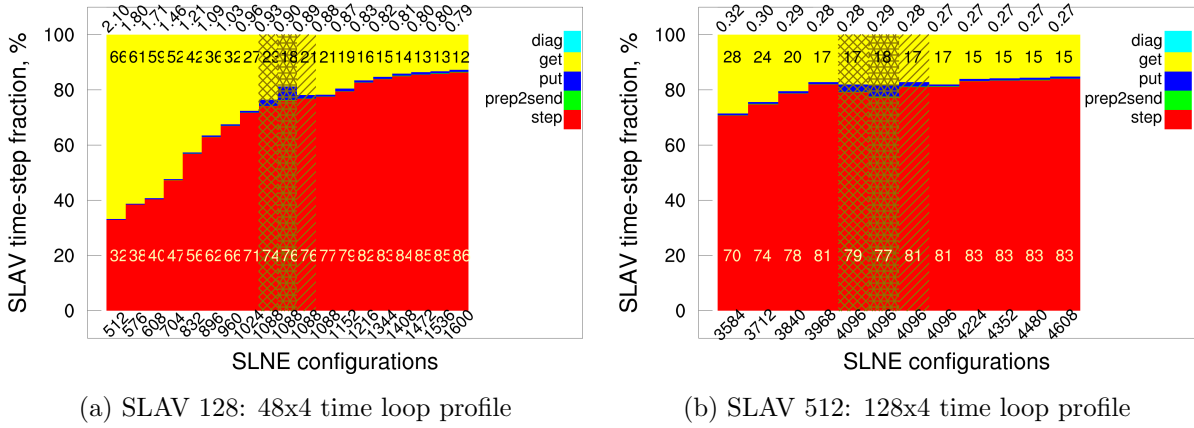


Figure 4. SLAV time step fraction with respect to the number of o3a3 SLNE computational cores (captions under the columns); SLNE configurations with additional exchanges are highlighted by stripes over the bars

the second and third experiments it was increased by 25 and 50, respectively. The additional exchanges and interpolations were performed in both directions. Therefore, in the first experiment, the total number of arrays exchanged between models was increased from 17 to 37. In the second and third experiments – to 67 and 117, respectively. Extra20, extra50 and extra100 will be used to indicate these configurations. The additional data exchanged between the models corresponded to the solar radiation surface heat flux. The obtained results of the experiments are presented in Tab. 4. The first row of the table lists the computational cores allocated for the SLNE model. The second row gives the performance (in terms of simulated years per day) of the o3a3 model in reference version with 17 exchanged arrays. Other rows of the table give the performance of extra20, extra50 and extra100 SLNE configurations with additional exchanges. The last column shows the average relative performance slowdown (given in %) of the corresponding o3a3 model configuration compared to the optimal model version.

Table 4. Parallel performance of optimal SLNE configurations in the reference version and in versions with increased number of data exchanged between models

conf.	288	576	1088	1536	2176	2880	4096	rel., %
optimal	1.24	2.41	4.44	6.03	8.63	11.06	14.13	–
extra20	–	2.40	4.39	5.95	8.63	10.36	13.85	1.9
extra50	1.20	2.33	4.37	5.86	8.37	10.34	13.25	3.9
extra100	1.22	2.32	4.24	5.74	8.12	9.99	13.63	4.9

As can be seen in Tab. 4, increasing the number of exchanged arrays by 20 leads to an insignificant slowdown of the model by a value of about several percent in comparison to reference (optimal) version of the model. In the case when the number of exchanged arrays significantly increases, the model performance slowdown is on average 5%, and for some configurations it reaches 7% slowdown. Note that only one experiment was performed for each configuration. Therefore, the results presented in the table may not be statistically significant. However, they show relatively small overheads associated with the data exchange and interpolation compared to the one time step component integration time.

The results of model profiling using ParProf in the version with increased number of exchanged arrays are shown in Fig. 4. These configurations are highlighted by stripes over the bar. Stripes inclined to the right corresponds to the extra20 configuration. Other cross-stripe patterns match the extra 50 (square tiles) and the extra100 (triangular tiles) configurations. The bars without stripes correspond to the reference (optimal) o3a3 SLNE version with 1088 (Fig. 4a) and 4096 (Fig. 4b) computational cores.

Parallel profile of SLNE components shown in Fig. 3–Fig. 4 gives a better understanding of the development of load imbalance costs. Additional exchanges lead to an increase in the time fraction of the `put` routine sending data from SLAV to NEMO. A similar result was obtained for the o2a2 SLNE configuration, that is shown in Fig. 3. Additional exchanges lead to extra costs of SLAV’s `put` routine and NEMO `sbc_rcv` routine. Change in the average computation time of one time step in this experiment is not substantial and is equal to 3.1% (it increases from 1.095 s. to 1.13 s.)

3. SLNE Diagnostics

3.1. Interpolation Accuracy

To evaluate the accuracy of interpolation of the data exchanged between SLNE components, we performed so-called ping-pong test. This test implies that the data are passed forth and back between SLAV and NEMO sequentially without being modified in these models. Each exchanging of data between components consists of a mapping operation that interpolates the data using bilinear interpolation scheme. In the experiment we used o1a1 version of SLNE model where data exchange is performed every time step in both models. In o1a1 configuration 128 computational cores are granted to SLAV and 768 cores are allocated for NEMO. At the initial moment of time, the data exchanged by the models corresponded to the downward solar radiation surface flux (see Fig. 5a). This heat flux has a sharp structure because it depends on cloudiness, cloud liquid water content, surface albedo, aerosol mixing ratio, etc.

Figure 5b shows the appropriate heat flux after performing of 840 exchanges from SLAV to NEMO and 840 exchanges in the opposite direction. Thus, the flux was interpolated 1680 times from the regular latitude-longitude grid to the tripolar ORCA025 grid and back. This number of exchanges corresponds to the 14 simulated days.

It can be seen that the solution is significantly smoothed. Figure 5c–Figure 5d show the difference of these fields after performing two (Fig. 5c) and 1680 (Fig. 5d) procedures of mapping. It can be seen that after performing 1680 procedures of mapping, no spurious extremum and abnormal anomalies due to the presence of the coastline and the water surface–sea ice interface appeared in test data. Global mean downward surface solar radiation used as test data decreases from 171.64 to 171.17 W/m² (0.27%).

3.2. Results of Numerical Simulations

Concurrent execution of multiple components may induce numerical instability, which can be caused by the following conditions: inconsistent coastal line of atmosphere and ocean models, coupling data errors, imbalance of physical processes at the components interface, etc. The development of numerical instability due to these errors and the importance of correctly coupling the components is shown in [2, 26, 35, 74]. To understand the feedbacks between components of

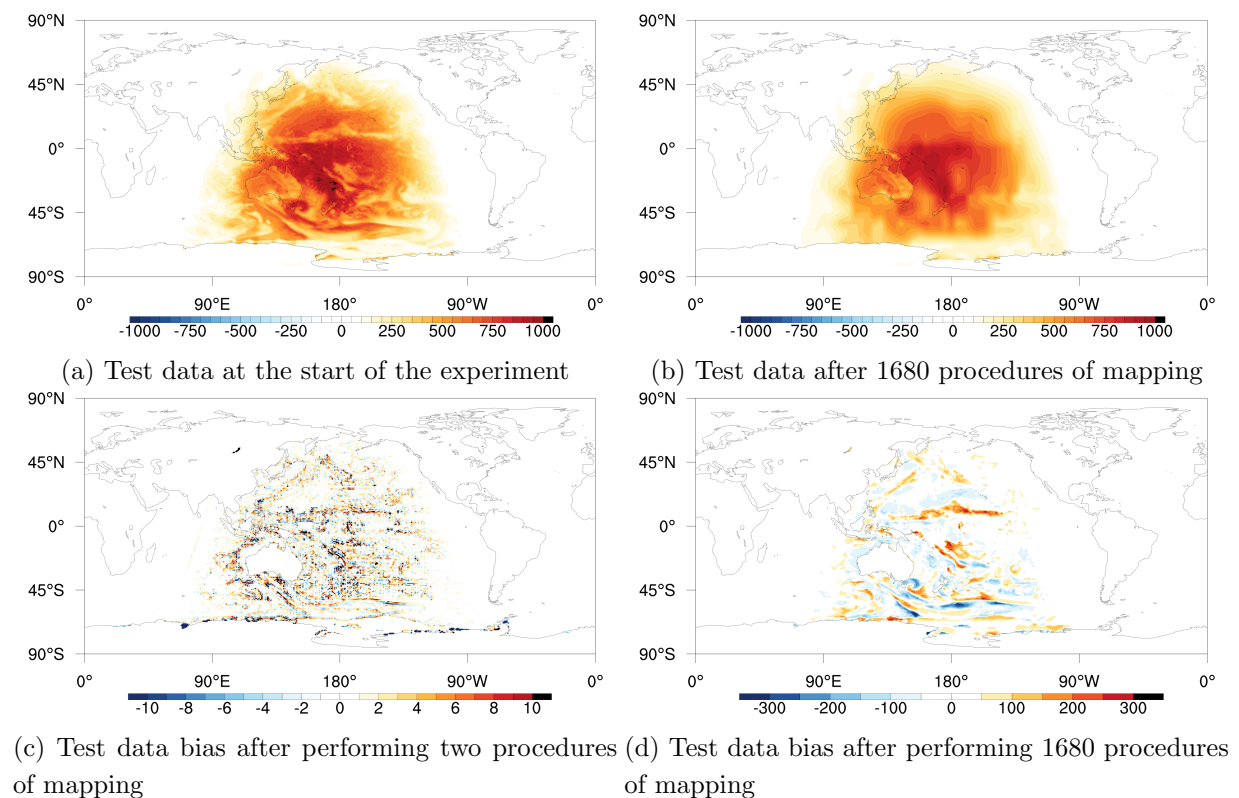


Figure 5. Results of ping-pong test where the test data corresponds to the solar radiation heat flux on the Earth's surface (W/m^2)

the coupled model at different time scales several experiments were performed. The start date in all experiments corresponded to 00 UTC on November 11, 2021.

In the first experiment, SLNE model was run for 14 days. Figure 6a shows the surface temperature after 14 days of SLNE integration. Figure 6b illustrates the surface temperature bias with respect to the atmospheric and ocean state corresponding to 00 UTC November 25, 2021 and obtained using the assimilation technology developed at RMHS [58]. A similar methodology was used to prepare the initial state of the ocean and sea ice.

In Fig. 6b, it can be seen that the surface temperature on land differs significantly from the observed temperature. This is due to the nonlinear nature of simulated processes: the lead time exceeds the hydrodynamic limit of predictability. Above sea ice, the bias is most noticeable near the water–sea ice boundary. The ocean surface temperature bias for 14 days of model integration in some regions reaches 2° Celsius, but, in general, it is not large. Numerical instability and large errors near the shoreline do not appear in the model. Errors near the sea ice boundary are probably due to an insufficiently accurate method for computation of atmospheric surface heat fluxes. The error reduction can be expected in case of implementation of the mosaic approach to representing surface heterogeneity.

SLNE model in version 01a1 has been integrated for a period of up to 3 years. Simulations showed that there were no significant systematic errors in the coupled model. The averaged sea ice extent in the Northern and Southern hemispheres remains approximately the same. The seasonal variations of surface temperature of sea ice and ocean in tropics are consistent with the reanalysis. A more detailed study of the medium-range and seasonal predictability of the model will be performed in the next studies.

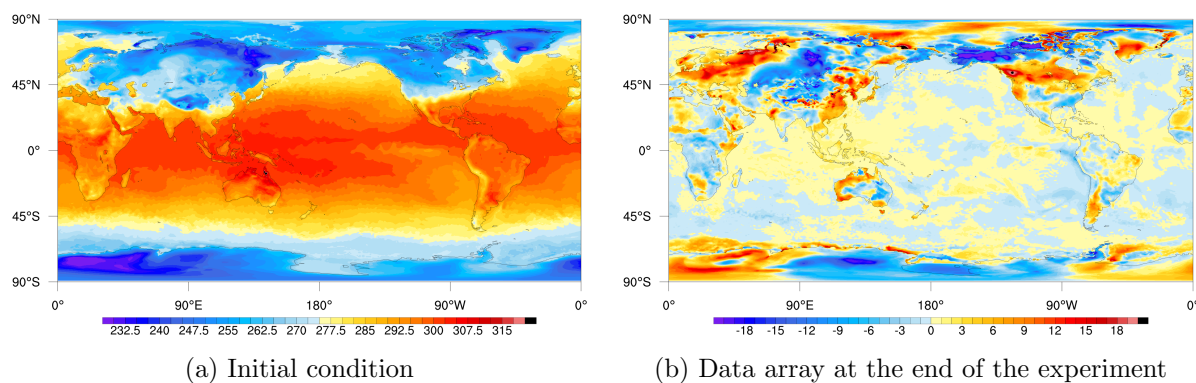


Figure 6. Test data array corresponding to the initial moment of time to the solar radiation on the Earth’s surface (W/m^2)

Conclusion

The paper presents the first version of SLNE coupled model. This model consists of an atmospheric model SLAV072L96, an ocean model NEMO4 and a sea ice model SI3. The models are coupled using OASIS3-MCT software. We propose three SLNE configurations that differ in the coupling period.

The total duration of a coupled model simulation can be separated into two parts for each component: a computing time and a waiting time during which a component waits for boundary conditions. An efficient use of the computing resources requires synchronizing of the component computational speed. In order to find these model configurations, we performed several experiments. It was found that in optimal parallel configurations SLAV model has to wait for data from NEMO. The fraction of SLAV’s waiting time is about 10%. Nevertheless, this time is not significant because SLAV requires 6 to 7 times fewer computational resources than NEMO.

The optimal SLNE configurations are run using from from 224 to 4096 computational cores of Cray XC40-LC HPC system installed at the Main Computer Center of Federal Service for Hydrometeorology and Environmental Monitoring. Coupled model scales quite well: scaling efficiency of about 85% on 4000 computational cores in comparison to the configuration using on 224 cores. Unfortunately, in our study, we were limited to 5000 computational cores.

The results of simulations with lead time ranges from 14 to 1100 days showed the absence of numerical instability and significant systematic errors of surface fields. Further research will focus on coupled model tuning and studying the accuracy of simulation.

Acknowledgements

The author is grateful to Yury Resnyanskii, Boris Strukov and Alexandr Zelen’ko for NEMO–SI3 model configuration and initial data for it. The author is appreciative of valuable comments from Mikhail Tolstykh, Gordey Goyman, Andrey Kuleshov and Konstantin Belyaev. The research presented in Sections 1 and 2 was carried out with the support of the Russian Science Foundation, Project No. 22-11-00053. The research presented in Section 3 was supported by the Moscow Center of Fundamental and Applied Mathematics at INM RAS (Agreement with the Ministry of Education and Science of the Russian Federation No. 075-15-2022-286).

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Aleksandrov, V.V., Arkhipov, P.L., Parkhomenko, V.P., Stenchikov, G.L.: Globalnaia model sistemy okean-atmosfera i issledovanie ee chuvstvitelnosti k izmeneniiu kontsentratsii CO₂ [A global model of the ocean-atmosphere system and the study of its sensitivity to changes in CO₂ concentration]. *Izvestiia AN SSSR. Fizika Atmosfery i Okeana* 19(5), 451–458 (1983) (in Russian).
2. Beljaars, A., Dutra, E., Balsamo, G., Lemarié, F.: On the numerical stability of surface–atmosphere coupling in weather and climate models. *Geoscientific Model Development* 10(2), 977–989 (2017). <https://doi.org/10.5194/gmd-10-977-2017>
3. Byrne, N.J., Shepherd, T.G., Polichtchouk, I.: Subseasonal-to-seasonal predictability of the Southern Hemisphere eddy-driven jet during austral spring and early summer. *J. of Geop. Res.: Atmospheres* 124, 6841–6855 (2019). <https://doi.org/10.1029/2018JD030173>
4. Callendar, G.S.: The artificial production of carbon dioxide and its influence on temperature. *Q.J.R. Meteorol. Soc.* 64, 223–240 (1938). <https://doi.org/10.1002/qj.49706427503>
5. Chamberlin, T.C.: A group of hypotheses bearing on climatic changes. *J. Geol.* 5, 653–683 (1897). <http://www.jstor.org/stable/30054630>
6. Chou, M.-D., Suarez, M.J.: A solar radiation parameterization (CLIRAD-SW) for atmospheric studies. NASA Tech. Memo. 10460(15) (1999). <https://ntrs.nasa.gov/api/citations/19990060930/downloads/19990060930.pdf>
7. Croll, J.: *Climate and Time in Their Geological Relations*. Edinburgh: Adam and Charles Black (1885).
8. Davis, P., Ruth, C., Scaife, A.A., Kettleborough, J.: A large ensemble seasonal forecasting system: GloSea6. In: *AGU Fall Meeting Abstracts*, vol. 2020, pp. A192-05 (2020).
9. Doose, K.: Modelling the future: climate change research in Russia during the late Cold War and beyond, 1970s–2000. *Climatic Change* 171(6) (2022). <https://doi.org/10.1007/s10584-022-03315-0>
10. Doyle, J.D., Hodur, R.M., Chen, S., *et al.*: Tropical cyclone prediction using COAMPS-TC. *Oceanography* 27(3), 104–115 (2014). <https://doi.org/10.5670/oceanog.2014.72>
11. Ďurán, I.B., Geleyn, J-F., Váňa, F.: A Compact Model for the Stability Dependency of TKE Production–Destruction–Conversion Terms Valid for the Whole Range of Richardson Numbers. *J. Atmos. Sci.* 71, 3004–3026 (2014). <https://doi.org/10.1175/JAS-D-13-0203.1>
12. Edwards, P.N.: History of climate modeling. *WIREs Clim Change* 2, 128–139 (2011). <https://doi.org/10.1002/wcc.95>

13. Eyring, V., Bony, S., Meehl, G.A., *et al.*: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geosci. Model Dev.* 9, 1937–1958 (2016). <https://doi.org/10.5194/gmd-9-1937-2016>
14. Fadeev, R.Yu., Ushakov, K.V., Tolstykh, M.A., Ibrayev, R.A.: Design and development of the SLAV-INMIO-CICE coupled model for seasonal prediction and climate research. *Rus. J. of Num. An. and Math. Mod.* 33(6), 333–340 (2018). <https://doi.org/10.1515/rnam-2018-0028>
15. Fadeev, R.Yu., Tolstykh, M.A., Volodin, E.M.: Climate version of the SL-AV global atmospheric model: development and preliminary results *Russ. Meteorol. Hydrol.* 44(1), 13–22 (2019). <https://doi.org/10.3103/S1068373919010023>
16. Fadeev, R.Y., Alipova, K.A., Koshkina, A.S., *et al.*: Glacier parameterization in SLAV numerical weather prediction model. *Rus. J. of Num. An. and Math. Mod.* 37(4), 189–201 (2022). <https://doi.org/10.1515/rnam-2022-0016>
17. Fadeev, R.Yu.: Wind gustiness parameterization and long-range weather prediction. *Proc. of Hydrometcentre of Russia* 2(388), 35–55 (2023) (in Russian). <https://doi.org/10.37162/2618-9631-2023-2-35-54>
18. Fraedrich, K., Kirk, E., Luksch, U. Lunkeit, F.: The Portable University Model of the Atmosphere (PUMA): stormtrack dynamics and low-frequency variability. *Meteorologische Zeitschrift* 14, 735–745 (2005). <https://doi.org/10.1127/0941-2948/2005/0074>
19. Fultz, D.: *Dynamics of Climate*. New York: Pergamon Press, 71–77 (1960).
20. Gerard, L., Piriou, J., Brožková, R., *et al.*: Cloud and Precipitation Parameterization in a Meso-Gamma-Scale Operational Weather Prediction Model *Mon. Wea. Rev.* 137, 3960–3977 (2009). <https://doi.org/10.1175/2009MWR2750.1>
21. Global Deterministic Prediction System (GDPS): Update from version 4.0.1 to version 5.0.0. Canadian Meteorological Centre Tech. Note 59 (2015). https://collaboration.cmc.ec.gc.ca/cmc/cmci/product_guide/docs/tech_notes/technote_gdps-500_20151215_e.pdf
22. Golubeva, E.N., Ivanov, U.A., Kuzin, V.I., Platov, G.A.: A numerical modeling of the world ocean circulation with allowance for the upper quasi-homogeneous layer. *Oceanologia* 32(3), 395–405 (1992).
23. Golubeva, E.N., Platov, G.A.: On improving the simulation of Atlantic water circulation in the Arctic Ocean. *J. Geophys. Res.* 112, C04S05 (2007). <https://doi.org/10.1029/2006JC003734>.
24. Gradov, V.S., Platov, G.A.: Overview of SCM Coupler and Its Application for Constructing Climate Models. *Supercomputing Frontiers and Innovations* 10(1), 58–76 (2023). <https://doi.org/10.14529/jsfi230106>
25. Guérémy, J.-F., Dubois, C., Viel, C., *et al.*: Documentaton of the METEO-FRANCE seasonal forecasting system 8 ECMWF Copernicus report. <http://www.umr-cnrm.fr/IMG/pdf/system8-technical.pdf>






-
26. Hallberg, R.: Numerical instabilities of the ice/ocean coupled system. CLIVAR Exchanges 19(69), 38–42 (2014). <https://doi.org/10.1016/j.ocemod.2008.05.005>
 27. Hersbach, H., Bell, B., Berrisford, P., *et al.*: The ERA5 global reanalysis. Quart. J. Roy. Meteor. Soc. 146(730), 1999–2049 (2020). <https://doi.org/10.1002/qj.3803>
 28. Hirahara, S., Kubo, Y., Yoshida, T., *et al.*: Japan Meteorological Agency/Meteorological Research Institute Coupled Prediction System Version 3 (JMA/MRI-CPS3). Journal of the Meteorological Society of Japan. Ser. II, 101(2), 149–169 (2023). <https://doi.org/10.2151/jmsj.2023-009>
 29. Hoskins, B.: The potential for skill across the range of the seamless weather-climate prediction problem: a stimulus for our science. Q.J.R. Meteorol. Soc. 139, 573–584 (2013). <https://doi.org/10.1002/qj.1991>
 30. Hunke, E.C., Lipscomb, W.H., Turner, A.K., *et al.*: CICE: the Los Alamos Sea Ice Model documentation and software users manual, version 5.1. Technical report LA-CC-06-012. Los Alamos National Laboratory: Los Alamos, NM. 2015. <http://www.ccpo.odu.edu/~klinck/Reprints/PDF/cicedoc2015.pdf>
 31. Ibrayev, R.A., Ushakov, K.V., Khabeev, R.N.: Eddy-resolving $1/10^\circ$ model of the World Ocean. Izvestiya, Atmosphere and Ocean Phys. 48, 37–46 (2012). <https://doi.org/10.1134/S0001433812010045>
 32. Kalmykov, V.V., Ibrayev, R.A., Kaurkin, M.N., Ushakov, K.V.: Compact modeling framework v3.0 for high-resolution global ocean–ice–atmosphere models. Geosci. Model Dev. 11, 3983–3997 (2018). <https://doi.org/10.5194/gmd-11-3983-2018>
 33. Kasahara, A., Washington, W.M.: NCAR global general circulation model of the atmosphere. Mon. Weather Rev. 95, 389–402 (1967).
 34. Larson, J., Jacob, R., Ong, E.: The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models. Int. J. High Perf. Comp. App. 19(3), 277–292 (2005). <https://doi.org/10.1177/1094342005056115>
 35. Lemarie, F., Blayo, E., Debreu, L.: Analysis of ocean–atmosphere coupling algorithms: Consistency and stability. Procedia Computer Science 51, 2066–2075 (2015). <https://doi.org/10.1016/j.procs.2015.05.473>
 36. Lin, H., Merryfield, W.J., Muncaster, R., *et al.*: The Canadian Seasonal to Interannual Prediction System Version 2 (CanSIPsv2). Weather and Forecasting 35(4), 1317–1343 (2020). <https://doi.org/10.1175/WAF-D-19-0259.1>
 37. MacLachlan, C., Arribas, A., Peterson, K.A., *et al.*: Global seasonal forecast system version 5 (GloSea5): a high-resolution seasonal forecast system. Quart. J. Roy. Meteor. Soc. 141, 1072–1084 (2015). <https://doi.org/10.1002/qj.2396>
 38. Madec, G., Bell, M., Blaker, A., *et al.*: NEMO Ocean Engine Reference Manual. Zenodo (2023). <https://doi.org/10.5281/zenodo.8167700>
 39. Madec, G., Imbard, M.: A global ocean mesh to overcome the North Pole singularity. Climate Dynamics 12, 381–388 (1996). <https://doi.org/10.1007/BF00211684>

40. Magnusson, L., Bidlot, J.-R., Bonavita, M., *et al.*: ECMWF activities for improved hurricane forecasts. *Bulletin of the American Meteorological Society* 100(3), 445–458 (2019). <https://doi.org/10.1175/BAMS-D-18-0044.1>
41. Manabe, S., Bryan, K.: Climate calculations with a combined ocean-atmosphere model. *J. Atmos. Sci.* 26, 786–789 (1969). [https://doi.org/10.1175/1520-0469\(1969\)026<0786:CCWACO>2.0.CO;2](https://doi.org/10.1175/1520-0469(1969)026<0786:CCWACO>2.0.CO;2)
42. Meleshko, V.P., Matyugin, V.A., Sporyshev, P.V., *et al.*: MGO general circulation model (version MGO-03 T63L25). *Proc. of Voeikov Main Geop. Obs.* 571, 5–87 (2014) (in Russian).
43. Mirvis, V.M., Meleshko, V.P., Lvova, T.Yu., *et al.*: Forecast experiments based on MGO coupled ocean-atmosphere model. *Proc. of Voeikov Main Geop. Obs.* 583, 129–148 (2016) (in Russian).
44. Mlawer, E.J., Taubman, S.J., Brown, P.D., *et al.*: RRTM, a validated correlated-k model for the longwave. *J. Geophys. Res.* 102(16), 663–682 (1997). <https://doi.org/10.1029/97JD00237>
45. Moiseev, N.N., Aleksandrov, V.V., Tarko, A.M.: Opyt sistemnogo analiza i eksperimenty s model'yu [Experience with systems analysis and experimentation with models]. *Chelovek i biosfera*, Moscow, Nauka (1983) (in Russian).
46. Molod, A., Salmun, H.: A global assessment of the mosaic approach to modeling land surface heterogeneity. *J. Geophys. Res.* 107(D14) (2002). <https://doi.org/10.1029/2001JD000588>
47. Phillips, N.A.: The general circulation of the atmosphere: a numerical experiment. *Q.J.R. Meteorol. Soc.* 82, 123–164 (1956). <https://doi.org/10.1002/qj.49708235202>
48. Piacentini, A., Maisonnave, E., Jonville, G., *et al.*: A parallel SCRIP interpolation library for OASIS. https://oasis.cerfacs.fr/wp-content/uploads/sites/114/2021/08/GLOBE_WN_Piacentini_Parallel_SCRIP_cmhc_18_34_2018.pdf
49. Platov, G., Krupchatnikov, V., Martynova, Y., *et al.*: A new earths climate system model of intermediate complexity, PlaSim-ICMMG-1.0: description and performance. *IOP Conference Series: Earth and Environmental Science* 96(1), 012005 (2017). <https://doi.org/10.1088/1755-1315/96/1/012005>
50. Russell, R.J.: *Climatic change through the ages*. United States Department of Agriculture, ed. *Climate and Man*. Washington: U.S. Government Printing Office, 67–97 (1941).
51. Saha, S., Moorthi, S., Wu, X., *et al.*: The NCEP climate forecast system version 2. *Journal of Climate* 27, 2185–2208 (2014). <https://doi.org/10.1175/JCLI-D-12-00823.1>
52. Shashkin, V.V., Fadeev, R.Yu., Tolstykh, M.A., *et al.*: Simulation of stratosphere processes with the atmosphere general circulation model SLAV072L96. *Rus. Meteorol. and Hydrol.* 49, 5–20 (2023). <https://doi.org/10.3103/S1068373923060018>
53. Schneider, S.H., Dickinson, R.E.: Climate modeling. *Rev. Geophys. Space Phys.* 12(3), 447–493 (1974). <https://doi.org/10.1029/RG012i003p00447>

-
54. Schwartz, C., Garfinkel, C.I.: Troposphere–stratosphere coupling in subseasonal–to–seasonal models and its importance for a realistic extratropical response to the Madden-Julian oscillation. *J. of Geop. Res.: Atmospheres* 125, e2019JD032043 (2020). <https://doi.org/10.1029/2019JD032043>
55. SCRIP project at GitHub. <https://github.com/SCRIP-Project/SCRIP>
56. Smith, G.C., Bélanger, J., Roy, F., *et al.*: Impact of Coupling with an Ice–Ocean Model on Global Medium-Range NWP Forecast Skill. *Monthly Weather Review* 146(4), 1157–1180 (2018). <https://doi.org/10.1175/MWR-D-17-0157.1>
57. Stepanov, V.N., Resnyanskii, Y.D., Strukov, B.S., *et al.*: Large-scale Ocean Circulation and Sea Ice Characteristics Derived from Numerical Experiments with the NEMO Model. *Russ. Meteorol. Hydrol.* 44, 33–44 (2019). <https://doi.org/10.3103/S1068373919010047>
58. Strukov, B.S., Resnyanskii, Y.D. Zelenko, A.A.: Relaxation Method for Assimilation of Sea Ice Concentration Data in the NEMOLIM3 Multicategory Sea Ice Model. *Russ. Meteorol. Hydrol.* 45, 96–104 (2020). <https://doi.org/10.3103/S1068373920020053>
59. Tarasevich, M., Sakhno, A., Blagodatskikh, D., *et al.*: Scalability of the INM RAS Earth System Model. In: Voevodin, V., Sobolev, S., Yakobovskiy, M., Shagaliev, R. (eds) *Supercomputing. RuSCDays 2023. Lecture Notes in Computer Science*, vol. 14388, pp. 202–216. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-49432-1_16
60. Tarasova, T., Fomin, B.: The Use of New Parameterizations for Gaseous Absorption in the CLIRAD-SW Solar Radiation Code for Models. *J. Atmos. and Oceanic Tech.* 24(6), 1157–1162 (2007). <https://doi.org/10.1175/JTECH2023.1>
61. The Working Group on Numerical Experimentation (WGNE), <https://wgne.net/>
62. Thompson, B., Sanchez, C., Heng, B.C.P., *et al.*: Development of a MetUM (v 11.1) and NEMO (v 3.6) coupled operational forecast model for the Maritime Continent – Part 1: Evaluation of ocean forecasts. *Geosci. Model Dev.* 14, 1081–1100 (2021). <https://doi.org/10.5194/gmd-14-1081-2021>
63. Tolstykh, M.A., Volodin, E.M., Kostykin, S.V., *et al.*: Development of the multiscale version of the SL-AV global atmosphere model. *Rus. Meteorol. and Hydrol.* 40, 374–382 (2015). <https://doi.org/10.3103/S1068373915060035>
64. Tolstykh, M.A., Shashkin, V.V., Fadeev, R.Yu., Goyman, G.S.: Vorticity-divergence semi-Lagrangian global atmospheric model SL-AV20: dynamical core. *Geoscientific Model Development* 10(5), 1961–1983 (2017). <https://doi.org/10.5194/gmd-10-1961-2017>
65. Tolstykh, M.A., Fadeev, R.Yu., Shashkin, V.V., *et al.*: Multiscale Global Atmosphere Model SL-AV: the Results of Medium-range Weather Forecasts. *Russ. Meteorol. Hydrol.* 43, 773–779 (2018). <https://doi.org/10.3103/S1068373918110080>
66. Tolstykh, M.A., Fadeev, R.Yu., Shashkin, V.V., *et al.*: The SLAV072L96 system for long range meteorological forecasts. Submitted to *Rus. Meteorol. and Hydrol.* (2023).

67. Tsyrunikov, M.D., Svirenko, P.I., Gayfulin, D.R., *et al.*: Development of the data assimilation scheme of the hydrometcentre of Russia. Proc. of Hydrometcentre of Russia 4(374), 112–126 (2019) (in Russian).
68. Valcke, S.: The OASIS3 coupler: a European climate modelling community software. Geosci. Model Devel. 6, 373–388 (2013). <https://doi.org/10.5194/gmd-6-373-2013>
69. Vancoppenolle, M., Rousset, C., Blockley, E., *et al.*: SI3 – Sea Ice modelling Integrated Initiative – The NEMO Sea Ice Engine. Zenodo (2023). <https://doi.org/10.5281/zenodo.7534900>
70. Volodin, E.M., Lykossov, V.N.: Parametrization of Heat and Moisture Transfer in the Soil–Vegetation System for Use in Atmospheric General Circulation Models: 1. Formulation and Simulations Based on Local Observational Data. Izv., Atm. and Oceanic Phys. 34, 402–416 (1998).
71. Volodin, E.M., Mortikov, E.V., Kostykin, S.V., *et al.*: Simulation of the present day climate with the climate model INMCM5. Clim. Dyn. 49, 3715–3734 (2017). <https://doi.org/10.1007/s00382-017-3539-7>
72. Volodin, E.M.: Possible Climate Change in Russia in the 21st Century Based on the INMCM5-0 Climate Model. Rus. Meteorol. and Hydrol. 47(5), 327–333 (2022). <https://doi.org/10.3103/S1068373922050016>
73. Vorobyeva, V., Volodin, E.: Evaluation of the INM RAS climate model skill in climate indices and stratospheric anomalies on seasonal timescale Tellus A: Dynamic Meteorology and Oceanography 73(1), 1892435. <https://doi.org/10.1080/16000870.2021.1892435>
74. Zhang, S., Xu, S., Fu, H., *et al.*: Toward Earth system modeling with resolved clouds and ocean submesoscales on heterogeneous many-core HPCs. National Science Review 10(6) (2023). <https://doi.org/10.1093/nsr/nwad069>
75. WCRP Coupled Model Intercomparison Project (CMIP), <https://www.wcrp-climate.org/wgcm-cmip>
76. Wood, N., Staniforth, A., White, A., *et al.*: An inherently mass-conserving semi-implicit semi-Lagrangian discretization of the deep-atmosphere global non-hydrostatic equations. Q.J.R. Meteorol. Soc. 140, 1505–1520 (2014). <https://doi.org/10.1002/qj.2235>

Quantum-Chemical Study of Gas-Phase 5/6/5 Tricyclic Tetrazine Derivatives

Vadim M. Volokhov¹ , Vladimir V. Parakhin²  Elena S. Amosova¹ ,
David B. Lempert¹ , Tatiana S. Zyubina¹ 

© The Authors 2023. This paper is published with open access at SuperFri.org

The most important task for specialists in the field of energy-intensive compounds is the search for new high-energy density materials and the study of their properties. This paper continues the study of series of tetrazines condensed with different types of azoles and presents the results of study of molecule structure of high-energy 5/6/5 tricyclic 1,2,3,4- and 1,2,4,5-tetrazines annelated with nitro-substituted imidazoles. The enthalpies of formation of the given molecules in the gaseous phase have been determined by high-performance quantum-chemical calculations by various calculation methods within the Gaussian 09 program package: G4, G4MP2, ω B97XD/aug-cc-pVTZ, CBS-4M, B3LYP/6-311+G(2d,p), M062X/6-311+G(2d,p). Different calculation methods and approaches have been compared in terms of their accuracy and time consumption. In addition, vibrational IR spectra have been calculated for the given compounds, and the correspondence of characteristic absorption frequencies to key fragments and functional groups of the structures has been determined. Enthalpy of formation of one of the studied substances (4220 kJ/kg) is the highest one among enthalpies of formation of energy-intensive bis(nitroazolo)tetrazines calculated up to date.

Keywords: high-performance computing, quantum-chemical calculations, enthalpy of formation, high-energy materials, tetrazines, nitroimidazoles, azides.

Introduction

The development of flat condensed polynitrogen polynuclear structures and their functionalization with explosophoric groups is one of the promising trends in the search for new high-energy density materials (HEDMs) [1–4]. Particularly, it is fused tricyclic tetrazines annelated with azoles that have a high energy potential [5, 6]. Since the energy properties of HEDMs largely depend on their enthalpy of formation (ΔH°_f), the accuracy of its determination is very important. Previously, we made a research of approaches to calculating ΔH°_f for several series of tetrazines condensed with nitropyrrroles and nitrotriazoles [7, 8]. It is obvious that derivatives of this class of tricyclic compounds containing nitroimidazole nuclei are also of interest, since the first synthesized representative of tricyclic imidazotetrazines, 3,8-diazido-2,9-dinitroimidazo[1,2-d:2',1'-f][1,2,3,4]tetrazine showed quite promising energy characteristics [9]. It is worth mentioning that in recent decades a significant progress has been made in the synthesis of energy-intensive nitroimidazoles [10, 11].

In the search for new high-energy materials, quantum chemical calculations are becoming increasingly important. In the largest computing centers, up to 40% of computing time is spent on quantum chemical calculations. Such calculations enable researchers to save time and material resources needed to produce the required quantities of the studied substances and to carry out labor-intensive thermochemical studies. They also help to design new molecules of promising compounds that have not yet been obtained in practice and to determine the physicochemical parameters of the mentioned substances with high accuracy. For example, the most accurate calculated values of the enthalpy of formation ΔH°_f are obtained by quantum-chemical calcula-

¹Federal Research Center of Problems of Chemical Physics and Medicinal Chemistry of the Russian Academy of Sciences, Chernogolovka, Moscow Region, Russian Federation

²N.D. Zelinskiy Institute of Organic Chemistry of the Russian Academy of Sciences, Moscow, Russian Federation

tions based on ab initio approaches. In the fundamental work of Curtiss [12], a thorough analysis of the accuracy of quantum-chemical calculations of thermochemical quantities on the example of 454 structures was performed using the G4 method within the Gaussian software, and it was shown that the average deviation of the calculation results from the experimental values in this case was only 0.8 kcal/mol, which for high-enthalpy substances was less than 1%. Over the past 15 years, many studies [13–15] of the approaches to calculate ΔH°_f for a wide range of substances of various classes, using high-level calculations, have shown that methods of the Gaussian family, in particular G4, are effective to estimate ΔH°_f of high-energy polynitrogen compounds.

Therefore, in order to make a correct assessment of the energy potential of tricyclic tetrazine derivatives annelated with nitroimidazoles, the main purpose of this work was to establish the enthalpy of formation by various quantum chemical methods in the gas phase under normal conditions ($\Delta H^\circ_{f(g)}$) and to detect regularities in the dependence of this value on molecular structure of isomeric 5/6/5 tricyclic 1,2,3,4- and 1,2,4,5-tetrazines annelated with dinitroimidazole (the general formula $C_6N_{10}O_8$) as well as isomeric diazido derivatives (the general formula $C_6N_{14}O_4$) (Fig. 1).

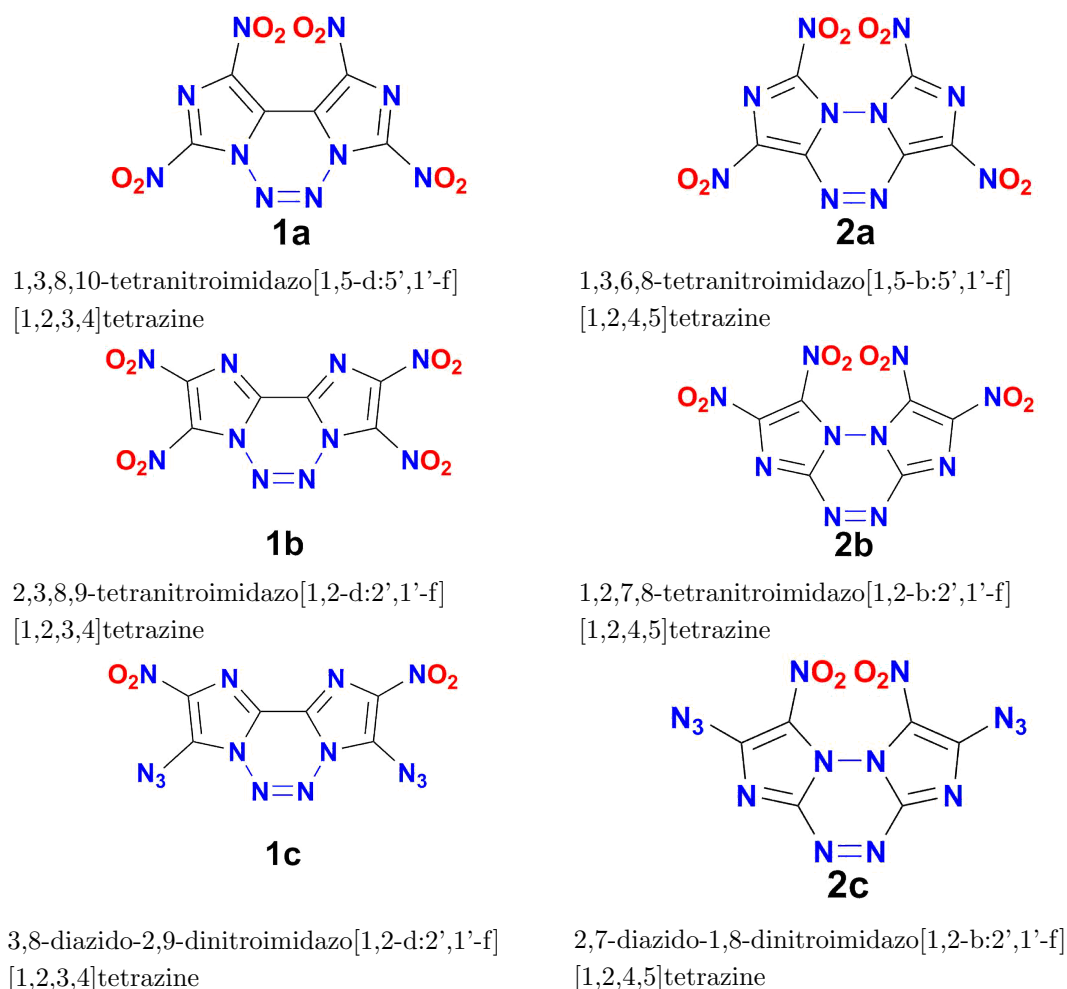


Figure 1. Tricycle molecules under study

1. Calculation Method

To calculate physicochemical properties and thermochemical parameters of the compounds under study, we used the following quantum-chemical methods within the Gaussian 09 program package [16]: hybrid density functionals B3LYP [17, 18], M062X [19] and ω B97XD [20] that includes the empirical variance, with basis sets 6-311+G(2d,p) and aug-ccpVTZ correspondingly, and also the composite G4 and G4MP2 methods [12, 21, 22] and CBS-4M [23, 24] developed by Petersons group. The geometry of the studied molecules was obtained by fully optimizing all geometric parameters using the density functional theory by the ω B97XD/aug-cc-pVTZ method (Fig. 2). The subsequent calculation of vibrational frequencies using the analytical first and second derivatives without taking into account the correction for anharmonicity (absence of imaginary frequencies) confirmed the stability of the obtained configurations. The IR absorption spectra have been calculated at the ω B97XD/aug-cc-pVTZ level introducing a scaling factor of 0.956 to improve the agreement with experiment as recommended in [25].

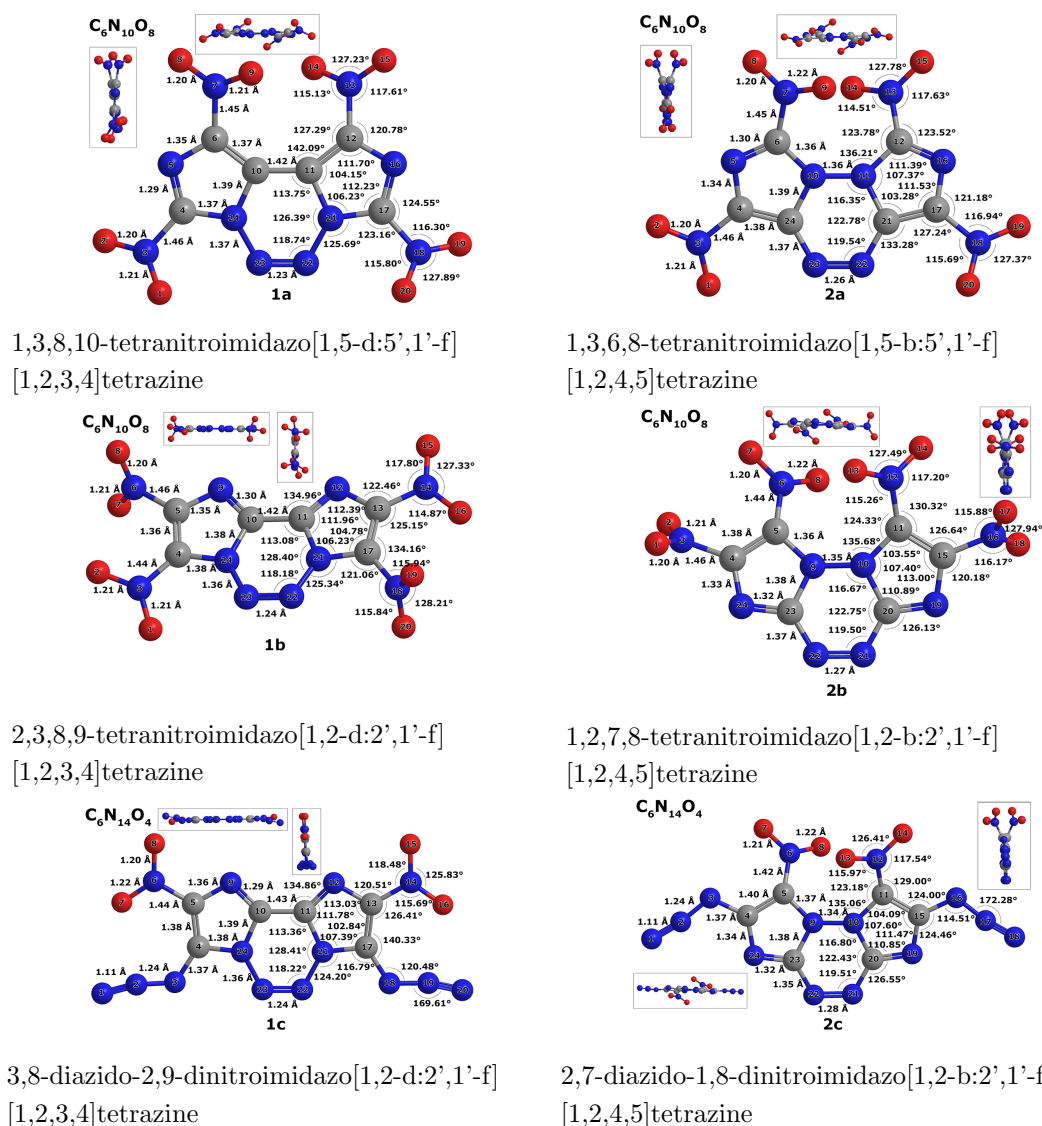
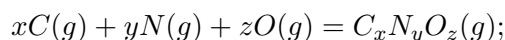


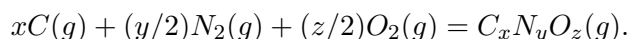
Figure 2. Structures (from different angles) and the most significant geometric parameters (in Å and °) of molecules **1a–c** and **2a–c**

In this work, we used two variants of calculating $\Delta H^\circ_{f(g)}$ of a substance of the general formula $C_xN_yO_z$, based on the energy balance of reactions involving the compound under study:

1) $\Delta H^\circ_{f(g)}(\text{I})$ – using ΔH° of the reaction of atomization of the studied compound:



2) $\Delta H^\circ_{f(g)}(\text{II})$ – using ΔH° of the reaction of formation of the studied compound from simple substances:



It should be noted that using two independent calculation schemes allows us to quantitatively compare the applied quantum-chemical calculation methods.

2. Results and Discussion

2.1. Enthalpy of Formation

For the structures under study, the oxygen saturation coefficient α ($\alpha = 2z/(4x + y)$ for $C_xH_yN_wO_z$) and the nitrogen mass fraction N% are 0.666 and 41.2%, respectively, in the case of isomers **1**, **2a–b**, and 0.333 and 59.0% in the case of isomeric diazides **1**, **2c**. Table 1 and Figure 3 present values of the enthalpy of formation of molecules **1a–c** and **2a–c** in the gas phase obtained by quantum-chemical calculations.

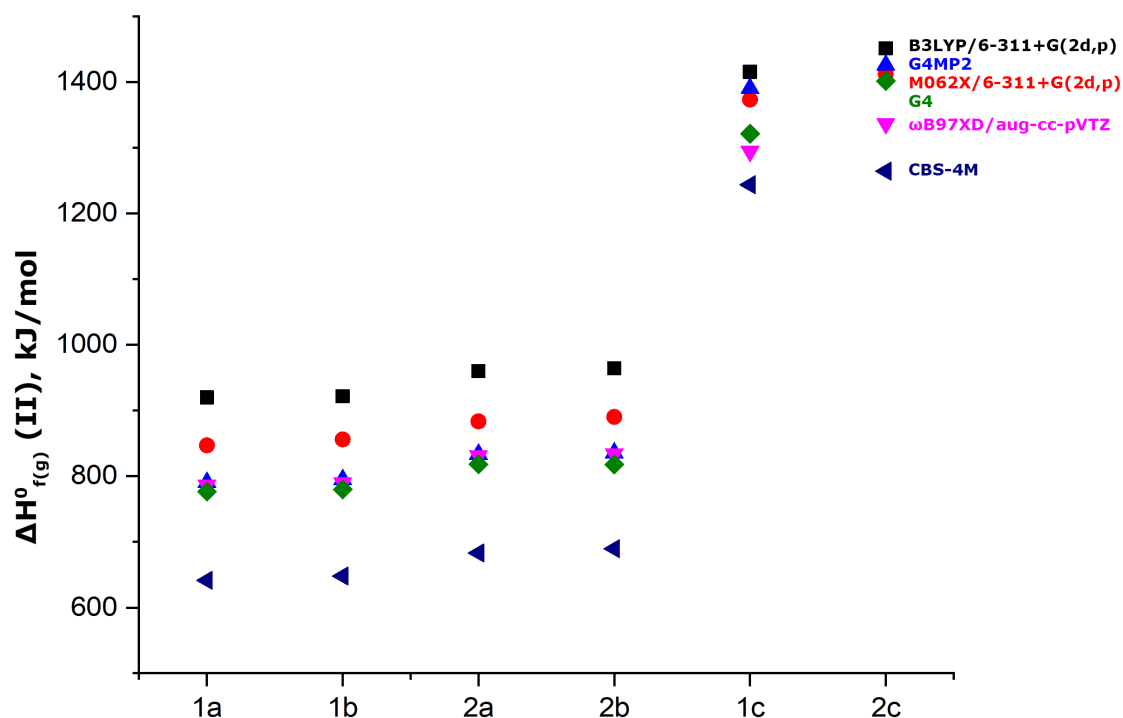


Figure 3. Values of the enthalpy of formation ($\Delta H^\circ_{f(g)}$) of molecules **1a–c** and **2a–c** in the gas phase calculated by different quantum-chemical methods

Table 1. Enthalpy of formation ($\Delta H^\circ_{f(g)}$) of molecules **1a–c** and **2a–c** in the gas phase calculated by different quantum-chemical methods

Calculation method	Enthalpy of formation, $\Delta H^\circ_{f(g)}$, kJ/mol					
	$C_6N_{10}O_8$			$C_6N_{14}O_4$		
	1a	2a	1b	2b	1c	2c
B3LYP/6-311+G(2d,p)	922.74 ^a	962.75 ^a	924.55 ^a	967.31 ^a	1442.41 ^a	1478.33 ^a
M062X/6-311+G(2d,p)	919.84 ^b	959.85 ^b	921.65 ^b	964.41 ^b	1415.34 ^b	1451.27 ^b
G4MP2	942.98 ^a	979.59 ^a	952.30 ^a	986.38 ^a	1498.04 ^a	1537.07 ^a
ω B97XD/aug-cc-pVTZ	846.79 ^b	883.40 ^b	856.11 ^b	890.19 ^b	1373.30 ^b	1412.33 ^b
G4	809.65 ^a	852.83 ^a	813.90 ^a	854.87 ^a	1394.94 ^a	1430.36 ^a
CBS-4M	790.48 ^b	833.66 ^b	794.73 ^b	835.70 ^b	1390.55 ^b	1425.96 ^b
	807.82 ^a	852.43 ^a	811.48 ^a	855.26 ^a	1375.62 ^a	1417.95 ^a
	786.39 ^b	830.99 ^b	790.04 ^b	833.82 ^b	1294.27 ^b	1336.59 ^b
	777.36 ^a	818.89 ^a	780.56 ^a	818.83 ^a	1315.56 ^a	1395.85 ^a
	776.40 ^b	817.92 ^b	779.89 ^b	817.86 ^b	1321.22 ^b	1401.51 ^b
	752.46 ^a	793.96 ^a	758.90 ^a	800.47 ^a	1381.72 ^a	1402.28 ^a
	641.60 ^b	683.09 ^b	648.03 ^b	689.60 ^b	1243.70 ^b	1264.26 ^b

^a $\Delta H^\circ_{f(g)}$ (I) calculated using the atomization reaction of the compound under study

^b $\Delta H^\circ_{f(g)}$ (II) calculated using the formation reaction of the compound under study from simple substances

Calculations of $\Delta H^\circ_{f(g)}$ by formulae I and II for each of the **1a–c** and **2a–c** compounds result in close values when the most high-level combined quantum-chemical G4 method is used (difference is no more than 1–6 kJ/mol). In the case when other methods are used to calculate the enthalpy of formation, difference between values of $\Delta H^\circ_{f(g)}$ (I) and $\Delta H^\circ_{f(g)}$ (II) slightly increases: 4–19 kJ/mol (0–2%) for G4MP2 method, 21–81 kJ/mol (3–6%) for ω B97XD/aug-cc-pVTZ, 3–27 kJ/mol (0–2%) for B3LYP/6-311+G(2d,p), 96–125 kJ/mol (9–11%) for M062X/6-311+G(2d,p) and 111–138 kJ/mol (11–16%) for CBS-4M. It should be noted that values of $\Delta H^\circ_{f(g)}$ (I) are higher than those of $\Delta H^\circ_{f(g)}$ (II) (see Tab. 1), with the only exception for corresponding values of the **1c** and **2c** structures calculated by the G4 method. It is known that the calculated values of the enthalpy of formation, as a rule, are higher than the experimental data, and therefore it can be assumed that quantum-chemical calculations using reaction II might be more accurate than calculations using reaction I. From the physical point of view, this seems quite logical, since the structure of the wave function in the case of the second reaction is divided into a smaller number of fragments.

In the case of calculations by the ω B97XD/aug-cc-pVTZ, G4MP2, M062X/6-311+G(2d,p), B3LYP/6-311+G(2d,p) and CBS-4M methods, the values of the enthalpy of formation $\Delta H^\circ_{f(g)}$ (II) differ from the ones obtained using the referential G4 method by 1–2%, 2–5%, 1–10%, 4–18% and 6–17%, respectively. Thus, it can be preliminarily concluded that the methods that give the closest results to the ones obtained by the G4 methods for **1a–c** and **2a–c** compounds are ω B97XD/aug-cc-pVTZ and G4MP2. At the same time, the time required to complete G4 calculations is much longer than the calculation time using the ω B97XD/aug-cc-pVTZ and G4MP2 methods. Therefore, the latter two methods can be recommended for calculating $\Delta H^\circ_{f(g)}$ of structures of the type under study.

As it can be seen in Tab. 1 and Fig. 3, $\Delta H^\circ_{f(g)}$ values for the pairs of isomers **1a** and **1b**, **2a** and **2b**, with the same central cycle, are almost equal (the difference is within 1 kcal/mol), which is quite understandable, since the structures of the mentioned isomers contain an equal number of $C-NO_2$ groups, as well as $N-N$, $N=N$, $C-N$, $C=N$, $C-C$, and $C=C$ bonds. This is due to the fact that the structures of these isomeric pairs contain the same six-membered ring isomer (1,2,3,4-tetrazine for **1a** and **1b**, 1,2,4,5-tetrazine for **2a** and **2b**), and the imidazole rings annelated with them are not isomeric.

At the same time, comparison of isomers pairs **1a** and **2a**, **1b** and **2b** shows that the values of $\Delta H^\circ_{f(g)}$ for them slightly differ, namely, by ~ 45 kJ/mol, and for a pair of **1c** and **2c**, by ~ 35 kJ/mol, and this difference is obviously caused by the type of the central cycle, on which the structure is based. Isomers **1a–1c** based on 1,2,3,4-tetrazine contain a $C-C$ bond, which greatly reduces the enthalpy of formation, while there is no such bond in similar isomers **2a–2c** based on symmetrical 1,2,4,5-tetrazine. Therefore, as can be seen from the results of calculations (see Tab. 1), $\Delta H^\circ_{f(g)}$ of isomer **2a** is greater than that of **1a**, respectively, $\Delta H^\circ_{f(g)}$ of structure **2b** is greater than that of **1b**, and $\Delta H^\circ_{f(g)}$ of **2c** is greater than that of **1c**.

Noticeable changes in the thermochemical characteristics occur when the nitro group in the imidazole rings is replaced by the azide group, i.e., in transition from structure **1b** to **1c** and from structure **2b** to **2c**, which leads to a significant increase in $\Delta H^\circ_{f(g)}$, by 541 and 584 kJ/mol, respectively. This increase in the enthalpy of formation is quite natural and is conditioned, firstly, by removal of the oxygen-containing $-NO_2$ group, which reduces $\Delta H^\circ_{f(g)}$, and, secondly, on the contrary, by introduction of the endothermic substituent $-N^- - N^+ \equiv N$, which significantly enlarges $\Delta H^\circ_{f(g)}$ [26–28].

Earlier Shreeve and coauthors [9] described the synthesis of compound **1c**, and calculated $\Delta H^\circ_{f(g)}$ for this structure by the method of isodesmic reactions at the G2 level. The result of their calculation was 1390.8 kJ/mol, which is very close to our calculation results at the G4MP2 level (1390.5 kJ/mol), but in comparison with the value of $\Delta H^\circ_{f(g)}$ calculated by the high-level G4 method (1321.2 kJ/mol), it is obviously slightly overestimated (by 70 kJ/mol, i.e., by 5%).

2.2. IR Spectra and Frequency Analysis

Since most of the compounds under study have not yet been synthesized, providing IR spectra for them could be of great help for their future identification during synthesis. We also performed a quantum chemical analysis of vibrational spectroscopy in the gas phase for structures **1a–c** and **2a–c** (Fig. 4, Tab. 2). The intense absorption bands in the region of $1627\text{--}1581\text{ cm}^{-1}$ and $1392\text{--}1356\text{ cm}^{-1}$ can be attributed to asymmetric and symmetric stretching vibrations of the NO_2 nitro groups, respectively. The most intense peaks at $\sim 2240\text{ cm}^{-1}$ in the spectra of structures **1c** and **2c** can be attributed to asymmetric stretching vibrations of N_3 azido groups, while the peaks at $\sim 1230\text{ cm}^{-1}$ can be attributed to symmetric stretching vibrations of N_3 azido groups. The characteristic absorption frequencies of functional groups determined in our calculations correspond to the typical values given in the literature [29–32].

3. Computational Details

In this work, a number of computational sources have been used for the quantum-chemical calculations. The use of the Gaussian package led to some limitations in the calculations, since the parallelization within the program is implemented inefficiently. Test calculations (optimization

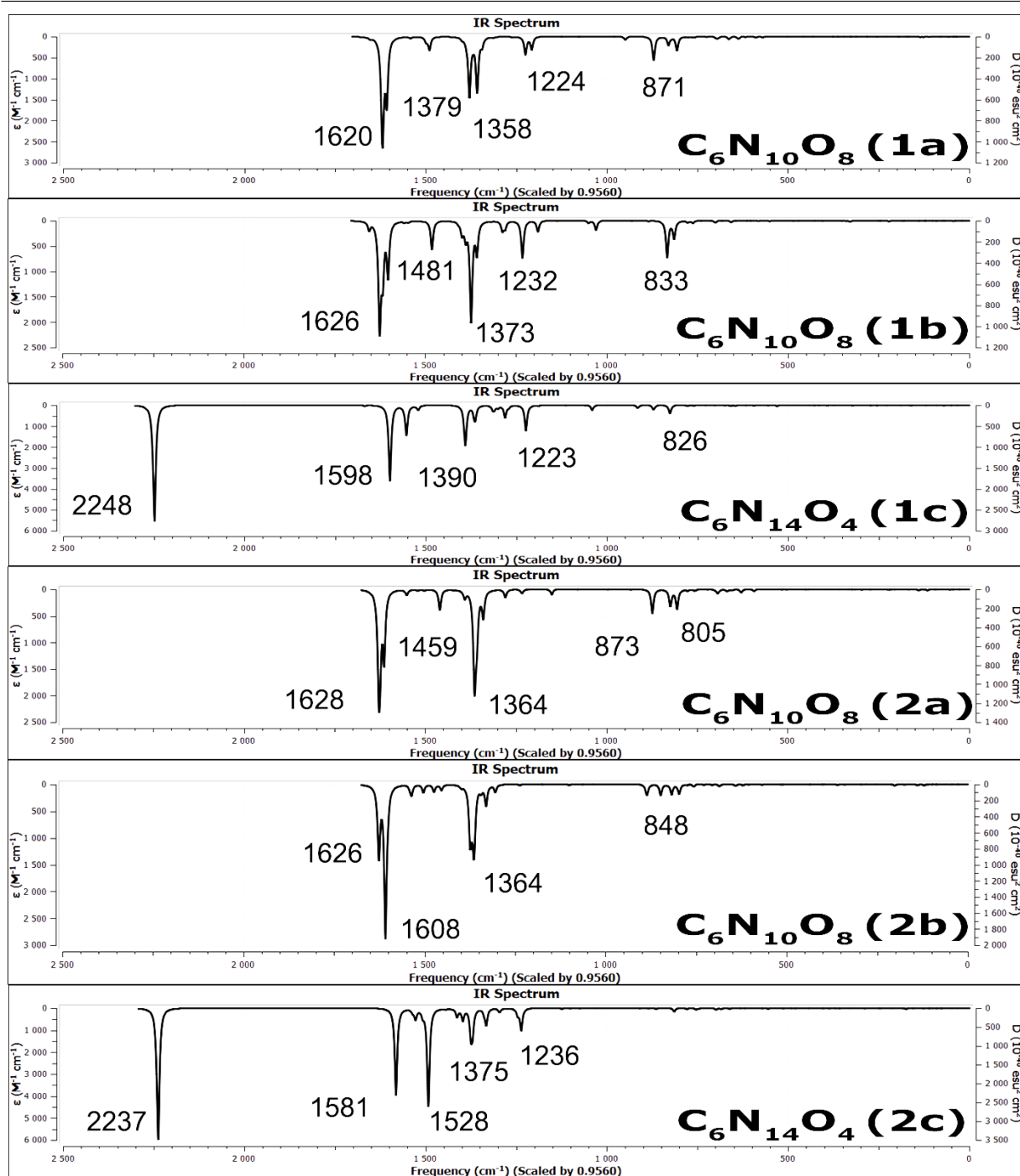


Figure 4. IR absorption spectra for structures 1a–c and 2a–c, calculated by the ω B97XD/aug-cc-pVTZ method

of a 28-atom molecule by the B3LYP method) on various resources carried out in the course of our previous work showed that a significant change in performance occurs only when increasing the number of computational cores from 1 to 8. A further increase in the number of cores does not lead to the expected rise of performance. Additional comparison of calculation by the G4MP2 method on 8 and 32 cores confirmed the previous observations and conclusions, and in this work we used 8 cores per task for each calculation method. Calculations by the B3LYP/6-311+G(2d,p) and CBS-4M methods were performed on the local computational resources of FRC PCP MC RAS. For other calculations, we used high-performance computing resources

Table 2. Allocation of the IR spectra of structures **1a–c** and **2a–c**

Compound	Frequency, cm^{-1}	Allocation of absorption frequencies in the IR spectrum
1a	1620–1607	asymm. stretch. vibrations NO_2
	1489	stretch. vibrations $C - N$ in imidazole rings
	1378–1356	symm. stretch. vibrations NO_2
	1224	asymm. stretch. vibrations $N - N$ in tetrazine ring
	1207	symm. stretch. vibrations $N - N$ in tetrazine ring
1b	1654	stretch. vibrations $C - C$ in tetrazine ring
	1625–1602	asymm. stretch. vibrations NO_2
	1481	stretch. vibrations $C - N$ and $C - C$ in imidazole rings
	1388–1357	symm. stretch. vibrations NO_2
	1231	stretch. vibrations $C - N$ and $N - N$ in tetrazine ring
	1189	asymm. stretch. vibrations $N - N$ in tetrazine ring
	1028	asymm. stretch. vibr. $N - N$ in tetrazine ring, deform. vibr. in imidazole rings
1c	2248	asymm. stretch. vibrations N_3
	1598	asymm. stretch. vibrations NO_2
	1552	asymm. stretch. vibrations $C - C$ and $C - N$
	1392	symm. stretch. vibrations NO_2
	1362	symm. stretch. vibrations $C - N$ and NO_2
	1280	symm. stretch. vibrations N_3 and deform. vibrations of imidazole rings
	1223	symm. stretch. vibrations N_3 and deform. vibrations of triazole rings
	1040	asymm. stretch. vibrations $N - N$ in tetrazine ring
	914	deform. vibrations in tetrazine ring
2a	1627–1624	asymm. stretch. vibrations NO_2
	1550	stretch. vibrations $C - C$ in imidazole rings
	1459	stretch. vibrations $C - N$ and $N - N$ in imidazole rings
	1390	stretch. vibrations $C - N$ in imidazole rings
	1364–1357	symm. vibrations NO_2
	1339	stretch. vibrations $C - N$ and $N - N$ in imidazole rings
2b	1626–1608	asymm. stretch. vibrations NO_2
	1503–1473	stretch. vibrations $C - C$ and $C - N$ in imidazole rings
	1453	stretch. vibrations $N - N$ in tetrazine ring
	1364	symm. stretch. vibrations NO_2 and $C - C$ in imidazole rings
	1362–1304	stretch. vibrations $C - N$ in tetrazine and imidazole rings
	847–796	angular vibrations NO_2 , stretch. vibrations $C - N_{NO_2}$
2c	2237	asymm. stretch. vibrations N_3
	1581	asymm. stretch. vibrations NO_2
	1527	asymm. deform. vibrations in tetrazine and imidazole rings
	1492	asymm. stretch. vibrations $N_{N_3} - C$
	1413	symm. stretch. vibrations $C - N$ and $N - N$
	1397	asymm. stretch. vibrations $C - N$ and $N_{NO_2} - C$
	1374	symm. stretch. vibrations NO_2
	1370	asymm. vibrations $C - N$
	1332	symm. stretch. vibr. NO_2 and asynch. deform. vibr. in imidazole rings
	1245	deform. vibrations of imidazole rings
1236	symm. stretch. vibrations N_3	

of the Lomonosov Moscow State University. Computation time varied from several hours for M062X/6-311+G(2d,p) method to several months by G4 method.

Conclusions

Quantum-chemical methods for calculating the enthalpy of formation of tricyclic bis(imidazo)tetrazines **1a–c** and **2a–c** in the gas phase at different calculation levels have been systematically studied, which made it possible to determine $\Delta H^\circ_{f(g)}$ for the given substances and discover its dependence on the structure of the compounds. The performed studies have shown that $\Delta H^\circ_{f(g)}$ of isomers with identical substituents and a central tetrazine ring (pairs **1a, b** and **2a, b**) are almost equal. At the same time, replacement of 1,2,3,4-tetrazine in the structure with 1,2,4,5-tetrazine, i.e., in transition from **1a, b** to **2a, b** (as well as from **1c** to **2c**) leads to increase in the calculated $\Delta H^\circ_{f(g)}$ by 8 (13) kcal/mol. Moreover, the replacement of one nitro group in each imidazole ring of the tricycle structure by an azide substituent, i.e., in transition from **1a, b** to **1c** and from **2a, b** to **2c** leads to a significant increase in $\Delta H^\circ_{f(g)}$ by 129–140 kJ/mol, as a result of which it is possible to achieve values of the enthalpy of formation up to 1400 kJ/mol (4220 kJ/kg), which is the largest value of $\Delta H^\circ_{f(g)}$ among energy-intensive bis(nitroazolo)tetrazines calculated up to date.

IR spectra have been calculated for the tricycles under study **1a–1c** and **2a–2c**, and the frequencies of the characteristic vibrations bands have been allocated to the corresponding structural fragments of the compounds under study, primarily to nitro groups (asymm. 1625–1580 cm^{-1} and symm. 1390–1350 cm^{-1}) and to azido groups (asym. 2250–2240 cm^{-1} and sym. 1225–1235 cm^{-1}).

Acknowledgements

The work was performed using the equipment of the Center for Collective Use of Super High-Performance Computational Resources of the Lomonosov Moscow State University [33–35] (projects 2065 and 2312) and computational resources of FRC PCP MC RAS. V. M. Volokhov and E. S. Amosova performed quantum-chemical research in accordance with the State Order, state registration No. AAAA-A19-119120690042-9. Calculations by resource-intensive methods were supported by Russian Science Foundation, project No. 23-71-00005. D. B. Lempert assessed the energy potential in accordance with the State Order, state registration No. AAAA-A19-119101690058-9. T. S. Zyubina performed calculation and analysis of the IR spectra and atom displacements in accordance with the State Order, state registration No. AAAA-A19-119061890019-5. V. V. Parakhin was engaged in the formulation of a scientific problem, literature review, analysis of the results, writing and editing the article.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Zlotin, S.G., Churakov, A.M., Egorov, M.P., *et al.*: Advanced energetic materials: novel strategies and versatile applications. *Mendeleev Commun.* 31(6), 731–749 (2021). <https://doi.org/10.1016/j.mcom.2021.06.001>

[//doi.org/10.1016/j.mencom.2021.11.001](https://doi.org/10.1016/j.mencom.2021.11.001)

2. Gao, H., Zhang, Q., Shreeve, J.M.: Fused heterocycle-based energetic materials (2012–2019). *J. Mater. Chem. A* 8, 4193–4216 (2020) <https://doi.org/10.1039/c9ta12704f>
3. Wu, J.T., Xu, J., Li, W., Li, H.B.: Coplanar Fused HeterocycleBased Energetic Materials. *Propellants Explos. Pyrotech.* 45, 536–545 (2020). <https://doi.org/10.1002/prop.201900333>
4. Voronin, A.A., Fedyanin, I.V., Churakov, A.M., *et al.*: 4H-[1,2,3]Triazolo[4,5-c][1,2,5]oxadiazole 5-oxide and its salts: promising multipurpose energetic materials. *ACS Appl. Energy Mater.* 3, 9401–9407 (2020). <https://doi.org/10.1021/acsaem.0c01769>
5. Chavez, D.E., Bottaro, J.C., Petrie, M., Parrish, D.A.: Synthesis and thermal behavior of a fused, tricyclic 1,2,3,4-tetrazine ring system. *Angew. Chem. Int. Ed.* 54, 12973–12975 (2015). <https://doi.org/10.1002/ange.201506744>
6. Tang, Y., Kumar, D., Shreeve, J.M.: Balancing excellent performance and high thermal stability in a dinitropyrazole fused 1,2,3,4-tetrazine. *J. Am. Chem. Soc.* 139, 13684–13687 (2017). <https://doi.org/10.1021/jacs.7b08789>
7. Volokhov, V.M., Amosova, E.S., Volokhov, A.V., *et al.*: Quantum-chemical calculations of physicochemical properties of high enthalpy 1,2,3,4- and 1,2,4,5-tetrazines annelated with polynitroderivatives of pyrrole and pyrazole. Comparison of different calculation methods. *Comput. Theor. Chem.* 1209, 113608 (2022). <https://doi.org/10.1016/j.comptc.2022.113608>
8. Volokhov, V.M., Parakhin, V.V., Amosova, E.S., *et al.*: Quantum-chemical calculations of the enthalpy of formation of 5/6/5 tricyclic tetrazine derivatives annelated with nitrotriazoles. *Russ. J. Phys. Chem. B*, in print (2024).
9. Tang, Y., He, Ch., Yin, P., *et al.*: Energetic functionalized azido/nitro imidazole fused 1,2,3,4-tetrazine. *Eur. J. Org. Chem.* 2273–2276 (2022). <https://doi.org/10.1002/ejoc.201800347>
10. Xie, C., Pei, L., Cai, J., *et al.*: Imidazolebased energetic materials: A promising family of Nheterocyclic framework. *Chem. Asian J.* 17, e202200829 (2022).
11. Lai, Y., Liu, Y., Huang, W., *et al.*: Synthesis and characterization of pyrazole- and imidazole-derived energetic compounds featuring ortho azido/nitro groups. *FirePhysChem.* 2, 140–146 (2022). <https://doi.org/10.1016/j.fpc.2021.09.003>
12. Curtiss, L.A., Redfern, P.C., Raghavachari, K.: Gaussian-4 theory. *J. Chem. Phys.* 126, 084108 (2007). <https://doi.org/10.1063/1.2436888>
13. Nirwan, A., Ghule, V.D.: Estimation of heats of formation for nitrogen-rich cations using G3, G4, and G4 (MP2) theoretical methods. *Theor. Chem. Acc.* 137, 1–9 (2018). <https://doi.org/10.1007/s00214-018-2300-6>
14. Suntsova, M.A., Dorofeeva, O.V.: Use of G4 theory for the assessment of inaccuracies in experimental enthalpies of formation of aromatic nitro compounds. *J. Chem. Eng. Data.* 61, 313–329 (2016). <https://doi.org/10.1021/acs.jced.5b00558>

15. Glorian, J., Han, K.T., Braun, S., Baschung, B.: Heat of formation of triazole-based salts: prediction and experimental validation. *Propellants Explos. Pyrotech.* 46, 124–133 (2021). <https://doi.org/10.1002/prop.202000187>
16. Frisch, M.J., Trucks, G.W., Schlegel, H.B., *et al.*: Gaussian 09, Revision B.01. Gaussian, Inc., Wallingford CT (2010).
17. Becke, A.D.: Densityfunctional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* 98(4), 5648–5652 (1993). <https://doi.org/10.1063/1.464913>
18. Johnson, B.J., Gill, P.M.W., Pople, J.A.: The performance of a family of density functional methods. *J. Chem. Phys.* 98(4), 5612–5626 (1993). <https://doi.org/10.1063/1.464906>
19. Zhao, Y., Truhlar, D.G.: The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals. *Theor. Chem. Acc.* 120, 215–241 (2008). <https://doi.org/10.1007/s00214-007-0310-x>
20. Chai, J.-D., Head-Gordon, M.: Long-range corrected hybrid density functionals with damped atom-atom dispersion corrections. *Phys. Chem. Chem. Phys.* 10, 6615–6620 (2008). <https://doi.org/10.1039/B810189B>
21. Curtiss, L.A., Redfern, P.C., Raghavachari, K.: Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* 127, 124105 (2007). <https://doi.org/10.1063/1.2770701>
22. Curtiss, L.A., Redfern, P.C., Raghavachari, K.: Gn theory. *Comput. Mol. Sci.* 1, 810–825 (2011). <https://doi.org/10.1002/wcms.59>
23. Montgomery Jr., J.A., Frisch, M.J., Ochterski, J.W., Petersson, G.A.: A complete basis set model chemistry. VII. Use of the minimum population localization method. *J. Chem. Phys.* 112, 6532–6542 (2000). <https://doi.org/10.1063/1.481224>
24. Petersson, G.A., Malick, D.K., Wilson, W.G., *et al.*: Calibration and comparison of the Gaussian-2, complete basis set, and density functional methods for computational thermochemistry. *J. Chem. Phys.* 109, 10570–10579 (1998). <https://doi.org/10.1063/1.477794>
25. CCCBDB Vibrational Frequency Scaling Factors. <https://cccbdb.nist.gov/vsfx.asp>, accessed: 2022-11-10
26. Klapoetke, T.M., Krumm, B.: Azide-containing high energy materials in organic azides: syntheses and applications, eds. S. Brase, K. Banert. 391-100. Chichester: Wiley (2010).
27. Tang, Y., Shreeve, J.M.: Nitroxy/AzidoFunctionalized triazoles as potential energetic plasticizers. *Chem. Eur. J.* 21, 7285–7291 (2015). <https://doi.org/10.1002/chem.201500098>
28. Luk'yanov, O.A., Parakhin, V.V., Shlykova, N.I., *et al.*: Energetic N-azidomethyl derivatives of polynitro hexaazaisowurtzitanes series: the most highly enthalpy analogues of CL-20. *New J. Chem.* 44, 8357–8365 (2020). <https://doi.org/10.1039/D0NJ01453B>

29. Slovetskii, V.I.: Infrared absorption spectra of aliphatic nitro-compounds and their derivatives. *Russ. Chem. Rev.* 40, 393–405 (1971). <https://doi.org/10.1070/RC1971v040n04ABEH001925>
30. Slovetskii, V.I.: IR spectra of nitro compounds. *Bull. Acad. Sci. USSR, Div. Chem. Sci.* 19, 2086–2091 (1970). <https://doi.org/10.1007/BF00861473>
31. Socrates, G.: *Infrared and Raman Characteristic Group Frequencies: Tables and Charts*, John Wiley & Sons, Chichester, 3th edn, (2004).
32. Pretsch, E., Buhlmann, Ph., Badertscher, M.: *IR Spectroscopy, in Structure Determination of Organic Compounds. Tables of Spectral Data*, Springer-Verlag, Berlin, Heidelberg, 5th edn, pp. 307–373 (2020). https://doi.org/10.1007/978-3-662-62439-5_7
33. Voevodin, V.V., Antonov, A.S., Nikitenko, D.A., *et al.*: Supercomputer Lomonosov-2: Large scale, deep monitoring and fine analytics for the user community. *Supercomput. Front. Innov.* 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>
34. Voevodin, V., Zhumatiy, S., Sobolev, S., *et al.*: Practice of “Lomonosov” Supercomputer. *Open Syst.* 7, 36–39 (2012) (in Russian).
35. Nikitenko, D., Voevodin, V., Zhumatiy, S.: Deep analysis of job state statistics on “Lomonosov-2” supercomputer. *Supercomput. Front. Innov.* 5(2), 4–10 (2019). <https://doi.org/10.14529/jsfi180201>

MOUSE2: Molecular Ordering Utilities for Simulations, Edition 2

Mikhail K. Glagolev¹ , Anna A. Glagoleva¹ ,
Valentina V. Vasilevskaya^{1,2} 

© The Authors 2023. This paper is published with open access at SuperFri.org

The progress in spatial and temporal scales of molecular simulations attainable with modern supercomputers makes the processing of the simulation data a challenging task in itself. One of the most important applications is the simulation of living systems, which are based on polymers, as well as simulation of polymer systems in material sciences. The behavior of many polymer systems is determined by local ordering of polymer chains, which on many occasions contain helical motifs. This ordering can be hard to quantify visually and using standard tools. To overcome these problems, we have developed an original toolkit to look into orientational and especially chiral ordering in polymer systems, which can quantify the orientational ordering of polymers based on their spatial proximity as well as assess the stiffness, helical and superhelical ordering based on polymer connectivity. The proposed software is aimed at balancing flexibility and computational efficiency. The quantitative order parameters can be useful to quantify various types of self-organization observed in coarse-grained as well as all-atom particle simulations. The utilities can be tailored to meet specific user requirements.

Keywords: molecular simulation, polymers, helicity, simulation data processing, order parameters, performance optimization.

Introduction

In recent decades, particle-based computer simulations have become a vital tool to advance scientific knowledge and get valuable insights into the behavior of complex systems [18]. Simulations can be used for high-throughput screening, to check the influence of conditions that cannot be easily achieved experimentally, and are also important for validation of the models.

As most of the considered systems are chaotic [64], meaningful results cannot be obtained only from the visual analysis of the snapshots, but require considering a statistically representative ensemble of configurations [17]. Therefore, the data from particle simulations has to be quantified, i.e. reduced to a relatively small set of comprehensible parameters.

Molecular ordering and self-assembly are the focus of computer simulations due to their great importance to many areas of research, including physics, chemistry, biology and medicine, and because computer simulations are the perfect tool to explore the relationships between molecular-level phenomena and practically important behavior of the systems.

Analysis of the results of particle simulations is implemented in multiple software tools, including those with open-source code. For example, one can use the algorithms included in popular simulation engines such as LAMMPS [57] or GROMACS [1] to calculate the required quantities in the course of the simulation or process the trajectories afterwards. In the case of post-processing, the range of possible tools is wider and includes multiple toolkits and libraries. Among the latter are MDAnalysis [45], VMD [37], Travis [10], PyLAT [36], Pteros [62] and MDTraj [44]. However, in many cases, the data analysis is performed by the unpublished in-house code, which hinders the reproducibility of the results.

In this manuscript we share a set of software tools which were used to describe the behavior of various complex polymer systems [2, 19–27], mostly representing the class of so-called conformationally asymmetric polymers. The parameters calculated by our tools exploit the ability of

¹A. N. Nesmeyanov Institute for Organoelement Compounds RAS, Moscow, Russian Federation

²Chemistry Department, M. V. Lomonosov Moscow State University, Moscow, Russian Federation

computer simulations to directly access the structure of the systems. This way, one can quantify the patterns that can be observed only in computer simulations and explore the relationship between the parameters of the system, its molecular structuring and its experimentally observed properties. This can reveal underlying processes responsible for complex phenomena observed in polymer systems.

The order parameters which are calculated by the proposed tools are independent of the atomistic details of the system and capture its general behavior. To analyze the results of detailed (e.g. atomistic) simulations, one can either select specific particles to represent the spatial positions of monomer units or apply a systematic coarse-graining procedure, i.e. consider the centers of mass of the units as their positions. The latter option can be implemented through available open-source coarse-graining software, for example, the VOTCA toolkit [54], or done within the MDAnalysis framework.

The order parameters are also designed to work without prior knowledge about the ordering in the system. For example, the estimation of helical parameters based on the autocorrelation function does not depend on the orientation of the helical fragments. The local ordering parameter and lamellar ordering parameter also do not depend on the direction of ordering in the system. However, some parameters, such as the backbone correlation distance or the local ordering cutoff range, are left to the user's discretion. It is worth noting that these spatial parameters can be efficiently employed to assess the scale of the spatial correlation in the systems and can be an important tool in assessing the soundness of the results in terms of size-related artifacts [17].

The utilities are written in Python3 [58], which is an easy-to-learn and flexible interpreted programming language. To make our code convenient for a wide range of researchers, we have refactored it and moved to use the popular and actively developed MDAnalysis library [45] to process the data input, providing an interface to input the data from many simulation packages, including LAMMPS, GROMACS, AMBER, HOOMD, Tinker, as well as data in the standard PDB format. The format of the data is recognized automatically by MDAnalysis based on filename extensions. The MDAnalysis also provides intrinsic support for processing of the time series. For the analysis of aggregation, we use the NetworkX library [32]. The input parameters are processed by the Argparse library. The output is provided in standard Python dictionary structures with a short description and results for all of the timesteps. It can be printed out in JSON [52] data format, to be assessed directly or processed further. In some of the utilities, Matplotlib [38] is used to plot the results.

Computing local structural properties is challenging because they scale as $O(N^2)$, where the number of particles N can be as large as 10^5 – 10^6 [4]. While some of the calculation-intensive algorithms could be reasonably implemented only using NumPy [34], we tried to optimize the code and use NumPy-based vectorisation wherever possible. For example, to calculate the list of neighbors for a selected atom, the distances between the atom and all of the other atoms are calculated, with the values exceeding the cutoff masked using the `numpy.ma` module. The indices of the atoms that do not satisfy the neighbor criteria are then filtered out using the `numpy.ma.compress` function. Similar approach, implying vectorized computations with masking out the irrelevant values was used to calculate the bond autocorrelation function as well as the superhelical twist parameters.

In the following sections, we describe the individual algorithms and their applicability to structural analysis, and in the end, we describe the performance optimization techniques employed.

1. Algorithms

1.1. Backbone Bonds Autocorrelation and Helicity

Helix is one of the main ways of polymer packing which is widespread in synthetic and biological macromolecules [15, 39, 48]. The helical packing combines the connectivity of the monomer units in the macromolecule with a high density [48]. The helical symmetry can be determined by optimization of the contacts between the polymer chains [11, 39], by steric limitations on the dihedral angles between the neighboring units along the backbone [20], and can emerge spontaneously [13, 16]. Many pure synthetic polymers, such as polylactic acid, crystallize in a helical form, which determines their physical and chemical properties. In biological macromolecules, the helical secondary structure serves as a basis for self-organization and determines the structuring at the higher levels [29, 30, 35, 43, 60]. During the protein folding, the formation of the local structure, including the helical motifs, provides the entrance to the so-called “folding funnel”, immensely reducing the entropy of the macromolecule [3, 6, 7]. It was shown that local helical structure of a macromolecule can significantly affect the probability of knot formation [63].

The analysis of the secondary helical structures in the soft matter is complicated due to the local nature of the correlations. Therefore, the algorithm must be agnostic in terms of the orientation of the helical fragments, and the length of the analyzed correlations could be limited.

In [21] we have proposed the autocorrelation function of the backbone bonds to quantify the local helical ordering in the macromolecules:

$$C(k) = \frac{1}{N-k} \sum_{i=1}^{N-k} |(\mathbf{d}_i \mathbf{d}_{i+k})|, \quad (1)$$

where \mathbf{d}_i designates the normalized vector, connecting the atoms of the i -th backbone bond, and N is the length of the backbone. From the analysis of the resulting autocorrelation functions, it is possible to determine the actual number of monomer units per helix turn and to evaluate the stability of the structure quantitatively.

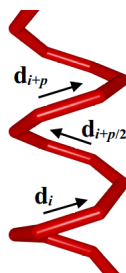


Figure 1. The bond vectors in a helix with an even number of monomer units per turn

For a perfect macromolecular helix with p monomer units per helix turn, bond length b , and pitch d , the $C(k)$ would depend on those values according to the following relationship:

$$C(k) = (1 - d^2/b^2p^2) \cos(2\pi k/p) + d^2/b^2p^2. \quad (2)$$

To evaluate the spatial parameters of the helical structure and its stability, one can fit $C(k)$ with a function in the form

$$y(x) = (A \cdot \cos(2\pi x/p) + B) \cdot e^{-\beta x} \quad (3)$$

to obtain the number of monomer units per turn and the persistence length $1/\beta$. It is worth noting that the amplitude of the oscillations and the shift correspond to the squared cosine and sine of the helix angle respectively:

$$1 - d^2/b^2p^2 = \cos^2(\alpha), \quad (4)$$

$$d^2/b^2p^2 = \sin^2(\alpha). \quad (5)$$

The decay parameter β can be used as a measure of the stability of the helical structure. Alternatively, one can use the value of $C(p)$, which corresponds to correlations between the backbone at the neighboring turns of the helix. The observed values can also be compared to the perfect crystalline form of the polymer, as was done in [28] for polylactic acid.

To use the implementation, the user can call the `bond_autocorrelations.py` tool, provide the input data file, the maximum value of k , and specify whether the correlations between the bonds connecting the beads with different residue IDs shall be taken into account. With the `--plot` option the graph is shown, and with the `--fit` option, the curve is fitted with the SciPy [59] `curve_fit` function, and the values of p , B and β from (3) are calculated.

1.2. Local Bond Orientations

Due to entanglement, the polymers can crystallize only partially, forming separate crystalline domains with different orientations. The degree of crystallinity together with the size of the domains and their mutual orientation can greatly affect the thermal, mechanical and barrier properties of the polymers.

In many biological systems, a complex local arrangement of the polymer chains is observed, when the backbones align at a certain non-zero angle to each other, forming a helix-coil structure or a helical multiplet. Such an arrangement can be caused by steric reasons [55], solvophobic [12] or electrostatic [40] interactions. Through these mechanisms, which make the basis of the so-called ‘‘chirality transfer’’, the local helicity can cause the twisting of the polymer filaments [50, 61], which, in its turn, can limit the growth of the bundles composed of such filaments [30, 61].

Therefore, studying the ordering effects at the molecular level can provide valuable insights. To establish the validity of computer simulations [5, 17] it is important to compare the characteristic scale of the ordering with the size of the simulation cell.

To evaluate the local ordering, we have implemented the algorithm to calculate the angles χ between the bonds, when the distance between the midpoints of the bonds lies in the interval between the user-defined minimum r_{min} and maximum r_{max} distances.

To avoid the complexity, we suggest using the bonds in the molecular dataset as the vectors. In the case of polymers, these can be the vectors connecting the positions of 1-st or higher-order neighbors along the polymer chain. To analyze the macromolecules which have a helical structure, one can determine the vectors as the ones connecting the i -th and $i + p$ -th monomer units, where p is the number of monomer units per helix turn, and p can be determined from bond autocorrelation analysis as described above. Alternatively, the ordering of helical macromolecules can be studied using the vectors, connecting the centers of mass of each p monomer units along the polymer backbone.

The values can then be used to calculate the 2-nd Legendre polynomial:

$$S = \frac{3 \langle \cos^2 \chi \rangle - 1}{2} \quad (6)$$

which can serve as the order parameter. To calculate the spatial scale of ordering, one can check the dependence of the value of the parameter on the cutoff radius. The minimum reasonable value of the cutoff radius is determined by the density of the system and can be of the order of the first coordination sphere. To avoid the finite-size effects, the maximum value must not exceed half the size of the simulation cell. The values of the angles between the neighboring bonds and helical segments were used in [23] to investigate the mechanically induced ordering of polylactic acid, and in [2] to study the structuring of polymer globules.

To use the implementation, the user can call the `local_alignment.py` tool, and provide the input data files, the minimum (`--r_min`) and maximum (`--r_max`) distance between the midpoints of the vectors which will be taken into account. With the `--histogram` option, the distributions of the parameters in terms of $\cos^2\chi$, $\cos\chi$ and χ , will be calculated. The distributions for $\cos^2\chi$, normalized by the total area and by the solid angle (so that the distribution is uniform for the isotropic case) can be plotted by using the `--plot` option. Studying the distributions of χ can be insightful for many biological and biomimetic systems, where the neighboring helical segments prefer to align at a certain angle to each other [55].

1.3. Orientation of Copolymer Blocks in Lamellae

The procedure described in this section is applied to the systems of diblock copolymers with a stiff block which undergoes microphase separation into the lamellar phase. Microphase separation is an important phenomenon which occurs in melts and concentrated solutions of block copolymers with incompatible blocks and results in the formation of micro- and nano-domains of various shapes [8, 33, 42]. This phenomenon finds its practical application in a wide range of nanostructured materials where the domains are endowed with predefined properties: those can be electronic or ionic conductivity, optical properties, etc. [41, 53].

Block copolymers containing a stiff block and a flexible block demonstrate the properties of both flexible coil-coil diblock copolymers and liquid crystals. The self-assembly of such block copolymers results in novel unique morphologies distinct from the classic morphologies formed by coil-coil diblock copolymers [14, 47] and can include various forms of lamellae. In particular, it was found that for block copolymers with stiff blocks the latter can be smectically or nematically ordered [46]. However, the orientation of the blocks is meaningful in relation to the lamellar plane. To calculate the orientational order parameters of the stiff blocks relative to the normal vector of the lamellae plane, we have implemented the algorithm described in [49] and used it to study the ordering of helix-coil block-copolymers [19].

First, it should be taken into account that the lamellae can take random orientation in the simulation cell, so the procedure of determining the vector normal to the lamellar planes is applied. For each polymer chain, the vectors connecting the centers of mass of incompatible blocks are calculated, normalized to have a unit length and placed to the origin of coordinates. The resulting unit vectors are denoted as $\mathbf{e}_i = (e_{ix}, e_{iy}, e_{iz}) (|\mathbf{e}_i| = 1)$. We find the gyration tensor of those vectors as follows:

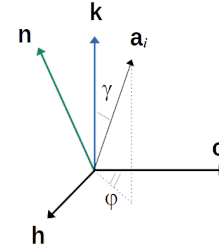
$$I_{\alpha\beta} = \frac{1}{n_{ch} * N_{stiff}} \sum_{i=1}^{n_{ch} * N_{stiff}} e_{i\alpha} e_{i\beta}. \quad (7)$$

In this equation, n_{ch} is the number of polymer chains in the system, N_{stiff} is the number of units in the stiff block. The eigenvalues and corresponding eigenvectors are calculated for this

tensor. The lamellar normal vector \mathbf{k} is denoted as the eigenvector which corresponds to the largest eigenvalue.



(a) Schematic representation of several molecules of the helix-coil diblock copolymer within the lamella, obtained in [19]



(b) Coordinate system related to lamellar domains as proposed in [49]

Figure 2. Schematic representation of the lamellar domain and the corresponding coordinate system. Blue arrows are the vectors connecting the centers of mass of the coil and helix block (in Fig. 2a) and the derived lamellar normal (in Fig. 2b). Green arrows are the vectors connecting the ends of the helix block (in Fig. 2a) and the director of these blocks (in Fig. 2b)

The orientational order parameter L_k is introduced as the average orientation of the stiff blocks with respect to the lamellar normal:

$$L_k = \left\langle \frac{3}{2} (\mathbf{a} \cdot \mathbf{k})^2 - \frac{1}{2} \right\rangle. \quad (8)$$

Here, \mathbf{a} is the unit vector showing the direction of each stiff block, i.e. the vector between the stiff block ends, and the averaging is performed over all the polymer chains.

The modulus of L_k tends to zero if the orientations of the stiff blocks are not correlated and is close to unity when the stiff blocks orient perpendicularly to the lamellae surface.

We also introduce the director \mathbf{n} of all the stiff blocks. It is defined via the same procedure as \mathbf{k} , using the vectors connecting the ends of the stiff blocks \mathbf{a}_i normalized to the unity length ($|\mathbf{a}_i| = 1$) instead of \mathbf{e}_i in (7). The plane comprising the vectors \mathbf{n} and \mathbf{k} is denoted as a tilt plane. We calculate the tilt angle θ of the stiff blocks using three order parameters:

$$\tan 2\theta = \frac{V}{L_k - 0.5P_k}, \quad (9)$$

$$P_k = \langle \sin^2 \gamma \cos 2\varphi \rangle, \quad (10)$$

$$V = \langle \sin 2\gamma \cos \varphi \rangle. \quad (11)$$

In (10) and (11), γ is the angle between the lamellar normal \mathbf{k} and \mathbf{a}_i ; φ is the angle between the vector \mathbf{c} , which belongs to the tilt plane and is perpendicular to the lamellar normal ($\mathbf{c} \perp \mathbf{k}$), and the projection of the vector \mathbf{a}_i on the plane formed by the vector \mathbf{c} and the normal to the tilt plane \mathbf{h} (Fig. 2 and ref. [49]).

The parameter P_k describes the biaxial distribution of the stiff blocks within the lamellae; V is a coefficient in the non-diagonal term of the tensor order parameter of the system [49] and describes the tilt of its main axis with respect to the lamellar normal. The algorithm is

implemented in `lamellar_alignment.py` tool. The user should provide an input data file and, optionally, the names of the block types.

The output of the calculation includes the order parameter L_k which is the measure of the correlation between the directions of the stiff blocks with each other and with the lamellar plane and the tilt angle θ of the stiff blocks.

1.4. Superhelical Twist

The formation of superhelical structures is common in the self-organization of biological macromolecules. The local helical structure of polymer chains can lead to a preferential alignment angle between the neighboring polymer chains, as discussed in section 1.2. In helical multiplets, this causes the twisting of the helical polymer chains around each other. The spatial dimensions of this type of structure depend on the spatial parameters of individual helices [31]. To assess the handedness of these superhelices, one can connect uniformly points of the helical tube with the vectors and calculate the dihedral angles formed by these vectors in the range $(-\pi; \pi]$. If the length of the vectors is commensurate to the period of the superhelix, the dihedral angles will have preferentially negative values, corresponding to the left-hand superhelices, or positive values, corresponding to right-hand superhelices [2]. The algorithm quantitatively supported the visual observations reported in [2], and the results were presented in the Supporting Information of [2].

For a set of the integer values of k , provided by the user, the algorithm calculates the dihedral angles between the triplets of consecutive vectors connecting the monomer units of the backbone: $(r_i, r_{i+k}), (r_{i+k}, r_{i+2k}), (r_{i+2k}, r_{i+3k})$, where r_j is the position of the j -th monomer unit. By default, the dihedral angle values are calculated using those values of i , where the start points and end points of all three vectors belong to the same molecule.

To calculate the values of the described dihedral angles, the user can call the `backbone_twist.py` utility, and provide the list of the values of k , and the criteria for backbone atom selection in terms of MDAnalysis `select_atoms` function. With the `--plot` option, the distributions for all of the values of k will be plotted using the NumPy histogram function and PyPlot.

1.5. Aggregation

The algorithm calculates the list of the aggregates at each timestep. The atom is defined as belonging to a particular aggregate if it has at least one neighboring atom belonging to the same aggregate.

At a given timestep, a list of neighbors for each atom is created. The atoms are considered neighbors if they lie within the cutoff distance from each other. The user can switch between the distance matrix calculation function of the MDAnalysis library or our algorithm based on NumPy (see the Performance Optimization section). The algorithm then loops through the atoms and for each atom loops through its neighbors, adding the pairs of the atoms and their neighbors as the edges of the NetworkX graph object [32].

The iterative deepening depth-first search algorithm is then applied to the graph [56]. The atom indices returned by this search are then stored as an individual aggregate, and removed from the list of all atom indices. The procedure is repeated until the list of all atom indices is empty.

The function can be called using `aggregates.py` tool with the data file and the cutoff distance as input. The output contains lists of the aggregates in a form of the lists of atoms in each aggregate. Such lists are generated for each timestep.

2. Performance Optimization

The performance of the toolkit is achieved by employing NumPy vectorization for computationally intensive operations. For example, to calculate the list of neighbors for a selected atom, the distances between the atom and all of the other atoms are calculated. The resulting squared distances are then compared to the maximum distance between the neighbors, and the mask for the `numpy.ma` module is generated. The indices of the atoms that do not satisfy the neighbor criteria are then filtered out using the `numpy.ma.compress` function.

To vectorize the calculation of the bond autocorrelation functions, two arrays of bond vector coordinates are calculated, which are shifted by the autocorrelation distance k :

```
b1 = np.pad(b, ((0, k), (0, 0)), constant_values = 1.)
b2 = np.pad(b, ((k, 0), (0, 0)), constant_values = 1.)
```

The residue identifiers, associated with the bonds, are then also padded, to check whether the result is meaningful, and the autocorrelation function is calculated for all of the bonds:

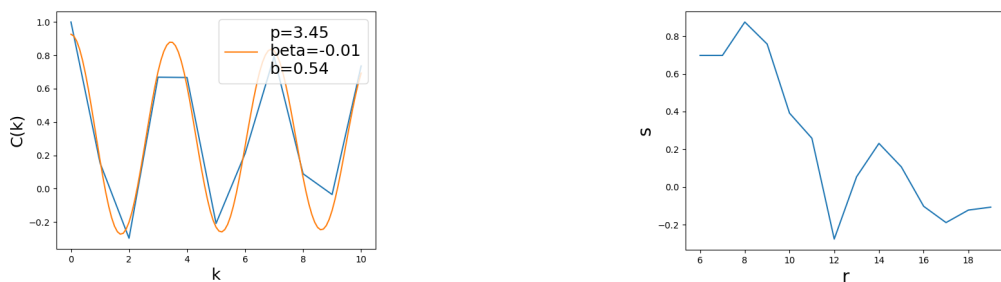
```
if not different_molecules:
    # Pad the residue id arrays
    resid1 = np.pad(bond_resids, (0, k), constant_values = 0)
    resid2 = np.pad(bond_resids, (k, 0), constant_values = 0)
    # Take into account only molecules with same residue id
    valid = np.logical_and(valid, np.equal(resid1, resid2))
# Mask is True for the values that are not valid
mask = np.logical_not(valid)
# Calculate the correlation values for all the bonds
c = (np.sum(np.multiply(b1, b2), axis = 1)
     / np.linalg.norm(b1, axis = 1)
     / np.linalg.norm(b2, axis = 1))
```

To calculate the dihedral angles associated with the superhelical twist, a similar scheme is performed first to calculate the vectors forming the dihedral angles, and then the values of the dihedral angles themselves.

3. Usage Examples

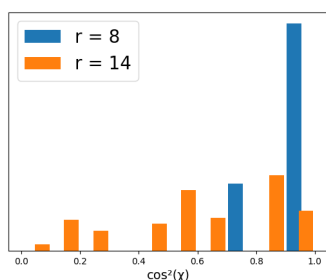
The examples directory of the project repository currently contains two Python scripts, which demonstrate using some of the functions. The systems for analysis are taken from the RSCB Protein Data Bank (PDB) [9]. We use the `bond_autocorrelations.py` tool to detect the parameters of the helices in a de novo protein with the PDB identifier 3H5G [51]. The script invokes the `bond_autocorrelations.py` command through the `os.system()` call, corresponding to the command line usage scenario. The `bond_autocorrelations.py` finds the average number of residues per helix turn as $p = 3.45$, which is close to the values for the α -helix, and the positive shift of the autocorrelation function, which depends on the elongation of the helix, as $b = 0.54$.

The very low value of the decay parameter $\beta = -0.01$ is indicative of a well-defined helical structure, which is consistent with the authors' conclusions about its stability.



(a) Autocorrelation function $C(k)$ of the helical fragments in 3H5G [51] molecules

(b) Average order parameter s of the helical fragments alignment in 6OCK [65] depending on the distance r between the midpoints of the helices



(c) Distributions of the angles between the helical fragments in 6OCK [65] at $r = 8$ and $r = 14$

Figure 3. Results for processing example proteins with helical structures from the RSCB PDB [9]

For the second protein from the PDB, with the identifier 6OCK [65], the example script invokes the `local_alignment.py` function, corresponding to the scenario where the user can build her/his own tools upon the functions provided by our package. The script calculates the average order parameter s , corresponding to the alignment of the helices, depending on the distance between their midpoints. The helices are taken according to the information provided in the original data bank record, and only those helices that are equal or longer than 10 residues are taken into account.

The distributions of the angles between the helices are then calculated for the main maximum at $r = 8$ and a small maximum at $r = 14$. The plots show that the helices located closely and belonging to the same domain are mostly aligned with each other, while at larger distances there is a wide distribution of mutual orientations of the helices, which can result in fluctuations of the average local alignment parameter s , leading to local extrema.

4. Availability and System Requirements

The tools described above can be downloaded free of charge, used and modified according to the conditions of the standard GNU GPL v.3 license. To use the software, the user needs to download it to her/his computer. Currently we provide the option to download the software from the main GitHub repository, or, alternatively to use the PyPI package. In the latter case the pip package manager automatically installs the libraries and packages upon which our software is built: MDAnalysis, NumPy, NetworkX, SciPy, Matplotlib. The instructions on how to install the software and a short user guide are available in the README.md file, and are displayed at the

GitHub page of the project. The unit tests for the algorithms are provided with the package and are located in the tests directory of the GitHub repository. We have developed, tested and used the software with the Linux platforms used by our group, and have not tested the installation and functioning of the software on other platforms. In case of the Windows operating system, the installation shall be possible using the PyPI and/or Conda package managers, and we will be happy to cooperate with the users to make the software work on other platforms.

Conclusions

Our package offers several tools for the analysis of molecular dynamics simulations and is available for download from GitHub under the GNU General Public License, and from PyPI. With the help of MDAnalysis library, it can process the output of multiple simulation software. The algorithms were optimized for performance and with the modern hardware can process the systems containing up to 10^6 particles.

Although it is impossible to implement all of the usage scenarios, the tools can be extended and the functions can be called from the Python code. We are open to cooperation and will consider implementing other usage scenarios.

We are planning to build on these utilities and add more functionality in the course of our future research.

Acknowledgements

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Contract No. 075-03-2023-642).

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Abraham, M.J., Murtola, T., Schulz, R., *et al.*: GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1-2, 19–25 (Sep 2015). <https://doi.org/10.1016/j.softx.2015.06.001>
2. Abramova, A., Glagolev, M., Vasilevskaya, V.: Structured globules with twisted arrangement of helical blocks: Computer simulation. *Polymer* 253, 124974 (Jun 2022). <https://doi.org/10.1016/j.polymer.2022.124974>
3. Adamcik, J., Mezzenga, R.: Amyloid Polymorphism in the Protein Folding and Aggregation Energy Landscape. *Angewandte Chemie International Edition* 57(28), 8370–8382 (Jul 2018). <https://doi.org/10.1002/anie.201713416>
4. Allen, M.P., Tildesley, D.J.: *Computer simulation of liquids*. Oxford University Press, Oxford, United Kingdom, second edition edn. (2017)
5. Arora, A., Morse, D.C., Bates, F.S., Dorfman, K.D.: Commensurability and finite size effects in lattice simulations of diblock copolymers. *Soft Matter* 11(24), 4862–4867 (2015). <https://doi.org/10.1039/C5SM00838G>

6. Baldwin, R.L., Rose, G.D.: Is protein folding hierarchic? II. Folding intermediates and transition states. *Trends in Biochemical Sciences* 24(2), 77–83 (Feb 1999). [https://doi.org/10.1016/S0968-0004\(98\)01345-0](https://doi.org/10.1016/S0968-0004(98)01345-0)
7. Baldwin, R.L., Rose, G.D.: Is protein folding hierarchic? II. Folding intermediates and transition states. *Trends in Biochemical Sciences* 24(2), 77–83 (Feb 1999). [https://doi.org/10.1016/S0968-0004\(98\)01345-0](https://doi.org/10.1016/S0968-0004(98)01345-0)
8. Bates, F.S., Fredrickson, G.H.: Block Copolymer Thermodynamics: Theory and Experiment. *Annual Review of Physical Chemistry* 41(1), 525–557 (Oct 1990). <https://doi.org/10.1146/annurev.pc.41.100190.002521>
9. Berman, H.M., Westbrook, J., Feng, Z., *et al.*: The Protein Data Bank. *Nucleic Acids Research* 28(1), 235–242 (01 2000). <https://doi.org/10.1093/nar/28.1.235>
10. Brehm, M., Thomas, M., Gehrke, S., Kirchner, B.: TRAVIS—A free analyzer for trajectories from molecular simulation. *The Journal of Chemical Physics* 152(16), 164105 (Apr 2020). <https://doi.org/10.1063/5.0005078>
11. Cajamarca, L., Grason, G.M.: Geometry of flexible filament cohesion: Better contact through twist? *The Journal of Chemical Physics* 141(17), 174904 (Nov 2014). <https://doi.org/10.1063/1.4900983>
12. Cejas, M.A., Kinney, W.A., Chen, C., *et al.*: Thrombogenic collagen-mimetic peptides: Self-assembly of triple helix-based fibrils driven by hydrophobic interactions. *Proceedings of the National Academy of Sciences* 105(25), 8513–8518 (Jun 2008). <https://doi.org/10.1073/pnas.0800291105>
13. Chaudhuri, D., Mulder, B.M.: Spontaneous Helicity of a Polymer with Side Loops Confined to a Cylinder. *Physical Review Letters* 108(26), 268305 (Jun 2012). <https://doi.org/10.1103/PhysRevLett.108.268305>
14. Chen, J.T., Thomas, E.L., Ober, C.K., Mao, G.p.: Self-Assembled Smectic Phases in Rod-Coil Block Copolymers. *Science* 273(5273), 343–346 (Jul 1996). <https://doi.org/10.1126/science.273.5273.343>
15. Claessens, M.M.A.E., Semmrich, C., Ramos, L., Bausch, A.R.: Helical twist controls the thickness of F-actin bundles. *Proceedings of the National Academy of Sciences* 105(26), 8819–8822 (Jul 2008). <https://doi.org/10.1073/pnas.0711149105>
16. Ermilov, V.A., Vasilevskaya, V.V., Khokhlov, A.R.: Secondary globular structure of copolymers containing amphiphilic and hydrophilic units: Computer simulation analysis. *Polymer Science Series A* 49(1), 89–96 (Jan 2007). <https://doi.org/10.1134/S0965545X07010129>
17. Frenkel, D.: Simulations: The dark side. *The European Physical Journal Plus* 128(1), 10 (Jan 2013). <https://doi.org/10.1140/epjp/i2013-13010-8>
18. Gartner, T.E., Jayaraman, A.: Modeling and Simulations of Polymers: A Roadmap. *Macromolecules* 52(3), 755–786 (Feb 2019). <https://doi.org/10.1021/acs.macromol.8b01836>

19. Glagolev, M.K., Glagoleva, A.A., Vasilevskaya, V.V.: Microphase separation in helix-coil block copolymer melts: computer simulation. *Soft Matter* 17(36), 8331–8342 (2021). <https://doi.org/10.1039/D1SM00759A>
20. Glagolev, M.K., Vasilevskaya, V.V.: Liquid-Crystalline Ordering of Filaments Formed by Bidisperse Amphiphilic Macromolecules. *Polymer Science, Series C* 60(1), 39–47 (Sep 2018). <https://doi.org/10.1134/S1811238218010046>
21. Glagolev, M.K., Vasilevskaya, V.V., Khokhlov, A.R.: Compactization of rigid-chain amphiphilic macromolecules with local helical structure. *Polymer Science Series A* 52(7), 761–774 (Jul 2010). <https://doi.org/10.1134/S0965545X10070102>
22. Glagolev, M.K., Vasilevskaya, V.V., Khokhlov, A.R.: Formation of fibrillar aggregates in concentrated solutions of rigid-chain amphiphilic macromolecules with fixed torsion and bend angles. *Polymer Science Series A* 53(8), 733–743 (Aug 2011). <https://doi.org/10.1134/S0965545X11080037>
23. Glagolev, M.K., Vasilevskaya, V.V.: Coarse-grained simulation of molecular ordering in polylactic blends under uniaxial strain. *Polymer* 190, 122232 (Mar 2020). <https://doi.org/10.1016/j.polymer.2020.122232>
24. Glagolev, M.K., Vasilevskaya, V.V., Khokhlov, A.R.: Effect of Induced Self-Organization in Mixtures of Amphiphilic Macromolecules with Different Stiffness. *Macromolecules* 48(11), 3767–3774 (Jun 2015). <https://doi.org/10.1021/acs.macromol.5b00188>
25. Glagolev, M.K., Vasilevskaya, V.V., Khokhlov, A.R.: Induced liquid-crystalline ordering in solutions of stiff and flexible amphiphilic macromolecules: Effect of mixture composition. *The Journal of Chemical Physics* 145(4), 044904 (Jul 2016). <https://doi.org/10.1063/1.4959861>
26. Glagolev, M.K., Vasilevskaya, V.V., Khokhlov, A.R.: Domains in mixtures of amphiphilic macromolecules with different stiffness of backbone. *Polymer* 125, 234–240 (Sep 2017). <https://doi.org/10.1016/j.polymer.2017.08.009>
27. Glagolev, M. K., Vasilevskaya, V.V., Khokhlov, A.R.: Self-organization of amphiphilic macromolecules with local helix structure in concentrated solutions. *The Journal of Chemical Physics* 137(8), 084901 (Aug 2012). <https://doi.org/10.1063/1.4745480>
28. Glova, A.D., Melnikova, S.D., Mercurieva, A.A., *et al.*: Grafting-Induced Structural Ordering of Lactide Chains. *Polymers* 11(12), 2056 (Dec 2019). <https://doi.org/10.3390/polym11122056>
29. Grason, G.M.: Chirality Transfer in Block Copolymer Melts: Emerging Concepts. *ACS Macro Letters* 4(5), 526–532 (May 2015). <https://doi.org/10.1021/acsmacrolett.5b00131>
30. Grason, G.M.: Chiral and achiral mechanisms of self-limiting assembly of twisted bundles. *Soft Matter* 16(4), 1102–1116 (2020). <https://doi.org/10.1039/C9SM01840A>
31. Grosberg, A.Y.: Theory of the cholesteric mesophase in a solution of chiral macromolecules. *Soviet Physics Doklady* 25, 638 (1980)

32. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring Network Structure, Dynamics, and Function using NetworkX. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) Proceedings of the 7th Python in Science Conference. pp. 11–15. Pasadena, CA, USA (2008)
33. Hajduk, D.A., Harper, P.E., Gruner, S.M., *et al.*: The Gyroid: A New Equilibrium Morphology in Weakly Segregated Diblock Copolymers. *Macromolecules* 27(15), 4063–4075 (Jul 1994). <https://doi.org/10.1021/ma00093a006>
34. Harris, C.R., Millman, K.J., van der Walt, S.J., *et al.*: Array programming with NumPy. *Nature* 585(7825), 357–362 (Sep 2020). <https://doi.org/10.1038/s41586-020-2649-2>
35. Ho, R.M., Li, M.C., Lin, S.C., *et al.*: Transfer of Chirality from Molecule to Phase in Self-Assembled Chiral Block Copolymers. *Journal of the American Chemical Society* 134(26), 10974–10986 (Jul 2012). <https://doi.org/10.1021/ja303513f>
36. Humbert, M.T., Zhang, Y., Maginn, E.J.: PyLAT: Python LAMMPS Analysis Tools. *Journal of Chemical Information and Modeling* 59(4), 1301–1305 (Apr 2019). <https://doi.org/10.1021/acs.jcim.9b00066>
37. Humphrey, W., Dalke, A., Schulten, K.: VMD: Visual molecular dynamics. *Journal of Molecular Graphics* 14(1), 33–38 (Feb 1996). [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5)
38. Hunter, J.D.: Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* 9(3), 90–95 (2007). <https://doi.org/10.1109/MCSE.2007.55>
39. Ji, X.Y., Zhao, M.Q., Wei, F., Feng, X.Q.: Spontaneous formation of double helical structure due to interfacial adhesion. *Applied Physics Letters* 100(26), 263104 (Jun 2012). <https://doi.org/10.1063/1.4731199>
40. Kornyshev, A.A., Lee, D.J., Leikin, S., Wynveen, A.: Structure and interactions of biological helices. *Reviews of Modern Physics* 79(3), 943–996 (Aug 2007). <https://doi.org/10.1103/RevModPhys.79.943>
41. Lazzari, M., Liu, G., Lecommandoux, S. (eds.): Block copolymers in nanoscience. Wiley-VCH ; John Wiley [distributor], Weinheim : Chichester (2006), oCLC: ocm69486595
42. Leibler, L.: Theory of Microphase Separation in Block Copolymers. *Macromolecules* 13(6), 1602–1617 (Nov 1980). <https://doi.org/10.1021/ma60078a047>
43. Li, M.C., Ousaka, N., Wang, H.F., *et al.*: Chirality Control and Its Memory at Microphase-Separated Interface of Self-Assembled Chiral Block Copolymers for Nanostructured Chiral Materials. *ACS Macro Letters* 6(9), 980–986 (Sep 2017). <https://doi.org/10.1021/acsmacrolett.7b00493>
44. McGibbon, R., Beauchamp, K., Harrigan, M., *et al.*: MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* 109(8), 1528–1532 (Oct 2015). <https://doi.org/10.1016/j.bpj.2015.08.015>
45. Michaud-Agrawal, N., Denning, E.J., Woolf, T.B., Beckstein, O.: MDAAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry* 32(10), 2319–2327 (Jul 2011). <https://doi.org/10.1002/jcc.21787>

46. Olsen, B., Segalman, R.: Self-assembly of rod-coil block copolymers. *Materials Science and Engineering: R: Reports* 62(2), 37–66 (Jul 2008). <https://doi.org/10.1016/j.mser.2008.04.001>
47. Olsen, B.D., Segalman, R.A.: Nonlamellar Phases in Asymmetric Rod-Coil Block Copolymers at Increased Segregation Strengths. *Macromolecules* 40(19), 6922–6929 (Sep 2007). <https://doi.org/10.1021/ma070976x>
48. Olsen, K., Bohr, J.: The generic geometry of helices and their close-packed structures. *Theoretical Chemistry Accounts* 125(3-6), 207–215 (Mar 2010). <https://doi.org/10.1007/s00214-009-0639-4>
49. Osipov, M.A., Gorkunov, M.V., Berezkin, A.V., *et al.*: Molecular theory of the tilting transition and computer simulations of the tilted lamellar phase of rod-coil diblock copolymers. *The Journal of Chemical Physics* 152(18), 184906 (May 2020). <https://doi.org/10.1063/5.0005854>
50. Paavilainen, S., Rg, T., Vattulainen, I.: Analysis of Twisting of Cellulose Nanofibrils in Atomistic Molecular Dynamics Simulations. *The Journal of Physical Chemistry B* 115(14), 3747–3755 (Apr 2011). <https://doi.org/10.1021/jp111459b>
51. Peacock, A., Stuckey, J., Pecoraro, V.: Switching the chirality of the metal environment alters the coordination mode in designed peptides. *Angewandte Chemie International Edition* 48(40), 7371–7374 (2009). <https://doi.org/10.1002/anie.200902166>
52. Pezoa, F., Reutter, J.L., Suarez, F., *et al.*: Foundations of JSON Schema. In: *Proceedings of the 25th International Conference on World Wide Web*. pp. 263–273. International World Wide Web Conferences Steering Committee, Montral, Qubec, Canada (Apr 2016). <https://doi.org/10.1145/2872427.2883029>
53. Ruokolainen, J., Mkinen, R., Torkkeli, M., *et al.*: Switching Supramolecular Polymeric Materials with Multiple Length Scales. *Science* 280(5363), 557–560 (Apr 1998). <https://doi.org/10.1126/science.280.5363.557>
54. Rhle, V., Junghans, C., Lukyanov, A., *et al.*: Versatile Object-Oriented Toolkit for Coarse-Graining Applications. *Journal of Chemical Theory and Computation* 5(12), 3211–3223 (Dec 2009). <https://doi.org/10.1021/ct900369w>
55. Straley, J.P.: Theory of piezoelectricity in nematic liquid crystals, and of the cholesteric ordering. *Physical Review A* 14(5), 1835–1841 (Nov 1976). <https://doi.org/10.1103/PhysRevA.14.1835>
56. Tarjan, R.: Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing* 1(2), 146–160 (Jun 1972). <https://doi.org/10.1137/0201010>
57. Thompson, A.P., Aktulga, H.M., Berger, R., *et al.*: LAMMPS – a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications* 271, 108171 (Feb 2022). <https://doi.org/10.1016/j.cpc.2021.108171>

58. Van Rossum, Guido, Drake, Fred L.: Python 3 Reference Manual. CreateSpace, Scotts Valley, CA
59. Virtanen, P., Gommers, R., Oliphant, T.E., *et al.*: SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17(3), 261–272 (Mar 2020). <https://doi.org/10.1038/s41592-019-0686-2>
60. Wang, H.F., Yang, K.C., Hsu, W.C., *et al.*: Generalizing the effects of chirality on block copolymer assembly. *Proceedings of the National Academy of Sciences* 116(10), 4080–4089 (Mar 2019). <https://doi.org/10.1073/pnas.1812356116>
61. Yang, Y., Meyer, R.B., Hagan, M.F.: Self-Limited Self-Assembly of Chiral Filaments. *Physical Review Letters* 104(25), 258102 (Jun 2010). <https://doi.org/10.1103/PhysRevLett.104.258102>
62. Yesylevskyy, S.O.: Pteros 2.0: Evolution of the fast parallel molecular analysis library for C++ and python. *Journal of Computational Chemistry* 36(19), 1480–1488 (Jul 2015). <https://doi.org/10.1002/jcc.23943>
63. Zhao, Y., Rothrl, J., Besenius, P., *et al.*: Can Polymer Helicity Affect Topological Chirality of Polymer Knots? *ACS Macro Letters* 12(2), 234–240 (Feb 2023). <https://doi.org/10.1021/acsmacrolett.2c00600>
64. Zhou, H.b., Wang, L.: Chaos in Biomolecular Dynamics. *The Journal of Physical Chemistry* 100(20), 8101–8105 (Jan 1996). <https://doi.org/10.1021/jp953409x>
65. Zielinski, K., Sekula, B., Bujacz, A., Szymczak, I.: Structural investigations of stereoselective profen binding by equine and leporine serum albumins. *Chirality* 32(3), 334–344 (2020). <https://doi.org/10.1002/chir.23162>

Digital Twins in Large-Scale Scientific Infrastructure Projects

Denis V. Kosyakov^{1,2}, *Mikhail A. Marchenko*²

© The Authors 2023. This paper is published with open access at SuperFri.org

The article provides an overview of publications on the topic of Digital Twins of large-scale scientific infrastructure. History, basic concepts and definition of Digital Twins are given. Main terminology in the field of big science and large-scale scientific infrastructure is also described. In Russian practice, the large-scale scientific infrastructure projects are often referred to as “megascience installations”. Such installations usually include facilities for research in areas such as astronomy and high-energy physics. The research infrastructure is a complex of construction facilities, engineering systems, precise control and measuring equipment, characterized by high complexity and strict requirements for all operational parameters. In addition, these facilities are associated with high operating costs, are sensitive to minor changes in their condition and environmental conditions, and carry the risk of data loss during long-term and unique experiments. Then, information about the use of Digital Twins in large scale astrophysical projects and also for particle accelerators control and tuning is provided. Potential areas of application of Digital Twins in large projects of scientific infrastructure are summarized. Necessary information about the Siberian Circular Photon Source (SKIF, in Russian) is given. On the basis of the review, and goals and objectives for the Digital Twin of the SKIF are determined. An analysis of the necessary computing resources and data storage volume is also carried out.

Keywords: digital twins, neural networks, supercomputing modelling, large-scale scientific infrastructure, Siberian Circular Photon Source.

Introduction

The term of “Digital Twin” (DT) was coined by Michael Grieves at the beginning of the 21st century. He originally introduced the concept during a presentation at University of Michigan in 2002, at an industry event dedicated to the creation of a Product Lifecycle Management (PLM) center. Grieves later expanded this idea into a course of lectures and in a white paper [1], as well as in a follow-up 2016 paper co-authored with John Vickers [2].

Already in 2011, the first journal article appeared on this topic [3]. It explored how digital twins can be effectively used to predict the lifespan of an aircraft structure. In 2012, NASA formalized the definition of DTs and highlighted their potential applications in the aerospace industry [4]. The period up to 2014 is considered to be the incubation stage of DT research [5]. At this stage of development, researchers began to explore the broader implications and potential applications of DTs, laying the groundwork for further advances in the field.

In his initial presentation, Grieves created the concept of “The Conceptual Ideal for PLM”. This early description already included all basic elements of a digital twin, such as real space, virtual space, flow of data from the real to the virtual space, flow of information from the virtual to the real space, and virtual subspaces. The driving principle of the model was the existence of two interconnected systems: a physical system that has always existed, and a new virtual system that contains all the information about the physical system. This connection created a mirror effect between the real and virtual spaces. The inclusion of PLM in the title emphasized that the concept was not limited to a static view, but rather encompassed the dynamic relationship between the two systems throughout the product lifecycle. This connection persists as the

¹Russian Research Institute of Economics, Politics and Law in Science and Technology, Moscow, Russian Federation

²Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russian Federation

system goes through four main phases: design, production, operation (maintenance/support) and disposal.

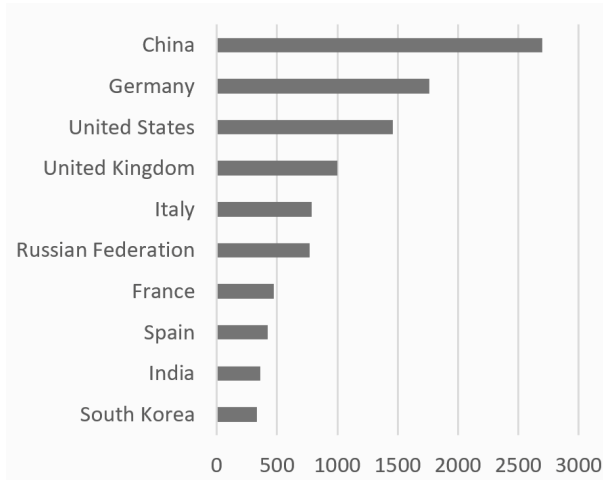
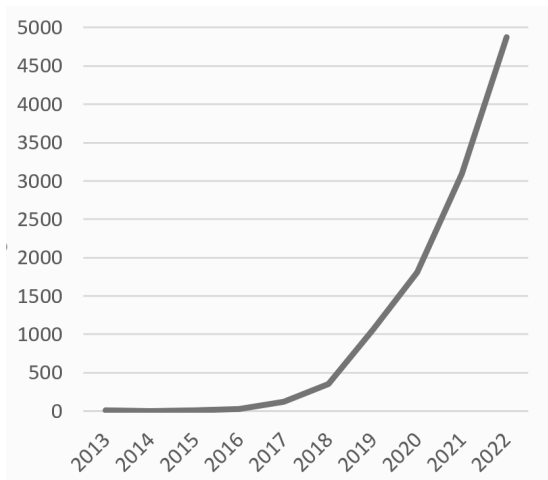
The concept was based on rapid advances in the performance and availability of computer systems, the development of computer-aided design (CAD) systems [6] and computer simulation systems [7]. In the first phase, the focus was on creating an accurate digital representation of physical objects, primarily through the use of CAD. This approach allowed to simulate the behavior of a real object using its DT at all stages of its life cycle. These simulations were based on or compared to the real characteristics of a physical object under real conditions [8]. This emphasis on accuracy and reproducibility has allowed DTs to serve as tools for studying and optimizing an object's performance throughout its lifetime.

The further evolution of digital twin technology is closely linked to advances in various related fields, including sensors and monitoring systems, the Internet of Things (IoT), industrial control systems (ICS), and the development of Industry 4.0 [9]. Advances in sensor technology and monitoring systems have made it possible to collect real-time data on physical objects that can be used to update and optimize their digital counterparts. The development of the concept of the Internet of Things [10], especially in the context of industrial production [11], has facilitated the integration of DTs into connected ecosystems, where different devices and systems can communicate and exchange information with each other. Advances in ICS have contributed to more efficient and automated production management, which has further facilitated the adoption of DTs in manufacturing processes. The concept of Industry 4.0, with its emphasis on cyber-physical systems [12], highlighted the importance of DTs as a means of bridging the gap between the physical and virtual worlds. In addition, significant advances in data collection, storage, and processing technologies have played an important role in processing large amounts of data generated by DTs [13].

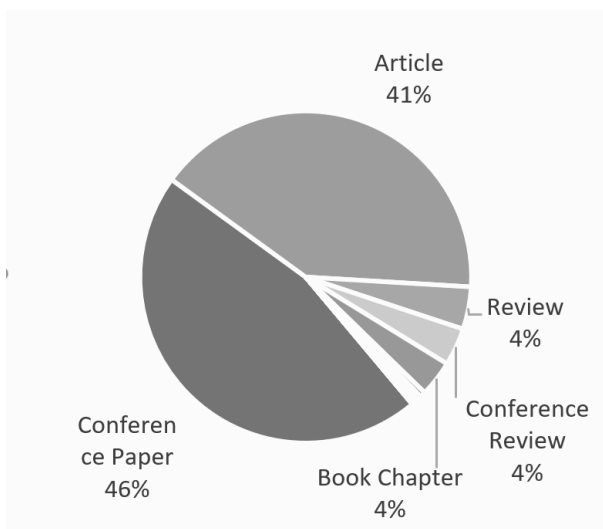
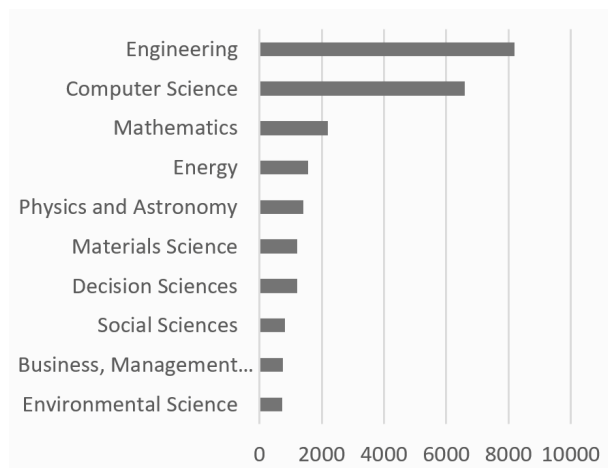
Recent advances in the field of artificial intelligence, namely in neural network technologies, have given another impulse to the development of the DT concept [14, 15]. Neural networks make it possible to build computer simulations of complex physical systems on the basis of training on a stream of real data according to the "black box" principle, which is especially useful if it is impossible or highly complex to use more traditional numerical simulations [16]. Thus, artificial intelligence technologies are one of the main components influencing the development of the DT concept [17].

In the graph showing the dynamics of the number of publications according to the Scopus database data (Fig. 1a), you can see several inflection points, a noticeable acceleration occurred in 2018, an explosive growth in the number of publications followed after 2020. Currently, this aspect is on the rise, coming out on top among other technologies. More than 17% of DT-related publications also contain one of the AI-related terms in keywords. Both leading developed countries and new industrial leaders, such as China, India, and South Korea, are interested in the topic – number of publications for the entire period is given on Fig. 1b.

Significant research interest in DT is noted in the field of product lifecycle management [18], industrial production [5, 19]. Kritzinger and colleagues note that at the time of writing the review, a significant number of studies were conceptual in nature with little addition to analysis or planning of real use cases. Particular attention is paid to the use of DT at the operation stage [20]. Applications of data centers in Smart City systems are developing [21, 22]. As expected, there is a convergence of concepts and technologies for Building Information Modeling (BIM) and DT [23–25] (Fig. 1c).



(a) Number of publications about DT summarized by all categories (b) Contribution of the top 10 countries to the publication flow about DT



(c) Number of publications about DT in the top 10 thematic categories (d) Distribution of the number of publications about DT by document type

Figure 1. Analysis of the topic of DTs based on publications indexed in Scopus

The novelty of the topic and the activity of researchers in this field, the high rates of scientific communication form a demand for holding thematic conferences and publishing relevant materials, which is why a significant part of publications on the topic is published in conference proceedings (46%), which is significantly higher than the average level (14.5%) (Fig. 1d). The share of reviews (4%) is slightly below the average for all topics (5.7%), which, probably due to relative novelty.

The subject of this review is the application of DT concepts and technologies in large-scale scientific infrastructure projects, which are often referred to as “megascience installations” in Russian practice. Such installations usually include facilities for research in areas such as astronomy and high-energy physics. The research infrastructure is a complex of construction facilities, engineering systems, precise control and measuring equipment, characterized by high complexity and strict requirements for all operational parameters. In addition, these facilities are associated

with high operating costs, are sensitive to minor changes in their condition and environmental conditions, and carry the risk of data loss during long-term and unique experiments.

Given the rapidly expanding application of DTs in the industries listed above, it can be assumed that these concepts and technologies should be widely used in the design, monitoring, control, management, and maintenance of large-scale scientific infrastructure. Integrating DTs into such facilities has the potential to improve the efficiency and reliability of these infrastructure while reducing operating costs and minimizing the risks associated with data loss and environmental factors.

The article is organized as follows. Section 1 is devoted to history, basic concepts and definition of Digital Twins. In Section 2 we describe main terminology in the field of big science and large-scale scientific infrastructure. Section 3 contains information about the use of Digital Twins in large scale astrophysical projects. In Section 4 we provide information on application of Digital Twins for particle accelerators control and tuning. Section 5 gives brief information about the Siberian Circular Photon Source (SKIF, in Russian). In Section 6 one can find goals and objectives for Digital Twin of the SKIF. Section 7 provides analysis of necessary computing resources and data storage volume for the Digital Twins of the SKIF. Conclusion summarizes the potential areas of application of Digital Twins in large projects of scientific infrastructure.

1. Basic Concepts and Definitions

According to the Russian State Standard [53], a digital twin is a system comprising a digital model of a product and bidirectional information connections with the product or its components. Let us investigate the previous publications on this topic that by many ways have made technological basis for the Standard.

Grieves and Vickers gave the following definition of DT: “The DT is a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level” [2]. As the main focus of this concept was on production and the product life cycle, they also introduced the concepts of prototype DT and instance DT with corresponding requirements. There are many attempts to supplement or expand this definition [5, 9, 16, 18–20]. The authors note that due to the multiplicity of concepts and solutions in different application areas, as well as the substitution or inclusion of other concepts in the DT concept, there is a significant diversity in the understanding and interpretation of this concept. In the definition given by Glaessgen and Stragel in 2012 [4], a DT consists of three main elements: a physical product, a virtual product, and flows data linking physical and virtual products.

According to the monograph [59], DT is a set of approaches and solutions designed to solve the problem that the increasing complexity of modern systems, as well as multi-component and multi-functional products, is outpacing the growth in the capabilities of tools for their design, manufacture and safe maintenance. And the solution to this problem lies in combining a range of digital technologies to offer more efficient means of modeling, designing, building and maintaining such complex systems. These tools must comprehensively describe the object, seamlessly integrate with each other, ensuring digital continuity of the product creation environment, work in close information connection with the subject of modeling and track all phases of its development (phases of the life cycle). The main element in this definition is integration: integration of digital technologies, integration of modeling tools based on continuous information exchange

at the data level, integration of the model and modeling object based on information exchange in near real time, integration of modeling at all stages of its life cycle.

From this definition and other interpretations, the following key attributes of DTs can be distinguished [18]:

- real-time reflection: DTs include both physical and virtual spaces, with the virtual space serving as a precisely synchronized and accurate reflection of the physical space;
- interoperability and convergence: this function can be viewed from three perspectives:
 - interaction and convergence in physical space: DTs provide comprehensive integration of all flows, elements, and services, ensuring the interconnection of data generated at different stages;
 - interoperability and convergence between log/archive and real-time data: DTs depend on a variety of data sources, including expertise and real-time information from all implemented systems, enabling deeper analysis and more efficient use of data;
 - interaction and convergence between physical and virtual spaces: in DTs, physical and virtual spaces are interconnected by seamless channels that allow for easy interaction between the two spaces;
- self-development: DTs support real-time data updates, allowing for continuous improvement of virtual models by comparing them in parallel with their physical counterparts.

Although “digital twin” is the most commonly used term, “digital model” and “digital shadow” are often used interchangeably to refer to digital representations of physical objects. This trio of terms corresponds to the evolution of the concept: “digital model” represents a virtual copy of a real physical object, “digital shadow” is its synchronous “reflection”, updating its state based on data about a real object in real time, while a full-fledged “digital twin” also provides data transfer to control systems of a real object, allowing to correct its state. Some authors use the term “digital twin” to refer to “digital model” or “digital shadow”.

As we can see, the most important element of the DT concept is the digital representation of a real or projected object in cyberspace. At the same time, if the initial concept assumed a high degree of detail and completeness of this model, later supplemented by the requirement of full-fledged behavior modeling, then there were disagreements about the information relationships between the real object and its DT almost immediately. Later, as the concept extended to a wide range of applications, the idea of the required degree of detail and completeness of the digital model also changes. Taking into account the specifics of the tasks to be solved, it is often sufficient to model only specific aspects of physical objects and systems in one way or another, including on the basis of statistical models [26] or machine learning methods [14, 27].

2. On Terminology in the Field of Big Science and Large-Scale Scientific Infrastructure

The term “megascience” was legitimized in 1992, when the OECD created the Megascience Forum [28]. This forum was then renamed the Global Science Forum. The term “megascience” is actively used in Russian practice, both in official documents and in scientific literature. Of the 127 results that Scopus provides for the request TITLE-ABS-KEY (“megascience”), 76 belong to Russian authors.

In foreign literature, the umbrella term “Big Science” has become more widespread (987 publications in Scopus), and this term refers to any large-scale projects, including those not

related to unique scientific facilities [29, 30]. In the context of large scientific facilities or other examples of research infrastructure and related research projects, the term Large Scale Research Infrastructure is used [31]. At the same time, this term is rarely used in the case of discussing specific projects of Big Science, which somewhat complicates the task of finding publications related to such projects and the use of DT technologies in them.

In this regard, the search was organized considering the main scientific areas in which research infrastructure of the corresponding class exist, are being built or designed. Such areas primarily include astronomy, particle accelerators, colliders, fusion reactors, and neutrino detectors. The search was also carried out by the names of the most well-known projects in these areas. The search results showed an extremely small number of publications about DTs in large-scale scientific infrastructure projects.

3. Digital Twins for Astronomical Complexes

DT technologies are used or planned for use in three major astrophysical projects. The Australian Square Kilometre Array Pathfinder (ASKAP) project is a radio telescope consisting of 36 parabolic antennas, each 12 metres in diameter, distributed in two dimensions with baselines up to 6 kilometres [32]. Each antenna is equipped with a monitoring system that measures parameters such as temperature, voltage, orientation and system status. DT technologies are used to create a realistic virtual representation of everything a complex that reflects the current state, allows displaying log/archive and current data on monitoring indicators. The developed DT can be used both for controlling the complex and scientific experiments, as well as for creating realistic visualizations, including for the purposes of public communication [33]. This project is part of the international Square Kilometre Array (SKA) project, the other part of which is the MeerKAT telescope [34] is based in South Africa. The use of DT technologies for the effective operation of these complexes is discussed in the report [35].

China's Five-hundred-meter Aperture Spherical radio Telescope (FAST), also known as the Tianyang (Celestial Eye), is the largest filled-aperture radio telescope in the world, located in southern China, officially commissioned in 2020 [36]. An important structural element of the FAST telescope is a flexible network of cables that supports the structure of the active reflector and allows the geometry of the active reflector to be changed. This network is equipped with more than 500 sensors for condition monitoring. Structural changes in the network associated with material fatigue are a key technical problem that determines the condition of the entire telescope. A DT of this cable network, based on data received from highly sensitive sensors and other measuring equipment, is designed to monitor and predict the state of the system [37]. The DT uses a physical model made using the ANSYS software package and provides an increase in efficiency and a reduction in the cost of supporting the operation of this complex.

The overall design of the FAST telescope includes other equally important components, respectively, there is a common task of monitoring the technical condition, supporting the operation of other units and components, as well as presenting information about the project in the media and other areas of scientific communication. In the article [38], the method of rapid prototyping used in the creation of a DT of the entire complex, which is a detailed 3D model, as well as a conceptual technical scheme for the use of the data center for the above purposes, is presented. The authors note that for the safe and efficient operation of a plant of this class, along with operational data, it is necessary to collect information about the state of the environment, such as temperature, humidity, atmospheric pressure, wind speed and direction, light

intensity, visibility, cloud cover and precipitation. This data can help optimize the operation of the complex in various weather conditions and prevent the negative consequences of dangerous natural phenomena. A prototype of the FAST's DT for the purpose of automated control of the telescope is also discussed in [39].

The extended ROentgen Survey with an Imaging Telescope Array (eROSITA) X-ray telescope project, which is the main instrument on board the Spectrum-Roentgen-Gamma (SRG) mission [40], uses DT technology in a completely different context. In [41] they describe the creation of a data center based on real data obtained during the mission. This DT is used to tune and test the algorithm for detecting clusters of galaxies and active galaxies galactic nuclei. This high-level simulation dramatically increases the understanding of real-world data and enhances the analysis capabilities of the complex arrays of signal sources observed by eROSITA.

4. Application of Digital Twins for Particle Accelerators

The booster is an integral part of the Fermi National Laboratory (Fermilab) accelerator complex in the USA and provides a flux of low-energy neutrinos for the MicroBooNE experiment [42]. The Gradient Magnet Power Supply (GMPS) is an important subsystem of this accelerator complex and is implemented in the form of four power supplies evenly distributed across the Fermilab Booster. Each feeds one of four complete gradient magnets, which are responsible for controlling and accelerating the 400 MeV proton beam that Booster receives from the linear accelerator to an energy of 8 GeV. GMPS operates in a 15 Hz cycle between injection and beam output. Currently, there is an operator in the control loop setting targets for a proportional-integral-differential control circuit that applies compensating offsets for GMPS. The goal is to create a reinforcement learning-based control circuit to optimize the control process [43]. A DT of the Booster-GMPS system was used to train, calibrate, and validate the model, created on the basis of real data collected during 6 months of operation of the complex based on a long chain of elements of short-term memory (LSTM): one of the architectures of recurrent neural networks.

The Compact Linear Collider (CLIC) is a projected accelerator that is being developed as a complement to CERN's accelerator complex. Its goal is to collide electrons and positrons head-on at energies of up to a few teraelectronvolts (TeV). In order to optimally use its physical potential, the CLIC is supposed to be built and operated in three stages at collision energies of 380 GeV, 1.5 TeV and 3 TeV, respectively, on a section with a length of 11 to 50 kilometres [44]. In the design, construction and operation of such equipment, the spatial alignment of the focusing beams of charged particles of magnetic assemblies is critical. Small deviations of the electromagnetic axes lead to significant errors in the due to the large size of the structures and the decrease in quality or failure of experiments. In the CLIC project, it is necessary to provide spatial alignment of several thousand large assemblies larger than a meter within the target combined standard uncertainty of 12 metres.

The authors in [45] found several gaps in knowledge that limit this possibility. Among them was the lack of uncertainty definitions to compensate for the thermal error applied to correct the instability of assembly dimensions, after metrology, as well as in assembly and alignment time. A new methodology was developed that used a combination of Monte Carlo modelling and high-precision traceable reference measurements to quantify the uncertainty of the various thermal expansion models used. The authors believe that this methodology can be used to create high-precision DTs with known uncertainty not only for accelerators, but also for other large

infrastructure facilities with high accuracy requirements even in the case of complex transient modes of operation. Such data centers can help in optimizing the operation of complex equipment and making important decisions.

In [46] authors discuss the prospects for the development of the bERLinPro accelerator recuperator project at Helmholtz-Zentrum Berlin (HZB), which was officially completed in 2020. This accelerator complex will get a new life as part of the Sealab (Superconducting RF Electron Accelerator Laboratory) project. It is assumed that Sealab will become a test site not only for physical experiments and development of technical solutions, but also an ideal object for research in the field of new control schemes for such complexes based on DT technologies and machine learning.

At the IPAC'23 – 14th International Particle Accelerator Conference, held in May 2023 in Venice, three reports were announced, touching on the topic of DTs in solving various problems in high-energy physics. The first report concerns the creation of a DT of the Karlsruhe Research Accelerator (KARA) with information about the energy system within the framework of the ACCESS (ACcelerator Energy System Stability) project in Energy Lab 2.0 in a simulation environment in real time. The goal of the project is to test energy solutions that can be applied to accelerators in a secure and flexible virtual environment, without interfering with the experiments conducted at KARA, while maintaining high accuracy [47].

The second report [48] discusses the creation of software tools for automatic tuning and alignment of the electron source and beam transport line based on machine learning methods and special simulation in the Test Brookhaven National Laboratory Accelerator Center. In this case, it is assumed that the model will be applied to DTs of electron beams in the Sirepo simulation environment, replicating the characteristics of real beams collected through the Bluesky system.

The third report [49] focuses on the implementation of a new product lifecycle management platform at CERN, combining log/archive data and data from different systems and external partners into a common coherent framework. One of the main goals of this platform is to create the foundation for DTs of current and future accelerators with the goal of radically reducing development time, operation and maintenance costs.

In a presentation at the ICALEPCS Data Science and ML Workshop in 2021, Orali Edelen and her colleagues presented a conceptual vision of the capabilities of artificial intelligence, machine learning, and DT technologies at the SLAC National Accelerator Laboratory (Stanford Linear Accelerator Center) in USA [50]. It notes that in the complex of the X-ray free-electron laser Linac Coherent Light Source (LCLS), about 400 hours a year are spent on setting up the equipment, for different experiments the configuration changes several times a day, the tuning cycle takes about 30 minutes. The cost of one hour of the experiment is estimated at 30 thousand US dollars, so the losses associated with long, mainly manual cycles of equipment tuning can be estimated at 12 million dollars.

At the same time, the creation of fast and accurate models that could provide automation of tuning processes is complicated by the need to use computationally complex models, many insignificant but tending to accumulate uncertainties, various kinds of fluctuations and noise, implicit dependencies, nonlinear effects and instability. In this regard, the functioning of the complex depends on the daily and constant work of operators, who ensure the performance of tasks for monitoring and correcting the operation of equipment. The authors of the report see a way out in the active use of machine learning technologies, real-time modeling, including the

creation of DTs in the tasks of anomaly detection and fault prediction, diagnostics, automated control and optimization.

5. Siberian Circular Photon Source (SKIF)

The Siberian Circular Photon Source (SKIF, in Russian), a 4+ generation synchrotron radiation source, is under development near Novosibirsk, in the Koltsovo settlement. The SKIF is comprised of an complicated accelerator complex, including a 200 MeV linear electron accelerator, a full-energy synchrotron booster, and a storage ring. This facility will feature a 3 GeV relativistic electron storage with a 476 metres perimeter and an ultra-small calculated horizontal natural emittance of 73.2 pm-rad, enabling it to generate synchrotron radiation beams of maximum brightness in the 100 eV to 100 keV range at 30 experimental stations. Particularly, for photon energies around $\sim 1\text{--}5$ keV, the emittance of the source reaches the wave diffraction limit, enhancing the spatial coherence of the Synchrotron Radiation (SR) and broadening the complex's research capabilities [51, 52].

It is known that SR, the electromagnetic radiation emitted by relativistic charged particles on curved trajectories, serves as a versatile tool for cutting-edge interdisciplinary research and technology applications in various critical economic sectors, contributing to technological sovereignty. SR sources, cyclic electron accumulators with several GeV of energy and orbital lengths from several hundred meters to kilometers, generate intense beams of charged particles with minimal phase volume emittance. These beams, moving in a transverse magnetic field, produce powerful and bright radiation, channeled to experimental stations for diverse research purposes.

Globally, dozens of SR sources are utilized for research in physics, chemistry, biology, medicine, geology, archaeology, materials science, and applied radiation applications, representing the most numerous class of ultra-relativistic energy electron beam accumulators. These facilities, operating in a collective access mode, provide infrastructure to various user organizations based on open competition outcomes. The key user characteristics of SR sources hinge prominently on the brightness and coherence of the emitted radiation. The brightness of an SR source refers to the intensity of the photon flux it produces, while coherence pertains to the uniformity and phase correlation of the radiation waves. The greater the brightness and the higher the coherent fraction of the photon flux, the superior is the “quality” of the facility. These attributes significantly enhance the facility's appeal to researchers from various external organizations. High brightness enables the study of faster processes and finer structures, while high coherence is crucial for techniques like coherent diffraction imaging and holography. Therefore, facilities that exhibit higher levels of brightness and coherence are more desirable for cutting-edge research, attracting a broader spectrum of scientific inquiries and explorations.

6. Goals and Objectives for Digital Twin of the SKIF

The DT for the SKIF aims to unify diverse mathematical and machine learning models, assimilate Big Data telemetry for identifying SKIF's structures and parameters, optimize product tuning using specific criteria, and integrate infrastructure components like software interfaces, digital test sites, and visualization tools.

The primary principles of SKIF's DT, as outlined [5, 18], include real-time reflection encompassing both physical and virtual spaces, with the virtual space accurately mirroring the

physical space. Furthermore, it supports self-improvement by updating models parameters in real time, allowing continuous enhancement of mathematical models through parallel comparison with their physical counterparts. Figure 2 illustrates the interaction between the DT, SKIF, the automated control system (ACS) and the human operator.

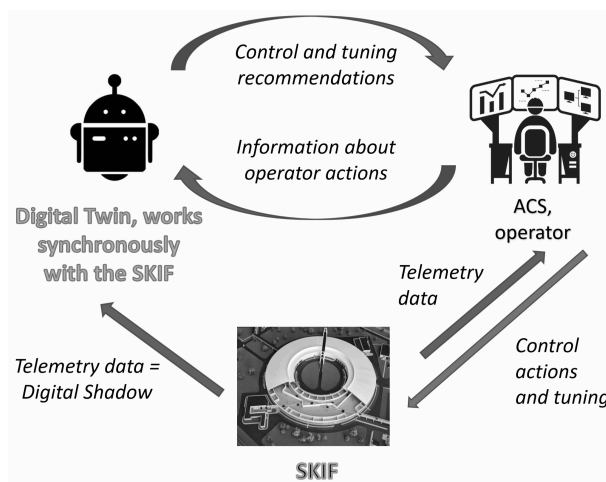


Figure 2. Scheme of interaction between the DT, SKIF, ACS and human operator

A dedicated computing environment for its DT is necessary, consisting of a computing platform for mathematical models and data integration, HPC servers and data storage. The computing platform capabilities are as follows: collecting and assimilating telemetry data, real-time control and tuning of the SKIF, making virtual experiments simulating critical operating modes, storing log/archive data archives, making experiment and data management, providing information security, making public demonstrations and users training.

The platform contains of a set of detailed mathematical models for precise virtual experiments and parametric analysis of phenomena. Purpose of the models: simulation of heat transfer, electromagnetism, radiation-sample-detector interactions, thermomechanical stresses and deformations, and stability and seismic resistance, etc. To make computations using detailed mathematical models, a HPC cluster is needed.

The set of models is central approach to conduct virtual experiments allowing for the simulation of various scenarios in a controlled virtual environment. This aspect is especially beneficial for tests that are impractical or risky to perform physically. Predictive analytics form a crucial part of these models, encompassing performance forecasting and anomaly prediction. These analytics aid in preemptively identifying potential operational challenges or deviations, facilitating timely maintenance and upgrades. This foresight is instrumental in scheduling equipment modernization and repairs, ensuring the facility's state-of-the-art status and optimal functionality.

The planning of experiments is also refined through these models. By simulating different parameters and conditions, they aid in optimizing experimental procedures, leading to more effective and efficient research outcomes. Additionally, the models play a pivotal role in tuning experimental data, a process vital for the calibration of instruments and validation of experimental results, ensuring their accuracy and reliability.

The platform includes Deep Machine Learning models and surrogate methods for fast computations. The surrogate solver, a blend of a Deep Machine Learning solver and a set of rapid mathematical models, will operate online in real physical time, using detailed mathematical

models results as synthetic datasets. It has a property of self-teaching through telemetry data and synthetic dataset assimilation. This solver will work primarily for SKIF tuning and control.

Data assimilation is a critical process, integrating and analyzing information through detailed mathematical models that run offline on a supercomputer cluster. Figure 3 presents the data assimilation scheme in the SKIF’s DT.

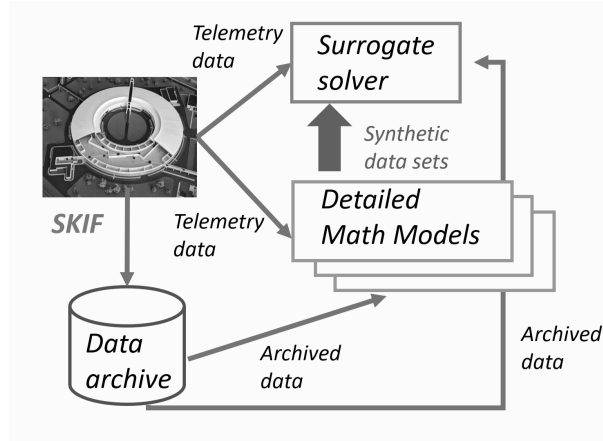


Figure 3. Scheme of data assimilation in SKIF’s DT

The platform also comprises a 3D Building Information Model (BIM) of the SKIF, SKIF’s documentation and research method descriptions.

The control scheme of the SKIF, as shown in Fig. 4, demonstrates how the DT technology integrates with the facility’s operational management. This integration facilitates real-time monitoring and adjustment of the SKIF’s parameters, ensuring its performance is maintained at an optimal level. The DT acts as an intellectual decision-support tool, enhancing the precision and efficiency of the facility’s operations, and allowing for swift responses to operational changes or anomalies. This holistic approach to managing the SKIF underscores the DT’s role in elevating the facility’s operational capabilities.

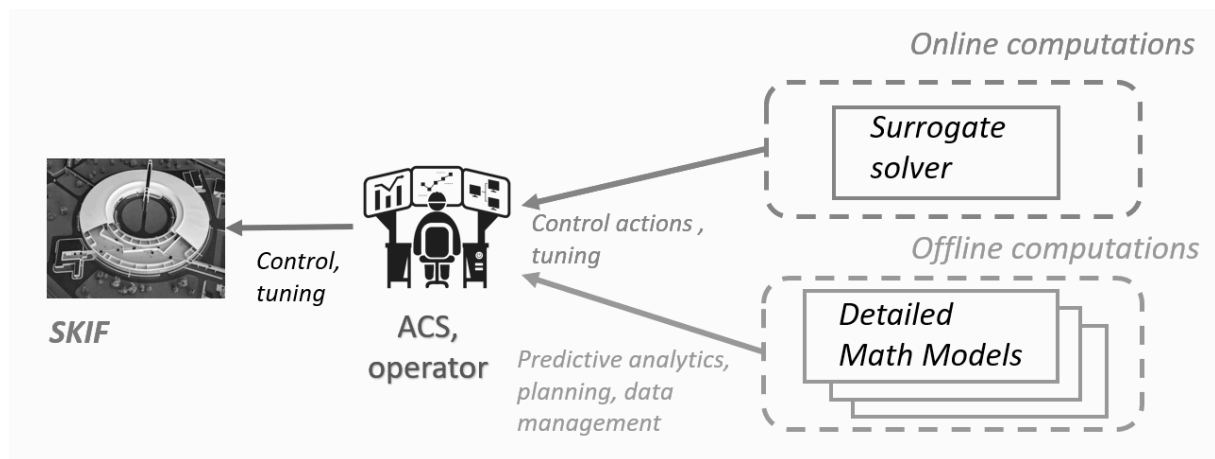


Figure 4. Scheme of controlling the SKIF by the DT

7. Analysis of Necessary Computing Resources and Data Storage Volume

Key computing tasks for SKIF encompass on-the-fly data processing through pipelines for sampling and optimization of data acquisition parameters, deciphering crystal structures from X-ray diffraction data, addressing the inverse problem of spectroscopy, and engaging in pattern recognition, classification, and image processing [51, 52]. The tasks also include applying deep learning neural network technologies, conducting mathematical modeling in quantum mechanics and engineering calculations, developing customized algorithms and software systems for the local community, and employing DT technologies for precise control and tuning.

The solution of the tasks assigned to the SKIF research team will be impossible without the utmost level of informatization and intellectualization of the processes of obtaining new scientific knowledge at this facility with the widespread use of AI technologies in data management chains in order to quickly obtain the final results of the study.

One of the key aspects that ensure high economic efficiency and scientific effectiveness of Large-Scale Scientific Infrastructure is a correctly selected and implemented data flow management policy based on modern information technologies. According to statistics, it is the Large-Scale Scientific Infrastructure that are among the most demanding users of HPC resources and structured storage of large amounts of information. Examples of such infrastructure are nuclear-physical facilities such as the Large Hadron Collider (LHC) or the International Thermonuclear Reactor (ITER), which is currently under construction. This should come as no surprise: Large-Scale Scientific Infrastructure generates Big Data.

The importance of this problem is becoming more and more apparent, which is reflected in the regular appearance of specialized articles, reviews, and entire thematic journal issues. For example, in a recently published analytical review [55], prepared by the IT managers of the four leading US synchrotron centers (APS, NSLS II, ALS and SLAC), forecasts the pace of development of the needs of the US national network of Large-Scale Scientific Infrastructure over the next few years. According to this forecast, by 2028, U.S. synchrotron sources will exceed the exabyte threshold (quintillion, or 10^{18} bytes) in terms of the amount of experimental data generated per year, and the requirements for peak performance of local computing systems will approach, respectively, exascale (quintillion floating-point operations per second).

According to the authors of another analytical review [56, 57], synchrotron crystallography and, in particular, macromolecular crystallography for biomedicine and pharmaceuticals, as well as computational tomography and other X-ray imaging techniques will be the key scientific areas shaping trends for ultra-high-performance IT infrastructure. An extremely resource-intensive experimental method is the recently developing resolving serial crystallography – microsecond for 4th generation synchrotron radiation sources and nanosecond for X-ray free electron lasers [58].

The IT infrastructure of the SKIF should include at least two layers. First layer combines automation and computing resources localized at experimental stations directly connected to information generators e.g., a high-speed two-dimensional detectors during a synchrotron experiments. The elements of the automation complex distributed among the experimental stations of the synchrotron radiation are entrusted with the role of experiment management, data collection in accordance with the experiment configuration, and transmission of primary “raw” experimental data to the DPC for subsequent processing and storage.

Second layer combines centralized computing facilities: resources hosted in the Data Processing Center (DPC) of the experimental stations. These resources are entrusted with the func-

tion of numerical mathematical processing of data obtained during the synchrotron experiment, placing these data, as well as the results of their processing in file storage, as well as providing access to them to the participants of the research collaboration.

The core switch of the SKIF's DPC should provide for the presence of 100–200 Gigabit ports to maintain high-speed communication with specialized or external computing resources, such as the existing Siberian Supercomputer Center of the Siberian Branch of the RAS at the Institute of Computational Mathematics and Mathematical Geophysics SB RAS or the promising Lavrentiev Supercomputer Center in Novosibirsk Akademgorodok.

The detailed status of the SKIF project at the current stage allows us to carry out initial assessments of the needs of experimental stations in data center computing resources and data storage volume:

- the rate of “hot” experimental data from one critical detector is 100–200 Gigabits per second;
- volumes of “hot” detector experimental data for short-term storage 200–240 Terabytes per day;
- the total bandwidth of the “hot” detector data storage system is 30–40 Gigabytes per second;
- the speed of experimental data from the “slow” detector equipment of synchrotron radiation stations is 10–20 Gigabits per second;
- the total peak rate of “slow” data from all six experimental stations is 200–250 Gigabits per second;
- the volume of “slow” experimental data for all six stations is 20–30 Terabytes per day;
- the volume of long-term (year) “warm” storage of experimental data is 4–8 Petabytes per year;
- universal multicore CPUs 275–350 Teraflops with double precision FP64;
- GPU accelerators 750–850 Teraflops with double precision FP64.

Besides, all experimental data, as well as telemetry data, must be stored eternally in accordance with the FAIR principle [54].

In the above values, a portion of the computing resources is allocated to operate the SKIF's DT. According to preliminary estimates of the SKIF's information complexity and based on the experience of developing DTs at Peter the Great St. Petersburg Polytechnic University by the team of A.I. Borovkov (see, for example, [59]), to carry out offline computations using a DT requires about 100 Teraflops of computing power on universal multi-core CPUs. To conduct online simulations primarily using neural network models, the digital twin requires about 10–30 Teraflops on GPU accelerators. At the same time, to implement a continuous mode of additional training of neural network models, about 100 Teraflops are required on GPU accelerators.

Conclusion

The review showed a slight but growing interest in the concept of DTs in the field of large-scale scientific infrastructure. There is a reason to believe that, at least in a number of cases, digital models with different bases are created in these tasks, connected by data flows with physical objects, which is quite consistent with the concept of a DT, but the appropriate terminology is not used. There is a growing interest in using machine learning technologies to create such digital models in these areas. For example, a query like TITLE-ABS-KEY (“particle accelerator*” AND (“neural network*” OR “machine learning”)) yields 415 results in Scopus,

most of which have been in recent years. It can be assumed that as the concept of DTs is accepted in the scientific and engineering communities associated with big science and related installations, the number of studies and cases of real application of DT technology will grow rapidly, which has already been observed in other applied areas.

Analysis of the literature shows that potential areas of application of DT in large projects of scientific infrastructure include:

- optimization of general management and reporting, increasing the efficiency of external communications with management and financing organizations, society through the creation of realistic three-dimensional models of scientific facilities with the accumulation of data on the state of the facilities themselves and the environment, experiments and main results;
- monitoring, maintenance, anomaly detection, prediction of equipment failures based on the collection and processing of data from sensor networks, comparison with the results of simulations using physical models or neural networks trained on real data;
- management of the life cycle of complex scientific equipment through the creation and constant updating of its DT, which provides consistent storage and processing of log/archive and current information about its components and their condition, routine and urgent work on configuration, repair, and updating of equipment;
- development, verification, fine-tuning of equipment and/or software for event detection based on DTs of experimental spaces or observed objects and phenomena;
- creation and support of automatic control systems that provide optimization of operating modes, reduction of time for setting up and changing configurations, reduction of delays in the generation of control signals.

Conceptualizing these applications will accelerate and optimize the adoption of DT technologies, and provide attention to the potential benefits and possible drawbacks of the technology, and the risks associated with its application. An essential issue related to this conceptualization is the correct definition of the object, the virtual embodiment of which is the DT. We have already seen examples of DTs of the systems and objects under study, which is markedly different from the classical interpretations of this technology. Probably, in the case of using a data center for automated control of complex objects, we can say that a virtual copy of the “operator control and control system control object control object” system is being formed, which also requires a significant rethinking of classical approaches.

Thus, DT technologies have significant potential in the design, construction and operation of large scientific infrastructure projects, providing the ability to virtually test various concepts and configurations, accelerate and improve the processes of setting up and optimizing equipment operation, monitoring the performance of critical components, predicting faults, which will ultimately lead to more scientific results at the same or lower cost.

As a perspective, the experience and results of making the DT of the SKIF may be used in the development of a universal platform for digital twins of other Large-Scale Research Infrastructure.

Acknowledgements

Authors thank Yan Zubavichus, Vladimir Poteryaev, Stanislav Shakirov, Igor Marinin, Maxim Gorodnichev, Yuriy Medvedev and Ruslan Permyakov for fruitful discussion.

The work was supported by the state project of ICMMG SB RAS 0251-2022-0005.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Grieves, M.: Digital Twin: Manufacturing Excellence Through Virtual Factory Replication. White Paper, pp. 1–7 (2014).
2. Grieves, M., Vickers, J.: Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches. pp. 85–113. Springer (2016). https://doi.org/10.1007/978-3-319-38756-7_4
3. Tuegel, E.J., *et al.*: Reengineering Aircraft Structural Life Prediction Using a Digital Twin. Int. J. Aerospace Eng. Article 154798. (2011). <https://doi.org/10.1155/2011/154798>
4. Glaessgen, E.H., Stargel, D.S.: The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles (2012).
5. Tao, F., *et al.*: Digital Twin in Industry: State-of-the-Art. IEEE Trans. Ind. Inform. 15(4), 2405–2415 (2019). <https://doi.org/10.1109/TII.2018.2873186>
6. O’Connell, C.: CAD/CAM (Computer-Aided Design/Computer-Aided Manufacturing). Sci. & Tech. Libraries, Routledge 7(4), 127–154 (1987). https://doi.org/10.1300/J122v07n04_13
7. Borrelli, A., Wellmann, J.: Computer Simulations Then and Now: An Introduction and Historical Reassessment. N.T.M. 27(4), 407–417 (2019). <https://doi.org/10.1007/s00048-019-00227-6>
8. Boschert, S., Rosen, R.: Digital Twin-The Simulation Aspect. In: Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers. pp. 59–74. Springer (2016). https://doi.org/10.1007/978-3-319-32156-1_5
9. Fuller, A., *et al.*: Digital Twin: Enabling Technologies, Challenges and Open Research. IEEE Access 8, 108952–108971 (2020). <https://doi.org/10.1109/ACCESS.2020.2998358>
10. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A Survey. Comput. Netw. 54(15), 2787–2805 (2010). <https://doi.org/10.1016/j.comnet.2010.05.010>
11. Sisinni, E., *et al.*: Industrial Internet of Things: Challenges, Opportunities, and Directions. IEEE Trans. Ind. Inform. 14(11), 4724–4734 (2018). <https://doi.org/10.1109/TII.2018.2852491>
12. Lee, J., Bagheri, B., Kao, H.-A.: A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems. Manuf. Lett. 3, 18–23 (2015). <https://doi.org/10.1016/j.mfglet.2014.12.001>
13. Philip Chen, C.L., Zhang, C.-Y.: Data-intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data. Inf. Sci. 275, 314–347 (2014). <https://doi.org/10.1016/j.ins.2014.01.015>

14. Lermer, M., Reich, C.: Creation of Digital Twins by Combining Fuzzy Rules with Artificial Neural Networks. In: IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, October 14-17, 2019. pp. 5849–5854. IEEE (2019). <https://doi.org/10.1109/IECON.2019.8926914>
15. Tarkhov, D.A., Malykhina, G.F.: Neural Network Modelling Methods for Creating Digital Twins of Real Objects. *Journal of Physics: Conference Series* 1236(1), 012056 (2019). <https://doi.org/10.1088/1742-6596/1236/1/012056>
16. Kaur, M.J., Mishra, V.P., Maheshwari, P.: The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action. In: Farsi, M., Daneshkhah, A., Hosseinian-Far, A., Jahankhani, H. (eds) *Digital Twin Technologies and Smart Cities. Internet of Things*. pp. 3–17. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-18732-3_1
17. Rathore, M.M., *et al.*: The Role of AI, Machine Learning, and Big Data in Digital Twinning: A Systematic Literature Review, Challenges, and Opportunities. *IEEE Access* 9, 32030–32052 (2021). <https://doi.org/10.1109/ACCESS.2021.3060863>
18. Tao, F., *et al.*: Digital Twin-driven Product Design, Manufacturing and Service with Big Data. *Int. J. Adv. Manuf. Tech.* 94(9-12), 3563–3576 (2018). <https://doi.org/10.1007/s00170-017-0233-1>
19. Kritzinger, W., *et al.*: Digital Twin in Manufacturing: A Categorical Literature Review and Classification. *IFAC-PapersOnLine* 51(11), 1016–1022 (2018). <https://doi.org/10.1016/j.ifacol.2018.08.474>
20. Errandonea, I., Beltran, S., Arrizabalaga, S.: Digital Twin for Maintenance: A Literature Review. *Comput. Ind.* 123, 103316 (2020). <https://doi.org/10.1016/j.compind.2020.103316>
21. Mohammadi, N., Taylor, J.E.: Smart City Digital Twins. In: 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017, Honolulu, HI, USA, November 27 – Dec. 1, 2017. pp. 1–5. IEEE (2018). <https://doi.org/10.1109/SSCI.2017.8285439>
22. Deng, T., Zhang, K., Shen, Z.-J.M.: A Systematic Review of a Digital Twin City: A New Pattern of Urban Governance Toward Smart Cities. *J. Manag. Sci. Eng.* 6(2), 125–134 (2021). <https://doi.org/10.1016/j.jmse.2021.03.003>
23. Deng, M., Menassa, C.C., Kamat, V.R.: From BIM to Digital Twins: A Systematic Review of the Evolution of Intelligent Building Representations in the AEC-FM Industry. *J. Inf. Technol. Constr.* 26, 58–83 (2021). <https://doi.org/10.36680/J.ITCON.2021.005>
24. Lu, Q., *et al.*: From BIM Towards Digital Twin: Strategy and Future Development for Smart Asset Management. *Stud. Comput. Intell.* 853, 392–404 (2020). https://doi.org/10.1007/978-3-030-27477-1_30
25. Sepasgozar, S.M.E., *et al.*: Lean Practices Using Building Information Modeling (BIM) and Digital Twinning for Sustainable Construction. *Sustainability (Switzerland)* 13(1), 1–22 (2021). <https://doi.org/10.3390/su13010161>

26. Regis, A., *et al.*: Physic-based vs Data-based Digital Twins for Bush Bearing Wear Diagnostic. *Wear* 526-527, 204888 (2023). <https://doi.org/10.1016/j.wear.2023.204888>
27. Sarkar, P.: Digital Twin Modeling Using Machine Learning and Constrained Optimization. *Medium* (2022). <https://towardsdatascience.com/digital-twin-modeling-using-machine-learning-and-constrained-optimization-401187f2a382>, accessed: 2023-04-10
28. Jacob, M., Hallonsten, O.: The Persistence of Big Science and Megascience in Research and Innovation Policy. *Sci. Public Policy* 39(4), 411–415 (2012). <https://doi.org/10.1093/scipol/scs056>
29. Borner, K., Silva, F.N., Milojevic, S.: Visualizing Big Science Projects. *Nat. Rev. Phys.* 3(11), 753–761 (2021). <https://doi.org/10.1038/s42254-021-00374-7>
30. Cramer, K.C., Hallonsten, O.: Big Science and Research Infrastructures in Europe. 2020, pp. 1–264. Edward Elgar Publishing, Cheltenham, UK (2020). <https://doi.org/10.4337/9781839100017>
31. D’Ippolito, B., Ruling, C.-C.: Research Collaboration in Large Scale Research Infrastructures: Collaboration Types and Policy Implications. *Res. Policy* 48(5), 1282–1296 (2019). <https://doi.org/10.1016/j.respol.2019.01.011>
32. Johnston, S., *et al.*: Science with ASKAP: The Australian Square-Kilometre-Array Pathfinder. *Exp. Astron.* 22(3), 151–273 (2008). <https://doi.org/10.1007/s10686-008-9124-7>
33. Bednarz, T., *et al.*: Digital Twin of the Australian Square Kilometre Array (ASKAP). In: SIGGRAPH Asia 2020 Posters. Article 15. ACM (2020). <https://doi.org/10.1145/3415264.3425462>
34. Jonas, J.L.: MeerKAT - The South African Array with Composite Dishes and Wide-Band Single Pixel Feeds. *Proc. IEEE* 97(8), 1522–1530 (2009). <https://doi.org/10.1109/JPROC.2009.2020713>
35. Taljaard, C., Chrysostomou, A., Van Zyl, A.N.: Sculpting a Maintenance Twin for SKA. In: Modeling, Systems Engineering, and Project Management for Astronomy IX, vol. 11450, pp. 114500F. SPIE (2020). <https://doi.org/10.1117/12.2562337>
36. Nan, R., *et al.*: The Five-Hundred-Meter Aperture Spherical Radio Telescope (FAST) Project. *Int. J. Mod. Phys. D* 20(6), 989–1024 (2011). <https://doi.org/10.1142/S0218271811019335>
37. Li, Q.-W., *et al.*: Prognostics and Health Management of FAST Cable-Net Structure Based on Digital Twin Technology. *Res. Astron. Astrophys.* 20(5) (2020). <https://doi.org/10.1088/1674-4527/20/5/67>
38. Wen, J., *et al.*: Rapid Modeling Method for The Digital Twin of Five-Hundred-Meter Aperture Spherical Radio Telescope. *IAENG Int. J. Comput. Sci.* 49(2) (2022).
39. Zhang, Q., Wu, P., Zhao, Z.: Design and Application of Digital Twin System Architecture for Large Radio Telescope. *Jisuanji Jicheng Zhizao Xitong/Comput. Integr. Manuf. Syst. CIMS* 27(2), 364–373 (2021). <https://doi.org/10.13196/j.cims.2021.02.005>

40. Predehl, P., *et al.*: The eROSITA X-ray Telescope on SRG. *Astron. Astrophys.* 647 (2021). <https://doi.org/10.1051/0004-6361/202039313>
41. Seppi, R., *et al.*: Detecting Clusters of Galaxies and Active Galactic Nuclei in an eROSITA All-Sky Survey Digital Twin. *Astron. Astrophys.* 665 (2022). <https://doi.org/10.1051/0004-6361/202243824>
42. Jones, B.J.P.: The Status of the MicroBooNE Experiment. *J. Phys.: Conf. Ser.* 408 (2013). <https://doi.org/10.1088/1742-6596/408/1/012028>
43. Kafkes, D., Schram, M.: Developing Robust Digital Twins and Reinforcement Learning for Accelerator Control Systems at the Fermilab Booster. In: *Proc. of the 12th International Particle Accelerator Conference*. pp. 2268–2271. JACoW Publishing, Geneva, Switzerland (2021). <https://doi.org/10.18429/JACoW-IPAC2021-TUPAB327>
44. Abramowicz, H., *et al.*: Higgs Physics at the CLIC Electron-Positron Linear Collider. *Eur. Phys. J. C* 77(7), article 475 (2017). <https://doi.org/10.1140/epjc/s10052-017-4968-5>
45. Doytchinov, I., *et al.*: Thermal Effects Compensation and Associated Uncertainty for Large Magnet Assembly Precision Alignment. *Precis. Eng.* 59, 134–149 (2019). <https://doi.org/10.1016/j.precisioneng.2019.06.005>
46. Neumann, A., *et al.*: bERLinPro Becomes SEALab: Status and Perspective of the Energy Recovery Linac at HZB. In: *Proc. 13th International Particle Accelerator Conference*. pp. 1110–1113. JACoW Publishing, Geneva, Switzerland (2022). <https://doi.org/10.18429/JACoW-IPAC2022-TUPOPT048>
47. Mahshid, M.Z., *et al.*: Realization of an Energy System-Informed Digital Twin of the KARA Accelerator at KIT in a Real-Time Simulation Environment: the ACCESS Project. In: *14th International Particle Accelerator Conference* (2023).
48. Giles, A., *et al.*: Operation the Accelerator Test Facility Linac Transport Beamline by Using Artificial Intelligence and Machine Learning Methods. In: *14th International Particle Accelerator Conference* (2023).
49. Ansel, A., *et al.*: A New Product Lifecycle Management Platform for CERN’s Accelerator Complex and Beyond. In: *14th International Particle Accelerator Conference* (2023).
50. Edelen, A.: AI/ML and Its Operational Challenges at SLAC’s Accelerators and Collaborating Facilities (2021).
51. Schaefer, K.I. (ed.): *Technological Infrastructure of the SKIF Center for Common Use*. Novosibirsk (2022). <https://disk.yandex.ru/d/1SBhHph2rgbeBg>, accessed: 2023-04-10
52. *Scientific Program of the SKIF Center for Common Use: Key Areas of Research at the First Stage Experimental Stations and the Concept of Infrastructure Development until 2035*. Novosibirsk, 439 p. (2023). <https://disk.yandex.ru/d/gxEIdsjIa1IvHw>, accessed: 2023-04-10
53. GOST R 57700.37-2021: *Computer Models and Modeling. Digital Twins of Products. General Provisions*.

54. Wilkinson, M., Dumontier, M., Aalbersberg, I., *et al.*: The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Sci. Data* 3, 160018 (2016). <https://doi.org/10.1038/sdata.2016.18>
55. Schwarz, N., Campbell, S., Hexemer, A., *et al.*: Enabling Scientific Discovery at Next-Generation Light Sources with Advanced AI and HPC. In: *Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI*, SMC 2020, *Comm. Comp. Inf. Sci.*, vol. 1315, pp. 145–156. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63393-6_10
56. Wang, C., Steiner, U., Sepe, A.: Synchrotron Big Data Science. *Small* 14(46), 1802291 (2018). <https://doi.org/10.1002/smll.201802291>
57. Wagner, H.: Data Handling and Storage. *Synchrotron Radiation News* 32(3), 2–3 (2019). <https://doi.org/10.1080/08940886.2019.1618682>
58. Ponsard, R., Janvier, N., Kieffer, J., *et al.*: RDMA Data Transfer and GPU Acceleration Methods for High-Throughput Online Processing of Serial Crystallography Images. *J. Synchrotron Radiat.* 27(5), 1297–1306 (2020). <https://doi.org/10.1107/s1600577520008140>
59. Prokhorov, A., Lysachev, M., Borovkov, A. (eds.): *Digital Twin. Analysis, Trends, World Experience*. First Edition. Moscow, Alliance Print, 401 p. (2020).