

Supercomputing Frontiers and Innovations

2024, Vol. 11, No. 1

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

Grid Computing Evolution in Scientific Applications M.A. Grigorieva, A.A. Klimentov	4
Development of Novel Nsp16 Inhibitors as Potential Anti-SARS-CoV-2 Agents K. Zhang, A.V. Sulimov, I.S. Ilin, D.C. Kutov, A.S. Taschilova, S. Jiang, T. Wang, V.B. Sulimov, Y. Xiao	51
Modeling the Recovery of the Earth's Gravitational Field from Satellite Measurements Using Parallel Computations A.S. Zhamkov, V.K. Milyukov, S.V. Ayukov, A.I. Filetkin, I.Yu. Vlasov, V.E. Zharov	67
WAVEWATCH III Hybrid Parallelization for Azov Sea Wave Modelling A.I. Sukhinov, E.A. Protsenko, S.V. Protsenko	81
Study of Thin Optical Films Properties Using High-performance Atomistic Simulation F.V. Grigoriev, V.B. Sulimov, A.V. Tikhonravov	97



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.

Grid Computing Evolution in Scientific Applications

Maria A. Grigorieva¹ , Alexei A. Klimentov² 

© The Authors 2024. This paper is published with open access at SuperFri.org

The advent of interconnected machines laid the foundation for utilizing distributed computing resources. Coinciding with rapid advancements in computing technologies and significant hardware innovations at the turn of the century, the field of science also experienced exponential growth. As supercomputers remained limited in availability and usage, distributed computing leveraged the potential of idle and dedicated resources, bridging the gap between scientists and computationally intensive projects. This paper provides a review of the evolutionary journey of grid computing in scientific applications, starting from the advancements in network connection technologies and concept of metacomputing and progressing to the current developments integrating cloud technologies with large-scale grids. The paper aims to outline the key milestones, advancements, and challenges encountered throughout this evolution, highlighting the potential of grid computing in enabling scientific breakthroughs and addressing future research directions. The most popular middleware systems are considered, as well as a description of scientific grid systems that existed in the past and are still in operation today is given. At the end of the article, we examined two of the most significant scientific discoveries that became possible largely thanks to grid technologies.

Keywords: grid, distributed computing, history, science, metacomputing, grid middleware, scientific grid, cloud computing.

Introduction

The advancement of distributed computing technologies was primarily driven by scientific endeavors. Since the early 21th century, the technologies faced significant challenges posed by international scientific projects in data-intensive fields such as particle physics, astronomy, computational chemistry, bioinformatics, and Earth and climate sciences. Subsequently, these technologies evolved and met these challenges by connecting existing heterogeneous computing resources distributed globally into unified environments through high-throughput networks and specialized middleware systems. This integration ensured secure, dependable, efficient, and transparent operations. As a result, massive amounts of data, reaching the scale of multipetabytes and even exabytes, became distributed and accessible to broad international communities. The advancement of big science propelled technological and software innovations, eventually leading to the creation of grid computing infrastructures.

This article aims to provide an overview of the key stages in the development of distributed computing systems within scientific domains. The paper is structured as follows:

- *Section 1* outlines the evolutionary journey of grid computing in scientific disciplines, highlighting significant milestones while also discussing advancements in network technologies.
- *Section 2* explores the historical background of interconnecting machines for computational purposes.
- *Section 3* gives an overview of metacomputing projects, emphasizing the stages of metacomputing and delving into the insights gained from these projects.
- Moving to *Section 4*, the focus is centered on crucial grid computing initiatives, including the I-WAY experiment, pioneering literature on grid computing, grid forums, and the shift towards Web 2.0 integration.

¹Lomonosov Moscow State University, Moscow, Russian Federation

²Brookhaven National Laboratory, Upton, US

- *Section 5* offers insights into grid middleware systems like HTCondor, Globus Toolkit, NIMROD, ARC, UNICORE, BOINC, which became the foundation for the efficient functioning of distributed systems.
- *Section 6* elaborates on various significant and captivating scientific grids such as TeraGrid, GridPP, WLCG, IGWN, among others.
- Over time, technologies have evolved to such a level that they can dictate their requirements to science. Hence, *Section 7* delves into exploring the integration of large grid systems with cloud technologies, using the WLCG as a case study example.
- The concluding *Section 8* illuminates two groundbreaking scientific discoveries that have been enabled through the innovative applications of grid computing technologies.

The topic of grid computing has a very long history, and it is so multifaceted that it does not allow for all its aspects to be covered within the framework of one article. Our objective is to present a broad perspective on the evolution of the technology, its various developmental stages, and to illustrate, using past and present grid systems as examples, the significance of this technology in scientific endeavors. It has established itself as a dependable bedrock for research, facilitating the continuous success of numerous major projects and fostering optimism for forthcoming discoveries.

Table 1 shows a representative sample of the most significant events in the evolution of grid computing, highlighted in this paper.

1. Distributed Computing and Network Throughput

Since a grid refers to computers connected by networks, the evolution of grid technologies is tightly connected with the significant increase of the network throughput.

In the 1960s, networks were primarily used for communication between government agencies and universities. These early networks were slow, with limited capacity and functionality. Thus, Ethernet was first developed in the early 1970s and initially it operated at a speed of *2.94 Mbps (megabit per second)*. However, over the years, Ethernet has evolved significantly and has seen several speed upgrades. In 1980, Ethernet was upgraded to support *10 Mbps*, which became known as 10Base-T. This was followed by the introduction of Fast Ethernet in 1995, which supported speeds of up to *100 Mbps*, also known as 100Base-T. Gigabit Ethernet was introduced in 1998, which allowed for even faster data-transfer rates of up to *1 Gbps (Gigabit per second)*. This was followed by 10 Gigabit Ethernet in 2002, which allowed for speeds of up to 10 Gbps, and more recently, by *40 Gigabit* and *100 Gigabit Ethernet*. So, the speed of Ethernet has evolved significantly from 2.94 Mbps in the 1970s to multi-gigabit speeds today. And of course, network speed contributed a lot in the development of distributed technologies [66].

To gain a deeper understanding of the history of distributed computing, it is essential to differentiate between the two types of networks: LAN (Local Area Network) and WAN (Wide Area Network). The main difference between LAN and WAN is their geographic scope, with LAN being limited to a small area and WAN covering a much larger area. This also affects the transmission speed and cost, as WANs often require high-speed and expensive communication links to connect various locations. Before the 2000s, WAN speeds had lagged behind LAN. While LAN technologies evolved rapidly with advancements such as Ethernet, WANs had to catch up with the advancements in fiber optics, routing protocols, and networking equipment to improve data transfer speeds over long distances. However, as networking technology has evolved WAN data rates have increased [16].

Table 1. Timeline of Grid Computing Evolution in Scientific Applications

Years	Technology/Event	Description
Early 1970s	ARPANET, NCP, FTP, TCP/IP, Ethernet	Development of protocols and networking technology that laid the groundwork for a connected world, enabling seamless communication and information exchange among computers.
Late 1980s–early 1990s	Gigabit Testbed and Metacomputing Projects, Condor, Web	Testing of high-speed network technology, developing projects that utilize distributed HPC for solving complex problems through parallel coordination. Emergence of the high-throughput computing system that helped in managing and scheduling computational jobs across distributed resources. Web technologies have been employed to create interfaces for accessing grid resources.
Mid–1990s	NIMROD, Globus Toolkit, UNICORE, I-WAY, Web Metacomputing	Development of grid middleware systems and projects showcased the feasibility, scalability, and advantages of grid computing, shaping its evolution.
Late 1990s	Grid book, Grid Forums, Web 2.0 and Web Services	Literature about grid computing, organization that worked on developing grid technologies, standards, and best practices to advance grid computing globally. Emergence of Web services have greatly influenced grid computing evolution by improving the accessibility, interoperability, and scalability of distributed computing resources.
2000–2010	Condor-G, ARC, Globus Toolkit 2-4, gLite, BOINC, NIMROD-G, Globus Online	Development and upgrading of notable grid middleware systems: additional features for managing and scaling tasks in grid environments, gradual transition to service-oriented platforms.
2011–now	Clouds over Grids, HTCondor, Unicore 7-9, Globus Toolkit decommissioning	Integration of cloud computing technologies with grids to improve scalability, flexibility, and resource utilization. Upgrading major grid middleware systems to support grid and cloud environments, transition from the older framework to more modern and streamlined approaches to distributed computing.

In the 1960s WANs connecting different organizations or regions were rare and typically used leased lines or dial-up connections. The data transmission rates were very low (about *56 kbps*). In 1970–80s the use of network protocols such as TCP/IP and X.25 for WANs started to emerge. Then, in the 1980s, the wide-scale deployment of dial-up modems and Integrated

Services Digital Network (ISDN) for WANs began, which improved the network speed compared to leased lines [74]. In the 1990s, when LANs started to use Fast and Gigabit Ethernet, WAN technologies like Asynchronous Transfer Mode (ATM) and Frame Relay replaced X.25. For example, ATMs offered speeds of up to *622 Mbps*. In the beginning of 2000s WANs moved to MPLS (Multiprotocol Label Switching) and VPN technologies. Fiber optic cables, which have much higher bandwidth than copper cables, also became more commonly used. MPLS could support multiple protocols and traffic types and offered speeds of *10 Gbps* and beyond. Over time, with the advancement of technologies like fiber optics, MPLS, and improvements in networking equipment, WAN speeds have also significantly improved and caught up with LAN speeds, now offering multi-Gbps and even 10 Gbps+ connections, narrowing the gap between LANs and WANs in terms of bandwidth capabilities.

Table 2 shows the evolution of LANs and WANs throughput over time.

Table 2. Evolution of LAN and WAN Throughput

Years	LAN	WAN
1980s	1–10 Mbps	a few Kbps to a few Mbps
1990s	100 Mbps (Fast Ethernet)	1.544–2.048 Mbps
Late 1990s to 2000s	1 Gbps (Gigabit Ethernet)	multiple to tens Mbps
2010s	10–100 Gbps	100 Mbps – 1 Gbps
Present	400 Gigabit Ethernet	10 Gbps and beyond

We are discussing the evolution of network technologies at the beginning of this paper as they have made a significant contribution to the development of distributed computing. Essentially, the development of grid technologies in international mega-science projects is closely connected with the evolution of WAN connections.

2. Early Ages of Distributed Computing

The idea of leveraging unutilized storage and CPU resources emerged when computers were first linked by networks. Initially, there were two directions for the evolution of remote computing. The first grew from the necessity of data exchange between military departments of the USA, and the second – from the introduction of the first mainframe machines.

The predecessor to the modern Internet was the ARPANET [59]. It appeared in 1967 as a project of the United States Department of Defense’s Advanced Research Projects Agency (DARPA). Its purpose was to create a network that could connect computers and share information between them, even in the event of a nuclear attack, which implies, above all, reliability and fault-tolerance in data transmission. The network throughput in the ARPANET was initially 50 Kbps. The first successful message sent on the Internet was in 1969, between two computers at the University of California, Los Angeles (UCLA) and the Stanford Research Institute (SRI). These machines were linked by a direct physical connection via a leased telephone line. Actually, by 1969, the first computers of different types were connected on a network, but it was not useful until the introduction of the NCP (Network Control Protocol) in 1970 – a primary protocol responsible for establishing connections, detecting errors, packetizing data, and routing

data across the network³. It enabled the first two host-host protocols, remote login (Telnet) and File Transfer Protocol (FTP). The ARPANET was declared operational in 1971. NCP was replaced by the Transmission Control Protocol/Internet Protocol (TCP/IP) in 1972, which is still used today as the standard protocol for the Internet. These technologies became the basis for developments over the next 20 years.

Another direction of the evolution of remote computing was related to the mainframe computers. They were first introduced in the 1960s and they were expensive, required specialized skills to operate, making them inaccessible to most individuals and businesses. To overcome this challenge, a system of remote computing was used that allowed users to access these mainframe computers from remote locations. That was another usage of the NCP protocol that allowed a user on one machine to login to another device via the Telnet protocol and upload a file. FTP allowed users to connect to a mainframe computer through a network connection and transfer files back and forth between their local computer and the mainframe. FTP quickly became a popular tool for remote computing and file sharing, and it remains in use today as a standard protocol for transferring files over the Internet.

Until the early 1970s, the dial-up connections using modems allowed users to connect to remote computers over telephone lines. This was a slow and unreliable method of communication. The emergence of Ethernet, FTP and TCP/IP protocols in the early 1970s laid the foundation for the evolution of remote computing by enabling remote access to data and resources and facilitating communication between different computer systems over a network. At the same time researchers began to understand that distributed computing would be difficult. As any message may be lost, corrupted, or delayed, the robust algorithms must be used in order to build a coherent system.

2.1. Linked Machines for Computations

In the early 1980s there were attempts to develop production systems that were supposed to ensure consistency, availability, and performance in distributed systems. And besides data exchange, there were early attempts at utilizing linked machines for computations. For example, the HTCondor (initially called Condor) project was developed in 1984 by Professor M. Livny at the University of Wisconsin, USA. It was a software package to manage workloads in the distributed computing environment. It allowed large numbers of computing tasks to be distributed across multiple machines, so that they could be executed in parallel. This advancement enabled the efficient processing of vast quantities of data at a significantly accelerated pace compared to using a single machine [58].

The idea behind the utilization of linked machines for computations was to aggregate idle computing resources that would otherwise be unused and use them to perform computationally intensive tasks. Condor used a variety of protocols for data processing on multiple machines, including FTP for file transfer, TCP/IP for reliable communication between devices, and other protocols such as UDP (User Datagram Protocol) for faster data transmission.

2.2. Emergence of World Wide Web

In the 1980s, software systems were developed with limited consideration for their interoperability with other systems. However, as the number and diversity of computer systems

³<https://www.rfc-editor.org/info/rfc33>

increased, the necessity of data exchange between them became more apparent. These systems utilized varying data formats and communication protocols, posing challenges in data sharing and information exchange. Consequently, this lack of interoperability impeded the development of new applications that could seamlessly operate across different systems. Hence, the demand for a standardized approach to connect diverse software systems became evident, leading to the emergence of “middleware”. Although the term “middleware” was not introduced until the mid-1990s, it denotes the need for a standardized solution to facilitate software system integration.

Things completely changed with the emergence of the World Wide Web (or Web)⁴. It is considered that Web was developed in 1989 at CERN by Tim Berners-Lee. However, actually, there was a whole scientific group involved in this development. The Web was described as a method for sharing information among geographically dispersed scientists: the CERN community at that time included more than 17000 scientists from over 100 countries. In Web infrastructure all hypertext documents were linked into an information system accessible from any node on the network [19].

The initial set of protocols and standards that were used to create the first Web system were limited, but still formed the foundation of the modern Web: HTML (Hypertext Markup Language), HTTP (Hypertext Transfer Protocol), URL (Uniform Resource Locator), TCP/IP, DNS (Domain Name System), FTP. The development of the Web created new demands for high-throughput computing that eventually revolutionized the way people access, share, and use information and resources.

3. Metacomputing

Another direction of the evolution of distributed computing in science was the idea of sharing supercomputing resources to address the growing demand for computing power in scientific and engineering fields. In the 1990s the scientific interests began to recognize that distributed supercomputing might achieve higher performance than individual supercomputers or clusters can provide [80]. So, various geographically distributed supercomputers can be connected by WANs in order to solve grand challenge problems in reasonable time. Such systems could be used, for example, to carry out combined climate and ocean modeling as demonstrated by using different types of the unique linked computing resources. This concept was called “metacomputing” and can be considered as the first prototype of grid computing. Basically, metacomputer can be defined as a network of heterogeneous, computational resources linked by software in such a way that they can be used as easily as a personal computer.

Supercomputers and high-performance computing (HPC) systems were often expensive, difficult to maintain, and not easily scalable to new applications. This made it difficult for individual researchers and small organizations to access the computing power they needed to conduct advanced studies and solve complex problems. Moreover, grand challenge problems run weeks and months even on supercomputers and clusters. To solve these issues, a wide variety of powerful batch execution systems such as LoadLeveler (descendant of Condor, 1994), LSF (Load Sharing Facility, 1992) [84], Maui (1999) [54], Portable Batch System (PBS, 1991) [51] were developed in the United States and spread throughout academia and business.

For instance, LoadLeveler was a system designed to effectively manage both serial and parallel jobs across a cluster of servers. This cluster, referred to as a LoadLeveler cluster, comprised

⁴<https://www.home.cern/science/computing/birth-web/short-history-web>

diverse machines or servers, such as desktop workstations utilized for batch jobs during idle periods, dedicated servers, and parallel machines [77].

The allocation of jobs to machines in the cluster was done by a scheduler. This allocation depended on the availability of resources within the cluster and various rules that could be set by the LoadLeveler administrator. When a user submitted a job using a job command file, the LoadLeveler scheduler tried to locate resources within the cluster to meet the job's requirements. Meanwhile, LoadLeveler's responsibility was to maximize the efficiency of the cluster. It achieved this by maximizing resource utilization while minimizing the turnaround time for jobs experienced by users.

LoadLeveler was used on several high-performance computing systems, including the IBM Blue Gene/L supercomputer, the Cray XT5h at Oak Ridge National Laboratory, and the IBM SP at the Max Planck Institute in Munich.

Despite significant progress in the development of batch execution systems, it can be assumed that the establishment of gigabit networks played a crucial role in the development of metacomputing at a global scale.

3.1. Gigabit Testbed Initiative

The Gigabit Testbed Initiative (USA, 1987–1995) [1] was a major effort by approximately forty organizations representing universities, telecommunication carriers, industry and national laboratories, and computer companies to create a set of very high-speed network testbeds and to explore their application to scientific research. This initiative started in 1990 and was funded by the National Science Foundation (NSF) and the Defense Advanced Research Projects Agency (DARPA).

At the time the Gigabit Testbed Project began, WAN speeds were in the 50 Kbps to 1.5 Mbps range and LAN aggregate speeds were typically 10 Mbps or less. So, we could argue that the creation of gigabit networks was a milestone which enabled the discovery of new possibilities for networked data transmission, and triggered the refinement of grid technologies.

“The coupling of computer network researchers, who have largely come from the field of computer science, with the carrier telecommunications community provided another important dimension of integration. The development of computer communications networks and carrier operated networks have historically proceeded along two separate paths with relatively little cross-fertilization. The testbeds allowed the two communities to work together, allowing each to better appreciate the problems and solutions of the other” (quote from The Gigabit Testbed Initiative Final Report, December 1996 [2]).

The five testbeds were geographically located around the US. Each testbed had a different set of research collaborators and a different overall research focus and objectives. At the same time, there were also common areas of research among the testbeds, allowing different solutions for a given problem to be explored. The Gigabit Testbed Initiative, by creating a new model for network research, has had a major impact on both education and industry.

3.2. Stages of Metacomputing Evolution

Metacomputing evolved in several stages: creating software to make users jobs executed on different machines easier (LAN); distribution of applications for seamless users interaction (LAN); and transition from LAN to WAN metacomputing systems.

3.2.1. Developing software to simplify the execution of users jobs on different machines

The first stage was in creating and harnessing the software to make the users jobs to be executed on different machines easier. “For any one project, a typical user might use a desktop workstation, a remote supercomputer, a mainframe supporting the mass storage archive, and a specialized graphics computer. Some users have worked in this environment for the past decade, using adhoc, custom solutions, providing specific capabilities at best, in most cases moving data and porting applications by hand from machine to machine. The goal of building a metacomputer is elimination of the drudgery involved in carrying out a project on such a diverse collection of computer systems” [78]. So, the first stage involved interconnecting the resources with high-performance networks, implementing a distributed file system, coordinating user access across the various computational elements, and making the environment seamless.

The metacomputer at the National Center for Supercomputing Applications (NCSA, USA) was one of the examples of the LAN metacomputer of the first stage. In 1987 NCSA embarked on a research project aimed at investigating various emerging computational technologies that held the potential to greatly influence the field of scientific computing. This initiative, named the Rivers (Research on Interactive Visual Environments) Project, focused on the development of hardware and software systems. These systems aim to transition high-end 3D visualization from a batch process to an interactive one, as well as enable visualization-based interactive steering of supercomputing simulations within a high-performance distributed environment [50].

Figure 1 demonstrates schematic architecture for the NCSA LAN Metacomputer. Upon

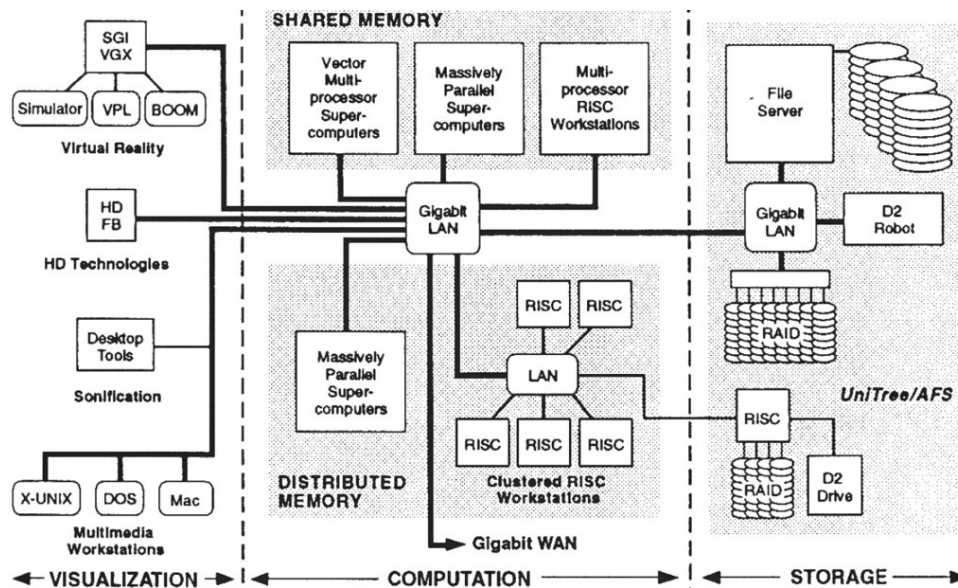


Figure 1. NCSA LAN Metacomputer [50]

closer examination, it is quite feasible to extend this diagram to modern grid and cloud computing setups, featuring distinct compartments for computation, storage, and a web interface for resource access. The pipelined structure of the visualization process required careful hardware system design. A balanced visualization system had to consider sustained bandwidth between distributed hosts, mass storage systems, host memory, and graphics accelerators. Shared memory architectures were also considered attractive for visualization. Powerful general-purpose processors were deemed necessary along with rendering engines due to computation-intensive calculations in the early stages of visualization. The distributed hardware system for visualization

under development within Rivers and NCSA included dual, three-tier schemes for computing and network strategies. High-level machines were connected by a high-speed network (with a peak bandwidth of 100 Mbps), while a middle-tier network (50–80 Mbps) handled messages and conventional network traffic. Graphics accelerators were utilized for rendering operations. The Rivers group developed a three-dimensional, interactive animation tool capable of real-time animation of complex polygon data sets. It was utilized to visualize data from diverse fields such as solid mechanics, atmospheric sciences, and astrophysics.

3.2.2. Distribution of applications for seamless users interaction

During the following stage, efforts were made to distribute applications, enabling users to work seamlessly, in addition to pooling resources by networks. The special software that allowed this to be done in a general way, as opposed to some custom solutions, was emerging since that time. This software is commonly referred to as “middleware”.

Several initiatives have resulted in valuable services for developers of metacomputing applications. To name a few:

- Legion (1992) – a software system that created system components based on a distributed object-oriented model [28]. It provided a high degree of flexibility in managing distributed resources;
- Message Passing Interface (MPI, 1994) – a standard protocol used for message-passing, particularly in distributed memory systems for parallel computing [31].

3.2.3. Transition to WAN metacomputing systems

The first two phases of the evolution of metacomputing predominately sought to harness LAN technology, given the apparent disparity in network capacity between LAN and WAN options. The capabilities of LAN-based metacomputers surpassed their WAN counterparts by roughly a year. The third stage of metacomputing entailed the establishment of a transparent national network, achieved through the implementation of WAN infrastructure and standardization of administrative, file systems, security, accounting, data transferring, and other levels to enable cooperation between multiple LANs.

For example, the High Performance Computing Center Stuttgart (HLRS) partnered with the Pittsburgh Supercomputing Center (PSC) and Sandia National Laboratories (SNL) were linked to establish a transatlantic wide area application test-bed in 1997 as part of the G7 initiative. The project aimed to couple HPC resources to create a powerful computing cluster with a theoretical peak performance of about 1 TFLOPS. The project utilized a high-speed network and a newly developed message-passing software, PACX-MPI⁵, for distributing a single application across multiple MPPs⁶. This setup was enhanced by integrating the visualization software COVISE⁷ for efficient data extraction directly from the application [73].

The project highlighted the need for faster networks for global work, the value of message-passing for distributed computing, challenges with standards in message-passing, limitations in

⁵PACX-MPI – extension of the Message Passing Interface (MPI) standard, which is widely used in distributed and parallel computing to allow communication between processes running on different computing nodes

⁶MPP – Massively Parallel Processing

⁷COVISE – visualization tool for analyzing scientific and engineering data in various fields such as computational fluid dynamics, structural mechanics, and geosciences

closely coupled applications like Navier-Stokes, and the lack of standards for resource management in metacomputing.

Another WAN metacomputing project aimed to create a global metacomputer by connecting supercomputers from Japan, the USA, Germany, and the UK, over a network spanning 10,000 miles. The optimization was achieved by using dedicated PVCs (Permanent Virtual Circuit) or improved routes. The project demonstrated the usability of this global metacomputer during Supercomputing 99 (SC99) conference in Portland with various demanding applications, showcasing the capability to link different types of machines like Cray-T3E and Hitachi SR8000, and successfully handle data-intensive tasks, such as processing output from experimental facilities [70].

3.3. Web Metacomputing

The rise of the World Wide Web brought a surge in Internet hosts. By the turn of the millennium, around 360 million computers were connected. This network of processors forms a potent parallel supercomputer, but many machines are underused, mainly used for basic tasks like email, file editing, and web browsing, resulting in idle time. Hence, leveraging this vast computing resource for cryptography, mathematics, and computational science problems is valuable.

To fill this gap, several alternative metacomputing projects had emerged to leverage the power of the Internet, one of which was Web metacomputing. This concept was related to the coupling of conventional machines. Web metacomputers were called distributed computing systems designed to run on top of the Web. These systems used web-based interfaces to allow users to access computing resources from different machines and locations.

Many of the issues that were addressed for LAN metacomputing systems, such as programmability, scheduling, and security, were considered in isolation and in well-controlled environments. But utilizing the Web as a parallel metacomputer introduced new challenges: there was no shared file system, no user who had accounts on all the potentially available machines, no common architecture, and the execution environment was dynamic.

Several projects were implemented to exploit the potential of the Web for metacomputing: ParaWeb (1995–1996) [23], Popcorn project (1996–1998) [69], Charlotte project (1998–2000) [14], SuperWeb project (1998–2000) [10].

For example, the SuperWeb project was a prototype of a distributed computing infrastructure that integrated hosts, brokers, and clients. Hosts registered a portion of their available computing resources (such as CPU time, memory, disk space, and bandwidth) with resource brokers. Client computations were subsequently assigned to registered resources by the broker. Additionally, an economic model was examined for trading computing resources within this project.

3.4. Lessons Learned

Metacomputing projects have provided valuable insights across networking, programming models, applications, and resource handling, shaping future directions for this field.

- Networking remains a crucial aspect, highlighting the need for faster global networks to support metacomputing on a larger scale efficiently. The limitations in bandwidth, espe-

cially for transatlantic or worldwide operations, underscore the importance of addressing network challenges for seamless metacomputing workflows.

- Dealing with different data representations, processor speeds, and communication speeds across various systems.
- Addressing load imbalance caused by differing processor speeds at the application level due to the lack of standardized ways to determine processor speed.
- Handling resource management and scheduling challenges, particularly in coordinating the availability of resources across multiple systems and networks.
- Recognizing the importance of a common file system for seamless access to data.
- Enhancing interoperability and setting message-passing standards are crucial for optimizing metacomputing applications.

Overall, these projects highlighted the complexities involved in integrating and coordinating resources in a metacomputing environment, emphasizing the need for standardized approaches and advanced resource management techniques.

By the beginning of 2000s, none of the Web metacomputing projects were widely adopted. One of the main reasons contributing to this issue was the lack of widespread standardization and interoperability between the different web metacomputing projects. Another crucial factor was the need for robust security mechanisms to ensure the privacy and confidentiality of sensitive data being processed on remote servers, which often proved to be a significant concern for users. As a result, many Web metacomputing projects were left largely unutilized, with researchers and practitioners eventually pivoting to other more promising alternatives.

However local and WAN metacomputing experiments demonstrated the possibility to create a new class of computing resources based on the coupling of the unique supercomputers. These metacomputing systems were the prototype of distributed systems, and such computing structure introduced some specific hurdles, including different data representations on each system, variable CPU speeds that led to load imbalancing, different communication speeds for internal messages and messages between systems, lack of a common file system and resource management [71].

4. Exploring Grid Computing Initiatives

4.1. Information Wide Area Year Experiment

The next step towards the evolution of distributed computing in scientific applications was the I-WAY (Information Wide Area Year) project, launched in 1995. It was a network connecting supercomputers, databases and advanced visualization devices, like virtual reality (VR) at 17 different sites within North America, interconnected through 10 high-bandwidth ATM networks of varying bandwidth (typically 45–155Mbps) and protocols, using different routing and switching technologies. The idea was not to build a network but to integrate existing high bandwidth networks. I-WAY was used by over 60 application groups for experiments in high-performance computing [35].

The logical structure of the I-WAY project is shown in Fig. 2.

It demonstrates that the significant portion of the I-WAY's physical networking infrastructure leveraged pre-existing smaller ATM research networks. These individual networks were interconnected through collaboration with various prominent network service providers.

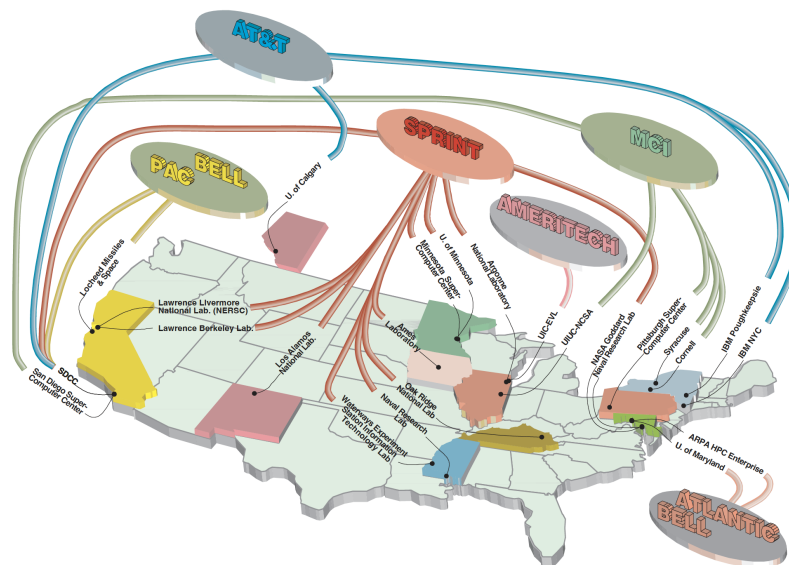


Figure 2. I-WAY Logical Structure [35]

I-WAY project identified several issues vital for designing future distributed computing environments:

- networking infrastructure should be relatively stable and persistent, because any disruption in the networking infrastructure can lead to loss of data, decreased performance, and reduced usability of the system;
- number of resources (people, machines, data) interconnected by the network should be large enough to create a critical mass of users;
- users should be involved with the computer scientists in development of experimental middleware. This can provide valuable feedback on the usability of the system, and also help identify any issues that may arise during implementation.

A major challenge of the I-WAY was providing a uniform environment across the geographically distributed and dispersed resources. To meet this challenge, researchers developed a middleware, called I-SOFT. This system was designed to run on dedicated I-WAY machines deployed at each participating site, and provided uniform authentication, resource reservation, process creation, and communication functions across I-WAY resources. It had a modular structure, with different components responsible for different functions. Some of the major components included a Resource Manager, which provided a uniform interface for managing resources, a Data Manager for a high-speed data transfer service, and an Execution Manager, which allowed users to submit and monitor jobs on remote systems.

The foundations of modern grid systems were laid in the I-WAY project. I-SOFT infrastructure was a precursor of the Globus Toolkit middleware framework, which became a de-facto standard of grid middleware in large-scale scientific applications for about two decades [43].

After the approach of connecting heterogeneous and distributed resources for collective use was tested in metacomputing systems, the development of distributed computing technology accelerated significantly. There was a need to provide a collection of solutions to problems that frequently come up when trying to build collaborative distributed applications: security, monitoring and resource discovery, access to computing and processing power, moving and managing data, deployment environments.

4.2. The Grid Book

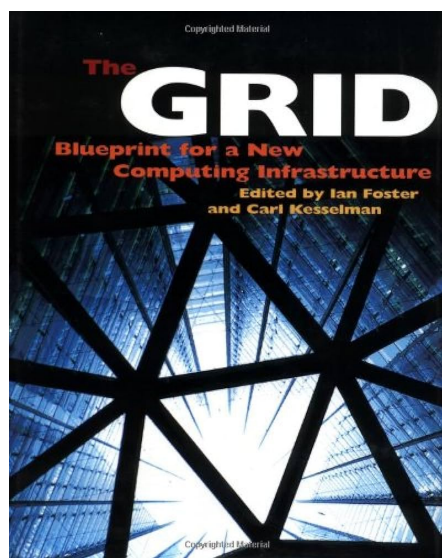


Figure 3. The Grid: Blueprint for a New Computing Infrastructure First Edition

Going back to the late 90's and early 2000s, the period of formation and development of distributed computing, the book “The Grid: Blueprint for a New Computing Infrastructure” by Kesselman, Foster (Fig. 3) was published, which had a catalyzing effect on the development of grid computing and the concept of grid was introduced in this monograph [42]. Kesselman and Foster were the winners of the British Computer Society's Lovelace Medal for their Grid work. They described the grid in the following words:

“The Grid is an emerging infrastructure that will fundamentally change the way we think – and use – computing. The word Grid is used by analogy with the electric power Grid, which provides pervasive access to electricity and, like the computer and a small number of other advances has had a dramatic impact on human capabilities and society. Many believe that by allowing all components of our information technology infrastructure – computational capabilities, databases, sensors,

and people – to be shared flexibly as true collaborative tools, the Grid will have a similar transforming effect, allowing new classes of application to emerge”.

Later on, Ian Foster, at one CERN Computing Seminar that was held in 2001, compared the Grid and the Web, called the Grid “The Web on Steroids”⁸. The Web is not yet a grid, with its open, general-purpose protocols that support access to distributed resources, it provides uniform access to HTML documents, but not the coordinated use of those resources to deliver negotiated qualities of service. On the other hand, grid ensures flexible, high-performant access to all resources, allowing on-demand creation of powerful virtual computing systems. So, whereas the Web is mainly focused on communication, grid computing enables resource sharing and collaborative resource interplay toward common business goals.

This book also provided an early reference to the term “grid middlewar” in the context of grid computing. The authors define grid middleware as “*software that bridges the gap between applications and lower-level network services, facilitating the development of distributed applications that span multiple administrative domains*”.

4.3. Grid Forums

A special organization called the Global Grid Forum (GGF) was created in 1999, which included academic institutions, as well as computer system manufacturers and software providers [13].

The GGF created and published a large number of documents on grid computing architecture, security, data management, and resource management, which were used as a reference for many grid projects worldwide. The GGF organized its meetings three times



⁸<http://cseminar.web.cern.ch/2001/0117/slides.pdf>

a year, often in conjunction with other conferences and events, primarily in North America, Europe, and Asia.

Since that time, scientific communities and industry started to look seriously at grid computing as a solution to resource federation problems. In 2002, the GGF and IBM introduced the Open Grid Services Architecture (OGSA). With this development, grid services were defined as a specialized type of web services, enabling standard internet protocol-based interaction with grid resources [79]. Beginning from 2002 large European, American, and international projects were established based on grid computing and many commercial and industrial grids were launched.

In 2004, the Enterprise Grid Alliance (EGA) consortium emerged as a significant competitor to GGF and attracted substantial industry players like Fujitsu Siemens Computers, Hewlett-Packard, Intel, NEC, Oracle, Sun Microsystems, and EMC.

To enhance collaboration and consolidate the grid community, GGF and EGA finally merged in 2006. This merger resulted in the formation of the Open Grid Forum (OGF)⁹, aimed at advancing grid and distributed computing environments.



4.4. Transition to Web 2.0

Another important milestone in the development of grid at the end of the millennium was the transition to Web 2.0 technologies and Web services in 1999. The term “Web 2.0” was introduced to refer to a new kind of web applications that use innovative architectures and toolkits to create very responsive and user-focused applications. This marked the progression from basic, static web pages to more dynamic ones with user-generated content. Additionally, the emergence of social media as a significant mode of internet communication was a hallmark of this second phase of internet development.

Web 2.0 was characterized by three key technologies: rich internet applications, web-oriented architecture, and the social web. There were two primary interface types that gained widespread acceptance: the “web as platform” interface or API (Application Programming Interface), which allowed companies to provide developers with access to data stores via APIs, and AII (Application Interaction Interface), which leveraged APIs and were classified as web applications.

The Web 2.0 and Grids Workshop at OGF19¹⁰ held in 2007 showcased how Web 2.0 and grid technologies intersect in the realm of e-Science. While both offer solutions, Web 2.0’s standout advantage lies in fostering collaboration and content creation. A key takeaway was the ease of use of Web 2.0, allowing seamless integration with existing resources, encouraging wider participation, and reaping associated benefits. Moreover, the grid’s role was highlighted in providing a robust foundation for Web 2.0 APIs, emphasizing the importance of the API ecosystem. The session underlined the commonalities between workflows and mashups, hinting at the need for further exploration.

Web 2.0 and API technology have contributed significantly to the evolution of grid computing by enabling the creation of more dynamic and interactive grid-based applications and services. Web 2.0 technologies such as AJAX and RESTful APIs have made it easier to develop and deploy grid applications that can interact with a range of web-based services and data sources, making grids more widely accessible and practical for a range of applications and uses.

⁹<https://gridcf.org/>

¹⁰<http://www.semanticgrid.org/OGF/ogf19/>

5. Grid Middleware

During the early 1990s, the task of enabling programs to communicate with one another across multiple machines presented a significant challenge, particularly when different hardware systems, operating systems, and programming languages were involved. Typically, programmers were required to construct an entire protocol framework from scratch using sockets, or alternatively, the programs were unable to establish a connection.

The concept of middleware emerged in response to the growing need for distributed computing systems and the challenges of integrating heterogeneous and complex applications [20].

In the early 90s, scientific literature began to see an increasing number of publications dedicated to middleware, discussing its essence, applications, research, and improvement. Thus, in publications of that time, middleware was defined as an infrastructure within which the development of distributed applications is possible, a middleware service – the APIs and protocols it supports and sits between the platform and the application. Middleware components are generic across applications and industries, run on multiple platforms, are distributed, and support standard interfaces and protocols. Middleware that is transparent with respect to a standard API is more easily accepted by the market, and applications using an existing API can use the new service without modification. Providing a set of standardized, reusable functionalities that can be leveraged to build more complex and sophisticated systems was the essence of the service concept.

Below are multiple application scenarios for middleware systems within extensive scientific projects:

- secure data sharing and transfer between organizations;
- high-performance data movement and processing;
- authentication and authorization of users and their data access;
- high-throughput computing, job scheduling and resource allocation for computing-intensive research;
- access to distributed resources for scientific workflows and simulations;
- integration of various middleware systems for a seamless, user-friendly interface;
- data staging, synchronization, and management for distributed applications;
- in-memory data processing, querying, and distributed caching and integration with big data frameworks for near-real-time analytics.

As the demand for interoperability between different platforms and technologies grew, open standards-based middleware solutions emerged. One such solution was CORBA (Common Object Request Broker Architecture, 1991), which provided a way for different software systems to communicate using a standardized interface. For example, the U.K. Distributed Aircraft Maintenance Environment (DAME) project that was set up in 2004 and applied grid technologies to the challenging problem of computer-based fault diagnosis, utilized CORBA as a middleware to support intercomponent communication among different software modules. In particular, DAME was working to diagnose faults in Rolls Royce aircraft engines, based on sensor data recorded at the rate of one gigabyte per engine per transatlantic flight. Additionally, Java RMI¹¹ was used for communication between distributed Java objects,



¹¹Java RMI – Remote Method Invocation – Java-specific technology relies on Java interfaces and allows Java objects to invoke methods on objects located on remote machines

and SOAP¹² – to enable remote collaboration and data sharing among its various stakeholders. Web Services enabled integration and interoperability of various components. The combination of these middleware technologies enabled DAME [55] to provide a flexible and scalable distributed environment for aircraft maintenance that improved efficiency and reduced downtime.

CORBA was a powerful middleware architecture that filled an important niche in distributed systems and object-oriented programming, but it was not well-suited to the needs of grid computing, and was outmatched by simpler, more focused solutions that were designed specifically for this field [61]. The reasons behind the waning popularity of CORBA are extensively examined in the publication “The Rise and Fall of CORBA: Valuable Lessons from its Mistakes” [52].

Other popular middleware solutions that emerged in the late 1990s and early 2000s were focused on various use-cases, some were designed to meet the needs of High-Throughput Computing (HTC), other – for High-Performance Computing (HPC), and there were even middleware developed specially for the in-memory and volunteer distributed computing.

Below we discuss several examples of general-purpose middleware systems. However, there are a lot of more specialized middleware, developed for more specific purposes: for data management, job scheduling, resource discovery, etc.

5.1. HTCCondor



Condor is a middleware system that focuses primarily on high-throughput computing (HTC), particularly for running large-scale batch jobs on a network of distributed computing resources.

The first version of Condor was released in 1988, and it quickly gained popularity among researchers and scientists who needed to run large-scale batch jobs on a network of distributed computing resources. “*Condor – A Hunter of Idle Workstations*” – that is the title of the program paper, describing the system design of the first version of Condor and its main principles [58]. The motivation of Condor was to maximize the utilization of workstations interconnected by high capacity networks with as little interference as possible between jobs it schedules and the activities of the people who own these workstations.

Figure 4 demonstrates the initial structure of Condor’s scheduler. Each machine had a local scheduler and background job queue for user job submissions. A central workstation housed a central coordinator, local scheduler, and background job queue. The central coordinator polled stations every two minutes to allocate remote cycles and manage pending background jobs. Local schedulers monitored station capacity and preempted background jobs for local user activity. Idle workstation capacity was allocated by the central coordinator to local schedulers with pending jobs, with local schedulers making decisions on job prioritization if multiple background jobs were waiting. Condor offered a comprehensive set of features including job management, scheduling policies, priority schemes, resource monitoring, and resource management. What set Condor apart from other so-called Resource Management Systems (RMS), was its unique architecture and mechanisms, enabling outstanding performance in environments where traditional RMS systems struggled – such as sustained high-throughput computing and opportunistic computing. In a high-throughput computing environment, the primary objective is to efficiently utilize all available network resources to provide computational power over extended periods with fault

¹²SOAP – Simple Object Access Protocol – protocol-independent, XML-based messaging protocol that is used for exchanging structured information in the implementation of web services

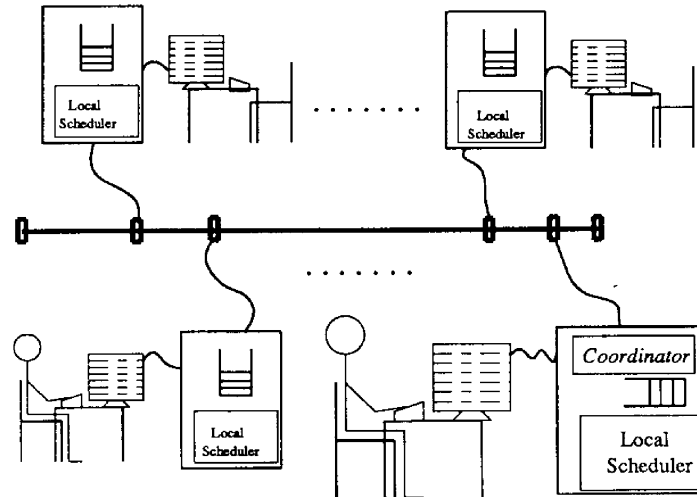


Figure 4. The Condor Scheduling Structure [58]

tolerance. Opportunistic computing, on the other hand, focuses on resource utilization whenever they are available, without requiring 100% resource availability. These two goals naturally complement each other. And Condor’s remarkable flexibility lied in its ability to seamlessly integrate High Throughput Computing (HTC) with volunteer computing, effectively harnessing idle resources. For example, consider the capability Condor offered to run jobs solely on desktop workstations during idle periods when both the keyboard and CPU remained idle. In the event of a user keystroke while a job was ongoing on a workstation, Condor adeptly shifted the job to another workstation, ensuring a smooth continuation from the point where it was interrupted. Additionally, Condor’s impressive mechanics enabled preemptive-resume scheduling for dedicated compute clusters.

This vital functionality empowered Condor to handle priority-based scheduling on clusters with utmost efficiency. Whenever a node in a dedicated cluster remained idle without a task, Condor could opportunistically allocate it for computing purposes.

Over the years, Condor had undergone several iterations and enhancements to keep up with changing computing requirements. In the early 2000s, with the growing interest in distributed computing and the need for such computing systems to support more diverse applications, the developers of Condor implemented a new version called Condor-G (where G stands for Grid) [45]. They added many features that made it easier to use Condor with grid standards such as the integration with Globus Toolkit services (Grid Resource Allocation and Management (GRAM), Globus Toolkits Global Access to Secondary Storage (GASS), Globus Toolkits Grid Security Infrastructure (GSI)), which allowed it to access remote computing infrastructures and provided users with an integrated and transparent interface to use grid resources.

Condor-G demonstrated its strengths when four mathematicians from Argonne National Laboratory, University of Iowa, and Northwestern University used it and several other technologies to solve a problem known as NUG30 challenge. The paper “Distributed Computing in Practice: The Condor Experience” describes this challenge as follows: “...a solution to NUG30 was discovered utilizing Condor-G in a computational run of less than one week. During this week, over 95,000 CPU hours were used to solve the over 540 billion linear assignment problems necessary to crack NUG30. Condor-G allowed the mathematicians to harness over 2500 CPUs at ten different sites ... spanning eight different institutions” [80]. Overall, the winning team

was able to achieve a 15x speedup compared to the NERSC baseline solution and a 6x speedup compared to the fastest solution submitted in the challenge prior to theirs.

Condor-G actively participated in various prominent projects, including the Grid Physics Network (GriPhyN), the International Virtual Data Grid Laboratory (iVDGL), the Particle Physics Data Grid (PPDG), the NSF Middleware Initiative (NMI), the TeraGrid, and the NASA Information Power Grid (IPG). Furthermore, as a founding member of the National Computational Science Alliance (NCSA) and a close collaborator of the Globus project, Condor-G played a pivotal role in advancing computational science and grid technologies.

In late 2012 Condor-G was upgraded and modernized to HTCondor (High-Throughput Computing Condor)¹³. HTCondor builds on the features and capabilities of Condor, adding new functionalities, performance improvements, and better support for current computing environments and architectures. Since the transfer, HTCondor has become a widely adopted solution for managing large-scale scientific workloads in the United States and around the world.

5.2. NIMROD – HPC in Plasma Physics

NIMROD, developed in 1995, was a tool for performing parameterized simulations over networks of loosely coupled workstations [29]. Using NIMROD the user could interactively generate a parameterized experiment. NIMROD then controlled the distribution of jobs to machines and the collection of results. In this context, this middleware was highly effective for research studies with users who have varying degrees of parallel programming skills. It was more suitable for individual researchers and small research groups who require access to parallel computing resources, but not for larger organizations and collaborations with complex workflows, data management needs, and secure access requirements.

In 2005 NIMROD was modernized to NIMROD/G (for Grid) [24] by a team at the Los Alamos Laboratory in the US. The project utilized funding from the Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) program. It became a highly specialized Grid middleware that had been designed specifically to meet the needs of the plasma physics simulation: it included functionality for modeling magnetohydrodynamics (MHD) and other relevant physical phenomena, a number of custom plugins and modules that were specifically tailored to the needs of plasma physicists and it used a custom job submission and management system that was optimized for large-scale, long-running simulations [49]. It provided a framework for running parallel processing jobs across multiple heterogeneous computing resources, as well as managing data transfer between those resources. While it might not be as broadly applicable to other scientific domains as some other middleware systems, it was highly effective for the specific use cases for which it was developed.

While NIMROD/G is no longer actively maintained, it was used in a variety of scientific and engineering projects over the years: DIII-D tokamak experiment at General Atomics in San Diego, California, USA; National Spherical Torus Experiment (NSTX) at Princeton Plasma Physics Laboratory in New Jersey, USA; Korea Superconducting Tokamak Advanced Research (KSTAR) experiment at the National Fusion Research Institute in Daejeon, South Korea; ITER project – an international collaboration to build a fusion energy prototype reactor in Cadarache, France.

¹³<https://htcondor.org/>

5.3. Globus Toolkit



Globus Toolkit is a middleware system specifically designed for distributed data-intensive scientific research applications, which require the transfer and management of large volumes of data across geographically distributed resources. In 1996 the Globus Alliance was formed to conduct R&D for the technology, standards and systems that form the grid. Alliance members eventually produced open-source software Globus Toolkit (GT) that evolved out of trying to solve real problems in real projects. It was a community-based, open-source set of services and software libraries that support grids [41].

The first version of the Globus Toolkit (GT1) emerged in the late 1998, during a time when the concept of grid computing was just beginning to emerge. The main functionality of GT1 was to provide a software infrastructure that could enable scientists and researchers to share distributed computing resources such as computing power and data storage over the Internet. GT1 was a relatively simple toolkit, with a limited set of functionality, focused mainly on providing low-level mechanisms for distributed computing, such as security and authentication mechanisms, resource discovery, and job submission and management.

The Globus Toolkit version 2 (GT2) emerged in 2002, and was a significant upgrade to GT1. GT2 was designed to make it easier to build and manage grid systems by providing a standardized set of middleware services and tools for building grid applications. GT2 introduced many new features and improvements over its predecessor GT1, including better support for authentication and authorization, support for multiple security mechanisms, and improved data transfer performance. It was widely adopted by many grid communities and projects, including the European DataGrid (EDG) and the GriPhyN project, among others.

In general, the four main GT protocols were:

- *Grid Security Infrastructure (GSI)* – authentication, authorization, policy, delegation. GSI had become a standard for grid security. It was based on the free SSLeay package and used X.509. It enabled “Single sign-on” (SSO) on the grid, user identity was guaranteed with a single certificate that helped to avoid frequent logins into various resources with different passwords;
- *Grid Resource Allocation Management (GRAM)* – for the remote allocation, reservation, monitoring, control of compute resources;
- *Grid Resource Information Service Protocol (GRIS)* – provided access to structure and state information, and
- *Grid File Transfer Protocol (GridFTP)* – high-performance data access and transport protocol.

The main challenge of the GT2 was its applicability for business processes: it was too much like a distributed batch system. New technologies required new concepts of grid middleware. Instead of functioning as a distributed batch system, it was supposed to be a network of services, small programs with standardized interfaces, geographically distributed. So, to meet the new requirements, a GT3 version was implemented in 2004, which was based on grid services.

In 2006, GT4 [39] was introduced with the addition of Web Services, XML-based mechanisms for describing, discovering, and invoking network services. GT4 provided a set of infrastructure services that implemented interfaces for managing computational, storage, and other resources. In many Globus deployments, such as TeraGrid, Open Science Grid, Cancer Bioinformatics Grid (caBIG), EGEE, LHC computing Grid, UK National Grid Service, China Grid, China National

Grid and NAREGI, these services were deployed to support a range of different application communities.

Effective use of high-speed networks for research required breaking the usability barriers that impede network use by non-expert users. Thus, the Center for Enabling Distributed Petascale Science (CEDPS) project launched the Globus Online project in 2009 [40], enabling the reliable high-performance research networking for the masses. Globus Online was a cloud-based service that provided data management and transfer capabilities to researchers and organizations who require data-intensive computing. If Globus Toolkit provided resource federation as a service, Globus Online was the cloud-hosted file transfer service, or software-as-a-service (SaaS) provider, taking the responsibility for managing the end-to-end data transfer processes, performed via GridFTP.

In 2018, after two decades of being a de-facto standard for grid applications, the Globus Toolkit, announced the end of support of the open-source project. The decommissioning of the Globus Toolkit affected a number of grid computing projects that relied on its technology. In response to the end-of-support of the Globus Toolkit the Grid Community Toolkit (GCT) was created¹⁴. It is an open-source fork by the Grid Community Forum (GCF) of the venerable Globus Toolkit created by the Globus Alliance. The GCT is derived from the Globus Toolkit, but is not the Globus Toolkit. Further, the GridCF is not a part of the Globus Alliance. Through this initiative, the following objectives are sought:

- Ensure the preservation of vital features, such as the Grid Security Infrastructure (GSI) and an open-source GridFTP implementation, by maintaining their quality and security.
- Foster collaboration among various grid organizations to distribute the responsibility of supporting these critical functionalities.
- Create an inclusive and open platform where external contributors can actively participate and enhance the software.

5.4. UNICORE

UNICORE (Uniform Interface to Computing Resources, 1997) [36] is an open-source middleware that provides a job submission and management system for grid computing.



Originally this project was initiated in the HPC domain. It is the project of the German Ministry for Research and Education (BMBF). Initial aim was in creating a network of supercomputing centres and providing a uniform interface while using existing technologies. UNICORE middleware was developed to meet

the specific requirements of complex distributed data-intensive applications that require high-performance computing environments. The implementation was based on Java and Java applets that ensure high portability. UNICORE was initially designed as a monolithic system, where various modules such as authentication, authorization, job submission, and data transfer are all tightly coupled. UNICORE provided a more restrictive job submission model where users must first register their application on the server.

UNICORE passed through many stages in the evolution:

- Initial focus on applications across multiple disciplines in Europe in 2001, driven by research projects sponsored by the European Commission. UNICORE saw deployments in

¹⁴<https://gridcf.org/gct-docs/latest/index.html>

several European high-performance computing centers, the creation of a High-Level API, and the introduction of interactive access to HPC systems through UNICORE.

- Transition to open-source in 2004 when the UNICORE software was made available under the BSD License via a SourceForge project.
- Shift towards Web Services-based architecture in 2007, followed by a focus on Data Oriented Services in 2011, emphasizing high-speed file transfers and enhancing support for data management and data-intensive applications within UNICORE.
- Major updates with UNICORE 7 in 2014, UNICORE 8 in 2020, introducing significant improvements.
- Ongoing development culminating in UNICORE 9 in 2022, representing a fully REST-based system.

Compared to the Globus Toolkit, UNICORE provides a more integrated and comprehensive interface to grid resources including an advanced security system that can support the access control policies and data privacy requirements of complex distributed applications. UNICORE middleware was designed for European grid infrastructures but is now used around the world. It was integrated with such highly-recognized scientific grid projects like, D-Grid and Human Brain Project.

One of the flagship UNICORE projects is Jupyter-JSC¹⁵, so-called “Supercomputing in Your Browser” (Jlich Supercomputing Centre (JSC)). It combines interactive supercomputing with UNICORE, offering researchers and analysts an efficient platform for data research, analysis, and visualization. Jupyter’s open-source, web-based design allows for diverse workflows and programming methods in a single interface. JupyterHub enables multi-user functionality, making it suitable for supercomputing centers. UNICORE ensures data access by integrating tools like XUADB¹⁶, UNICORE/X¹⁷, TSI¹⁸, allowing users to spawn Jupyter applications with their HPC accounts. At the JSC, Jupyter-JSC provides direct web access to start and connect Jupyter or JupyterLab. UNICORE manages jobs and provides reliable access and information output.

5.5. ARC

Advanced Resource Connector (ARC)¹⁹, in the past called NorduGrid middleware, is a middleware system established in 2002 to provide a management framework for distributed computing resources in the Nordic countries [57]. Scientific and academic computing in the Nordic countries exhibited a distinct characteristic with a multitude of small and medium-sized facilities, each differing in nature and ownership.



Considering this, the development of ARC had prioritized the need for a portable, compact, and manageable middleware solution with a focus on interoperability between different software and hardware systems that caters to both server and client sides. Notably, the first stable client version of ARC package occupied only 14 megabytes and was compatible with most Linux distributions, allowing installation at any accessible location by nonprivileged users. And for computing service, only three main processes are needed: file transfer service, Grid Manager and Local Information Service [3].

¹⁵<https://jupyter.jsc.fz-juelich.de/>

¹⁶UNICORE user database, manages access to resources

¹⁷Server, the main component, interacts with XUADB for authentication purposes

¹⁸Target System Interface – interface of the execution environment, launches tasks on the target system

¹⁹<https://www.nordugrid.org/arc/arc6/index.html>

ARC is designed to manage the execution of large numbers of independent tasks (HTC workloads), such as those commonly found in scientific computing and data-intensive research applications. However, it can be used to manage HPC workloads as well. Since 2002, ARC has continuously supported production-level systems and demonstrated its exceptional performance in demanding High Energy Physics (HEP) computing tasks. In fact, it is the pioneering middleware that has successfully provided services for such a massive global data processing endeavor.

Being a part of HEP distributed environment in ATLAS experiment at the LHC, ARC introduced several new features and possibilities in the traditional ATLAS computing model. Firstly, ARC allowed for pilot jobs that come with pre-cached input files, reducing the need for data transfer during job execution. Secondly, ARC enabled automatic migration of jobs between different sites, improving job allocation and resource utilization. Thirdly, it allowed for the integration of remote sites that do not have direct storage connectivity, thus expanding the pool of available computing resources. Next, ARC provided automatic brokering for jobs with specific resource requirements, facilitating efficient job allocation. Additionally, ARC offered an automatic data transfer model that allowed computing sites to participate in ATLAS' global task management system without centralized brokering or data transfer services. Finally, ARC included a powerful API with Python and Java bindings, making it easy to build new services for job control and data transfer [38].

The middleware leveraged well-established open source solutions such as OpenLDAP, OpenSSL, SASL, and the Globus Toolkit 2 (GT2) libraries. ARC introduced pioneering solutions that were crucial for a reliable and high-performing middleware, including the Grid Manager, ARC GridFTP server, information model and providers, User Interface and broker, an extended Resource Specification Language (xRSL), and a comprehensive monitoring system. These innovations provided by ARC enhanced the functionality and effectiveness of the middleware.

5.6. BOINC – Volunteer Computing



BOINC is a platform for Volunteer Computing (VC) that was developed in 2004. It is a client-server desktop grid middleware that has grown to power very large computational projects. BOINC clients request computing jobs to a central server and run them alongside other regular applications [11].

“Volunteer computing” is the use of consumer digital devices for high-throughput scientific computing. It can provide large computing capacity at low cost, but presents challenges due to device heterogeneity, unreliability, and churn. VC is a type of computing architecture that is best suited to High-Throughput Computing (HTC) workloads, where the goal is to complete a high volume of jobs rather than focusing on low job turnaround time. However, VC is less well-suited for workloads that require high memory or storage requirements, or where the ratio of network communication to computing is very high.

Compared to other forms of HTC, VC presents unique challenges that must be addressed by the platform. For example, VC computers are anonymous, untrusted, inaccessible, and uncontrollable. They may behave unpredictably and cannot be punished or stopped. Additionally, VC computers are heterogeneous in all hardware and software dimensions, which requires either multiple application versions or the use of virtualization and makes it challenging to estimate job

runtimes. Another challenge of VC is creating the resource pool, which involves recruiting and retaining volunteers. This requires incentive features such as teams, computing credit accounting, and screensaver graphics. Finally, the scale of VC is much larger, with potentially millions of computers and millions of jobs per day. Therefore, the server software must be efficient and scalable. BOINC platform takes into account all these challenges.

SETI@Home (1999 – 2020) was one of the largest projects that used BOINC as a middleware [12]. This project truly popularized distributed computing and showed that it could work. SETI@Home was an effort by the Search for Extraterrestrial Intelligence (SETI) at the University of California at Berkeley. The project was started in 1999 to analyze the radio telescope signals. There were over three million users who volunteered their idle computing resources in the search for extraterrestrial intelligence. Anyone who has an Internet connection and some spare CPUs can participate by running a free program that analyzes radio telescope data. The project was retired on March 31, 2020, after over 20 years of operation. The data collected by SETI@Home is now being archived for future analysis by astronomers.

Below, there are some other examples of grid projects that used BOINC middleware:

- **SimGrid (1999 – present time)** [26] – a distributed computing project that provides tools and libraries for distributed computing research, such as grid and cloud computing, high-performance computing, and peer-to-peer networks. The project provides a platform for researchers to model, simulate, and analyze various strategies for large-scale distributed systems. SimGrid runs on various computing systems.
- **Folding@home (2000 – present time)** [17] – a distributed computing project that simulates protein folding, misfolding, and related diseases. The project is run by a consortium of universities that has built one of the world’s largest and most powerful supercomputers, with the help of over 4 million volunteers who donate their unused computing power for research. The project consists of tens of thousands of CPUs and GPUs from around the world.
- **Climateprediction.net (2003 – present time)**²⁰ – a distributed project that uses global climate models to predict the possible impacts of climate change on the earth. The project uses hundreds of thousands of volunteers’ computers to run simulations of the atmosphere, ocean, and land surface. The project has been used to produce some of the most extensive surveys of the effects of climate change to date, and it has resulted in numerous published research papers.
- **Rosetta@home (2005 – present time)**²¹ – project focused on protein structure prediction and designs of new proteins to aid computational drug discovery. The project uses over 600,000 CPUs and GPUs from around the world.
- **Einstein@home (2005 – present time)** [8] – searching for gravitational waves from black holes, pulsars and other objects in the universe. The project uses data from the Laser Interferometer Gravitational-Wave Observatory (LIGO) detectors. It has distributed over 30 million work units to volunteer computers and currently uses over 200,000 devices to process data.
- **MilkyWay@home (2006 – present time)** [33] – simulations of the formation and evolution of galaxies, including the Milky Way. The project uses volunteer computing

²⁰<https://climateprediction.net/>

²¹<https://boinc.bakerlab.org/>

power for parameter studies, such as studying the effects of dark matter. MilkyWay@Home runs on over 100,000 CPUs and GPUs contributed by volunteers around the world.

- **GPUGRID.net (2007 –present time)**²² – a distributed computing project focused on molecular dynamics simulations. It performs pharmacological simulations, including simulations of protein-ligand interactions and drug design. The project runs on more than 5,000 GPUs from volunteers worldwide.
- **Asteroids@home (2008 – present time)** [82] – searching for unknown asteroids and comets by processing astronomical data using image-recognition algorithms. The project aims to assist professional astronomers in discovering asteroids that could pose a threat to our planet. Currently, the project runs on about 8,000 devices contributed by volunteers worldwide.

5.7. Specialized Middleware Systems in High-Energy Physics Projects

Grid middleware systems, like Globus Toolkit, Condor, UNICORE, ARC, gLite, play a vital role in enabling efficient data management, job execution, resource utilization, and collaboration in grid applications. The specific middleware systems utilized may vary depending on the requirements and objectives of the grid projects. In various projects, different combinations of middleware systems are used based on the specific needs and goals of each project.

For example, there were several different middleware packages available in the D-Grid project: Globus Toolkit, UNICORE, LCG/gLite, GridSphere and the Grid Application Toolkit (GAT). LIGO Data Grid utilized a combination of Globus Toolkit and Condor-G services. The widest variety of middleware systems, including custom middleware services, is in the WLCG – the largest grid infrastructure in the world. There are several scientific experiments connected to the WLCG: ATLAS, ALICE, CMS and LHCb – these are the largest international high-energy physics projects that brought notable contributions in the development of elementary physics. In the realm of experimentation, middleware technologies are utilized across various experiments, with some similarities and differences in their deployment as per specific experiment requirements.

The distinctive data handling and processing demands of each of the experiments conducted at the LHC have given rise to the development of separate distributed frameworks for data management, job submission, and execution by each respective experiment team:

- **PanDA (Production and Distributed Analysis)** is a workflow management system developed by the ATLAS experiment at CERN. It is designed to manage large-scale data processing and analysis tasks, and can be used with a variety of computing resources including grid, cloud, and HPC systems [60].
- **Rucio Distributed Data Management system** – a primary distributed data management system in ATLAS since 2014. Rucio is responsible for the allocation and placement of files across distributed storage systems, data transfer monitoring and optimization, and authentication and authorization of user access to data. Rucio also provides ATLAS with a centralized data catalog, where the metadata of all datasets and files are stored and indexed, making it easy for users to access the data they need. Furthermore, Rucio supports data preservation and long-term archiving, ensuring that the data produced by ATLAS remains accessible and usable for future scientific studies [47].

²²<https://gpugrid.net/>

- **DIRAC (Distributed Infrastructure with Remote Agent Control)** is a middleware system developed by the LHCb experiment at CERN. It is designed to manage distributed data processing and analysis tasks, and can be used with a variety of computing resources including grid, cloud, and HPC systems. It includes features such as job queuing, job monitoring, and data replication [25].
- **ALIEN (ALICE Environment)** is a distributed computing system developed by the ALICE experiment to manage large-scale data processing and analysis tasks, and can be used with a variety of computing resources including grid, cloud, and HPC systems. Some of its key features include support for user authentication and authorization, job prioritization, and data management [75].
- **Grid Control System (GlideinWMS)** is the primary system used for job submission and management on the grid in the Compact Muon Solenoid (CMS) experiment. It allows users to submit and monitor jobs to various computing resources around the world, including those provided by the WLCG. GlideinWMS is often used for large-scale Monte Carlo simulations, where virtual particles are generated and their interactions with the detector are simulated to predict the outcome of physical processes [76].

6. Instances of Scientific Grids

Numerous scientific grids have emerged since the beginning of 2000s. Each initiative has its own specifics and the scale of computing resources. In this section the largest and the most interesting grids are described.

TeraGrid (2001 – 2011) [27] was a collaboration between multiple supercomputing centers in the United States. In the beginning of 2000s it was the most ambitious high-performance grid project in the United States (Fig. 5²³). It offered over a petaflop of total compute capabilities and many different services and gateways to thousands of US scientists. The genesis of the TeraGrid project was the Distributed Terascale Facility (DTF). The DTF was conceived in October 2001 as a distributed high-performance computing grid that could link supercomputers as well as some specialized resources for data management and visualization. As of July 2006 the range of scientific disciplines in TeraGrid was from molecular bioscience, physics, astronomical sciences and chemistry to the neuroscience, communication systems and environmental biology. Until the beginning of the Large Hadron Collider (LHC) project, TeraGrid was the worlds largest, most comprehensive distributed cyberinfrastructure for open scientific research. The project was finished in 2011.

GridPP (2001 – still ongoing)²⁴ is a community of particle physicists and computer scientists based in the United Kingdom and at CERN. Initially it was formed to cater for the substantial computing demands of the Large Hadron Collider experiments, representing the UK in the Worldwide LHC Computing Grid (WLCG). As the GridPP community has grown and evolved, however, many user communities from a wide range of disciplines have taken advantage of the computing resources offered by GridPP.

The institutes that form the GridPP Collaboration are organized as follows: The UK Tier-1 center, which is hosted at the Rutherford Appleton Laboratory (RAL), 18 UK university-based

²³https://www.nsf.gov/news/mmg/media/images/teragrid_h.jpg

²⁴<https://www.gridpp.ac.uk/>

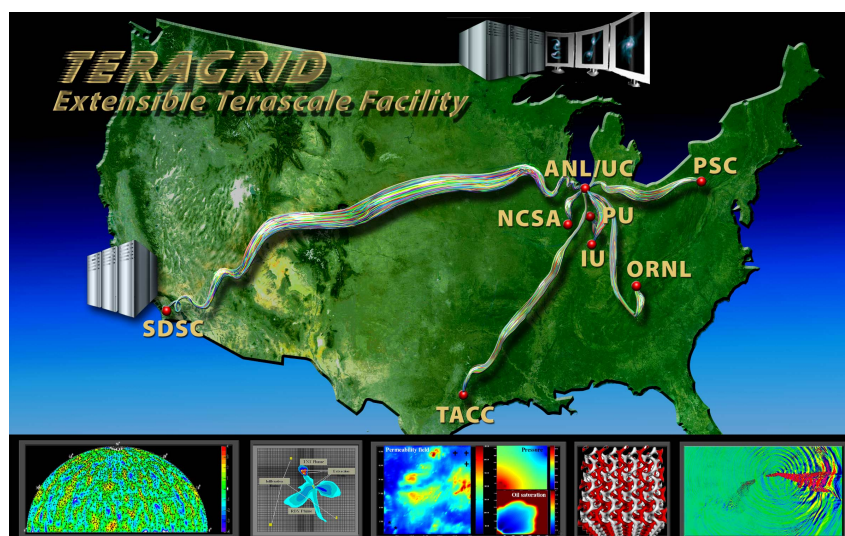


Figure 5. TeraGrid Members



Figure 6. GridPP Collaborating Institutes

Tier-2 sites, which are grouped into four Regional Tier-2s (LondonGrid, NorthGrid, ScotGrid, SouthGrid)²⁵.

As of the beginning of 2012, these GridPP sites collectively offer approximately 30,000 logical CPUs and around 29 PB of disk-based storage. The Deployment Board (DB) governs both the Tier-1 center and the regional Tier-2s²⁶.

GridPP supports research in physics (LSST, Fusion, Supersymmetry, LUCID, MAGIC) and particle physics (LHC, H1, BaBar, D0, SNO+), and other sciences (BioMed, EPIC, PRaVDA, Proteomics and Phylogenomics at QMUL, WISDOM).

²⁵<https://www.gridpp.ac.uk/about/collaborating-institutes/>

²⁶<https://www.gridpp.ac.uk/infrastructure/>

International Gravitational Wave Observatory (IGWN) Computing Grid (2002 – still ongoing) [9] is a distributed, high-throughput computing infrastructure based on HTCondor middleware. It facilitates the processing of scientific workflows across a geographically diverse set of resources. These resources include dedicated computing resources provided by IGWN member groups, as well as opportunistic resources from external partners. The grid is connected through a support infrastructure maintained in collaboration with the Open Science Grid (OSG). It also offers utilities for remote access to data and services, enhancing the capabilities of researchers and enabling collaboration within the IGWN community²⁷.

The LIGO project uses interferometric detectors to search for gravitational waves, ripples in space-time caused by the acceleration of massive objects. LIGO has been operational since 2002, with the first round of observations running from 2005-2007. The second round of observations began in 2015, and has resulted in several major gravitational wave detections, including the first observations of binary black hole mergers and binary neutron star mergers.

Actually, there are three detectors in the infrastructure of gravitational-wave astronomy: LIGO (United States), Virgo (Italy and France) and Kamioka Gravitational Wave Detector (KARGA, Japan). To successfully detect gravitational waves, a robust computational framework is necessary to not only support the detectors themselves but also the data analysis pipelines that are employed to locate and define gravitational wave signals. Virgo, LIGO and KAGRA collaborations joined efforts to move from partially interoperable owned computing resources to a wholly shared common computing infrastructure IGWN by adopting common tools and interfaces.

The Gravitational Wave community is ushering in a new era of computing with several transformative features. These include achieving full interoperability between the Virgo, LIGO, and KAGRA observatories, establishing a common and sustainable computing environment, standardizing a uniform runtime environment for offline pipelines, embracing scalability and heterogeneous resource utilization, and adopting mainstream, widely-used tools. These advancements pave the way for enhanced collaboration, increased computational efficiency, and optimized resource utilization, ultimately propelling gravitational wave research towards new frontiers of discovery.

Worldwide LHC Computing Grid (WLCG, 2002 – still ongoing) [22]: high energy physicists designing the LHC were the first to realize that they needed grids to federate computing systems at hundreds of sites to analyze the many petabytes of data to be produced by LHC experiments. Thus, they launched the EU-DataGrid (2001) project in Europe, the Particle Physics Data Grid (PPDG, 2002) and Grid Physics Network projects in the US – these two efforts led to the creation of the Open Science Grid (OSG, 2003) in the US, EGEE (Enabling Grids for E-science, 2004) and then EGI (European Grid Infrastructure, 2008) in Europe. The international LHC Computing Grid (LCG, 2006), which was later renamed to the Worldwide LCG Computing Grid (WLCG). This was done to better reflect the global scale and collaborative nature of the project.

WLCG is the largest grid infrastructure. At an early stage of the development of the LHC computing model (2000s), it was decided to combine the existing and newly constructed computing centers (more than 200) into a distributed data processing center and to do this in such a way that physicists from universities and scientific organizations of the participating countries

²⁷<https://computing.docs.ligo.org/guide/dhtc/>

would have equal opportunities to analyze information. As a result of the work of physicists, scientists, and IT engineers, WLCG was created (Fig. 7)²⁸.



Figure 7. WLCG sites

Today the WLCG is the largest academic distributed computing network in the world, consisting of about 300 computing centers in 42 countries. More than 10K scientists use these centers to analyze LHC data in search of new physics phenomena. In the WLCG, up to 3M physics jobs run daily, the total storage space exceeds 1 EB, the data processing results are archived, distributed between data processing and analysis centers, and go directly to the physicists workplace. Such a system can be compared with a huge computing complex the nodes of which are connected by high-speed Internet. Data transfer rates between centers are up to 20 GB/s (average value during a day and 35 GB/s at peak). Now the WLCG consists of four tiers: 0, 1, 2, and 3.

- *Tier 0:* CERN Data Center, responsible for data archiving, reconstruction, and reprocessing during LHC down-times.
- *Tier 1:* 13 large centers with CPU and storage capacity, providing round-the-clock support. They store data, perform reprocessing, distribute data to Tier 2s, and store simulated data.
- *Tier 2:* Universities and scientific institutes that store data and provide computing power for analysis tasks. There are around 170 Tier 2 sites globally.
- *Tier 3:* Computing facilities in universities and research labs, retrieving data from Tier 2 for processing and analysis.

WLCG has mesh structure where data centers are interconnected, allowing for efficient distribution of workloads and computing resources. This mesh structure provides high redundancy and robustness. In case of a data center failure, workloads can be automatically rerouted to other nodes in the network, ensuring continuous availability of services without major disruption.

Russian Data Intensive Grid (RDIG, 2003 – still ongoing) [21] is a national-level initiative in Russia that aims to facilitate and support large-scale data-intensive research projects. Initially it was established at the initiative of the Kurchatov Institute National Research Centre and the Joint Institute for Nuclear Research in Russia to participate in the distributed data processing of experiments at the LHC. Later, its objective evolved beyond high-energy physics to encompass a broader range of scientific disciplines, including astrophysics, bioinformatics, and climate modeling, among others.

²⁸<https://wlcg.web.cern.ch/using-wlcg/monitoring-visualisation>

RDIG connects the largest Russian scientific centers, such as IHEP (Institute for High Energy Physics), IMBP RAS (Institute of Mathematical Problems in Biology), ITEP (Institute of Theoretical and Experimental Physics), JINR (Joint Institute for Nuclear Research, Dubna), KIAM RAS (Keldysh Institute of Applied Mathematics), PNPI (Petersburg Nuclear Physics Institute), NRC KI (Kurchatov Institute, Russia's National Research Center), SINP MSU (Skobeltsyn Institute of Nuclear Physics at Moscow State University), SPBU (Saint Petersburg State University), NovSU (Novgorod State University).

Now RDIG is an important component of the IT infrastructure being developed in Russia, which ensures the functioning of mega-science projects, including providing storage for the national genetic information database and biorepository centers - key elements of the genetic research infrastructure.

International Lattice Data Grid (ILDG, 2003 – still ongoing) [56] is a global collaboration of lattice quantum chromodynamics (QCD) groups. It is designed to provide a unified framework for storing and sharing lattice QCD data sets in a secure, efficient, and standardized manner. ILDG initially emerged as a federation of interoperable yet independently operated infrastructures and services, often referred to as a “Grid of Grids” It consists of several regions (Australia, Japan, Continental Europe, UK, US). Each regional grid has the autonomy to implement and manage its services in a manner compatible with ILDG standards. For instance, the regional grids in Japan and Europe use rather different solutions for data storage: a global file system (GFARM) in case of Japan Lattice Data Grid (JLDG) [26], and several (distributed) storage elements with SRM interfaces being part of the Worldwide LHC Computing Grid in case of European Latfor DataGrid (LDG). One crucial responsibility of the regional grids is to acquire the required storage resources for their data.

ILDG's primary role is to facilitate community-wide user registration, promoting collaboration among its members. Additionally, ILDG maintains two working groups tasked with developing ILDG-wide metadata schemata and APIs for the regional grid services. Through these standardized specifications, data sharing and metadata searching become seamless and accessible across the entire ILDG community.

Several large LQCD collaborations in Japan have been working on QCD simulations using super-computers²⁹.

European LDG operates at the following institutions: DESY (Hamburg and Zeuthen, Germany), IN2P3 (Lyon, France), INFN (Bologna, Italy), JSC (Jülich, Germany), RUG (Groningen, Netherlands), SARA (Amsterdam, Netherlands)³⁰.

UK Lattice Field Theory region currently comprises 35 academics from 10 different UK institutions (Swansea, Glasgow, Plymouth, Liverpool, Southampton, Edinburgh, Imperial, Cambridge)³¹.

US Region operates at the resources of Jefferson Lab LQCD (JLAB), Fermilab Lattice QCD Facility, BNL clusters for USQCD.

Naregi National Research Grid Initiative, NAREGI (2003 – 2012) [68] is one of the major Japanese national IT projects to develop and use state-of-the-art, high-performance general-purpose supercomputers promoted by the Ministry of Education, Culture, Sports, Science and Technology. The Computational Nanoscience Center at the Institute for Molecular Science is actively utilizing grid computing technologies to conduct cutting-edge research in the

²⁹<https://www.jldg.org/system.html>

³⁰<https://hpc.desy.de/ldg>

³¹<https://generic.wordpress.soton.ac.uk/uklft/>

field of nano-science and nanotechnology simulation applications. This research aims to facilitate the discovery and development of novel materials and advanced nano-devices for future generations.

Enabling Grids for E-science (EGEE, 2004 – 2010) [46]. The main purpose of EGEE was to create a sustainable pan-European grid infrastructure to support scientific research and collaboration across different disciplines. EGEE involved participation from numerous countries across Europe and beyond. Some of the countries that participated in the project included Belgium, France, Germany, Italy, Netherlands, Spain, Switzerland, United Kingdom, Greece, Portugal, Russia, Taiwan, and many more. The project comprised a network of data centers that collectively formed a powerful computing grid. The exact number of data centers evolved over time as new institutions joined the project. EGEE shared its grid infrastructure with LCG, and the grid operations activity within EGEE had a substantial overlap with LCG operations.

EGEE encompassed a global infrastructure consisting of approximately 200,000 CPU cores, hosted collaboratively by over 300 centers worldwide. Throughout the duration of the project, around 13 million jobs were executed on the EGEE grid every month.

EGEE was completed in 2010 with the establishment of its successor project, the European Grid Infrastructure (EGI). This transition was driven by the need for a more sustainable and operational model for grid infrastructure.

Distributed European Infrastructure for Supercomputing Applications (DEISA, 2004 – 2011) [48] was a project funded by the European Commission to create a European-wide distributed supercomputing infrastructure for research in science and engineering. The project aimed to provide researchers with access to high-performance computing resources and expertise that they otherwise would not have had access to. The project involved 11 European supercomputing centers in seven countries, which were connected by a high-speed network. Prominent scientists throughout Europe are taking advantage of the combined power of supercomputers and corresponding global data management infrastructures in an effective and convenient manner. The emphasis is particularly on grand-challenge applications related to key scientific fields including material sciences, climate research, astrophysics, life sciences, and fusion-oriented energy research.

D-Grid (2004 – 2015) [44] was the German Grid Initiative (DGI) aimed at providing computer infrastructure for education and research in the field of e-Science. The initiative focused on utilizing grid computing technology. It commenced on September 1, 2005, with six community projects, an integration project (DGI), and various partner projects. For example, AstroGrid-D – a joint research project managed by the Astrophysical Institute Potsdam (AIP) focused on integrating German research facilities in astronomy into a unified nationwide infrastructure; C3-Grid – a project aiming to link distributed data archives in climate research; GDI-Grid – a project integrating geospatial data and grid technologies to enhance spatial data infrastructures; HEP-Grid – a project optimizing data analysis in High Energy Physics, Nuclear Physics, and Astroparticle Physics.

Open Science Grid (OSG, 2005 – still ongoing) [72] is a project funded by the National Science Foundation (NSF) that offers researchers distributed high throughput computing. Numerous institutions contribute their computer resources to the OSG, allowing users to run jobs during periods of idle activity. The scale of the OSG's capacity is impressive. By 2021, the OSG provided a staggering 1.1 million core hours, which is equivalent to utilizing 42 thousand

cores in a single day [6]. Currently, OSG includes 249 site and 153 institutes distributed all over the world (see Fig. 8)³².

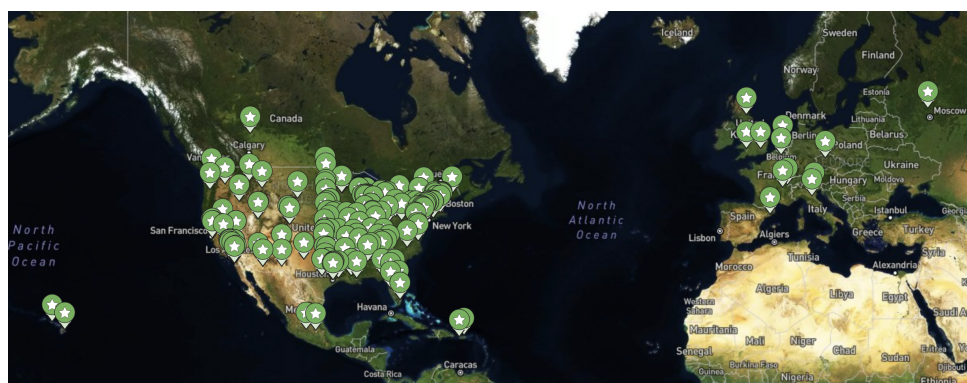


Figure 8. OSG sites

Each pool within the OSG is structured and managed to cater to specific research communities, such as campuses or multi-institutional collaborations. These pools utilize technologies and services offered by the core OSG Team. One notable pool is the Open Science Pool, which serves the entire US-associated open science community. Consequently, the Consortium encompasses all researchers, resources, individuals, and institutions that either benefit from or contribute to any of the OSG’s Fabric of Services (further detailed below).

Currently there are more than 100 scientific projects carried out within the OSG pool of resources. The vast majority of the CPU hours are occupied with physics, chemical science and bioinformatics projects.

Earth System Grid Federation (ESGF, 2006 – still ongoing) [30] is a collaborative set of international interlinked data centers that work together to support climate and Earth system science through data sharing and computational frameworks. ESGF has a global network of distributed data centers, each of which provides infrastructure and resources for scientific data management, access, and sharing. As of early 2022, there are over 40 participating data centers in the ESGF spread across more than 20 countries³³.

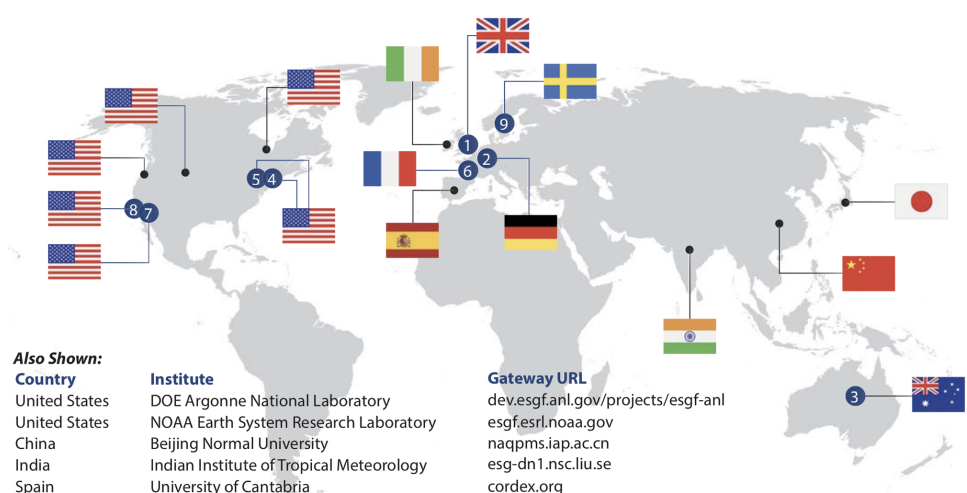


Figure 9. Major federated ESGF worldwide sites [5]

³²<https://map.opensciencegrid.org/map>

³³<https://esgf-index1.ceda.ac.uk/projects/esgf-ceda/>

Among them, CEDA (Centre for Environmental Data Analysis), based in the UK; DKRZ (Deutsches Klimarechenzentrum), located in Germany; IPSL (Institut Pierre Simon Laplace), a French research institute; LIU (Linkping University), based in Sweden; LLNL (Lawrence Livermore National Laboratory) and ORNL (Oak Ridge National Laboratory) metagrids – US-based metagrids offering data storage and computing resources for scientific research and data analysis in various fields including climate and energy; NCI (National Computational Infrastructure), a data center in Australia [5].

MammoGrid (2009 – 2013) [83] was a collaboration between several European research institutions and universities. The aim of the project was to develop a Europe-wide database of mammograms. MammoGrid was a distributed computing system that aims to improve the accuracy and efficiency of mammography screening. The system used machine learning algorithms to analyze mammograms and provide recommendations to radiologists. MammoGrid used to process large volumes of mammograms at a high speed, enabling healthcare providers to quickly and accurately identify potential breast cancer cases. This project represented an important milestone in the development of eHealth services and distributed computing infrastructure for medical imaging analysis.

European Grid Infrastructure (EGI, 2010 – still ongoing)³⁴ is a federation of national computing and storage resources across Europe. Its purpose is to foster advanced research,

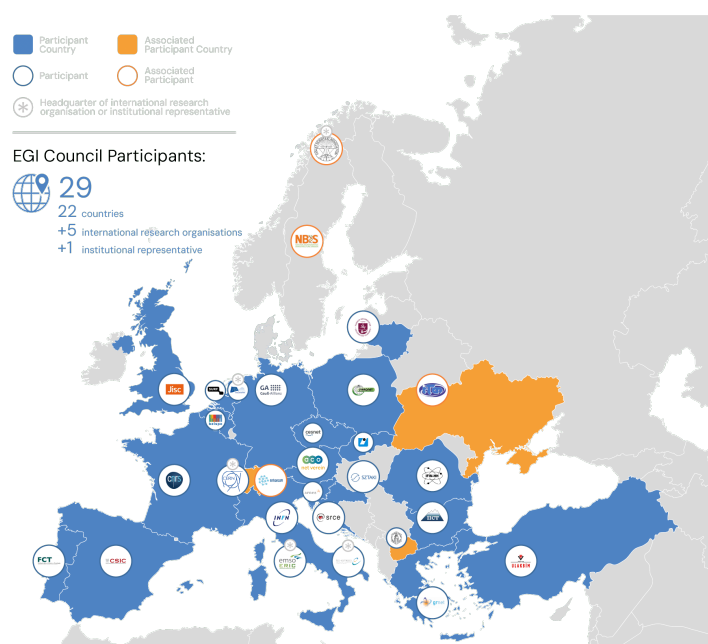


Figure 10. EGI Members

innovation, and knowledge transfer within the continent. EGI builds upon the substantial investments made by national governments and the European Commission over a span of more than a decade.

Currently EGI has 29 participants, 22 represented countries and 300+ represented organisations (see Fig. 10)³⁵.

EGI brings together a range of resources: a federated cloud infrastructure and HTC Platform (EGI HTC). With over 94,000 users, EGI is a significant player in the field, delivering 80 million

³⁴<https://www.egi.eu/>

³⁵<https://www.egi.eu/egi-federation/>

cloud CPU hours and 7 billion HTC CPU hours in 2023. The platform boasts 580 PB of online storage capacity. The EGI Federation draws support from nearly 300 data centers predominantly in European countries aligned with the EGI Council. It further benefits from resources in Canada, the USA, Latin America, North Africa, and the Asia-Pacific region.

EGI supports scientists in various research disciplines, including high-energy physics, astrophysics, computational chemistry, life sciences, earth sciences, fusion, and many others.

Partnership for Advanced Computing in Europe (PRACE, 2010 – still ongoing)³⁶ is a collaborative project that was established in 2010 to provide researchers with access to some of the most powerful supercomputers in Europe to carry out large-scale simulations and data analysis tasks in a wide range of scientific domains, including biochemistry, climate modeling, astrophysics, and high-energy physics. The collaboration consists of 26 members (Fig. 11), including most of the major EU nations³⁷. PRACE operates a distributed infrastructure that

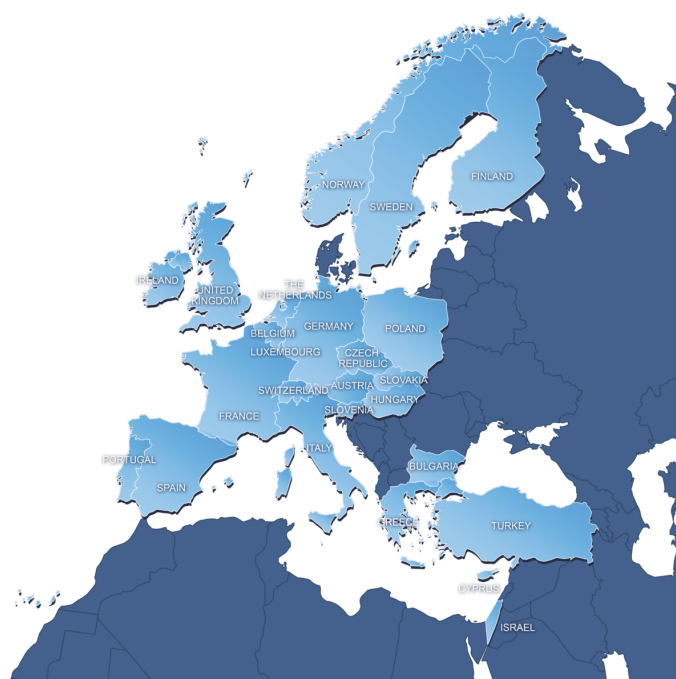


Figure 11. PRACE member countries

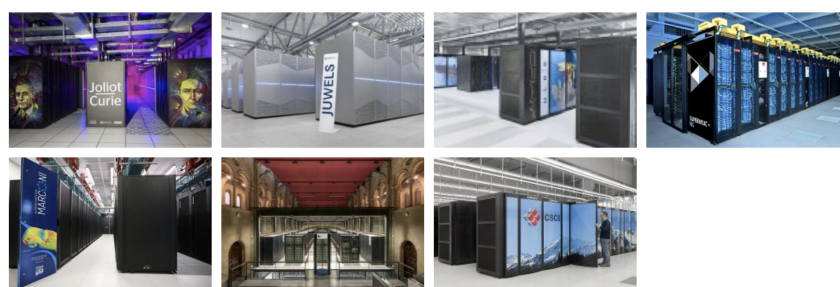


Figure 12. PRACE Tier-0 Centers

consists of a number of Tier-0 supercomputing centers (Fig. 12), located in different countries, that host some of the fastest and most powerful supercomputers in the world, including Joliot-

³⁶www.prace-ri.eu

³⁷<https://prace-ri.eu/about/members/>

Curie in France, JUWELS, HAWK and SuperMUC-NG in Germany, MARCONI in Italy, MareNostrum 4 in Spain, Piz Daint in Switzerland³⁸.

The PRACE infrastructure also includes Tier-1 centers that provide additional computing resources and services to support researchers' work.

7. Clouds over Grids

Among all grid applications, the WLCG is the largest and the most long-living grid project. When the WLCG was created, technologies were driven by science and adapted to the needs of Big Science. However, with the rapid technological growth in the last decade, the situation has changed dramatically, and now science is following technology. The same happened with the WLCG, which gradually began to adapt to the use of cloud resources.

Until recently, the computing model at the WLCG was built on the assumption that experiments are the "owners" of computing resources. To be more precise, the computing resources were operated by laboratories and university partners or deployed for a specific program, owned and operated by the host laboratory.

However, HEP experiments revised the initial strategy of utilizing only on-premises resources due to several reasons [53]:

- *Stochastic and bursty nature of computing activities and heterogeneous resources:* The computing activities of the HEP experiments can be categorized into two types: central production activities and analysis activities. Central production is planned and centrally managed, while analysis activities are more chaotic, involving submissions from multiple individuals and being less predictable. In both cases, computing activity occurs in bursts, influenced by the accelerator schedule and advancements in software and detector modernization. Observing a typical data processing for the on-premises resources of HEP experiments demonstrates the stochastic nature of compute demand and the cost implications of provisioning for peak capacity rather than steady-state usage. However, having peak capacity is necessary to perform significant amounts of computing within shorter timeframes, such as for new analyses with discovery potential or conference deadlines.
- *Multiple programs reaching their peak:* By 2025, several HEP programs, including the muon and neutrino programs, and the LHC, will be at the apex of their offline analysis. This stage requires significant computing power and resources.
- *Evolution of the experimental program:* The ongoing upgrades and new experiments in high-energy physics generate a higher demand for computing resources, which surpasses the expected performance gains from advancements in computing techniques and technologies. For example, the High-Luminosity LHC³⁹ and the Deep Underground Neutrino Experiment (DUNE)⁴⁰ are two upcoming programs in HEP that will generate massive amounts of data. The increased precision, event complexity, and luminosity of these programs alone will push computing needs nearly two orders of magnitude higher than current capabilities, generating exabytes of data.
- *The landscape of modern computing resources and the needs for them are dramatically different from the situation 20–30 years ago, when HEP applications were one of the main "consumers" of computing power in the global world.* For example, the LHC at CERN

³⁸<https://prace-ri.eu/infrastructure-support/prace-hpc-infrastructure/>

³⁹<https://www.home.cern/science/accelerators/high-luminosity-lhc>

⁴⁰<https://www.dunescience.org/>

generates about 50 petabytes per year at WLCG, while there is a large pool of computing resources outside of the HEP: commercial resources (Google, Amazon, Microsoft) and supercomputer centers that are hundreds of times more powerful than the WLCG consortium. Currently, a diverse range of projects are responsible for vast data generation. For instance, social media platforms like Facebook create around 250 petabytes of new data per year. Google's search index alone tracks a minimum of 62 petabytes of data per year. Over 260 petabytes of video content are uploaded to YouTube annually. [32].

Therefore, HEP experiments decided to move towards the usage of cloud technologies that imply on-demand allocation of the resource from a large pool of computing and storage resources, which can be accessed via standard protocols via an abstract interface.

Clouds can be built on top of many existing protocols (WDSL, SOAP), Web 2.0. Technologies (REST, RSS, AJAX) and implemented over existing grids leveraging more than a decade of community efforts in standardization, security, resource management, and virtualization support.

The adoption of cloud computing over grid systems at CERN was a gradual and systematic process, with different commercial cloud solutions being applied to various workflows. Now, let us provide a concise summary of the process involved in establishing cloud infrastructure within WLCG.

OpenStack at CERN Since 2013, CERN has deployed an OpenStack-based private cloud to manage resources across its main data centre in Meyrin, Switzerland, and a remote extension in Budapest, Hungary. The purpose of the use of OpenStack at CERN was to provide resources to scientists and researchers who are conducting experiments related to high-energy physics and particle acceleration.

In just its first two years of implementation, the OpenStack cloud tremendously expanded, supporting over 12,000 virtual machines that were spread across 5,500 compute nodes. The usage of this cloud environment was highly dynamic, witnessing an average daily creation and deletion of approximately 3,000 virtual machines. As of 2020, the total number of cores in use has surpassed 300,000, highlighting the remarkable growth and scalability achieved [18].

OpenStack helps CERN to flexibly and efficiently allocate computing resources, reducing management overhead and increasing productivity for the researchers. Additionally, OpenStack enables CERN to better manage cost and capacity planning as it supports the scaling of computing resources which is critical for big science.

OpenStack plays a crucial role in managing and provisioning the majority of CERN's computer center infrastructure, accounting for over 90%. This encompasses various essential functions such as physics processing, storage, databases, and the infrastructure supporting laboratory administration. To ensure robust resource management, accounting accuracy, and effective life-cycle tracking, the remaining hardware within the computer center is currently being integrated into Ironic.

Since 2011, CERN has actively developed and refined OpenStack, with over 1,000 commits. They primarily focused on Magnum, Nova, and Keystone projects. CERN shared their experiences at various events, including over 30 talks at OpenStack summits and regional gatherings like open Infrastructure days. Notably, CERN organized an OpenStack day in 2019 showcasing real-world applications in science, and in 2020, they hosted the Ironic mid-cycle meetup, fostering collaboration within the OpenStack community [37].

European HelixNebula

ATLAS experiment pioneered the usage of cloud computing as a flagship project during the first stages of the European HelixNebula initiative between 2012 and 2014. Helix Nebula is a consortium formed by public research institutions (CERN, EMBL⁴¹ and ESA⁴²) and several commercial providers of cloud services (T-Systems, Atos, CloudSigma, The Server Labs, Interoute, etc.) with the aim to make the private resources accessible by research institutions in Europe in a transparent way. During the first proof of concept, this project contributed over 40,000 CPU-days of Monte Carlo production throughput to the ATLAS experiment with marginal manpower required. CERN's experience, together with that of ESA and EMBL, is providing a great insight into the cloud computing industry and highlighted several challenges that are being tackled in order to ease the export of the scientific workloads to the cloud environments [65].



Atos and Microsoft Azure

In March 2015, CERN embarked on their first cloud activity project in collaboration with Atos, a globally renowned company providing information technology, consulting, system integration, and remote management services [34]. Atos, founded on July 1, 2011, is the number one company in Europe in the field of Operational Services and has served as a key IT partner of the International Olympic Committee (IOC) for over two decades. This project involved the provisioning of approximately 3,000 single-core virtual machines (VMs) for the ATLAS collaboration to process simulation jobs over a six-week period. The project demonstrated the ability to incorporate cloud infrastructure-as-a-service (IaaS) within the WLCG workflows to execute central processing unit (CPU) intensive tasks accessing data and software libraries remotely via technologies such as CVMFS and XRootD.



In parallel, Microsoft Azure performed an evaluation by conducting simultaneous deployments in three different data centers, two in the European Union and one in the United States, with collaborations from the ATLAS, CMS, and LHCb submitting simulation jobs.



HEPCloud The Fermilab Scientific Computing Division launched the HEPCloud project in June of 2015 with the objective of creating a comprehensive resource facility with a common interface for a variety of resources including local clusters, grids, high-performance computing, and both community and commercial clouds. The pilot project to assess the feasibility and capability of HEPCloud was initiated in 2016 and executed workflows from the CMS and NOvA experiments. In January of the same year, the project showcased the ability to increase global CMS resources by 58,000 cores from 150,000 cores (a 25% increase rate) to prepare for the Recontres de Moriond, one of the most significant international High Energy Physics (HEP) events [67].



The cost of commercial cloud resources was found to be comparable to on-site resources. This project suggested that steady-state computing costs can become more cost-effective as the cloud computing industry scales, benefiting from potential economies of scale. The flexibility in resource payment provides improved planning and cost efficiency. However, workflows with

⁴¹European Molecular Biology Laboratory

⁴²European Space Agency

intensive data processing might be more suitable for local resources. Therefore, it was shown that adopting a hybrid approach like the HEPCloud facility, which integrates both on-premises and off-premises resources into a unified virtual facility, offers the optimal flexibility to meet the dynamic requirements of the scientific community.

Deutsche Borse Cloud Exchange (DBCE)

Deutsche Borse Cloud Exchange (DBCE) were the vendors for the second cloud procurement in autumn 2016, providing 1,000 4-core VMs running simulation jobs for all LHC collaborations. Although established in May 2013, DBCE was discontinued in 2016 and operated as an international marketplace for the buying and selling of cloud resources. Unlike other commercial cloud providers, DBCE served as a broker to connect customers with IaaS providers in the DBCE marketplace where compute capacity was traded like any other commodity exchange. During this time, DBCE settled a contract agreement among CERN and five different cloud providers, namely ClouData, Cloud&Heat, DARZ, Innovo, and Ultimium [34].



T-Systems and Open Telekom Cloud

T-Systems⁴³ was allocated a contract for a Pre-Commercial Procurement (PCP) initiated by CERN to design, develop and pilot the Helix Nebula Science Cloud. This joint venture between industry, space, and science aimed to establish an ecosystem with open cloud services that seamlessly integrate science into a business environment. T-Systems and CERN conducted a three-month pilot to assess the capabilities of the Open Telekom Cloud, built on the OpenStack open-source architecture, for data and resource management between private and public clouds. CERN used Open Telekom Cloud to validate whether outsourced commercial cloud providers could process physics data flexibly through the deployment of one thousand virtual machines with associated cluster storage exceeding 500 terabytes.



Amazon Web Services and ATLAS

In 2018, the ATLAS experiment submitted a request for \$250,000 worth of Amazon Web Services (AWS) credits to establish the first virtual Tier 3 cluster on the cloud. The objective was to examine cloud-based solutions for physics analysis and simulation for the ATLAS experiment. The proposal was approved and funded in December 2019, with support from ATLAS and US ATLAS computing management teams. The proposed work, approved by LBNL and University of Texas at Arlington, included setting up ATLAS computing environments for physical analysis frameworks, Monte Carlo simulations, generation and production on the cloud, running sizable analysis jobs, Monte Carlo generation, and simulation for the ATLAS collaboration, and exploring an economic model for future ATLAS computing on the cloud.



Google and ATLAS Projects

The collaborative project between ATLAS and Google was initiated in 2017 to demonstrate a transparent use of commercial cloud resources for scientific experiments [15]. The next phase of the collaboration was launched



⁴³<https://www.t-systems.com/>

in 2020 [62]. The scope of this project was to integrate Google cloud resources (Storage and Compute) into the ATLAS distributed computing environment in order to enable ATLAS to explore different computing models in preparation for the High-Luminosity LHC, to allow ATLAS users to leverage the Google infrastructure for their analyses, and provide Google with real scientific use cases to enhance their cloud platform. The success of the above projects has triggered an interest and a 15-month ATLAS-Google project [63] was started in May 2022. Recently, long-term projects with Google and Amazon, supported by US ATLAS and California State University Fresno grants respectively were initiated. These projects have gone beyond infrastructure rental, fostering relationships through regular meetings and information exchange [64].

ATLAS has successfully demonstrated the advantage of integration of on-prem and commercial cloud resources as a universal discovery cyberinfrastructure which can be used by large international collaborations. The common perception amongst site administrators is that the cloud is more expensive compared to on-premise options, although cost analysis may be influenced by comparisons to dedicated compute instances rather than spot instances. Previous experiences with spot instances showed eviction rates of up to 15%, which is deemed unacceptable, particularly at multi VO (Virtual Organizations) sites. However, during this project, the observed eviction rate ranged between 1–2%, with 20% of failed jobs resulting from preemptions. Performance variations over time were found to be unpredictable in cloud environments, but the CPU provided by Google remained stable and experienced minimal downtime. The administrators expressed worries about vendor lock-in and emphasized the importance of maintaining flexibility and the ability to switch cloud providers. Notably, the solutions developed in this project are cloud-agnostic and have also been used on AWS at a smaller scale.

The integration of the ATLAS Google site into ATLAS distributed computing has been overwhelmingly successful, showcasing that elastic increase of on-prem resources with a commercial cloud at a large scale requires minimal operational effort. The current ATLAS workflow and data management tools have proven to be adequate for adjusting the configuration of the cloud site. Commercial cloud computing offers a new avenue for HEP computing, providing additional CPU resources on-demand, although network costs may pose a significant factor. Different ATLAS workflows have shown varying degrees of success regarding egress, with derivation production being the most resource-intensive. Resource bursting has been effectively demonstrated but comes at a notable cost. The project has also fostered parallel research and development efforts, allowing for the utilization of diverse resources such as GPUs and ARM CPUs on an elastic basis with seamless integration. The subscription pricing model employed in this project has proven to be suitable for ATLAS; however, future considerations regarding this model remain uncertain. Overall, the project has made significant advancements and laid the groundwork for the continued exploration of cloud-based solutions for ATLAS [81].

8. Grid Applications Ended Up with Greatest Scientific Discoveries

Grid computing, with its ability to connect large-scale computing resources around the world and make them available to researchers and scientists, has played a significant role in some of the greatest scientific discoveries and achievements of the last decade.

Some of the most significant scientific discoveries made possible by grid computing are the Higgs Boson discovery and the detection of gravitational waves.

8.1. Higgs Boson Discovery

In 2012, a team of researchers at CERN used the WLCG to identify the Higgs boson particle, which helps explain the origins of mass in the universe. The discovery was a major milestone in the field of particle physics and led to a Nobel Prize in Physics in 2013 (awarded to Francois Englert and Peter Higgs) [7].

CERN scientists, along with thousands of researchers from around the world, contributed to the discovery of the Higgs boson. Two experiments conducted at the LHC, called ATLAS and CMS, independently announced the discovery on July 4, 2012. These experiments involved analyzing the data from billions of proton-proton collisions to identify the telltale signature of the Higgs boson production.

The Higgs boson is associated with the Higgs field, a field permeating space that interacts with particles, giving them mass. Its discovery confirmed a crucial aspect of the Standard Model and provided experimental evidence for the Higgs mechanism, proposed by Francois Englert and Peter Higgs in the 1960s.

To observe the Higgs boson, the ATLAS and CMS experiments at the LHC required a substantial amount of data accumulation. More specifically, both collaborations analyzed a dataset of around 25–30 petabytes of collision data.

Dr. Fabiola Gianotti (Spokesperson of the ATLAS experiment in 2008/2013, General Director of CERN since 2014) said at a seminar on the discovery of the Higgs boson: *“We are observing a new particle with a mass of about 126 GeV. We would not have been able to process and analyze data so quickly if we had not used the grid. Centers in all countries participating in the experiment were involved in processing LHC data; practically it was a stress test for computing power, and the grid proved to be a highly efficient and reliable system.”*⁴⁴

8.2. Gravitational Waves Detection

In 2015, the Laser Interferometer Gravitational-Wave Observatory (LIGO) used grid computing to detect the first direct evidence of gravitational waves, a phenomenon predicted by the Einstein’s theory of general relativity nearly a century ago. Gravitational waves are ripples in the fabric of spacetime that travel at the speed of light, carrying information about extremely energetic cosmic events.

On September 14, 2015 (Event GW150914), LIGO made history by detecting the collision of two black holes, an event that emitted gravitational waves. This marked the first direct observation of gravitational waves, opening up an entirely new way to observe and study the universe.

While the scale of gravitational-wave data analysis may no longer be considered as “big data” by current standards, data management remains complex in a distributed HTC environment: LIGO/Virgo generates approximately 20TB of data per interferometer per observing year. For the discovery of gravitational waves in the particular event GW150914, LIGO utilized data spanning approximately 16 days and required 50 million CPU hours for computations⁴⁵.

For this groundbreaking achievement, the Nobel Prize in Physics was awarded in 2017 to three key scientists involved in the LIGO project: Rainer Weiss, Barry C. Barish, and Kip S. Thorne.

⁴⁴https://www.mpg.de/5882007/higgs_boson

⁴⁵<https://www.ligo.org/detections/GW150914/fact-sheet.pdf>

The discovery of gravitational waves opened up a new window of observation, allowing scientists to explore astrophysical phenomena that had previously been invisible. Since then, LIGO and other gravitational wave observatories have continued to detect gravitational waves from various cosmic events, including the merger of neutron stars and black holes. This field of research has furthered our understanding of black holes, neutron stars, and the nature of gravity itself.

Grid computing plays a pivotal role in LIGO's computational infrastructure, enhancing the capabilities of the LIGO Data Grid. In addition to dedicated clusters, LIGO has integrated external resources like the Open Science Grid (OSG) to introduce flexibility and accommodate diverse computational needs. OSG's elasticity allows LIGO to incorporate non-traditional systems alongside its existing infrastructure, catering to both regular workloads and dynamic tasks with efficiency.

Furthermore, LIGO leverages the Berkely Open Infrastructure for Network Computing (BOINC) to harness idle volunteer computers worldwide through projects like Einstein@Home(E@H). This distributed computing model enables LIGO to engage a vast network of distributed resources for computationally intensive tasks, such as Pulsar searches [4].

Conclusion

The following factors have played a crucial role in the development of distributed computing: the challenges of big science, the advancement of data transmission networks, and the development of middleware systems. Grid technologies have made significant scientific discoveries possible on a global scale, such as the discovery of the Higgs boson at CERN and of gravitational waves in the LIGO project. Countless specialists from around the world have contributed their intellectual efforts to the advancement of distributed technologies. Large international grid infrastructures were created, and middleware was developed to ensure the seamless functioning of major scientific projects.

Until the early 2000s, technology followed science, developed technological and software approaches meeting the needs of science. But afterward, when technology reached a new level, scientific projects have begun to adapt to modern technologies by integrating commercial cloud services and containerization into their grid infrastructure.

Despite the success of cloud technologies in both science and business, many existing scientific projects are built on grid infrastructures and will continue to operate for approximately 10 more years. As for new scientific projects currently being developed, some of them may utilize HPC (High-Performance Computing) resources, such as the International Thermonuclear Experimental Reactor (ITER), while others may opt for hybrid systems combining grid, HPC, and cloud technologies, like Nuclotron-based Ion Collider fAcility (NICA) or Deep Underground Neutrino Experiment (DUNE).

Therefore, despite the emergence of more modern technologies, grid computing still maintains its position thanks to the software developed over the past decades, successful application experience, and reliability.

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. Gigabit network testbeds. *Computer* 23(9), 77–80 (1990). <https://doi.org/10.1109/2.58220>
2. The Gigabit Testbed Initiative. Final Report (1996), http://www.cnri.reston.va.us/gigafr/Gigabit_Final_Rpt.pdf, accessed: 2024-02-16
3. ARC middleware. The NorduGrid Collaboration (2005), <http://www.nordugrid.org/documents/whitepaper.pdf>, Accessed: 2024-02-16
4. OSG helps LIGO scientists confirm Einstein’s unproven theory (2016), <https://osg-htc.org/spotlights/osg-helps-ligo-scientists-confirm-einsteins-last-unproven-theory.html>, accessed: 2024-04-27
5. 2017 ESGF Brochure (2017), <https://esgf.llnl.gov/esgf-media/pdf/2017-ESGF-Brochure.pdf>, accessed: 2024-04-29
6. OSPool Usage Hits Daily Record (2021), <https://osg-htc.org/news/2021/06/10/OSPool-Hits-Record-Utilization.html>, accessed: 2024-02-16
7. Aad, G., *et al.*: Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Physics Letters B* 716(1), 1–29 (2012). <https://doi.org/10.1016/j.physletb.2012.08.020>
8. Aasi, J., *et al.*: Einstein@Home all-sky search for periodic gravitational waves in LIGO S5 data. *Phys. Rev. D* 87, 042001 (Feb 2013). <https://doi.org/10.1103/PhysRevD.87.042001>
9. Abbott, B.P., *et al.*: LIGO: the laser interferometer gravitational-wave observatory. *Reports on Progress in Physics* 72(7), 076901 (jun 2009). <https://doi.org/10.1088/0034-4885/72/7/076901>
10. Alexandrov, A., Ibel, M., Schausser, K., *et al.*: SuperWeb: towards a global Web-based parallel computing infrastructure. In: *Proceedings 11th International Parallel Processing Symposium*. pp. 100–106. IEEE Comput. Soc. Press, Genva, Switzerland (1997). <https://doi.org/10.1109/IPPS.1997.580858>
11. Anderson, D.P.: BOINC: A Platform for Volunteer Computing. *Journal of Grid Computing* 18(1), 99122 (Mar 2020). <https://doi.org/10.1007/s10723-019-09497-9>
12. Anderson, D.P., Cobb, J., Korpela, E., *et al.*: SETI@home: An Experiment in Public-Resource Computing. *Commun. ACM* 45(11), 56–61 (nov 2002). <https://doi.org/10.1145/581571.581573>
13. Antonioletti, M., Krause, A., Paton, N.W.: An Outline of the Global Grid Forum Data Access and Integration Service Specifications. In: Pierson, J.M. (ed.) *Data Management in Grids*. pp. 71–84. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
14. Baratloo, A., Karaul, M., Kedem, Z., *et al.*: Charlotte: Metacomputing on the Web. *Future Generation Computer Systems* 15(5), 559–570 (1999). [https://doi.org/10.1016/S0167-739X\(99\)00009-6](https://doi.org/10.1016/S0167-739X(99)00009-6)

15. Barisits, Martin, Barreiro, Fernando, *et al.*, B.: The Data Ocean Project – An ATLAS and Google R&D collaboration. EPJ Web Conf. 214, 04020 (2019). <https://doi.org/10.1051/epjconf/201921404020>
16. Bauerfeld, W.: A tutorial on network gateways and interworking of LANs and WANs. Computer Networks and ISDN Systems 13(3), 187–193 (Jan 1987). [https://doi.org/10.1016/0169-7552\(87\)90025-0](https://doi.org/10.1016/0169-7552(87)90025-0)
17. Beberg, A.L., Ensign, D.L., Jayachandran, G., *et al.*: Folding@home: Lessons from Eight Years of Volunteer Distributed Computing. In: Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing. pp. 1–8. IPDPS '09, IEEE Computer Society, USA (2009). <https://doi.org/10.1109/IPDPS.2009.5160922>
18. Bell, T., Bompastor, B., Bukowiec, S., *et al.*: Scaling the cern openstack cloud. Journal of Physics: Conference Series 664, 022003 (12 2015). <https://doi.org/10.1088/1742-6596/664/2/022003>
19. Berners-Lee, T., Cailliau, R., Luotonen, A.t.: The World-Wide Web. Commun. ACM 37(8), 76–82 (aug 1994). <https://doi.org/10.1145/179606.179671>
20. Bernstein, P.A.: Middleware: A Model for Distributed System Services. Commun. ACM 39(2), 86–98 (feb 1996). <https://doi.org/10.1145/230798.230809>
21. Bogdanov, A., *et al.*: GRID processing and analysis of ALICE data at distributed Russian Tier2 centre: RDIG. J. Phys. Conf. Ser. 219, 072054 (2010). <https://doi.org/10.1088/1742-6596/219/7/072054>
22. Bonacorsi, D., Ferrari, T.: WLCG Service Challenges and Tiered architecture in the LHC era. In: Montagna, G., Nicosini, O., Vercesi, V. (eds.) IFAE 2006. pp. 365–368. Springer Milan, Milano (2007). https://doi.org/10.1007/978-88-470-0530-3_68
23. Brecht, T., Sandhu, H., Shan, M., *et al.*: ParaWeb: Towards World-Wide Supercomputing. In: Proceedings of the 7th Workshop on ACM SIGOPS European Workshop: Systems Support for Worldwide Applications. pp. 181–188. EW 7, Association for Computing Machinery, New York, NY, USA (1996). <https://doi.org/10.1145/504450.504484>
24. Buyya, R., Abramson, D., Giddy, J.: Nimrod/G: an architecture for a resource management and scheduling system in a global computational grid. In: Proceedings Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region. vol. 1, pp. 283–289 (2000). <https://doi.org/10.1109/HPC.2000.846563>
25. Casajus, A., Graciani, R., Paterson, S., *et al.*: DIRAC pilot framework and the DIRAC Workload Management System. Journal of Physics: Conference Series 219(6), 062049 (Apr 2010). <https://doi.org/10.1088/1742-6596/219/6/062049>
26. Casanova, H.: Simgrid: a toolkit for the simulation of application scheduling. In: Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid. pp. 430–437 (2001). <https://doi.org/10.1109/CCGRID.2001.923223>
27. Catlett, C., Allcock, W., Andrews, P., *et al.*: TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Applications, vol. 16, pp. 225–249 (01 2008)

28. Chapin, S.J., Katramatos, D., Karpovich, J., *et al.*: The Legion Resource Management System. In: Feitelson, D.G., Rudolph, L. (eds.) *Job Scheduling Strategies for Parallel Processing*. pp. 162–178. Springer Berlin Heidelberg, Berlin, Heidelberg (1999). https://doi.org/10.1007/3-540-47954-6_9
29. Chiappa, J.N., Steenstrup, M.E., Castineyra, I.M.: The Nimrod Routing Architecture. RFC 1992 (Aug 1996). <https://doi.org/10.17487/RFC1992>, accessed: 2024-02-16
30. Cinquini, L., *et al.*: The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data. *Future Generation Computer Systems* 36, 400–417 (Jul 2014). <https://doi.org/10.1016/j.future.2013.07.002>
31. Clarke, L., Glendinning, I., Hempel, R.: The MPI Message Passing Interface Standard. In: Decker, K.M., Rehmann, R.M. (eds.) *Programming Environments for Massively Parallel Distributed Systems*. pp. 213–218. Birkhäuser Basel, Basel (1994)
32. Clissa, L., Lassnig, M., Rinaldi, L.: How big is Big Data? A comprehensive survey of data production, storage, and streaming in science and industry. *Frontiers in Big Data* 6 (2023). <https://doi.org/10.3389/fdata.2023.1271639>
33. Cole, N., Desell, T., Lombrana González, D., *et al.*: Evolutionary Algorithms on Volunteer Computing Platforms: The MilkyWay@Home Project, pp. 63–90. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-10675-0_4
34. Cordeiro, C., Field, L., *et al.*: CERN Computing in Commercial Clouds. *Journal of Physics: Conference Series* 898(8), 082030 (oct 2017). <https://doi.org/10.1088/1742-6596/898/8/082030>
35. DeFanti, T.A., Foster, I., Papka, M.E., *et al.*: Overview of the I-Way: Wide-Area Visual Supercomputing. *The International Journal of Supercomputer Applications and High Performance Computing* 10(23), 123–131 (Jun 1996). <https://doi.org/10.1177/109434209601000201>
36. Erwin, D.W., Snelling, D.F.: UNICORE: A Grid Computing Environment. In: Sakellariou, Rizos and Gurd, John and Freeman, Len *et al.* (ed.) *Euro-Par 2001 Parallel Processing*. pp. 825–834. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
37. Ferguson, A.: Where are they now? Superuser Awards winner: CERN, <https://superuser.openinfra.dev/articles/cern-openstack-update/>, accessed: 2024-02-16
38. Filipi, A., Cameron, D., Smirnova, O., *et al.*: The Next Generation ARC Middleware and ATLAS Computing Model. *Journal of Physics: Conference Series* 396(3), 032039 (dec 2012). <https://doi.org/10.1088/1742-6596/396/3/032039>
39. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Jin, H., Reed, D., Jiang, W. (eds.) *Network and Parallel Computing*. pp. 2–13. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
40. Foster, I.: Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. *IEEE Internet Computing* 15(3), 70–73 (2011). <https://doi.org/10.1109/MIC.2011.64>

41. Foster, I., Kesselman, C.: Globus: a Metacomputing Infrastructure Toolkit. *The International Journal of Supercomputer Applications and High Performance Computing* 11(2), 115–128 (1997). <https://doi.org/10.1177/109434209701100205>
42. Foster, I., Kesselman, C., Tuecke, S.: Chapter 17 – The Open Grid Services Architecture, pp. 215–257. *The Morgan Kaufmann Series in Computer Architecture and Design*, Morgan Kaufmann, Burlington (Jan 2004). <https://doi.org/10.1016/B978-155860933-4/50022-5>
43. Foster, Ian and Kesselman, Carl: *The History of the Grid* 20(21), 22 (2010)
44. Freitag, S., Wieder, P.: *The German Grid Initiative D-Grid: Current State and Future Perspectives*, pp. 29–52. Springer London, London (2011). https://doi.org/10.1007/978-0-85729-439-5_2
45. Frey, J., Tannenbaum, T., Livny, M., *et al.*: Condor-G: a computation management agent for multi-institutional grids. In: *Proceedings 10th IEEE International Symposium on High Performance Distributed Computing*. pp. 55–63 (2001). <https://doi.org/10.1109/HPDC.2001.945176>
46. Gagliardi, F.: The EGEE European Grid Infrastructure Project. In: Daydé, Michel and Dongarra, Jack and Hernández, Vicente and *et al.* (ed.) *High Performance Computing for Computational Science – VECPAR 2004*. pp. 194–203. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/11403937_16
47. Garonne, V., Vigne, R., Stewart, G., *et al.*: Rucio – The next generation of large scale distributed system for ATLAS Data Management. *Journal of Physics: Conference Series* 513(4), 042021 (Jun 2014). <https://doi.org/10.1088/1742-6596/513/4/042021>
48. Gentzsch, W., Girou, D., Kennedy, A., *et al.*: DEISA – Distributed European Infrastructure for Supercomputing Applications. *Journal of Grid Computing* 9(2), 259–277 (Jun 2011). <https://doi.org/10.1007/s10723-011-9183-2>
49. Glasser, A.H., Sovinec, C.R., Nebel, R.A., *et al.*: The NIMROD code: a new approach to numerical plasma physics. *Plasma Physics and Controlled Fusion* 41(3A), A747 (mar 1999). <https://doi.org/10.1088/0741-3335/41/3A/067>
50. Haber, R.: Scientific visualization and the Rivers Project at the National Center for Supercomputing Applications. *Computer* 22(8), 84–89 (1989). <https://doi.org/10.1109/2.35205>
51. Henderson, R.L.: Job scheduling under the Portable Batch System. In: Feitelson, D.G., Rudolph, L. (eds.) *Job Scheduling Strategies for Parallel Processing*. pp. 279–294. Springer Berlin Heidelberg, Berlin, Heidelberg (1995). https://doi.org/10.1007/3-540-60153-8_34
52. Henning, M.: The Rise and Fall of CORBA: Theres a lot we can learn from CORBAs mistakes. *Queue* 4(5), 28–34 (jun 2006). <https://doi.org/10.1145/1142031.1142044>
53. Holzman, B., Bauerdick, L.A.T., Bockelman, B., *et al.*: HEPCloud, a New Paradigm for HEP Facilities: CMS Amazon Web Services Investigation. *Computing and Software for Big Science* 1(1), 1 (Nov 2017). <https://doi.org/10.1007/s41781-017-0001-9>

54. Jackson, D., Snell, Q., Clement, M.: Core Algorithms of the Maui Scheduler. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. pp. 87–102. Springer Berlin Heidelberg, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-45540-X_6
55. Jackson, T., Austin, J., Fletcher, M., *et al.*: Diagnostics and Prognostics on the Grid: the Distributed Aircraft Maintenance Environment project (DAME) (01 2004)
56. Karsch, F., Simma, H., Yoshie, T.: The International Lattice Data Grid – towards FAIR data. In: Proceedings of The 39th International Symposium on Lattice Field Theory PoS(LATTICE2022). p. 244. Sissa Medialab, Bonn, Germany (Feb 2023). <https://doi.org/10.22323/1.430.0244>
57. Kónya, B.: Advanced Resource Connector (ARC) – The Grid Middleware of the NorduGrid. In: Kranzlmüller, D., Kacsuk, P., Dongarra, J. (eds.) Recent Advances in Parallel Virtual Machine and Message Passing Interface. p. 10. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
58. Litzkow, M., Livny, M., Mutka, M.: Condor-a hunter of idle workstations. In: [1988] Proceedings. The 8th International Conference on Distributed. pp. 104–111. IEEE Comput. Soc. Press, San Jose, CA, USA (1988). <https://doi.org/10.1109/DCS.1988.12507>
59. Lukasik, S.: Why the Arpanet Was Built. IEEE Annals of the History of Computing 33(3), 4–21 (Mar 2011). <https://doi.org/10.1109/MAHC.2010.11>
60. Maeno, T., De, K., Wenaus, T., *et al.*: Overview of ATLAS PanDA Workload Management. Journal of Physics: Conference Series 331(7), 072024 (Dec 2011). <https://doi.org/10.1088/1742-6596/331/7/072024>
61. Maffeis, S., Schmidt, D.: Constructing reliable distributed communication systems with CORBA. IEEE Communications Magazine 35(2), 56–60 (1997). <https://doi.org/10.1109/35.565656>
62. Megino, F.B., Bawa Harinder, S., De, K., *et al.*: Seamless integration of commercial Clouds with ATLAS Distributed Computing. EPJ Web Conf. 251, 02005 (2021). <https://doi.org/10.1051/epjconf/202125102005>
63. Megino, F.B., De, K., Elmsheuser, J., *et al.*: Operational Experience and R&D results using the Google Cloud for High Energy Physics in the ATLAS experiment (2024). <https://doi.org/10.48550/arXiv.2403.15873>
64. Megino, F.B., *et al.*: Accelerating science: the usage of commercial clouds in ATLAS distributed computing (5 2023), <https://indico.jlab.org/event/459/contributions/11636/>, accessed: 2024-02-16
65. Megino, F.H.B., Jones, R., Kucharczyk, K., *et al.*: Helix nebula and cern: A symbiotic approach to exploiting commercial clouds. Journal of Physics: Conference Series 513(3), 032067 (jun 2014). <https://doi.org/10.1088/1742-6596/513/3/032067>
66. Metcalfe, R.M., Boggs, D.R.: Ethernet: distributed packet switching for local computer networks. Communications of the ACM 19(7), 395–404 (Jul 1976). <https://doi.org/10.1145/360248.360253>

67. Mhashilkar, P., Altunay, M., Berman, E., *et al.*: HEPCloud, an Elastic Hybrid HEP Facility using an Intelligent Decision Support System. EPJ Web of Conferences 214, 03060 (2019). <https://doi.org/10.1051/epjconf/201921403060>
68. Miura, K.: Overview of Japanese science Grid project NAREGI. Progress in Informatics (3), 67 (Apr 2006). <https://doi.org/10.2201/NiiPi.2006.3.7>
69. Nisan, N., London, S., Regev, O., *et al.*: Globally distributed computation over the Internet—the POPCORN project. In: Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No.98CB36183). p. 592601. IEEE Comput. Soc, Amsterdam, Netherlands (1998). <https://doi.org/10.1109/ICDCS.1998.679836>
70. Pickles, S., Brooke, J., Costen, F., *et al.*: Metacomputing across intercontinental networks. Future Generation Computer Systems 17(8), 911–918 (2001). [https://doi.org/10.1016/S0167-739X\(01\)00032-2](https://doi.org/10.1016/S0167-739X(01)00032-2), high Performance Computing and Networking
71. Pickles, S., Brooke, J., Costen, F., *et al.*: Metacomputing across intercontinental networks. Future Generation Computer Systems 17(8), 911–918 (2001). [https://doi.org/10.1016/S0167-739X\(01\)00032-2](https://doi.org/10.1016/S0167-739X(01)00032-2), high Performance Computing and Networking
72. Pordes, R., Petravick, D., Kramer, B., *et al.*: The Open Science Grid. Journal of Physics: Conference Series 78(1), 012057 (jul 2007). <https://doi.org/10.1088/1742-6596/78/1/012057>
73. Resch, M.M., Rantzau, D., Stoy, R.: Metacomputing experience in a transatlantic wide area application test-bed. Future Generation Computer Systems 15(5), 807–816 (1999). [https://doi.org/10.1016/S0167-739X\(99\)00028-X](https://doi.org/10.1016/S0167-739X(99)00028-X)
74. Rutkowski, A.M.: The Integrated Serviced digital Network: Issues and Options for the Future. Jurimetrics 24(1), 19–42 (1983), <http://www.jstor.org/stable/29761846>
75. Saiz, P., Aphetche, L., Buni, P., *et al.*: AliEnALICE environment on the GRID. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 502(23), 437–440 (Apr 2003). [https://doi.org/10.1016/S0168-9002\(03\)00462-5](https://doi.org/10.1016/S0168-9002(03)00462-5)
76. Sfiligoi, I.: glideinWMSa generic pilot-based workload management system. Journal of Physics: Conference Series 119(6), 062044 (Jul 2008). <https://doi.org/10.1088/1742-6596/119/6/062044>
77. Skovira, J., Chan, W., Zhou, H., *et al.*: The EASY – LoadLeveler API project. In: Feitelson, D.G., Rudolph, L. (eds.) Job Scheduling Strategies for Parallel Processing. pp. 41–47. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
78. Smarr, L., Catlett, C.E.: Metacomputing. Commun. ACM 35(6), 44–52 (jun 1992). <https://doi.org/10.1145/129888.129890>
79. Talia, D.: The Open Grid Services Architecture: where the grid meets the Web. IEEE Internet Computing 6(6), 67–71 (2002). <https://doi.org/10.1109/MIC.2002.1067739>

80. Thain, D., Tannenbaum, T., Livny, M.: Distributed computing in practice: the Condor experience. *Concurrency and Computation: Practice and Experience* 17(2-4), 323–356 (2005). <https://doi.org/10.1002/cpe.938>
81. The ATLAS Collaboration: Total cost of ownership and evaluation of Google cloud resources for the ATLAS experiment at the LHC (2024). <https://doi.org/10.48550/arXiv.2405.13695>
82. Ďurech, J., Hanu, J., Vano, R.: Asteroids@homeA BOINC distributed computing project for asteroid shape reconstruction. *Astronomy and Computing* 13, 80–84 (2015). <https://doi.org/10.1016/j.ascom.2015.09.004>
83. Warren, R., *et al.*: MammoGrid – a prototype distributed mammographic database for Europe. *Clinical Radiology* 62(11), 1044–1051 (2007). <https://doi.org/10.1016/j.crad.2006.09.032>
84. Xu, M.: Effective metacomputing using LSF Multicluster. In: *Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid*. pp. 100–105 (2001). <https://doi.org/10.1109/CCGRID.2001.923181>

Development of Novel Nsp16 Inhibitors as Potential Anti-SARS-CoV-2 Agents

*Kuojun Zhang*¹, *Alexey V. Sulimov*², *Ivan S. Ilin*², *Danil C. Kutov*²,
*Anna S. Taschilova*², *Sheng Jiang*¹, *Tianyu Wang*¹, *Vladimir B. Sulimov*²,
*Yibei Xiao*¹

© The Authors 2024. This paper is published with open access at SuperFri.org

Computer aided structural based approach was used to find inhibitors of SARS-CoV-2 nsp16 (2'-O-methyltransferase). Docking based virtual screening of three libraries, Enamine Coronavirus Library, Enamine Nucleoside Mimetics Library, and Chemdiv Nucleoside Analogue Library, was performed. In total, 39350 3D-structures of low molecular weight ligands were docked into a model of nsp16 prepared using the structure of 6WKQ complex from the Protein Data Bank. Docking was performed by the SOL docking program. For the best SOL scored ligands, the protein-ligand binding enthalpy was calculated using the PM7 semiempirical quantum-chemical method with the COSMO implicit solvent model. The most promising eleven compounds were purchased and their inhibitory activity against the recombinant viral nsp16 protein was measured using MST assay with Monolith NT.115. As a result, two compounds, Z195979162 and Z1333277068, from Enamine Coronavirus Library demonstrated dissociation constants K_d for nsp16/nsp10 complex equal to 2.0 and 5.0 μM . The relative stability of these ligands in their docked positions in the nsp16 S-adenosylmethionine (SAM) binding site was confirmed in the molecular dynamics simulations along 70 ns trajectories. Z195979162 and Z1333277068 compounds belong to two chemical classes: 1,4-disubstituted tetrahydropyridines and derivatives of pyrazole-5-carboxamide, respectively, and can be good starting points for further hit optimization in the field of nsp16 inhibitors design.

Keywords: nsp16, methyltransferase, virtual screening, protein-ligand binding, enzyme inhibitors, recombinant protein expression, SARS-CoV-2.

Introduction

The 2019 coronavirus disease (COVID-19) pandemic caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) still poses a great threat to the global public health. According to the WHO report, there have been 632 million confirmed cases and more than 6.5 million deaths worldwide, as of November 2022. SARS-CoV-2 is evolving and new infectious and antibody-resistant strains are emerging [17]. Although highly effective SARS-CoV-2 vaccines have been used globally, increasing number of breakthrough infections have been observed in fully vaccinated individuals, mainly due to compromised effectiveness against the emerging variants, such as Delta and Omicron [15, 28]. Nirmatrelvir, an inhibitor of SARS-CoV-2 main protease, which was granted FDA Emergency Use Authorization at the end of 2021 and discovered rationally, not by re-purposing of existing drugs represents the only direct-acting agent of such type in scarce arsenal of anti-coronavirus treatment [11, 32]. The public are hence still at high risk of being infected, and consequently the need for innovative and effective antivirals against SARS-CoV-2 and the emerging variants is still highly needed and unmet.

The fight against this disease includes various tasks, including the development of direct-acting antiviral drugs on therapeutic target proteins of the SARS-CoV-2 coronavirus. By the beginning of the pandemic, information about the mechanisms of coronavirus replication already existed thanks to research on SARS-CoV and MERS-CoV coronaviruses, which caused two major outbreaks of deadly pneumonia in 2002 and 2012, respectively [50, 51].

¹School of Pharmacy, China Pharmaceutical University, Nanjing, China

²Research Computer Center of Lomonosov Moscow State University, Moscow, Russian Federation

SARS-CoV-2's genome codes 29 proteins, several of which have been exploited as potential therapeutic targets for the development of new antivirals [18, 39, 46, 52, 53]. In this work, we chose nonstructural protein 16 (nsp16), the viral RNA 2'-O-Methyltransferase (2'-O-MTase) [45]. Nsp16 has been considered as an attractive target for antiviral therapeutics due to its key role in the formation of a viral RNA cap. Specially, nsp16 and nsp10 form a heterodimer and function to catalyze the transfer of a methyl group from its co-substrate S-adenosylmethionine (SAM, **1**) in Fig. 1 to the first transcribed nucleotide at the 2'-OH position, converting the cap-0 structure (featured as 7MeGpppA...) into the cap-1 structure (featured as 7MeGpppA2'-O-Me...) [13]. 2'-O-methylation is necessary for the formation of the cap structure in viral RNAs, which makes it possible to read the SARS-CoV-2 genetic information by the host translation mechanism. This 2'-O-MTase has been shown to be indispensable for the replication of coronaviruses in cell cultures [16, 25, 34, 35]. This methylation process plays an important role in the translation of viral proteins, protection of the nascent viral mRNA from exonucleases, and evasion of host immunity [10, 29]. SAM contains an activated methyl group and is a common co-substrate for methyl transfer reactions in living organisms. Thus, blocking the SAM binding site of nsp16 leads to inhibition of the methylation process catalyzed by this enzyme. Moreover, 2'-O-MTase/nsp16 is highly conserved among various coronavirus species, also making it a potential effective target for the development of broad-spectrum antivirals [10]. Furthermore, crystal structures of SARS-CoV-2 nsp16/nsp10 complex alone and in complex with RNA, SAM, or SAM analogs have been reported [24, 25, 27, 37, 48], which provides structural basis for structure-based inhibitor design and virtual screening approaches.

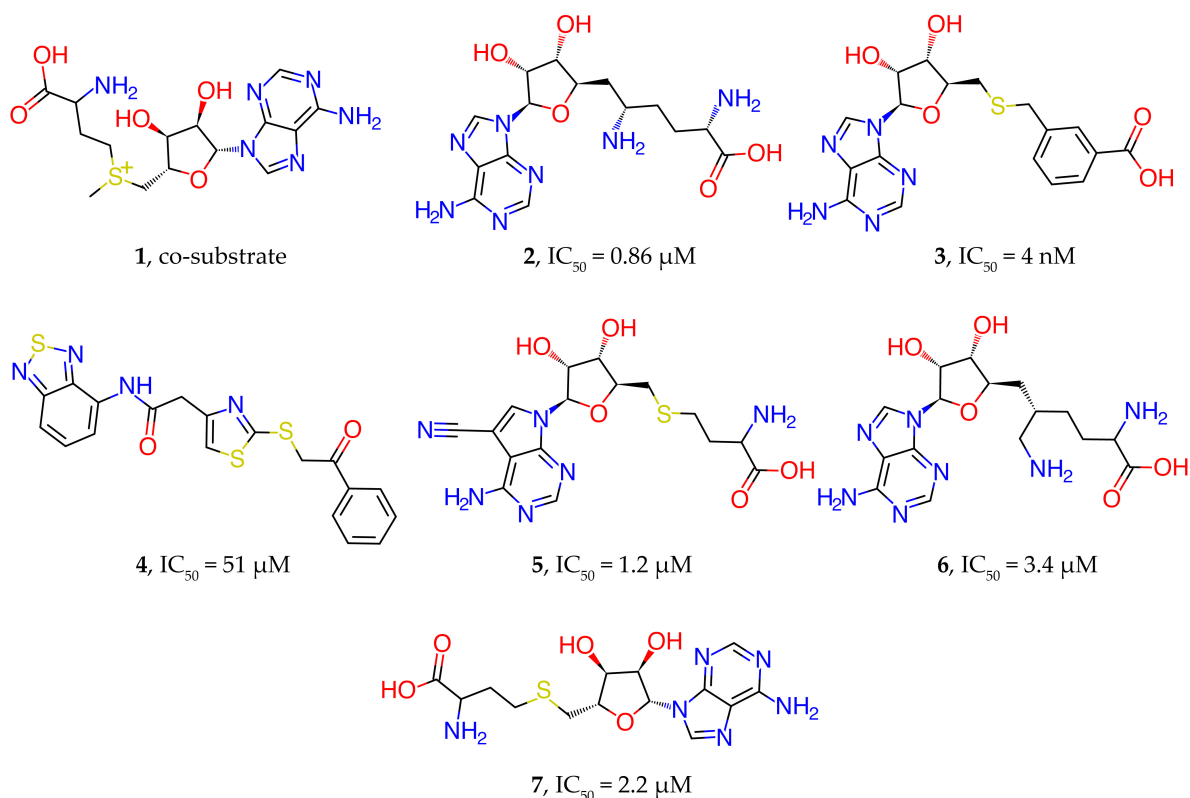


Figure 1. Experimentally confirmed nsp16 inhibitors

Nsp16 was studied during the outbreak of SARS-CoV pneumonia, which led to the identification of a strong suppressor of nsp16 methyltransferase activity, sinefungin **2** in Fig. 1 [12]. Sinefungin, like SAM, an endogenous cofactor of nsp16, binds in the same pocket. The activity of sinefungin against SARS-CoV-2 nsp16 was also confirmed by X-ray diffraction analysis [25]. Sinefungin was reported to be a potent 2'-O-MTase inhibitor against vaccinia virus, MERS-CoV, SARS-CoV and SARS-CoV-2. Recently, Bobileva et al. reported a series of sinefungin derivatives, exemplified by compound **3** (Fig. 1), by using bioisosteric substitution of the sulfonium and amino acid substructures aided by docking model [8]. Compound **3** exhibited high inhibitory potency against SARS-CoV-2 nsp16 and nsp14 with IC₅₀ values 4.0 and 8.0 nM, respectively. However, it is nonselective against human glycine N-methyltransferase (GNMT) (IC₅₀ = 8.6 nM). The main reason for the non-selectivity may be related to the very high similarity between the nsp16 inhibitors found and S-adenosylmethionine, the cofactor used by most methyltransferases.

Employing a high throughput virtual screening (vHTS) based on the docking of 7 million commercially available drug compounds and SAM analogues, Bobrovs et al. identified some inhibitors against nsp14 and nsp16 [9]. Among them, the best compound ZINC38661771 **4** (Fig. 1) with a sulfanylthiazole core exhibited inhibitory activity against nsp14 and nsp16 with IC₅₀ values of 118 and 51 μ M, respectively, but this compound was also active against GNMT with IC₅₀ value of 58 μ M.

Through cross-screening of SARS-CoV-2 nsp14 inhibitors against nsp10/nsp16 complex, Klima et al. identified two nsp16 inhibitors SS148 **5** and WZ16 **6** (Fig. 1) with IC₅₀ = 1.2 and 3.4 μ M, respectively [24]. The IC₅₀ value determined for S-adenosylhomocysteine (SAH) **7** (Fig. 1) used as a control was 2.2 ± 0.2 μ M. Also, in this work, the crystal structures of nsp10/nsp16 complex with the found inhibitors SS148 (PDB ID: 7R1T) and WZ16 (PDB ID: 7R1U) were presented.

At the moment, there are many studies using various computer modeling methods such as molecular docking, molecular dynamics, quantum chemical calculations, machine learning methods to identify nsp16 inhibitors. However, most of these studies do not include experimental validation of the inhibitory activity of the candidates found.

A large-scale study on the identification of nsp16 inhibitors was carried out in [14]. A 3D pharmacophore model based on the nsp16-sinefungin complex (PDB ID: 6WKQ) screened 48 million drug-like compounds from the ZINC database. The 24 best performing molecules were placed in the SAM binding pocket. Molecular dynamics simulations were performed for the top three compounds, and one compound **8** (see Fig. 2) with a triazine backbone was identified as a potential inhibitor of nsp16.

In a study [38], several inhibitors of nsp16 were identified by computer simulations, without confirmation of *in vitro* activity. Pharmacophore screening of the Bionet and Chemdiv databases was carried out, followed by molecular docking into the nsp16 active pocket by Glide, molecular dynamics simulations and binding free energy calculations. Based on results of virtual screening two compounds **9** and **10** were sorted out as potential nsp16 inhibitors (Fig. 2). These compounds can form a stable complex with the protein and form strong hydrogen bonds and salt bridges with the key amino acids Asp6897 and Asp6928.

Mohammad et al. discovered genquanin-6-C-beta-glucopyranoside (**11**, Fig. 2) as a potential nsp16 inhibitor via screening the North African Natural Products database for compounds that can interact with the nsp10 interface and disrupt the formation of the nsp10/nsp16 complex [30].

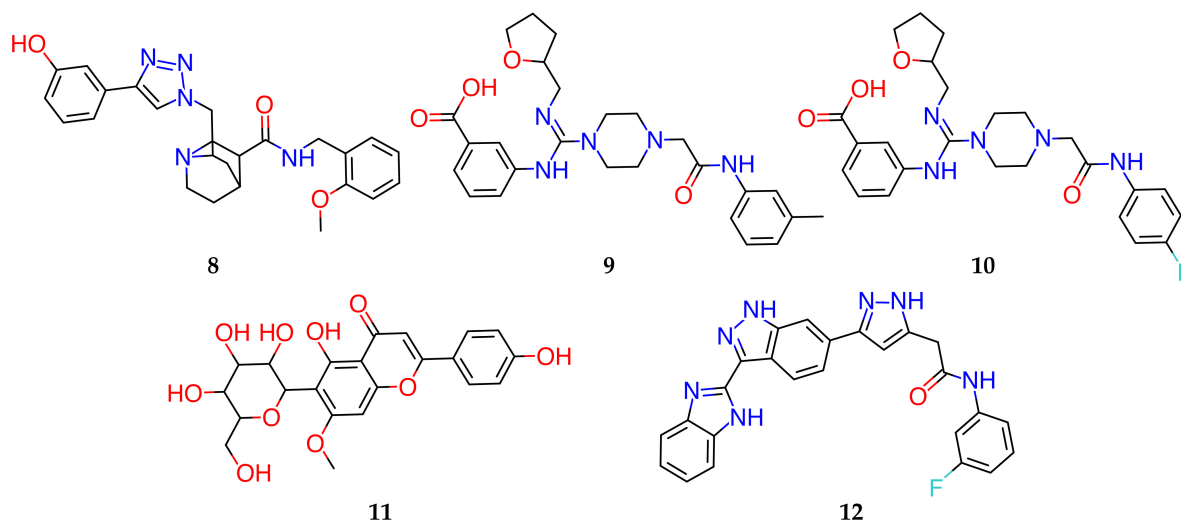


Figure 2. Inhibitors of the 2'-O-MTase activity of nsp10/nsp16 complex, predicted by molecular modeling in the literature, which have not yet received experimental confirmation of their inhibitory activity

The best binding affinity was predicted for four compounds using AutoDock Vina. Further, molecular dynamics simulations were performed for these four compounds and genquanin-6-C-beta-glucopyranoside showed the most stable complex with nsp10, but was not tested *in vitro*.

A search for inhibitors of 2'-O-MTase activity of the nsp10/nsp16 complex in the PubChem and anti-HIV databases was carried out using a multi-stage virtual screening protocol, including Lipinski's rule filtering, docking, and steered molecular dynamics [31]. As a result, the best candidate CID 135566620 (**12**, Fig. 2) from PubChem was predicted to possess inhibition constant in the submicromolar range. *In vitro* and *in vivo* testing to confirm the activity of this nsp10/nsp16 inhibitor candidate has not been performed. A virtual molecular docking was carried out to screen inhibitors against nsp10/nsp16 2'-O-MTase activity from a library of compounds that were found in the ZINC database, in plants of the Caatinga Brazilian biome, as well as structural analogues of SAM and sinefungin (SFG) at PubChem database [19]. Docking was performed using AutoDock Vina, and 24 potential 2'-O-MTase inhibitors were identified in all the studied sets of molecules, among which the most promising were 5 compounds from the ZINC database.

In this work, a virtual screening of chemical compounds from three libraries using docking was carried out, followed by a quantum-chemical calculation of the enthalpy of binding of a ligand to the protein in order to search for nsp16 inhibitors. The SAM-binding site of the enzyme was used as a target to inhibit the 2'-O-MTase activity. As a result, eleven compounds were selected for experimental validation of their binding to nsp16; for two of them, belonging to two chemical classes, 1,4-disubstituted tetrahydropyridines and derivatives of pyrazole-5-carboxamide, the binding affinity with K_d of 2.0 and 5.0 μM was experimentally confirmed. Molecular dynamics studies additionally confirmed stability of complexes of these two compounds with nsp16.

The article is organized as follows. Section 1 describes the modeling methods used in virtual screening, the study of the stability of inhibitors in docked positions using molecular dynamics, and experimental techniques for measuring the inhibitory activity of selected compounds. In section 2 we present results of our study. Conclusion summarizes the study and points directions for further work.

1. Materials and Methods

1.1. Virtual Screening

Virtual screening of three chemical libraries was performed using the SOL docking program [43, 44]. These libraries were: Enamine Coronavirus Library, Enamine Nucleoside Mimetics Library, and Chemdiv Nucleoside Analogue Library. Enamine libraries were downloaded in SDF format from here [2]. Chemdiv Nucleoside Analogue Library also in SDF format was downloaded from here [1]. Prior to ligand preparation, we filtered out PAINS (Pan Assay Interference Compounds) [6] molecules from all three libraries. The LigPrep tool [5] was applied to protonate ligands from these libraries and generate three-dimensional conformers for each ligand. In case of Enamine Coronavirus Library, from initial 16800 compounds 16561 organic molecules were left after PAINS filtering and 34405 unique 3D conformers were generated by LigPrep. For Enamine Nucleoside Mimetics Library, 276 of 290 compounds were retained during PAINS filtering which were converted into 898 conformers by LigPrep. For the third library, Chemdiv Nucleoside Analogue Library, PAINS filtering retained 2107 of 2600 initial compounds and 4047 conformers were generated for them. To sum up, 39350 unique 3D molecules were virtually screened against nsp16 in the frame of the study.

A model of the nsp16 protein was prepared using the protein structure from the 6WKQ crystal complex [37] stored in the Protein Data Bank (PDB) [7]. This complex consisted of the SARS-CoV-2 nsp16/nsp10 heterodimer with sinefungin crystallized in a SAM-binding site of nsp16 and possessed excellent resolution (1.98 Å) with only four missed residues on N-terminus. These residues were not repaired in our model due to their distant position from the site selected for docking. The protein model was prepared by removing the nsp10 part of the heterodimer and adding hydrogen atoms using the APlite program [26], which protonates protein atoms at pH 7.4. A grid of potentials was constructed by the SOLGRID module [43, 44]. The model was validated by docking known nsp16 inhibitors using the SOL program [43, 44]: sinefungin in a co-crystallized and partially neutralized form and S-Adenosyl-L-homocysteine extracted from another PDB complex, 6WJT. Positioning of sinefungin (in a partially neutralized form) resulted in the scoring-function of -7.07 kcal/mol and the reproduction of a crystalized pose (RMSD = 0.87 Å). Docking of S-Adenosyl-L-homocysteine gave scoring-function of -7.48 kcal/mol and the similar conformation observed in the complex of nsp16 with S-Adenosyl-L-homocysteine: RMSD between a docked pose and a quasi-native conformation obtained from alignment of 6WJT on 6WKQ was 0.97 Å. Details of the protein model preparation and validation are presented in [42].

The SOL docking program performed grid-based docking utilizing genetic algorithm for conformational sampling and physics-based scoring-function which contained interactions terms from the MMFF94 force field [21] and a desolvation term based on a simplified form of the Generalized Born solvation model [36]. The flexibility of a ligand was encoded by translation and rotation of a ligand as a rigid body and rotations of each rotatable bond. Since conformations of macrocycles were not sampled, they were generated preliminary for each ligand with such cycles at the stage of library preparation. The conformation of the protein was fixed during docking. Genetic algorithm used for a global optimization task was applied with the following parameters: population size was 30000, the number of generations was 1000, and the number of independent runs was 50. Independent runs helped to assess how well the conformational space of a given ligand was explored by docking. In other words, it gave one an opportunity to assess reliability of the found global energy minimum: if independent runs converged well and many of them resulted

in the same best solution, it increased certainty in the identified best docking pose. Technically, such convergence was estimated by cluster analysis. If RMSD between two poses was less than 1 Å, they were included in one cluster. Convergence was expressed in a small number of clusters and a high population for the first cluster, which contained the docked pose of the ligand with the lowest energy – the best docked pose. In each run, the best ligand docked pose corresponded to the most negative value of the objective function of the global optimization employed by the SOL docking algorithm. The objective function was the sum of the ligand energy in the protein field and the internal strain energy of the ligand calculated in the MMFF94 force field. To accelerate the virtual screening of libraries with thousands of compounds, this study used SOL installed on the Lomonosov-2 supercomputer [49] of Lomonosov Moscow State University with docking of each ligand on a separate computing core.

In order to evaluate the affinity of ligands for nsp16 better, the protein-ligand binding enthalpy ΔH_{bind} was calculated by MOPAC [4] using the PM7 quantum-chemical semiempirical method [40] for top scored ligands predicted by docking. The binding enthalpy was obtained by calculating three terms: the enthalpy of formation of a protein-ligand complex, the enthalpy of formation of an unbound ligand, and the enthalpy of formation of an unbound protein molecule. The first term was calculated in two stages including optimization of the PM7 energy of the protein-ligand complex by varying Cartesian coordinates of all ligand atoms from the initial geometry predicted by docking followed by calculation of enthalpy of formation of the protein-ligand complex for the optimized geometry using the PM7 method with the COSMO implicit solvation model [23]. To handle calculation for such big molecular system as a protein-ligand complex, MOZYME module [41] was applied. The similar procedure was performed for calculation of the enthalpy of formation of an unbound ligand. The enthalpy of formation of unbound protein was calculated for the fixed conformation used during docking. The final value of ΔH_{bind} was found according to the following equation:

$$\Delta H_{bind} = \Delta H_{complex} - \Delta H_{protein} - \Delta H_{ligand}, \quad (1)$$

$\Delta H_{complex}$ is the enthalpy of formation of the protein-ligand complex, $\Delta H_{protein}$ is the enthalpy of formation of an unbound protein, ΔH_{ligand} is the enthalpy of formation of an unbound ligand.

The final prioritization of compounds for purchasing and testing was performed with considering three criteria: the most negative values of the SOL scoring function and ΔH_{bind} , and geometric features of the ligand pose after optimization in PM7. The latter implies visual estimation of ligand complementarity to an active site of nsp16, ligand distortion and presence of key interactions between nsp16 and the ligand.

1.2. Molecular Dynamics Simulation

To illustrate stability of the nsp10/nsp16 complex with our two inhibitors, we performed a molecular dynamics (MD) simulation of the system along a relatively long trajectory of several dozen ns. Besides these two molecules, sinefungin and C692-0494 were simulated to obtain, respectively, positive and negative control. We used the NAMD [33] package (<http://www.ks.uiuc.edu/Training/Tutorials/namd>), the CHARMM36 all-atom additive force field, and NPT ensemble, when the number of the particle N, pressure P, and temperature T were constants. Initial poses were taken from docking studies.

Nsp16 protein preparation included specification of exact protonation state for histidine residues with replacing generic HIS notation with one of three options: HSD (neutral side group, proton on the ND1 atom), HSE (neutral side group, proton on the NE2 atom), HSP (charged histidine, proton on both ND1 and NE2). These states were determined manually via observation of the protein structure after protonation. Preparation of the PSF files was performed by VMD [3].

Ligands were parametrized in CHARMM general force field [47] using web tool: <https://cgenff.umaryland.edu/>. To facilitate trajectory analysis, the protein-ligand complex was moved to the origin. Solvation was carried out in VMDs solvated plugin with constructing $60 \times 60 \times 60 \text{ \AA}$ box with TIP3P water molecules and applying 8 \AA padding. The specified 8 \AA padding creates the water box sized such that in each direction from the atom with the largest coordinate in that direction there is an 8 \AA layer of water to provide enough space for the mobility of the protein-ligand complex. The autoionize plugin of VMD was applied to neutralize the system by addition of counterions. In total, each prepared system included about 40000 atoms with about 11500 water molecules. For complex nsp16 with C692-0494, two different solvation models were used: TIP3P and SPC.

The molecular dynamic simulation for each complex consisted of three steps: minimization, equilibration and production run. Firstly, minimization was performed for each complex and consisted of 5000 steps. After that, 1 ns of equilibration was followed by a 72 ns unrestrained production run at 310 K and 1 atm. Langevin dynamics was used for constant temperature control and constant pressure control. Periodic boundary conditions were applied and the time step of the MD trajectory was 1 fs. For all calculations resources of the Lomonosov-2 supercomputer [49] of Lomonosov Moscow State University were used. Calculation of equilibration step (1 ns) took about 70 minutes using 32 cores. To calculate full production run (72 ns), it took about 2800 minutes using 64 cores for each complex.

To analyse trajectories, the MDAnalysis library was applied [20]. A plot of RMSD change over time was prepared using the Matplotlib library [22]. RMSD is the root-mean square deviation of the corresponding ligand from its docked position in the enzyme.

1.3. Protein Production and Purification

The co-expressed nsp16 and nsp10 proteins were cloned into modified pETduet-1 vector, with N terminus possessing 6X His-tag in nsp10 while nsp16 bond to nsp10 without extra affinity tags. Briefly, *E. coli* BL21 (DE3) cells were transformed with the expression vector and grown at $37 \text{ }^\circ\text{C}$ in LB medium. After OD600 reached 0.6, the protein expression was induced by addition of IPTG to final concentration $500 \text{ }\mu\text{M}$ and the protein was expressed overnight at $18 \text{ }^\circ\text{C}$. The cells were harvested and the pellets were resuspended in a buffer (20 mM HEPES pH 8.0, 500 mM NaCl, 5 mM MgCl_2 , 10% glycerol) and homogenized with an ultrahigh-pressure cell disrupter at $4 \text{ }^\circ\text{C}$. The insoluble material was removed by centrifugation at 17000 rpm for 40 min. The protein complex was first purified by Ni-NTA affinity chromatography by wash buffer (20 mM HEPES pH 8.0, 500 mM NaCl, 5 mM MgCl_2 , 20 mM imidazole), then eluted by elution buffer (20 mM HEPES pH 8.0, 500 mM NaCl, 5 mM MgCl_2 , 300 mM imidazole). The proteins were purified by Superdex 10/300 columns. The nsp10/nsp16 complex was eluted as a single homogenous species in a final buffer containing 20 mM HEPES pH 8.0, 150 mM NaCl, 5 mM MgCl_2 .

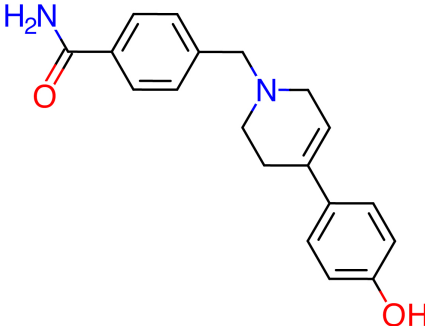
1.4. Determining Affinity of Compounds for Nsp16/Nsp10

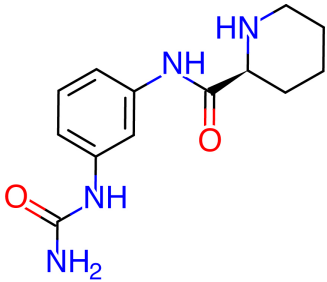
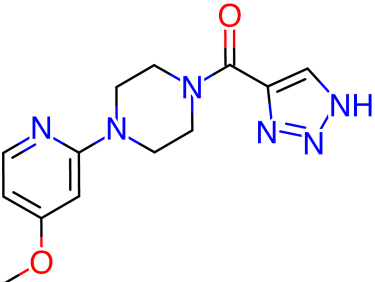
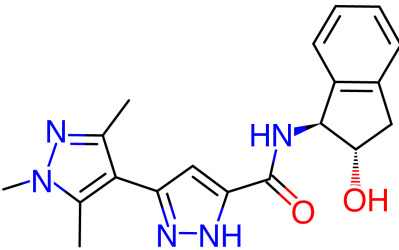
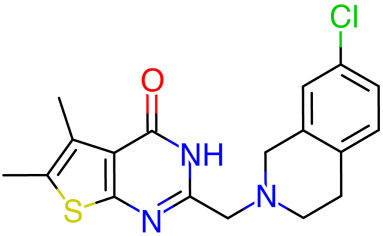
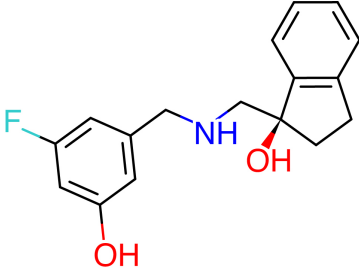
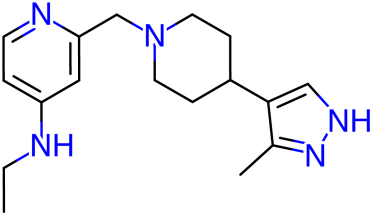
The purified nsp16/nsp10 protein complex was labeled with a protein labeling Kit (Monolith, RED-NHS 2nd Generation kit, Cat# MO-L011). In brief, 10 μM of protein was incubated with dye solution (30 μM) in the labeling buffer and the reaction was allowed to proceed at room temperature for 30 min. Each compound was dissolved in the Microscale thermophoresis (MST) reaction buffer containing 20 mM HEPES pH 8.0, 150 mM NaCl, 5 mM MgCl_2 . Two-fold serial dilutions started at specific concentration were made in 16 steps. The labeled protein at a final concentration of 20 nM was equally mixed into each reaction. The final reaction mixtures were loaded into premium capillary chips (Monolith Cat# MO-AK002) and measured on a Monolith NT.115 instrument at 100% excitation power and 40% MST power at 25 $^\circ\text{C}$.

2. Results

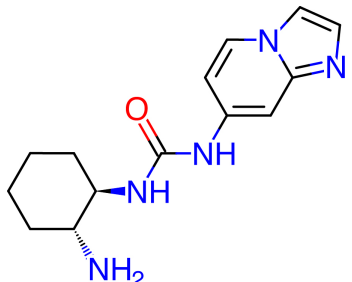
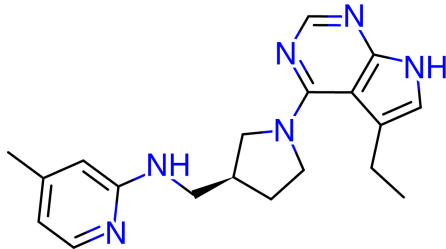
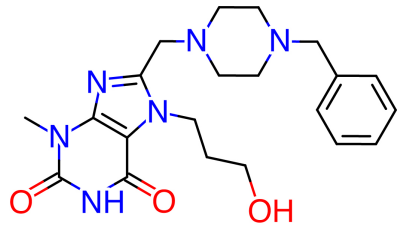
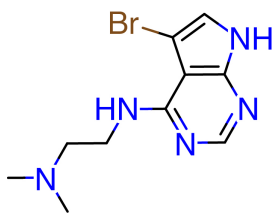
Virtual screening against nsp16 with docking into SAM-binding site of the enzyme was carried out. During in silico screening, a chemical space formed by three chemical libraries was selected: Enamine Coronavirus Library, Enamine Nucleoside Mimetics Library, and Chemdiv Nucleoside Analogue Library. After docking using SOL, 242 best scored compounds were selected for semiempirical quantum chemistry calculations to confirm or refute their affinity to nsp16. The selection criteria were described in [42]. Relying upon predicted values of the SOL score, binding enthalpies and binding modes, we selected 21 compounds for purchasing and further experimental testing. All 21 compounds were obtained, but 10 of them precipitated out of solution and their activity was not measured. For the remaining 11 compounds, dissociation constants were measured using Monolith NT.115. Structures of the measured 11 compounds and calculated values of the SOL scoring function and the binding enthalpy, as well as the measured dissociation constant K_d are given in Tab. 1.

Table 1. Best candidate inhibitors selected for experimental testing, their calculated SOL score, binding enthalpy ΔH_{bind} , and measured dissociation constant K_d

Compound	Structure	SOL score, kcal/mol	1-st cluster popul.	ΔH_{bind} , kcal/mol	K_d , μM
Z195979162		-5.24	40	-53.4	2.0

Compound	Structure	SOL score, kcal/mol	1-st cluster popul.	ΔH_{bind} , kcal/mol	K_d , μM
Z445470482		-5.54	15	-60.0	>20
Z1129470080		-6.04	36	-38.3	>20
Z1333277068		-5.24	47	-53.8	5.0
Z1538127913		-6.13	34	-40.7	>20
Z1715767396		-5.38	23	-53.0	>20
Z2045761676		-6.76	4	-30.8	>20

Development of Novel Nsp16 Inhibitors as Potential Anti-SARS-CoV-2 Agents

Compound	Structure	SOL score, kcal/mol	1-st cluster popul.	ΔH_{bind} , kcal/mol	K_d , μM
Z2396606359		-5.81	30	-54.5	>20
Z2968942047		-6.15	17	-55.4	>20
C692-0494		-5.27	10	-48.4	>20
FF01-2805		-5.03	37	-37.7	>20

To verify measured values of binding affinity of our 11 compounds with complex of nsp10/16, we performed the same MST measurements of SAM (native substrate of nsp10/16). Our MST measurements showed SAM binding affinity of 12 μM , which was consistent with reported values. We also measured activities of two published nsp16 inhibitors shown in Fig. 1: for compounds **4** and **6** the following values of K_d were obtained 123 μM and 1.4 μM , which are compatible with the published IC_{50} values, 51 μM and 3.4 μM , respectively.

The MD trajectories of inhibitors discovered in this study, Z195979162 ($K_d = 2.0 \mu\text{M}$) and Z1333277068 ($K_d = 5.0 \mu\text{M}$), are presented in Fig. 3 together with trajectories of a positive control – sinefungin ($\text{IC}_{50} = 0.86 \mu\text{M}$), and a negative control, C692-0494 ($K_d > 20 \mu\text{M}$).

In Fig. 3, we see a clear difference in the stability of the complex with sinefungin and with nsp16/nsp10 binders we found, compared to the inactive C692-0494 ligand which explicitly leaves the SAM-binding site of nsp16. The maximal values of RMSD for C692-0494, Z1333277068, Z1333277062, and sinefungin are 44.3, 6.13, 3.55, and 2.06 \AA , respectively. The mean values of RMSD calculated for the whole trajectory except for first 10 ns are of similar magnitude

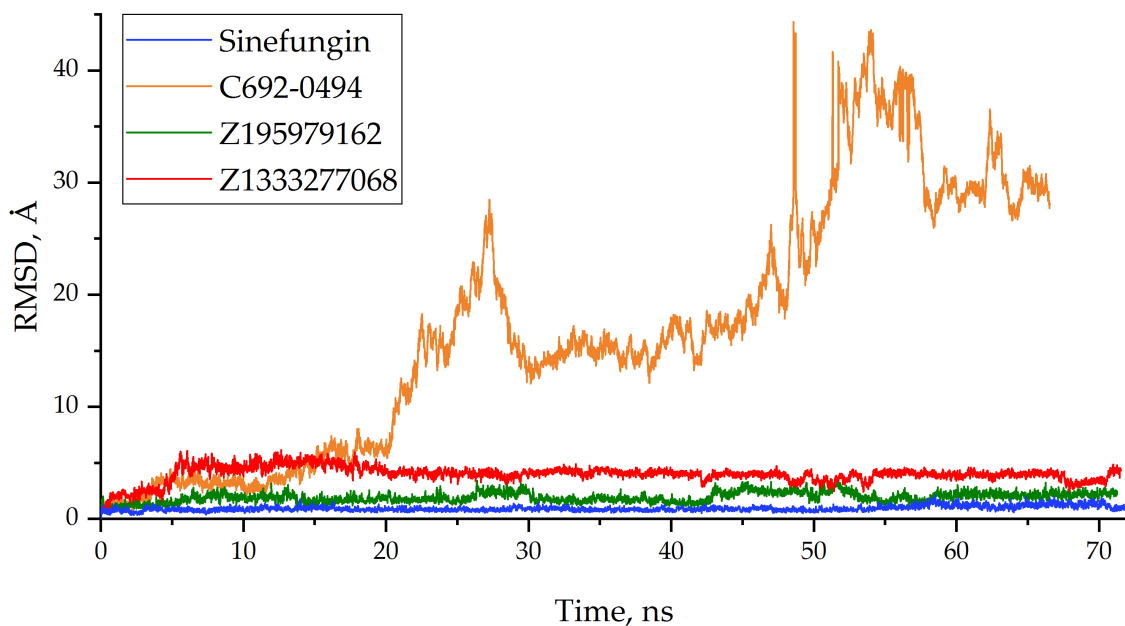


Figure 3. Stability of the nsp16 complex with each of four ligands along the MD trajectory. RMSD is the root-mean square deviation of the corresponding ligand from its docked position in the enzyme. Calculation of RMSD was performed only for heavy atoms of ligands

and equal 19.26, 4.07, 1.97, and 0.96 Å for C692-0494, Z1333277068, Z1333277062, and sinefungin, respectively. The latter shows a much stronger fixation near the docked position than the inhibitors Z1333277068 and Z1333277062. These values correspond to the observed relative 2'-O-MTase activity of these three inhibitors and the inactive ligand C692-0494. Notably, using two different water models, TIP3P and SPC, for C692-0494 resulted in similar molecular system behavior and a similar change in ligand RMSD along the trajectory.

Conclusion

COVID-19 continues to be associated with a significant health burden worldwide, and development of additional direct anti-SARS-CoV-2 treatments that act as alternatives to existing treatments is of great importance. Nsp16 protein provides important methylation reactions essential for the coronavirus life cycle and thereby represents a convenient target for which few actives are discovered to date. Using a computer-aided drug design approach, we managed to identify novel inhibitors of SARS-CoV-2 nsp16. To identify inhibitors, a virtual screening of three chemical libraries was performed using docking combined with semi-empirical quantum-chemical calculations of the enthalpy of binding of the ligand to the protein. The screened libraries contained 39350 unique 3D molecular structures. Relying upon predicted affinity values, we prioritized 11 compounds for *in vitro* testing. In MST assay, two inhibitors from Enamine Coronavirus Library, Z1333277062 and Z1333277068, were identified, which showed noticeable binding to nsp16/nsp10 complex with K_d values of 2.0 and 5.0 μM , respectively. The additional enzymatic assay might be applied in further steps to probe their inhibition profile. Compounds Z1333277062 and Z1333277068 belong to 1,4-disubstituted tetrahydropyridines and derivatives of pyrazole-5-carboxamide, respectively. In molecular dynamics studies, the complexes of these two inhibitors with nsp16 were stable for 70 ns of the simulated trajectory. Unlike existing

experimentally confirmed nsp16 inhibitors, the identified compounds do not possess structural patterns similar to the SAM co-substrate. Therefore, they may have a higher selectivity for off-target proteins than, for example, sinefungin. Thus, Z1333277062 and Z1333277068 can be good starting points for further hit expansion and optimization in the field of nsp16 inhibitors design.

Acknowledgments

The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University, including the Lomonosov super-computer [49].

This research was financially supported by the Russian Foundation for Basic Research (RFBR, Russia), Project No. 20–51–55001 (A.S., D.K., I.I., A.T., and V.S. are grateful to the RFBR), and the National Natural Foundation of China (NSFC, China), Project No. 82161138005 (S.J. and Y.X. are grateful to the NSFC). All experimental works were carried out at the expense of Project No. 82161138005. Molecular modeling (construction of atomistic model of nsp10/nsp16 complex, docking, quantum chemistry, and MD calculations) were carried out at the expense of Project No. 20–51–55001.

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. Chemdiv, nucleoside mimetics library, <https://www.chemdiv.com/catalog/focused-and-targeted-libraries/nucleoside-mimetics-library/>
2. Enamine, targeted libraries, <https://enamine.net/compound-libraries/targeted-libraries>
3. VMD: Visual Molecular Dynamics, <http://www.ks.uiuc.edu/Research/vmd/>
4. Stewart Computational Chemistry. MOPAC2016 (2016), <http://openmopac.net/MOPAC2016.html>
5. Schrodinger Release 2019-3: LigPrep (2019), <https://www.schrodinger.com/products/ligprep>
6. Baell, J.B., Holloway, G.A.: New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *Journal of Medicinal Chemistry* 53(7), 2719–2740 (2010). <https://doi.org/10.1021/jm901137j>
7. 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>
8. Bobieva, O., Bobrovs, R., Kaepe, I., *et al.*: Potent SARS-CoV-2 mRNA Cap Methyltransferase Inhibitors by Bioisosteric Replacement of Methionine in SAM Cosubstrate. *ACS Medicinal Chemistry Letters* 12(7), 1102–1107 (2021). <https://doi.org/10.1021/acsmchemlett.1c00140>



9. Bobrovs, R., Kanepe, I., Narvaiss, N., *et al.*: Discovery of SARS-CoV-2 Nsp14 and Nsp16 Methyltransferase Inhibitors by High-Throughput Virtual Screening. *Pharmaceuticals* 14(12) (2021). <https://doi.org/10.3390/ph14121243>
10. Chang, L.J., Chen, T.H.: NSP16 2'-O-MTase in Coronavirus Pathogenesis: Possible Prevention and Treatments Strategies. *Viruses* 13(4) (2021). <https://doi.org/10.3390/v13040538>
11. Citarella, A., Dimasi, A., Moi, D., *et al.*: Recent Advances in SARS-CoV-2 Main Protease Inhibitors: From Nirmatrelvir to Future Perspectives. *Biomolecules* 13(9) (2023). <https://doi.org/10.3390/biom13091339>
12. Decroly, E., Debarnot, C., Ferron, F., *et al.*: Crystal Structure and Functional Analysis of the SARS-Coronavirus RNA Cap 2'-O-Methyltransferase nsp10/nsp16 Complex. *PLOS Pathogens* 7(5), 1–14 (05 2011). <https://doi.org/10.1371/journal.ppat.1002059>
13. Decroly, E., Imbert, I., Coutard, B., *et al.*: Coronavirus Nonstructural Protein 16 Is a Cap-0 Binding Enzyme Possessing (Nucleoside-2'O)-Methyltransferase Activity. *Journal of Virology* 82(16), 8071–8084 (2008). <https://doi.org/10.1128/jvi.00407-08>
14. El Hassab, M.A., Ibrahim, T.M., Shoun, A.A., *et al.*: In silico identification of potential SARS COV-2 2'-O-methyltransferase inhibitor: fragment-based screening approach and MM-PBSA calculations. *RSC Advances* 11, 16026–16033 (2021). <https://doi.org/10.1039/D1RA01809D>
15. Feikin, D., Higdon, M., Abu-Raddad, L., *et al.*: Duration of effectiveness of vaccines against SARS-CoV-2 infection and COVID-19 disease: results of a systematic review and meta-regression. *The Lancet* 399(10328), 924–944 (02 2022). [https://doi.org/10.1016/S0140-6736\(22\)00152-0](https://doi.org/10.1016/S0140-6736(22)00152-0)
16. Ferron, F., Decroly, E., Selisko, B., Canard, B.: The viral RNA capping machinery as a target for antiviral drugs. *Antiviral Research* 96(1), 21–31 (2012). <https://doi.org/10.1016/j.antiviral.2012.07.007>
17. Gao, K., Wang, R., Chen, J., *et al.*: Methodology-Centered Review of Molecular Modeling, Simulation, and Prediction of SARS-CoV-2. *Chemical Reviews* 122(13), 11287–11368 (2022). <https://doi.org/10.1021/acs.chemrev.1c00965>
18. Gil, C., Ginex, T., Maestro, I., *et al.*: COVID-19: Drug Targets and Potential Treatments. *Journal of Medicinal Chemistry* 63(21), 12359–12386 (2020). <https://doi.org/10.1021/acs.jmedchem.0c00606>
19. Gomes, J.P.A., de Oliveira Rocha, L., Leal, C.E.Y., de Alencar Filho, E.B.: Virtual screening of molecular databases for potential inhibitors of the NSP16/NSP10 methyltransferase from SARS-CoV-2. *Journal of Molecular Structure* 1261, 132951 (2022). <https://doi.org/10.1016/j.molstruc.2022.132951>
20. Gowers, R., Linke, M., Barnoud, J., *et al.*: MDAnalysis: A Python package for the rapid analysis of molecular dynamics simulations. In: *SciPy*. pp. 98–105 (01 2016). <https://doi.org/10.25080/Majora-629e541a-00e>

21. Halgren, T.A.: MMFF VII. Characterization of MMFF94, MMFF94s, and other widely available force fields for conformational energies and for intermolecular-interaction energies and geometries. *Journal of Computational Chemistry* 20(7), 730–748 (1999). [https://doi.org/10.1002/\(SICI\)1096-987X\(199905\)20:7<730::AID-JCC8>3.0.CO;2-T](https://doi.org/10.1002/(SICI)1096-987X(199905)20:7<730::AID-JCC8>3.0.CO;2-T)
22. Hunter, J.: Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* 9, 90–95 (06 2007). <https://doi.org/10.1109/MCSE.2007.55>
23. Klamt, A., Schuurmann, G.: COSMO: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *Journal of the Chemical Society, Perkin Transactions 2* pp. 799–805 (1993). <https://doi.org/10.1039/P29930000799>
24. Klima, M., Khalili Yazdi, A., Li, F., *et al.*: Crystal structure of SARS-CoV-2 nsp10nsp16 in complex with small molecule inhibitors, SS148 and WZ16. *Protein Science* 31(9), e4395 (2022). <https://doi.org/10.1002/pro.4395>
25. Krafcikova, P., Silhan, J., Nencka, R., Boura, E.: Structural analysis of the SARS-CoV-2 methyltransferase complex involved in RNA cap creation bound to sinefungin. *Nature Communications* 11(1), 3717 (07 2020). <https://doi.org/10.1038/s41467-020-17495-9>
26. Kutov, D.C., Katkova, E.V., Sulimov, A.V., *et al.*: Influence of the method of hydrogen atoms incorporation into the target protein on the protein-ligand binding energy. *Bulletin of the South Ural State University. Series “Mathematical Modelling, Programming and Computer Software”* 10(3), 94–107 (2017). <https://doi.org/10.14529/mmp170308>
27. Lin, S., Chen, H., Ye, F., *et al.*: Crystal structure of SARS-CoV-2 nsp10/nsp16 2'-O-methylase and its implication on antiviral drug design. *Signal Transduction and Targeted Therapy* 5(1), 131 (12 2020). <https://doi.org/10.1038/s41392-020-00241-4>
28. Lipsitch, M., Krammer, F., Regev-Yochay, G., *et al.*: SARS-CoV-2 breakthrough infections in vaccinated individuals: measurement, causes and impact. *Nature Reviews Immunology* 22(1), 57–65 (12 2021). <https://doi.org/10.1038/s41577-021-00662-4>
29. Malone, B., Urakova, N., Snijder, E., Campbell, E.: Structures and functions of coronavirus replication/transcription complexes and their relevance for SARS-CoV-2 drug design. *Nature Reviews Molecular Cell Biology* 23(1), 21–39 (11 2021). <https://doi.org/10.1038/s41580-021-00432-z>
30. Mohammad, A., Alshawaf, E., Marafie, S.K., *et al.*: Molecular Simulation-Based Investigation of Highly Potent Natural Products to Abrogate Formation of the nsp10nsp16 Complex of SARS-CoV-2. *Biomolecules* 11(4), 573 (2021). <https://doi.org/10.3390/biom11040573>
31. Nguyen, H.L., Thai, N.Q., Li, M.S.: Identifying inhibitors of NSP16-NSP10 of SARS-CoV-2 from large databases. *Journal of Biomolecular Structure and Dynamics* 41(15), 7045–7054 (2023). <https://doi.org/10.1080/07391102.2022.2114941>
32. Owen, D.R., Allerton, C.M.N., Anderson, A.S., *et al.*: An oral SARS-CoV-2 Mpro inhibitor clinical candidate for the treatment of COVID-19. *Science* 374(6575), 1586–1593 (2021). <https://doi.org/10.1126/science.abl4784>

33. Phillips, J.C., Hardy, D.J., Maia, J.D.C., *et al.*: Scalable molecular dynamics on CPU and GPU architectures with NAMD. *The Journal of Chemical Physics* 153(4), 044130 (2020). <https://doi.org/10.1063/5.0014475>
34. Ramanathan, A., Robb, G.B., Chan, S.H.: mRNA capping: biological functions and applications. *Nucleic Acids Research* 44(16), 7511–7526 (06 2016). <https://doi.org/10.1093/nar/gkw551>
35. Romano, M., Ruggiero, A., Squeglia, F., *et al.*: A Structural View of SARS-CoV-2 RNA Replication Machinery: RNA Synthesis, Proofreading and Final Capping. *Cells* 9(5), 1267 (2020). <https://doi.org/10.3390/cells9051267>
36. Romanov, A.N., Jabin, S.N., Martynov, Y.B., *et al.*: Surface Generalized Born method: a simple, fast and precise implicit solvent model beyond the Coulomb approximation. *The Journal of Physical Chemistry A: Molecules, Clusters, and Aerosols* 108(43), 9323–9327 (2004). <https://doi.org/10.1021/jp046721s>
37. Rosas-Lemus, M., Minasov, G., Shuvalova, L., *et al.*: High-resolution structures of the SARS-CoV-2 2'-O-methyltransferase reveal strategies for structure-based inhibitor design. *Science Signaling* 13(651), eabe1202 (2020). <https://doi.org/10.1126/scisignal.abe1202>
38. Shi, L., Wen, Z., Song, Y., *et al.*: Computational investigation of potent inhibitors against SARS-CoV-2 2'-O-methyltransferase (nsp16): Structure-based pharmacophore modeling, molecular docking, molecular dynamics simulations and binding free energy calculations. *Journal of Molecular Graphics and Modelling* 117, 108306 (2022). <https://doi.org/10.1016/j.jmgm.2022.108306>
39. Silva, L.R., da Silva Santos-Junior, P.F., de Andrade Brandao, J., *et al.*: Druggable targets from coronaviruses for designing new antiviral drugs. *Bioorganic & Medicinal Chemistry* 28(22), 115745 (2020). <https://doi.org/10.1016/j.bmc.2020.115745>
40. Stewart, J.: Optimization of parameters for semiempirical methods VI: More modifications to the NDDO approximations and re-optimization of parameters. *Journal of molecular modeling* 19(1), 1–32 (1 2013). <https://doi.org/10.1007/s00894-012-1667-x>
41. Stewart, J.J.: Application of localized molecular orbitals to the solution of semiempirical self-consistent field equations. *International Journal of Quantum Chemistry* 58(2), 133–146 (1996). [https://doi.org/10.1002/\(SICI\)1097-461X\(1996\)58:2%3C133::AID-QUA2%3E3.0.CO;2-Z](https://doi.org/10.1002/(SICI)1097-461X(1996)58:2%3C133::AID-QUA2%3E3.0.CO;2-Z)
42. Sulimov, A., Kutov, D., Ilin, I., *et al.*: Novel Inhibitors of 2'-O-Methyltransferase of the SARS-CoV-2 Coronavirus. *Molecules* 27(9), 2721 (2022). <https://doi.org/10.3390/molecules27092721>
43. Sulimov, A.V., Kutov, D.C., Oferkin, I.V., *et al.*: Application of the Docking Program SOL for CSAR Benchmark. *Journal of Chemical Information and Modeling* 53(8), 1946–1956 (2013). <https://doi.org/10.1021/ci400094h>

44. Sulimov, V.B., Ilin, I.S., Kutov, D.C., Sulimov, A.V.: Development of docking programs for Lomonosov supercomputer. *Journal of the Turkish Chemical Society Section A: Chemistry* 7(1), 259–276 (Feb 2020). <https://doi.org/10.18596/jotcsa.634130>
45. Tazikeh-Lemeski, E., Moradi, S., Raoufi, R., *et al.*: Targeting SARS-COV-2 non-structural protein 16: a virtual drug repurposing study. *Journal of Biomolecular Structure and Dynamics* 39(13), 4633–4646 (2021). <https://doi.org/10.1080/07391102.2020.1779133>
46. Tiwari, V., Beer, J.C., Sankaranarayanan, N.V., *et al.*: Discovering small-molecule therapeutics against SARS-CoV-2. *Drug Discovery Today* 25(8), 1535–1544 (2020). <https://doi.org/10.1016/j.drudis.2020.06.017>
47. Vanommeslaeghe, K., Hatcher, E., Acharya, C., *et al.*: CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *Journal of Computational Chemistry* 31(4), 671–690 (2010). <https://doi.org/10.1002/jcc.21367>
48. Viswanathan, T., Arya, S., Chan, S.H., *et al.*: Structural basis of RNA cap modification by SARS-CoV-2. *Nature Communications* 11(1), 3718 (07 2020). <https://doi.org/10.1038/s41467-020-17496-8>
49. 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. *Supercomputing Frontiers and Innovations* 6(2), 4–11 (Jun 2019). <https://doi.org/10.14529/jsfi190201>
50. Walls, A.C., Park, Y.J., Tortorici, M.A., *et al.*: Structure, Function, and Antigenicity of the SARS-CoV-2 Spike Glycoprotein. *Cell* 181(2), 1735 (2020). <https://doi.org/10.1016/j.cell.2020.02.058>
51. Wrapp, D., Wang, N., Corbett, K., *et al.*: Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science* 367(6483), 1260–1263 (02 2020). <https://doi.org/10.1126/science.abb2507>
52. Wu, C., Liu, Y., Yang, Y., *et al.*: Analysis of therapeutic targets for SARS-CoV-2 and discovery of potential drugs by computational methods. *Acta Pharmaceutica Sinica B* 10(5), 766–788 (2020). <https://doi.org/10.1016/j.apsb.2020.02.008>
53. Wu, Y., Li, Z., Zhao, Y.S., *et al.*: Therapeutic targets and potential agents for the treatment of COVID-19. *Medicinal Research Reviews* 41(3), 1775–1797 (2021). <https://doi.org/10.1002/med.21776>

Modeling the Recovery of the Earth's Gravitational Field from Satellite Measurements Using Parallel Computations

Aleksandr S. Zhamkov¹, Vadim K. Milyukov¹ , Sergey V. Ayukov¹ ,
Aleksandr I. Filetkin¹, Igor Yu. Vlasov¹, Vladimir E. Zharov¹

© The Authors 2024. This paper is published with open access at SuperFri.org

Global models of the Earth's gravitational field, built from data collected by space geodetic missions, play a very important role in studying global processes across Earth's various geospheres. The paper is devoted to the development of a program for the recovery of the Earth's gravity field parameters. This program will enable in the future to process the results of measurements from the Russian satellite constellation and to build gravity field models of different spatial and temporal resolution. The recovery of the gravity field from satellite measurements is a rather resource-consuming computational process, and parallel computations are crucial for its optimization. This paper describes the mathematical model, the algorithm and the results of parallelization, as well as presents the results of the gravity field recovery using parallel computations working with real measurement data. The static model of the Earth's gravitational field MSU2024-1 was built using the GRACE-FO mission data for the whole year 2021. The model is decomposed to degrees and orders of $n = m = 120$ and presented in terms of geoid heights. We also compared the EGF recovery on a monthly interval using the GRACE-FO data obtained in this work with the CSR, GFZ, and JPL temporal models built at other world centers.

Keywords: Earth's gravity field, space gravimetry, gravity field recovery, parallel computations.

Introduction

Depending on the granularity of the representation and measurement methods of the Earth's gravity field (EGF), three components of the gravity field can be distinguished: local, regional and global. The local level of the EGF is provided by land and sea gravimetric surveys, as well as by airborne gravimetry. The regional component is represented by satellite altimetry of the Earth's surface and satellite laser ranging. The global gravity field is measured by satellite gravimetry. Of course, such division of the gravity field into three components is not strictly defined.

Satellite gravimetry began to develop since the beginning of this century and is represented by such missions as CHAMP (orbited during 2000–2010) [18], GRACE (2002–2017) [20], GOCE (2009–2013) [2] and GRACE Follow On (2018 to present time) [9, 11].

While the CHAMP and GOCE missions were single spacecraft, the GRACE and GRACE Follow On missions consist of a pair of identical spacecraft in the same near-polar Earth orbit, following each other at a distance of about 200 km. The main measured quantity in such satellite configuration is the inter-satellite range, which varies depending on the gravitational field of the terrain (oceans, mountains, etc.) over which the satellites of the constellation are currently flying. The measurements of the inter-satellite range provide the necessary information for the solution of the inverse problem of recovering the Earth gravity field. The main instrument that measures the inter-satellite range in the GRACE and GRACE-FO missions is a high-precision microwave interferometric system, which operates at a micrometer accuracy level [21, 22]. In parallel with the microwave system, the GRACE-FO mission satellites are equipped with a laser interferometric system, which allows to raise the measurement accuracy to a nanometer level [1].

¹Sternberg Astronomical Institute, M. V. Lomonosov Moscow State University, Moscow, Russian Federation

Twin satellite configurations have proven to be highly effective in providing high-resolution global EGF models, so there is a high probability of launching new missions of that type in the future. In particular, as a part of the Chinese TianQin space gravitational wave detector project [12], there are plans to launch the TianQin 2 constellation, similar to GRACE-FO, at the end of 2025 [25].

With the beginning of data collecting from near-Earth gravity missions, static global models of the EGF began to be developed. In addition, data from the GRACE and GRACE-FO missions allowed the production of monthly models, i.e., with a temporal resolution of one month. A complete and regularly updated list of such models is published on the website of the International Center for Global Earth Models (ICGEM) [8]. EGF models are represented as a series expansion of spherical harmonics, whose degrees and orders characterize the spatial resolution of the model, or, in other words, the detalization of the EGF representation. The first static EGF model produced using the CHAMP mission data, the EIGEN-1 model, was published in 2002 [19]. EGF models can be based either on data from a single mission or on a combination of data from different missions. As an example, one of the most recent (2023) EGF models is the WHU-SWPU-GOGR2022S model built from GOCE and GRACE data; this model is complete to degrees and orders of 300 [24].

There are static models of the EGF built from data from the GRACE missions only. The first such models are the ITG-Grace03 [14] and ITG-Grace2010s [15]. Both models are represented as a decomposition up to 180 spherical harmonics. The most recent model, which uses all GRACE observations since April 2002, is the ITSG-Grace2018s model [10, 16].

Monthly models of the EGF built from GRACE and GRACE-FO data are created in three international centers: Center for Space Research at University of Texas, Austin (CSR), Helmholtz Centre Potsdam German Research Centre for Geosciences (GFZ), and Jet Propulsion Laboratory (JPL). The solutions obtained in these centers are also available on the ICGEM website.

Designing a similar Russian space mission to measure the Earth's gravitational field is also being considered. Some research on this subject is being carried out in a number of scientific and industrial organizations of the country, not only on the development of measurement systems and scientific background, but also on the modeling of the EGF recovery from satellite measurements. The latter is a rather resource-consuming computational process. The number of unknown values in the recovery of the EGF parameters depends quadratically on the degree of decomposition. Therefore, when recovering the EGF up to degree $N_{max} = 200$, it will be necessary to operate with matrices of 40000×40000 and more. This requires hundreds of gigabytes of RAM to store such amount of data if double precision numbers are being used. In this regard, parallel computation in the EGF recovery problem plays a crucial role in optimizing the computational process. This paper presents the results of such modeling using parallel computing based on real data from the GRACE Follow On mission.

The paper is organized as follows. Section 1 presents the mathematical model of the gravity field recovery from satellite measurements. Section 2 describes the parallelization algorithm of the computational process. The results of the parallelization are given in Section 3. Section 4 contains the results of the EGF recovery. The conclusion summarizes the study.

1. Mathematical Model

The GRACE and GRACE-FO missions have provided the scientific community with a significant amount of inter-satellite measurements. In order to use this data to recover the EGF, it is crucial to have prior knowledge of the exact positions of the spacecraft in orbit, as well as their relative locations. The main objective is to assess the impact of all potential effects, including both gravitational and non-gravitational forces, on the motion of the spacecraft. Therefore a mathematical model of spacecraft motion has been developed to numerically integrate the equations of their orbital motion.

The main forces considered in modeling such motion include the non-sphericity of the geopotential, the attraction of the Moon, Sun and planets, solid tides in the Earth's body, ocean tides, resistance of the atmosphere, and solar radiation pressure. Each of these forces can cause on a GRACE-type spacecraft an acceleration greater than 10^{-9} m/s². Accelerations below this value will be compensated for by a non-gravitational acceleration compensation system and can be disregarded. It is worth noting that, when forming a complete mathematical model of inter-satellite measurements, it is necessary to take into account the relativistic effects of signal passage from one spacecraft to another.

As mentioned in the introduction, recovering the EGF is a resource-consuming procedure. Therefore, the idea of using parallel computations in structures that can be parallelized arises naturally. Examples of such structures include the non-sphericity of the EGF and ocean tides.

The nonsphericity of the Earth's gravitational potential is the main factor to be considered when modeling spacecraft motion with high precision in low-Earth orbit. Mathematically the non-sphericity of the EGF is determined by the Stokes coefficients. These coefficients form the base of its decomposition by Legendre polynomials. In this work, we utilize the Belikov and Taibatorov algorithm for decomposing the EGF [5]. The main advantage of the algorithm is its novel normalization of Legendre polynomials, which eliminates the need to recalculate coefficients during the recurrent procedure.

The expansion of the EGF in this algorithm is achieved using the following expressions:

$$\begin{aligned}
 U &= \frac{GM}{r} + \frac{GM}{r} \left[\sum_{n=2}^{N_{max}} \sum_{m=0}^n T_n^m (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) \right], \\
 T_n^m &= \frac{R}{r} \left[\cos \theta T_{n-1}^m - \frac{\sin \theta}{4} (T_{n-1}^{m+1} - 4T_{n-1}^{m-1}) \right], \\
 \frac{\partial T_n^m}{\partial \theta} &= -n \frac{R}{r} [\sin \theta T_{n-1}^m - 4T_{n-1}^{m-1}], \\
 T_0^0 &= 1, \quad T_n^{-1} = -\frac{1}{4} T_n^1,
 \end{aligned} \tag{1}$$

where U is the Earths gravitational potential; GM is the Earths gravitational parameter; R is the mean equatorial radius of the Earth; r is the geocentric distance to a given point; θ is the polar angle ($0 \leq \theta \leq \pi$); λ is the longitude ($0 \leq \lambda < 2\pi$); n and m are degree and order of the EGF expansion; C_{nm} and S_{nm} are dimensionless harmonic coefficients (the Stokes coefficients); N_{max} is the maximal degree of the expansion.

The coordinate derivatives of the gravitational potential give the acceleration of the satellite along the corresponding axes as functions of the new variables T_n^m :

$$\left\{ \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}, \frac{\partial U}{\partial z} \right\} = F \left(T_n^m, \frac{\partial T_n^m}{\partial \theta} \dots \right), \tag{2}$$

which simplifies the calculation of the accelerations.

The following are the expressions for the derivatives of geopotential with respect to spherical coordinates:

$$\begin{aligned}\frac{\partial U}{\partial r} &= -\frac{GM}{r^2} \sum_{n=2}^{N_{max}} \sum_{m=0}^n (n+1) T_n^m (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda), \\ \frac{\partial U}{\partial \theta} &= \frac{GM}{r} \sum_{n=2}^{N_{max}} \sum_{m=0}^n \frac{\partial T_n^m}{\partial \theta} (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda), \\ \frac{\partial U}{\partial \lambda} &= \frac{GM}{r} \sum_{n=2}^{N_{max}} \sum_{m=1}^n m T_n^m (C_{nm} \cos m\lambda - S_{nm} \sin m\lambda).\end{aligned}\tag{3}$$

The Cartesian derivatives of the EGF, which are used in the numerical integration of the satellites equations of motion, are obtained from (3) by the following relations:

$$\begin{aligned}\frac{\partial U}{\partial x} &= \sin \theta \cos \lambda \frac{\partial U}{\partial r} + \frac{\cos \theta \cos \lambda}{r} \frac{\partial U}{\partial \theta} - \frac{\sin \lambda}{r \sin \theta} \frac{\partial U}{\partial \lambda}, \\ \frac{\partial U}{\partial y} &= \sin \theta \sin \lambda \frac{\partial U}{\partial r} + \frac{\cos \theta \sin \lambda}{r} \frac{\partial U}{\partial \theta} + \frac{\cos \lambda}{r \sin \theta} \frac{\partial U}{\partial \lambda}, \\ \frac{\partial U}{\partial z} &= \cos \theta \frac{\partial U}{\partial r} - \frac{\sin \theta}{r} \frac{\partial U}{\partial \theta}.\end{aligned}\tag{4}$$

It is also worth noting that the Belikov and Taibatorov algorithm is a recurrent ‘‘column’’ type algorithm. This means that the right-hand side contains functions with the same index $n - 1$, as opposed to a ‘‘row’’ type algorithm. This property makes the algorithm fast and stable even for calculations with large values of n and allows it to be efficiently parallelized, since the summation limit N_{max} can reach the values $N_{max} = 200$.

The expression used to calculate the impact of ocean tides also employs double summation. The corrections to the Stokes coefficients take into account the effect of ocean tides in the same way as for solid tides. The formula for calculating these corrections is as follows:

$$[\Delta \bar{C}_{nm} - i \Delta \bar{S}_{nm}](t) = \sum_f \sum_{+}^{-} \left(C_{f,nm}^{\pm} \mp i S_{f,nm}^{\pm} \right) e^{\pm i \theta_f(t)},\tag{5}$$

where $C_{f,nm}^{\pm}$ and $S_{f,nm}^{\pm}$ are the harmonic amplitudes of the geopotential for the tidal component at frequency f [17]. These amplitudes are model-dependent. We used the FES2014 ocean tide model [4, 6, 13], which considers degrees and orders up to 100 for all tidal waves, except for long-period tidal waves, which are calculated up to and including the 50th degree.

Finally, the third part, which can be significantly optimized by introducing parallel computations, is directly related to the procedure of the EGF recovery performed with the least-squares method. During the recovery process, all observations made on the spacecraft are divided into small 30-minute intervals (arcs), and the same observation accuracy is assumed for each arc. Thus, the following normal equations can be formed for each arc:

$$\mathbf{N}_i \Delta \mathbf{x} = \mathbf{b}_i,\tag{6}$$

with

$$\mathbf{N}_i = \begin{pmatrix} l_{11} & l_{12} & \dots & l_{1n} \\ l_{21} & l_{22} & \dots & l_{2n} \\ \dots & \dots & \ddots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix}_i \quad (7)$$

and

$$\mathbf{b}_i = (b_1, b_2 \dots b_n)^T, \quad (8)$$

where the components are given by

$$\begin{aligned} l_{kj} &= \sum_{i=1}^m \omega_i a_k^{(i)} a_j^{(i)} \quad (k, j = 1, 2, \dots, n), \\ b_k &= \sum_{i=1}^m \omega_i a_k^{(i)} \Delta \xi_i \quad (k = 1, 2, \dots, n), \\ a_k^{(i)} &= \frac{\partial \ddot{r}_i^C}{\partial x_k} = \frac{(\mathbf{r}_1 - \mathbf{r}_2)}{D} \frac{\partial (\ddot{\mathbf{r}}_1 - \ddot{\mathbf{r}}_2)}{\partial x_k} \end{aligned} \quad (9)$$

for $x_k = C_{nm}$ and S_{nm} . Here $\mathbf{r}_{1,2} = \{X_{1,2}, Y_{1,2}, Z_{1,2}\}$ is the radius-vector of each satellite in the GCRS inertial frame; $\ddot{\mathbf{r}}_{1,2} = \{a_{x1,2}, a_{y1,2}, a_{z1,2}\}$ is the acceleration vector of each spacecraft; $D = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2}$ is the distance between spacecrafts; ω_i is the observation weights, such that $\sqrt{\omega_i} = 1/\sigma_i$, where σ_i is the standard deviation of the observations for the i -th arc; $\Delta \xi_i = \ddot{r}_i^O - \ddot{r}_i^C$ is the difference between the observed and modeled inter-satellite accelerations.

The normal equations for each arc (with a total duration of about 30 days for a monthly model) can be combined into a general system:

$$\mathbf{N} \Delta \mathbf{x} = \mathbf{b}, \quad (10)$$

where

$$\begin{aligned} \mathbf{N} &= \sum_{i=1}^I \frac{1}{\sigma_i^2} \mathbf{N}_i, \\ \mathbf{b} &= \sum_{i=1}^I \frac{1}{\sigma_i^2} \mathbf{b}_i, \end{aligned} \quad (11)$$

and I is the total number of arcs. This combination is done under the assumption that the covariance matrices of each group of observations are independent. The value σ_k is the measurement error of the observation group k . The solution of the system can be easily found in the form

$$\Delta \mathbf{x} = \mathbf{N}^{-1} \mathbf{b}. \quad (12)$$

Therefore, the calculation and summation can be performed in parallel for individual arcs.

2. Parallelization Algorithm

The mathematical model of the EGF recovery, as described above, includes models for calculating geopotential non-sphericity and ocean tides. Measurements of the execution times

of the program modules corresponding to these models have shown that these two algorithms are the most time-consuming. Since the EGF recovery algorithm is programmed in Fortran language, it was possible to use OpenMP parallelization tools [7] with the help of the PARALLEL DO/END PARALLEL DO construct, which was utilized to compute the ocean tide corrections (Fig. 1).

```

!$OMP PARALLEL DO PRIVATE(arg, ca, sa)
do j=0,100
do i=0,100

dC_oc(i,j) = 0.d0
dS_oc(i,j) = 0.d0

do ii=1,n_ocean

arg = ocean(ii,i,j,1)*tau +
:      (ocean(ii,i,j,2)-5.d0)*s +
:      (ocean(ii,i,j,3)-5.d0)*h +
:      (ocean(ii,i,j,4)-5.d0)*p +
:      (ocean(ii,i,j,5)-5.d0)*N +
:      (ocean(ii,i,j,6)-5.d0)*ps

sa = sin(arg)
ca = cos(arg)

dC_oc(i,j) = dC_oc(i,j) +
:      ocean(ii,i,j,7)*ca +
:      ocean(ii,i,j,8)*sa +
:      ocean(ii,i,j,9)*ca +
:      ocean(ii,i,j,10)*sa

dS_oc(i,j)=dS_oc(i,j) -
:      ocean(ii,i,j,7)*sa +
:      ocean(ii,i,j,8)*ca +
:      ocean(ii,i,j,9)*sa -
:      ocean(ii,i,j,10)*ca

enddo ! ii
enddo ! i
enddo ! j
!$OMP END PARALLEL DO

```

Figure 1. Parallelization of the part of the code responsible for calculating ocean tides

A more complex approach was used in the subroutine responsible for calculating the geopotential. In addition to the PARALLEL DO construct, there is a part of the code that adds up the values obtained in different execution threads, and this fragment is executed sequentially by the CRITICAL/END CRITICAL construct. The calculation of each type of derivative (dV_{dr} , dV_{dteta} , dV_{dlyam}) is done in its own execution thread (Fig. 2).

The partial sums dV_{dr} , dV_{dteta} , dV_{dlyam} are obtained separately in each thread (CRITICAL section) and recorded in the variables dV_{dr2} , dV_{dteta2} , dV_{dlyam2} .

The parallelization of the EGF recovery process is performed as follows: matrices with the calculated coefficients can be written to disk for each arc, or they can be summed up during the calculation and written to disk once at the end of the process. From a runtime point of view, it is better to sum the matrices at the end of the calculation, but this requires more memory. The program allows efficient parallelization, since the calculations for each arc are done independently. The program divides the whole range of arcs into N parts and starts N processes simultaneously for each part. After all the parts of the time interval have been processed and

```

!$OMP PARALLEL PRIVATE(dV_dr, dV_dteta, dV_dlyam)
!$OMP:          SHARED(dV_dr2, dV_dteta2, dV_dlyam2)

      dV_dr=0.0d0
      dV_dteta=0.0d0
      dV_dlyam=0.0d0

      dV_dr2=0.d0
      dV_dteta2=0.d0
      dV_dlyam2=0.d0

!$OMP DO
  do i=0,nmax
    do j=0,i

      dV_dr = dV_dr-fm*rs*rs*dble(i+1)*T(i,j)*
!$OMP:          (Cnk2(i,j)*cj(j) + Snk2(i,j)*sj(j))

      dV_dteta = dV_dteta+fm*rs*dT_dteta(i,j)*
!$OMP:          (Cnk2(i,j)*cj(j) + Snk2(i,j)*sj(j))

      dV_dlyam = dV_dlyam+fm*rs*dble(j)*T(i,j)*
!$OMP:          (Snk2(i,j)*cj(j) - Cnk2(i,j)*sj(j))

      enddo ! j
    enddo ! i
!$OMP END DO
!$OMP CRITICAL
      dV_dr2 = dV_dr2 + dV_dr
      dV_dteta2 = dV_dteta2 + dV_dteta
      dV_dlyam2 = dV_dlyam2 + dV_dlyam
!$OMP END CRITICAL
!$OMP END PARALLEL

```

Figure 2. Parallelization of the part of the code responsible for the geopotential calculation

the coefficient files have been written (this is the 1st processing stage), the procedure for reading the files with the partial coefficient matrices and their summation in RAM is started (the 2nd processing stage).

Previously calculated files are used to recover the EGF coefficients. This is done by the least-squares method (LSM) using the routines of the LAPACK library [3]. After each run of the LSM, a new file containing the EGF coefficients is written. The coefficients are written up to the degree N_{max} specified in the initialization file.

The parallelization of the EGF recovery process, implemented in the Fortran programming language, is universal for the following three cases:

- computing cluster;
- multi-core computer running Linux OS;
- multi-core computer running Windows OS.

The differences between the systems are localized in a single procedure that starts a child process by calling “system()”. Different values of environment variables are used for the “computing cluster” and the Windows OS environments, in other cases it is assumed that the program is started under the Linux OS.

Synchronization of parallel computations is performed by creating/deleting flag files (the process of the next computation stage waits for the end of the previous stage). The desired number of parallel processes is always selected at startup using command line arguments. The number of parallel processes depends on the available RAM, because each process running at

the 1st stage needs a certain amount of RAM to work, and all these processes are running simultaneously. For example, if the recovery is performed up to $N_{max} = 300$, then each process of the 1st stage would require 120 GB of RAM. If the computer only has 512 GB of RAM, then the stage can only be divided into 4 parallel processes.

3. Results of Parallelization

The performance of parallelization on computing nodes was tested by measuring the time $T(1)$ required to compute on one node and the time $T(N)$ required to compute on N nodes:

$$SPD = \frac{T(1)}{T(N)}, \quad (13)$$

$$Eff = \frac{SPD}{N} \times 100\%.$$

Here, the *SPD* parameter indicates how much faster the parallelized computation is compared to the single node variant. The *Eff* parameter is a measure of the parallelization efficiency and shows the impact of the overhead on the program execution time. The tests were performed on nodes each having an Intel Xeon E5-2680 v4 2.40GHz/3.3GHz processor with 28 cores and 128GB RAM.

Table 1 shows the speed-up of the program complex as a function of N (the number of computing nodes used). It should be noted that the results are influenced by the speed of reading and writing intermediate files, which can vary considerably. The initialization procedures cannot be parallelized and start to take up a larger and larger part of the total computation time with increasing N . This explains the decrease in efficiency with increasing number of processes. When $N > 16$, the time for reading intermediate files increases drastically. The decrease in efficiency for large N , which has the same origin, can also be observed in Tab. 2.

Table 1. Computation performance of the EGF recovery on the interval of 1 month, up to the maximal degree of the expansion $N_{max} = 96$ (1), time in seconds

N	$T(N)$	SPD	Eff
1	1080	1	100%
2	555	1.946	97.3%
4	305	3.541	88.5%
8	175	6.171	77.1%
16	116	9.310	58.2%
32	96	11.250	35.1%
64	112	9.642	15.0%
128	143	7.552	5.9%

Table 3 summarizes the computation time of the EGF recovery for a 60-day interval. It shows that the number of processes executing the 1st stage decreases as N_{max} increases, which is due to the RAM limit. The values are based on a specific computer configuration (Intel Xeon Gold 6148 x 2, 512GB RAM) and are not universal.

Table 2. Computation performance of the EGF recovery on the interval of 1 month, up to the maximal degree of the expansion $N_{max} = 180$ (1), time in seconds

N	$T(N)$	SPD	Eff
1	60867	1	100%
2	31023	1.962	98.1%
4	15712	3.874	96.8%
8	8253	7.375	92.2%
16	4549	13.380	83.6%
32	3100	19.634	61.3%
64	2757	22.077	34.5%
128	4006	15.194	11.9%

Table 3. The computation time of the EGF recovery with parallelization for a 60-day interval

N_{max}	Number of threads	1 st stage (hrs)	2 nd stage (hrs)	Total time (hrs)
200	15	0.42	0.33	0.75
220	12	0.75	0.5	1.25
240	9	1.17	0.83	2
260	6	1.67	1.5	3.17
280	5	2.33	2.17	4.5
300	4	3	3	6
320	3	5	7.5	12.5
340	2	9.83	8.33	18.17
360	2	12.5	13.25	25.75
380	1	31	19.67	50.67
400	1	36	30	66

4. Results of the EGF Recovery

The developed program allows to create models of the EGF with different temporal resolutions using real measurements of the inter-satellite range of the GRACE and GRACE-FO missions. These models can cover long time intervals, such as a year or more (static models), as well as short time intervals, such as monthly models. The latter can track global processes that change over time, such as mass movements in the atmosphere and oceans, melting of glaciers, and more.

Various methods can be used to graphically present the results of the EGF recovery. One such method is visualizing the distribution of the gravity field over the Earth's surface. In this case, the EGF is represented either in acceleration units or in geoid heights. Milligals (mGal) are typically used as the acceleration units, defined as $1 \text{ Gal} = 1 \text{ cm}/c^2$. The EGF expressed in geoid heights is the difference between a generalized surface of the Earth defined by the actual gravitational field and the ellipsoid of rotation defined by the C_{20} coefficient in the expansion of the gravitational potential (1). Accordingly, the height of the geoid is determined by the

expression

$$H(\varphi, \lambda) = R \left[\sum_{n=2}^{N_{max}} \sum_{m=0}^n P_{nm}(\sin \varphi) (C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)) - P_{20}(\sin \varphi) C_{20} \right], \quad (14)$$

where $P_{nm}(\sin \varphi)$ are the associated Legendre polynomials of degree n and order m ; φ, λ are the angular spherical coordinates of a given point at the Earth's surface; R is the mean radius of the Earth; C_{nm} and S_{nm} are the dimensionless Stokes coefficients. Geoid heights are usually expressed in meters.

The following presents the results of the Earth's gravity field recovery using real data from the GRACE-FO mission. These results were obtained using the program described in this paper.

Figure 3 shows the static model of the Earth's gravitational field MSU2024-1 built from GRACE-FO data for the whole year 2021. The model is decomposed to degrees and orders of $n = m = 120$ and presented here in terms of geoid heights.

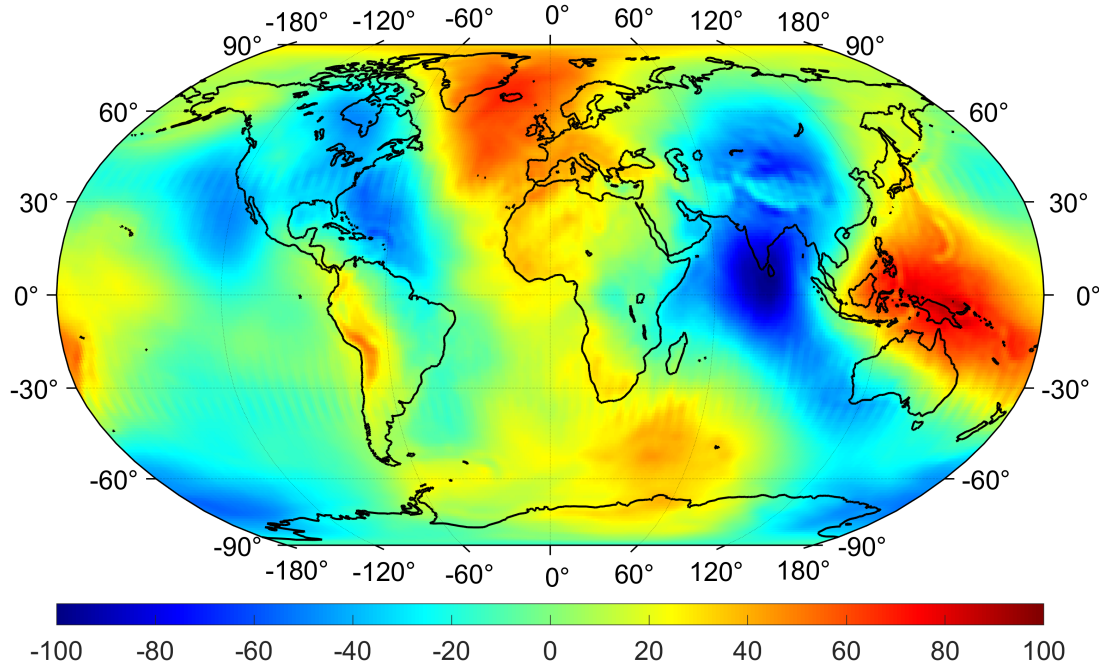


Figure 3. Results of the EGF recovery up to $n = m = 120$ using the GRACE-FO data for a one-year interval (the static model MSU2024-1). Colors correspond to geoid heights in meters

Several measures are used to compare different EGF models, including the degree difference variance (DDV). This quantity is defined as

$$\Delta\sigma_n^2 = R \sum_{m=0}^n (\Delta C_{nm}^2 + \Delta S_{nm}^2). \quad (15)$$

It represents the difference between two EGF models per degree, i.e., for each degree, the orders are summed. It is given in terms of geoid heights. The quantities ΔC_{nm} and ΔS_{nm} in (15) represent the difference between the Stokes coefficients of the reference (REF) and recovered (REC) models: $\Delta C_{nm} = (C_{nm})_{ref} - (C_{nm})_{rec}$ and $\Delta S_{nm} = (S_{nm})_{ref} - (S_{nm})_{rec}$.

Figure 4 shows a comparison between the EGF obtained from GRACE-FO data in this study and the reference models of CSR, GFZ, and JPL. As mentioned in the introduction, monthly temporal models are created in three international centers: the Center for Space Research at the

University of Texas, Austin (CSR), the Helmholtz Centre Potsdam German Research Centre for Geosciences (GFZ), and the Jet Propulsion Laboratory (JPL). Figure 4 displays the ‘gravity’ curve for the cumulative gravity signal per degree (in terms of geoid heights) of the reference EGF, excluding the second zonal harmonic C_{20} .

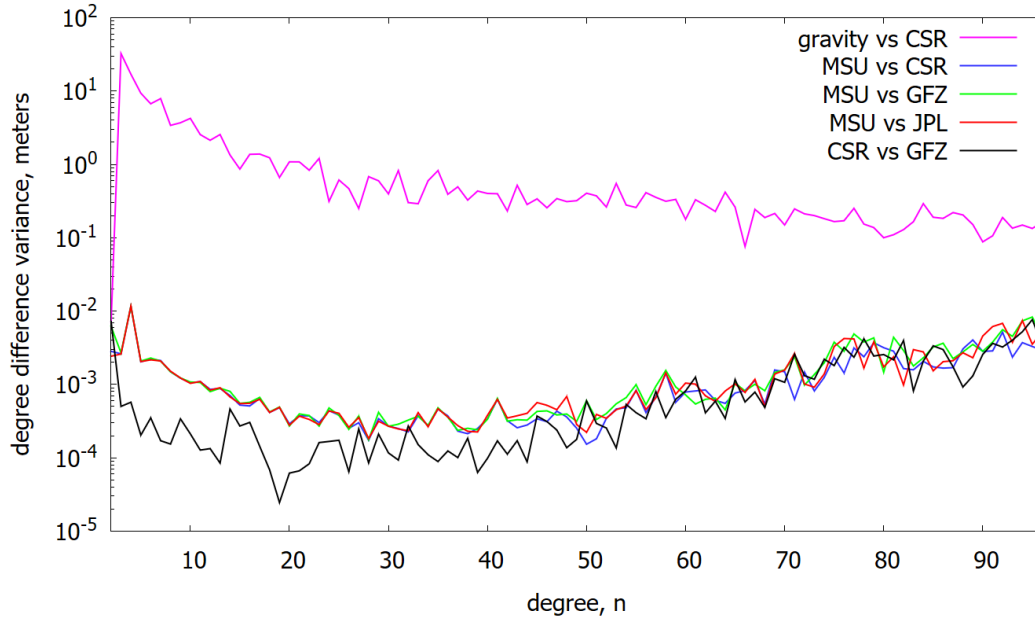


Figure 4. Comparison between the monthly MSU model and the CSR, GFZ and JPL reference temporal models expanded to degree $n = 96$. All four models are built using GRACE-FO data for January, 2021

The degree difference variance between the recovered and reference models shows a characteristic behavior of the errors of the spherical harmonic coefficients. Specifically, the differences decrease and reach a minimum value in the low-frequency and early mid-frequency parts of the spectrum (as seen, for example, in [23]), before increasing again. These differences result in minimum geoid heights of about 0.1 to 0.2 mm for degrees $n = 30$ to 50.

Conclusion

Global EGF models, built from data collected by space geodetic missions, play a very important role in studying global processes across Earth’s various geospheres. This paper presents the results of the EGF recovery using parallel computing based on real data from the GRACE Follow On mission. As degree of decomposition N_{max} increases, the recovery of EGF parameters becomes more time consuming and expensive in terms of memory requirements. This is because the number of parameters increases quadratically with N_{max} . For example, recovering the EGF up to a large degree $N_{max} \approx 200$ requires hundreds of gigabytes of RAM. Therefore, parallel computations are crucial for optimizing the computational process for the EGF recovery.

Parallelization is especially effective for code sections that involve summation. In this study, we utilized OpenMP parallelization tools to compute the geopotential non-sphericity and ocean tides models. Furthermore, we applied parallelization directly to the EGF recovery procedure, which can be significantly accelerated by dividing the calculation of the normal equation matrix into parts and then summing them up.

Testing showed that parallelizing the program for recovering the EGF over a one-month interval with $N_{max} = 96$ resulted in an 11-fold reduction in computation time. Similarly, recovering over a one-year interval with $N_{max} = 180$ resulted in a 22-fold improvement. It was found that the speed of access to the cluster's common filesystem was critical to recovering the EGF with large N_{max} . The optimal number of nodes for simultaneous processing strongly depends on the speed of reading files with matrix coefficients. Other factors determining the optimal number are the maximum degree N_{max} and the duration of the time interval to be processed.

With the help of the developed program the static model of the Earth's gravitational field MSU2024-1 was built using the GRACE-FO mission data for the whole year 2021. The model is decomposed to degrees and orders of $n = m = 120$ and presented in terms of geoid heights. We also compared the EGF recovery on a monthly interval using the GRACE-FO data obtained in this work with the CSR, GFZ, and JPL temporal models built at other world centers.

Further research will focus on software modifications to increase the spatial and temporal resolution of EGF models.

Acknowledgements

This work is supported by the Russian Science Foundation under Grant No. 23-42-00055.

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. Abich, K., Abramovici, A., Amparan, B., *et al.*: In-Orbit Performance of the GRACE Follow-on Laser Ranging Interferometer. Phys. Rev. Lett. 123, 031101 (Jul 2019). <https://doi.org/10.1103/PhysRevLett.123.031101>
2. Albertella, A., Migliaccio, F., Sans, F.: GOCE: The Earth Gravity Field by Space Gradiometry. Celestial Mechanics and Dynamical Astronomy 83, 1–15 (05 2002). <https://doi.org/10.1023/A:1020104624752>
3. Anderson, E., Bai, Z., Bischof, C., Blackford, S., *et al.*: LAPACK Users' Guide. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edn. (1999)
4. AVISO: Global tide FES 2014 Description. <https://www.aviso.altimetry.fr/en/data/products/auxiliary-products/global-tide-fes/description-fes2014.html>
5. Belikov, M.V., Taibatorov, K.A.: Efficient algorithm for calculating the Earth's gravitational potential and its first derivatives for solving satellite problems. Kinematics and physics of celestial bodies (in Russian) 6(2), 24–32 (1990)
6. Carrère, L., Lyard, F.H., Cancet, M., Guillot, A.: FES 2014, a new tidal model on the global ocean with enhanced accuracy in shallow seas and in the Arctic region. In: Geophysical Research Abstracts (2015), <https://api.semanticscholar.org/CorpusID:218081284>
7. Chandra, R., Dagum, L., Kohr, D., *et al.*: Parallel programming in OpenMP. Morgan Kaufmann (2001)

8. Ince, E.S., Barthelmes, F., Reißland, S., *et al.*: ICGEM – 15 years of successful collection and distribution of global gravitational models, associated services, and future plans. *Earth System Science Data* 11(2), 647–674 (2019). <https://doi.org/10.5194/essd-11-647-2019>
9. Kornfeld, R.P., Arnold, B.W., Gross, M.A., *et al.*: GRACE-FO: The Gravity Recovery and Climate Experiment Follow-On Mission. *Journal of Spacecraft and Rockets* 56(3), 931–951 (2019). <https://doi.org/10.2514/1.A34326>
10. Kvas, A., Behzadpour, S., Ellmer, M., *et al.*: ITSG-Grace2018: Overview and Evaluation of a New GRACE-Only Gravity Field Time Series. *Journal of Geophysical Research: Solid Earth* 124(8), 9332–9344 (2019). <https://doi.org/10.1029/2019JB017415>
11. Landerer, F.W., Flechtner, F.M., Save, H., *et al.*: Extending the Global Mass Change Data Record: GRACE Follow-On Instrument and Science Data Performance. *Geophysical Research Letters* 47(12), e2020GL088306 (2020). <https://doi.org/10.1029/2020GL088306>
12. Luo, J., Chen, L.S., Duan, H.Z., *et al.*: TianQin: a space-borne gravitational wave detector. *Classical and Quantum Gravity* 33(3), 035010 (jan 2016). <https://doi.org/10.1088/0264-9381/33/3/035010>
13. Lyard, F.H., Allain, D.J., Cancet, M., *et al.*: FES2014 global ocean tide atlas: design and performance. *Ocean Science* 17(3), 615–649 (2021). <https://doi.org/10.5194/os-17-615-2021>
14. Mayer-Gürr, T., Eicker, A., Ilk, K.H.: ITG-Grace03 Gravity Field Model. http://icgem.gfz-potsdam.de/tom_longtime (2007)
15. Mayer-Gürr, T., Kurtenbach, E., Eicker, A.: ITG-Grace2010 Gravity Field Model. http://icgem.gfz-potsdam.de/tom_longtime (2010)
16. Mayer-Gürr, T., Behzadpur, S., Ellmer, M., *et al.*: ITSG-Grace2018 - Monthly, Daily and Static Gravity Field Solutions from GRACE. *GFZ Data Services* (2018). <https://doi.org/10.5880/ICGEM.2018.003>
17. Petit, G., Luzum, B.: IERS conventions (2010). Tech. Rep. DTIC Document 36, 180 (01 2010), https://www.iers.org/SharedDocs/Publikationen/EN/IERS/Publications/tn/TechnNote36/tn36.pdf?__blob=publicationFile&v=1
18. Reigber, C., Lhr, H., Schwintzer, P.: CHAMP mission status. *Advances in Space Research* 30(2), 129–134 (2002). [https://doi.org/10.1016/S0273-1177\(02\)00276-4](https://doi.org/10.1016/S0273-1177(02)00276-4)
19. Reigber, C., Balmino, G., Schwintzer, P., *et al.*: A high-quality global gravity field model from CHAMP GPS tracking data and accelerometry (EIGEN-1S). *Geophysical Research Letters* 29(14), 37–1–37–4 (2002). <https://doi.org/10.1029/2002GL015064>
20. Tapley, B.D., Bettadpur, S., Watkins, M., Reigber, C.: The gravity recovery and climate experiment: Mission overview and early results. *Geophysical Research Letters* 31(9) (2004). <https://doi.org/10.1029/2004GL019920>
21. Tapley, B., Reigber, C.: The GRACE Mission: Status And Future Plans. *AGU Fall Meeting Abstracts -1*, 02 (11 2001)

22. Watkins, M.M., Bettadpur, S.V.: The GRACE Mission: The Challenges of Using Micron-Level Satellite-to-Satellite Ranging to Measure the Earth's Gravity Field. In: Physics, Environmental Science (2000), <https://api.semanticscholar.org/CorpusID:119013578>
23. Zhamkov, A., Milyukov, V.: Next Generation Gravity Missions to Address the Challenges of High-Precision Space Gravimetry. *Izvestiya, Physics of the Solid Earth* 57, 266–278 (03 2021). <https://doi.org/10.1134/S1069351321020130>
24. Zhao, Y., Li, J., Xu, X., *et al.*: Determination of static gravity field model by using satellite data of GOCE and GRACE. *Chinese Journal of Geophysics (in Chinese)* 66(6), 2322–2336 (2023). <https://doi.org/10.6038/cjg2022Q0615>
25. Zhou, H., Wang, P., Pail, R., *et al.*: Assessment of a near-polar pair mission for detecting the Earth's temporal gravity field. *Geophysical Journal International* 234(2), 852–869 (03 2023). <https://doi.org/10.1093/gji/ggad107>

WAVEWATCH III Hybrid Parallelization for Azov Sea Wave Modelling

Alexander I. Sukhinov¹ , Elena A. Protsenko² , Sofya V. Protsenko² 

© The Authors 2024. This paper is published with open access at SuperFri.org

The article examines potential applications of WAVEWATCH III (WW3), the third-generation wind-wave model. This study delves into the implementation of hybrid parallelization (MPI-OpenMP) and the development of multiple-cell grids tailored for the Azov Sea region. It elucidates fundamental equations of the model, their discretization, and software execution. The multiple-cell grid strategy employs high-resolution cells within the region of interest, gradually increasing cell density in other areas to optimize memory consumption. A 6-level multiple-cell grid was specifically crafted for the Azov Sea, with an algorithm outlined for its generation incorporating two refinement methods. This algorithm enables the creation of refined multiple-cell grids near shorelines at varying levels, along with the capability to refine grid structures in arbitrary zones. Additionally, the article presents hybrid parallelization techniques for the wave spectral component (MPI-OpenMP), assessing scalability in both MPI and hybrid deployments. The WW3 model offers a multigrid option facilitating parallel operation of subdomains akin to domain decomposition, while ensuring parallelization of each subnet via the component decomposition method.

Keywords: multiple-cell grid, WAVEWATCH III, hybrid parallelization, parallel numerical implementation.

Introduction

Modern simulations of ocean wave motion are becoming increasingly intricate and resource-intensive. Among the primary tools in this domain is WW3, an advanced wave modeling system engineered to accurately and efficiently replicate wave phenomena across oceans.

The WW3 model serves as a numerical tool for wave prediction in oceans and other aquatic environments. Developed and maintained by the National Center for Environmental Prediction (NCEP), a division of the National Oceanic and Atmospheric Administration (NOAA) in the United States, the latest iteration, WAVEWATCH III version 6.07, was released in 2019 and is accessible on GitHub.

WW3 enjoys widespread usage and recognition in forecasting ocean wave motion. However, like any modeling system, it both has its strengths and weaknesses. WW3 excels in delivering highly precise forecasts of wave motion, particularly over extended temporal and spatial scales. The model accounts for numerous factors influencing wave formation, including wind, currents, tides, and other parameters, rendering it versatile for a multitude of applications. By incorporating various wave components, such as wind waves and tidal waves, WW3 enhances realism across diverse scenarios. Additionally, WW3 offers various configurations and customization options tailored to specific tasks, encompassing global and regional forecasting endeavors [1–5].

The model might demand substantial computational resources, particularly when employing finer spatial resolutions or conducting global forecasts. Limitations may emerge at regional levels, particularly in regions characterized by intricate coastline geometries or proximity to continents. Additionally, WW3 exhibits a restricted prognostic horizon, potentially complicating the accurate prediction of waves over extended periods.

¹Don State Technical University, Rostov-on-Don, Russian Federation

²Chekhov Taganrog Institute (Branch of Rostov State University of Economics), Taganrog, Russian Federation

In summary, notwithstanding these constraints, WW3 retains its efficacy as a robust tool for wave prediction and finds broad utility across diverse domains of oceanography and maritime safety [6].

The WW3 model offers various grid configuration options tailored to different forecasting requirements and scales. The adoption of multiple-cell grids within WW3 has emerged as a pivotal factor, elevating standards of accuracy and adaptability in modeling practices. Multiple-cell grids afford the potential for enhanced granularity and adaptability in regional coverage, a critical aspect for proficient wave motion forecasting. In this paper, we examine the significance of employing multiple-cell grids within WW3 and their consequential impact on augmenting spatial resolution and precision in wave modeling within the Azov Sea region [7].

The article is organized as follows. Section 1 is devoted to the description of the WW3 model, the governing equations of the model and their discretization are presented. Section 2 describes the algorithm for generating a Multiple-Cell Grid implemented in WW3, based on this algorithm, a Multiple-Cell Grid was built to simulate processes in the Azov Sea. Section 3 includes a description of The main elements of the WW3 program. Section 4 demonstrates numerical experiments conducted on the basis of WW3 for the Azov Sea, including Simulation results of the direction and period of the peak wave, maximum wave height in the Azov Sea. Section 5 describes the parallel implementation of WW3 based on the MPI-OpenMP hybrid approach. Conclusion summarizes the study and points directions for further work.

1. Materials and Methods

1.1. Governing Equation in WW3 Model

The principles governing quasi-homogeneous waves find localized application in scenarios where the approach involves a gradual alteration in flow and depth conditions. This application yields the subsequent relationships [7]:

$$\sigma^2 = gk \tanh kd, \tag{1}$$

$$\omega = \sigma + \mathbf{k} \cdot \mathbf{U}. \tag{2}$$

The variables presented are defined as follows: \mathbf{k} represents the wavenumber vector, where k denotes the wavenumber, and θ signifies the direction perpendicular to the wave crest or spectral component, equating to the direction of \mathbf{k} . $\sigma = 2\pi fr$ stands for the relative or natural frequency, while $\omega = 2\pi fa$ represents the absolute frequency. Moreover, d signifies the average water depth, and \mathbf{U} indicates the flow velocity, averaged over both depth and time on the scale of individual waves.

The definition of \mathbf{k} and ω within the wave phase function implies the conservation of the number of wave crests:

$$\frac{\partial \mathbf{k}}{\partial t} + \nabla \omega = 0. \tag{3}$$

The density spectrum characterizing the dispersion of surface height F is contingent upon all independent phase parameters, denoted as $F(\mathbf{k}, \sigma, \omega)$, exhibiting spatial and temporal variability across scales exceeding those of individual waves.

The primary spectrum encompasses the wave number-directional spectrum $F(k, \theta)$. The WW3 output comprises the frequency-directional spectrum $F(f_r, \theta)$. Diverse spectra can be derived from $F(k, \theta)$:

$$F(f_r, \theta) = \frac{\partial k}{\partial f_r} F(k, \theta) = \frac{2\pi}{c_g} F(k, \theta), \quad (4)$$

$$F(f_a, \theta) = \frac{\partial k}{\partial f_a} F(k, \theta) = \frac{2\pi}{c_g} \left(1 + \frac{\mathbf{k} \cdot \mathbf{U}}{kc_g}\right)^{-1} F(k, \theta), \quad (5)$$

$$c_g = \frac{\partial \sigma}{\partial k} = n \frac{\sigma}{k}, \quad n = \frac{1}{2} + \frac{kd}{\sinh 2kd}, \quad (6)$$

where \mathbf{c}_g is the group velocity.

The formula governing wave propagation is as follows:

$$\frac{DN}{Dt} = \frac{S}{\sigma}. \quad (7)$$

The expression utilizes D/Dt to denote the total derivative, with S indicating the cumulative impact of sources and absorbers on the spectrum F .

The equilibrium equation governing the spectrum $N(k, \theta; \mathbf{x}, t)$ takes the following format:

$$\frac{\partial N}{\partial t} + \nabla_x \cdot \dot{\mathbf{x}}N + \frac{\partial}{\partial k} \dot{k}N + \frac{\partial}{\partial \theta} \dot{\theta}N = \frac{S}{\sigma}, \quad (8)$$

$$\dot{\mathbf{x}} = \mathbf{c}_g + \mathbf{U}, \quad (9)$$

$$\dot{k} = -\frac{\partial \sigma}{\partial d} \frac{\partial d}{\partial s} - \mathbf{k} \cdot \frac{\partial \mathbf{U}}{\partial s}, \quad (10)$$

$$\dot{\theta} = -\frac{1}{k} \left[\frac{\partial \sigma}{\partial d} \frac{\partial d}{\partial m} + \mathbf{k} \cdot \frac{\partial \mathbf{U}}{\partial m} \right], \quad (11)$$

where $\mathbf{c}_g = (c_g \sin \theta, c_g \cos \theta)$, s is the coordinate in the direction θ , and m is the coordinate perpendicular to s [7-8].

The standard source terms employed in WW3 are delineated as follows:

$$S = S_{ln} + S_{in} + S_{nl} + S_{ds} + S_{bot} + S_{db} + S_{tr} + S_{sc} + S_{ref}. \quad (12)$$

The components within the equation consist of the source term denoted as S , where S_{ln} represents the linear input term facilitating model initialization and enhancing the initial wave growth's realism. Additionally, S_{in} pertains to the interaction between the atmosphere and waves, while S_{nl} characterizes nonlinear wave interaction. Furthermore, S_{ds} corresponds to wave interaction with the ocean, predominantly encompassing crashing waves, and S_{bot} denotes wave interaction with the bottom. S_{db} serves as an additional criterion for faults during extremely low water conditions, while S_{tr} represents the term governing triadic wave interactions. Moreover, S_{sc} accounts for wave scattering by bottom elements, and S_{ref} signifies the energy sources for infragravity waves [9].

Various combinations of these fundamental source terms are available, with nonlinear wave interactions feasibly modeled via discrete interaction approximation.

1.2. Discretization

If equation (8) is directly solved, it results in a noticeable reduction in spectral resolution in shallow water. This decrease in resolution can be mitigated by solving the equation on a grid featuring variable wavenumbers, implicitly accommodating changes in the kinematic wavenumber due to shallow depths. Such a grid of wavenumbers corresponds to a spatially and temporally consistent grid of relative frequencies. The respective local grid of wavenumbers can be computed directly from the varied frequency grid and dispersion relation (1), thereby becoming dependent on the local depth d . To facilitate cost-effective calculations of S_{nl} and ensure proper segregation of rising frequencies, frequency sampling utilizes exponentially increasing increments. This approach ensures that the variable frequency resolution is proportionate to the local frequency [10, 11]:

$$\sigma_{m+1} = X_\sigma \sigma_m. \quad (13)$$

The impact of employing a spatially variable grid will be examined in relation to equation (8). Representing a grid featuring a variable wavenumber as κ , the equilibrium equation is expressed as:

$$\frac{\partial N}{\partial t c_g} + \frac{\partial \dot{x}N}{\partial x c_g} + \frac{\partial \dot{y}N}{\partial y c_g} + \frac{\partial \dot{\kappa}N}{\partial \kappa c_g} + \frac{\partial \dot{\theta}N}{\partial \theta c_g} = \frac{S}{\sigma c_g}, \quad (14)$$

$$\dot{\kappa} \frac{\partial k}{\partial \kappa} = c_g^{-1} \frac{\partial \sigma}{\partial d} \left(\frac{\partial d}{\partial t} + \mathbf{U} \cdot \nabla_x d \right) - \mathbf{k} \cdot \frac{\partial \mathbf{U}}{\partial s}. \quad (15)$$

In WW3, equation (14) is resolved utilizing the fractional step method. The initial step manages temporal alterations in depth and the corresponding adjustments in the wavenumber grid. By isolating the effects of transient changes in water level, the grid remains constant, while the depth becomes quasi-stationary for subsequent fractional steps. Additional fractional steps account for spatial propagation, intra-spectral propagation, and source conditions.

This division into multiple stages ensures efficient vectorization and parallelization concurrently. Furthermore, temporal partitioning enables the utilization of distinct partial or dynamically adaptable time intervals across various segmented stages of the model.

In the upstream 2nd order non-oscillatory advection scheme (UNO2), the wave action's interpolated value at the mid-flux (MF) point for the cell face between cell $i - 1$ and cell i is defined as

$$N_{i-}^{MF} = N_C + \frac{1}{2} \text{sgn}(N_D - N_C) \left(1 - \left| \dot{\phi} \right| \Delta t / \Delta \phi \right) \min(|N_U - N_C|, |N_C - N_D|). \quad (16)$$

In the context provided, $i-$ represents the index pertaining to the cell face situated between cell $i - 1$ and cell i ; $\dot{\phi}$ denotes the propagation velocity at the boundary between points i and $i - 1$; the indices U , C , and D signify the Upstream, Central, and Downstream cells, respectively. When $\dot{\phi} > 0$, $U = i - 2$, $C = i - 1$, $D = i$ for the cell face positioned between cell $i - 1$ and cell i . Conversely, if $\dot{\phi} \leq 0$, then $U = i + 1$, $C = i$, $D = i - 1$.

The UNO scheme, implemented on multiple-cell grids, demonstrates stability, conservativeness, and shape retention.

The process of updating the flow and cell values is executed based on the following formulas:

$$\tilde{F}_{i-} = \phi \dot{N}_{i-}^{MF}; \quad N_i^{n+1} = N_i^n + \frac{\Delta t}{\Delta \phi} \left(\tilde{F}_{i-} - \tilde{F}_{i+} \right), \quad (17)$$

where \tilde{F}_{i-} denotes the flux between grid points with indices i and $i - 1$ in ϕ -space, while \tilde{F}_{i+} represents the flow at the cell face between cell i and cell $i + 1$. Its estimation can be achieved by utilizing the average flow value, akin to equation (16), albeit with the substitution of i by $i + 1$.

UNO2 schemes are characterized by conservativeness, shape preservation, and immunity to numerical fluctuations. They also maintain the component correlation of the transferred variables. UNO2 is implemented within the multiple-cell grid framework to facilitate spatial and interspectral propagation in WW3 [14].

2. Multiple-Cell Grid Generation Algorithm

A multiple-cell grid employs high-resolution cells in designated regions while incrementally increasing cell density elsewhere to optimize memory utilization. Let's establish the minimum length of the grid side as 1, and denote the side length of the grid at the n -th level as 2^{n-1} . Consequently, a 4-level multiple-cell grid encompasses four grid tiers with side lengths of 1, 2, 4, and 8. The largest grid within WW3 is referred to as the base grid. Figure 1 illustrates the layout of the 4-level coastal refined multiple-cell grid.

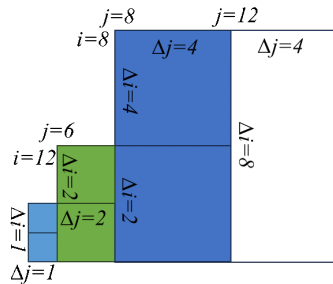


Figure 1. 4-level refined multiple-cell grid

The resolution of bottom bathymetry data for generating a multiple-cell grid should adhere to the same minimal selection criteria as that of a single-level grid.

The fundamental grid must satisfy the following criteria: the grid can be scaled at a 1:2 ratio, the depth of the grid must exceed the minimum depth requirement of the WW3 model, grid points near shorelines should belong to a level 1 grid. We introduce the notion of a limited radius, representing a distance within which a ground grid of a level other than the first is prohibited.

These restrictions impose a boundary on the grid, wherein ground grids beyond a certain distance (limited radius) are disallowed, except for the level 1 grid. As illustrated in Fig. 1, the limited radius for a 4-level grid is $1+2+4 = 7$, for a 5-level grid it is 15, and for a 6-level grid it is 31. It is also evident that the bounded radii of N -level grids are given by $L_{Nlim} = 2^{N-1} - 1$.

If the base grid meets the specified constraints, it will be included in the multi-cell grid file. Conversely, if the base grid fails to meet the constraints, the large grid will be evenly subdivided into four 3-level grids and assessed against the same criteria until further division is no longer feasible.

The grid generation procedure follows a recursive approach, wherein grid cells undergo continuous division until further division is unfeasible. Thus, the grid generation process can be executed using a recursive loop structure, allowing users to adjust the multi-cell grid level.

In the recursive loop, the grid generation process functions iteratively. If the grid fails to meet the specified constraints, the function will invoke itself until either the constraints are satisfied

or the grid becomes unsplitable. Figure 2 illustrates the algorithm of the program designed to generate a multi-cell grid for the Azov Sea.

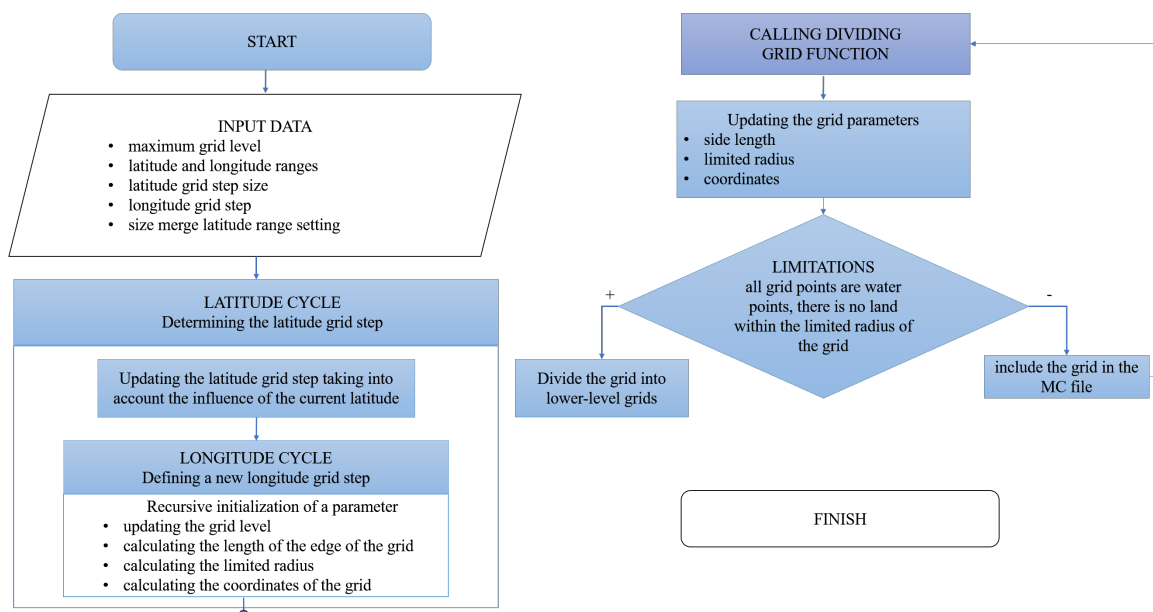


Figure 2. Algorithm for organizing recursive cyclic coastal refinement

To achieve a flexible form of grid refinement, we utilize a curve to delineate the area necessitating refinement, while imbuing hierarchical attributes into the curve. Consequently, grid refinement can be implemented to the specified level within any designated area (refer to Fig. 3).

In order to impose constraints on the refinement of an arbitrary area, it becomes necessary to modify the last of the three constraints in the coastal zone refinement method. We postulate that the N -level grid is encompassed by the N -level curve and lies beyond the scope of the $(N - 1)$ -level curve. If a grid intersects with a curve, it is assumed to be encircled by the curve. Accordingly, the following constraints are appended to elucidate an arbitrary area: grids must fall within the boundaries of curves of the same level and must lie outside the boundaries of curves of a lower level, except for the base grid, the base grid must lie beyond the boundaries of any level of curve, if a curve merely traverses through a grid, the grid is also considered to be within the boundaries of the curve.

Under these constraints, refinement of any arbitrary domain can be achieved. To ensure that the multiple-cell grid scales only in ratios of 1:2, additional restrictions must be adhered to when defining a refined area using a curve: curves must not intersect, any lower-level curve must be contained within the boundaries of the upper-level curve, the distance between the n th level curve and the $(N - 1)$ -level curve must exceed the length of the side of the n -th level grid (to prevent the two curves from being too close).

The ray method is employed to ascertain whether a grid lies within the curve. A ray, parallel to the x -axis and extending positively from the estimated grid, is introduced. If the number of intersections between the ray and the curve is odd, the estimated point lies inside the curve; if the number of intersections is even, the estimated point is outside the curve. In cases where the ray passes tangentially to the curve, no intersection is counted.

A ray is projected from the point S , which is subsequently intersected by curves, resulting in segments delineated on both sides of the curve, as depicted in Fig. 4.

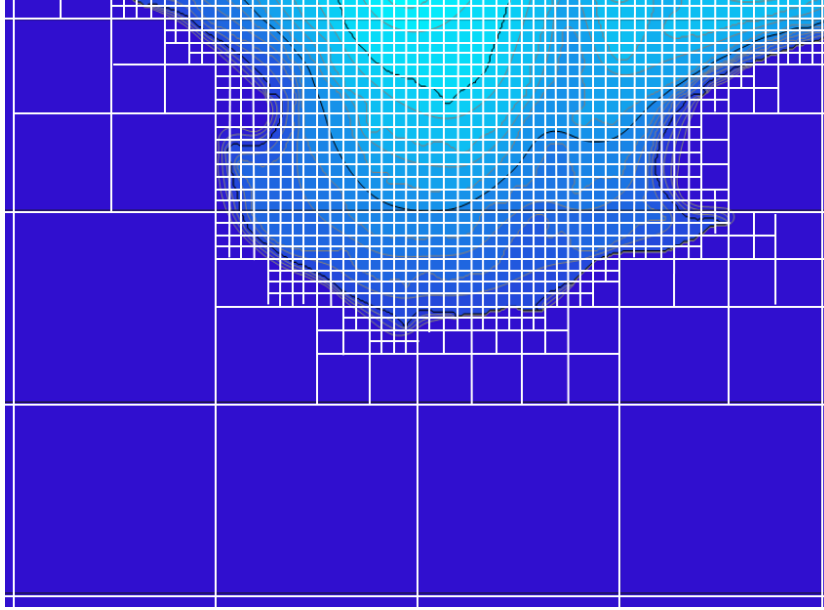


Figure 3. The scheme of using curves to refine the grid

Given the finite extent of the curve, it is straightforward to establish that the outermost line segment should invariably lie outside the curve. In instances where the number of segments is even, the line segment containing point P coincides spatially with the outermost line segment (thus outside the curve). Conversely, if the number of segments is odd, point S resides inside the curve.

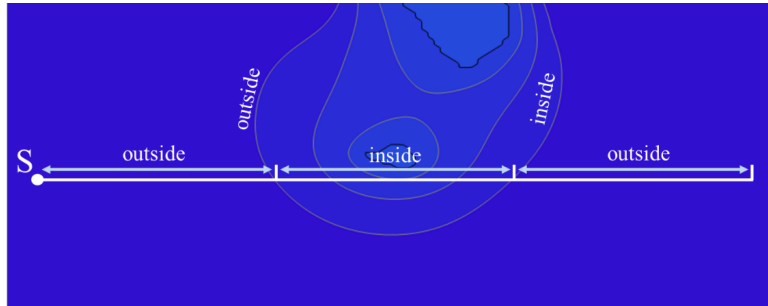


Figure 4. The ray method used to determine whether a point is inside a curve

In the application of the ray method, a polygon serves as an approximation for the curve. Consequently, it becomes essential to tally the intersections occurring between the ray and each edge comprising the polygon. The formula utilized to estimate these intersections is as follows:

$$\begin{cases} y_s \geq \min(y_a, y_b), \\ y_s \leq \max(y_a, y_b) : \text{intersection}, \\ x_s < x_i. \end{cases} \quad (18)$$

In the equation provided, (x_p, y_p) represent the coordinates of the estimated point, while (x_a, y_a) and (x_b, y_b) denote the coordinates of the edge's endpoints. Additionally, (x_i, y_i) stand for the coordinates of the intersection point where the straight line intersects the ray, and where the segment AB is situated.

The curve maintains continuity and differentiability throughout its entirety. However, the polygon is composed of vertices and edges, rendering the ray method ineffective when the ray

intersects the polygon's vertex or coincides with its edge. To rectify these inaccuracies, the mathematical expression for estimating intersections is adjusted to the following equation:

$$\begin{cases} y_s > \min(y_a, y_b), \\ y_s \leq \max(y_a, y_b) : \text{intersection}, \\ x_s < x_i. \end{cases} \quad (19)$$

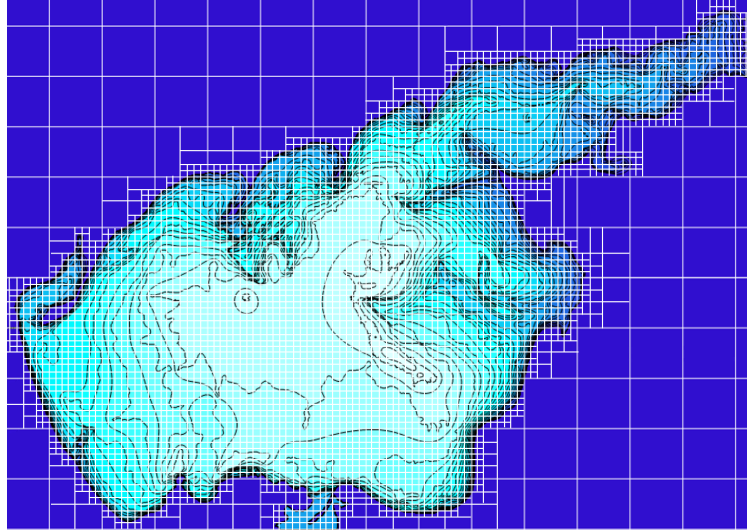


Figure 5. Multiple-cell grid for the Azov Sea

Figure 5 illustrates the multiple-cell grid constructed for the Azov Sea. The algorithm facilitates the creation of multiple-cell grids featuring varying levels of granularity, thereby enabling enhanced modeling of wave dynamics across diverse regions within the sea and coastal areas.

3. WW3 Model Structure

The fundamental component of WW3 resides in its wave model subroutine, accessible via a standalone shell or any other software necessitating real-time updates of wave data. Complementary programs encompass a grid preprocessor, a tool for generating synthetic initial conditions, versatile software shells tailored for individual or multiple-cell grid applications, along with two input preprocessors and postprocessors facilitating grid and point output data manipulation. Figure 6 delineates the relational diagram depicting the foundational data flow within the system.

The data files are denoted by the .ww3 file extension. For the multiple-cell grid wave module, the extension ww3 multi indicates a distinct grid. The WW3 grid preprocessor generates the mod def.ww3 model definition file, containing details regarding the date, obstacles, and parameters governing both physical and numerical methodologies. Initial conditions for the wave model are encapsulated in a restart file named restart.ww3. Additionally, the w3wave procedure produces restart files labeled restartn.ww3.

Boundary conditions are sourced from the nest.ww3 file by the WW3 model and generated for subsequent runs in nestn.ww3. Output data is written to files such as out grd.ww3, out pnt.ww3, track o.ww3, and partition.ww3. In versions employing distributed memory, a test output file named testnnn.ww3 is employed, with nnn representing the processor number, commencing from 1.

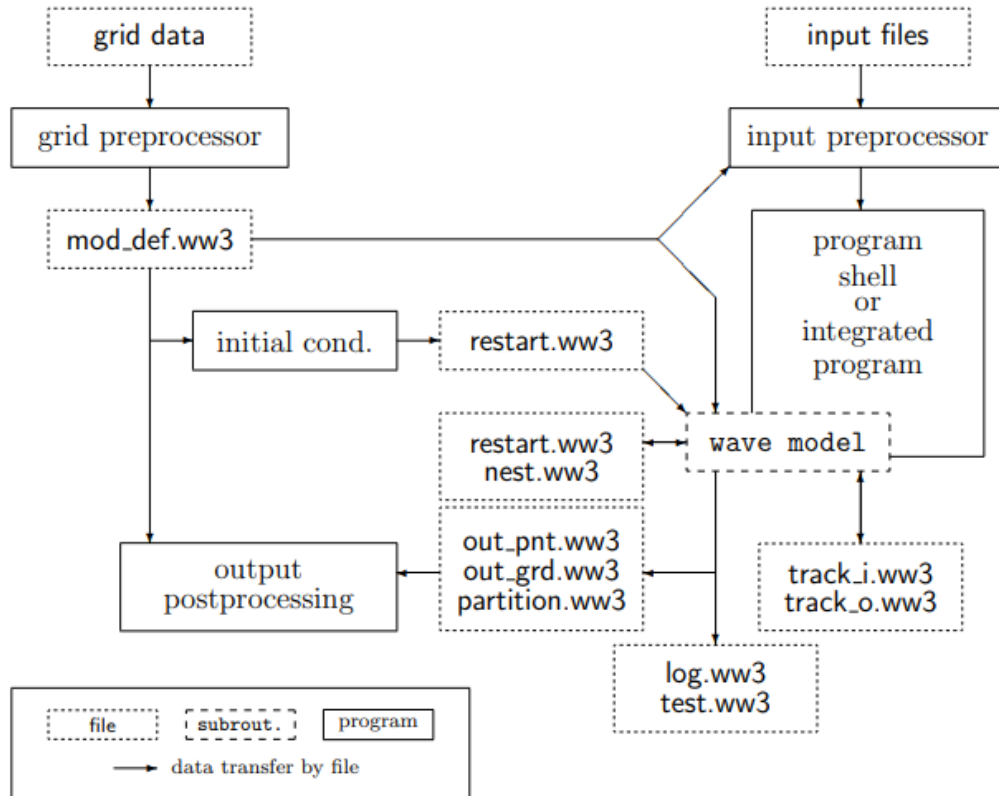


Figure 6. The main elements of the WW3 program

4. WW3 Simulation Results

The Azov Sea was chosen as a testing ground for evaluating the accuracy of the WW3 model. The peculiarity of the Azov Sea is its weather conditions, which are characterized by steady winds, significantly different for the winter and summer seasons. The winter season is chosen because stable winds from the north-west direction prevail on average in winter. The action of the wind along a fixed direction occurs for a long time, which makes it possible to study wave spectra at various accelerations.

The computational areas approximating the configuration of the shores and the bathymetry of real marine basins are determined in the predictive model on regular latitude-longitude grids. The bathymetry required for calculations and the corresponding land-sea mask for each of the basins are constructed using navigation maps. The resulting sampling of the spatial grid is 1.2' in both directions (approximately 2 km). The spectral resolution of the model is 24 directions. The time step for integrating the functions of sources and sinks of wave energy is 5 s, the time steps for energy transmission over the spectrum were 120 s. The wind direction varies in the range from west-northwest to northwest.

The WW3 model was verified on the basis of a regressive analysis of data obtained from calculations and observations from the database Coastal array of monthly average wave heights in the Azov Sea. Figure 7 shows the average (Hsing) and maximum wave heights (Wlen), along with the average wavelength (HsingMAX) by month in the Azov Sea. Furthermore, the directions, periods of peak waves, and maximum wave height for the Azov Sea were calculated based on WW3. The data obtained from the model aligns with the Unified State Information System on the Situation in the Oceans. Notably, the highest wave heights were recorded in the central



Figure 7. Simulation results of the maximum wave height in the Azov Sea

region, while the lowest occurred north of Taganrog Bay, indicating minimal storm activity. Wave direction and intensity in the Azov Sea correlate with prevailing winds and their speed. During colder months, maximum wave heights can reach up to 3 meters (averaging around 0.8 meters), with the Siberian anticyclone influencing wavelengths in the central sea area, reaching 14 meters or more from October to April, often accompanied by ice cover from December to March. In contrast, warmer seasons witness prevailing southwesterly winds, with wave heights typically around 0.2 meters. The discrepancies in the numerical estimates of the disturbance between the data of the WW3 spectral wave model and alternative measurements are within the accuracy of reproducing the corresponding characteristics, the patterns of spatial distribution at the selected scale are very close.

Based on the WW3 spectral model, the forecast of wind wave parameters in the Azov Sea was carried out. The predictive meteorological fields necessary for calculations (wind speed, water and air temperature) are represented by the products of the medium-term meteorological forecast systems of the Hydrometeorological Center of Russia and NCEP/NOAA. The general grid of winds in the ww3-multi process is used as the input grid. To do this, you need to generate Wind.input and mod-def.input (Fig. 8).

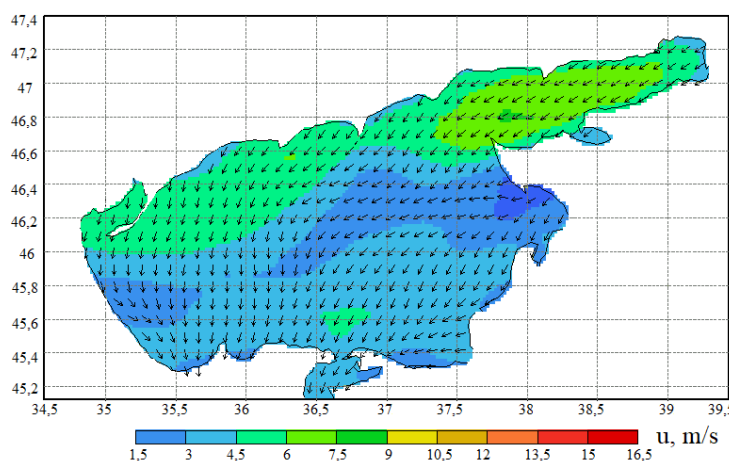


Figure 8. The wind speed at an altitude of 10 meters at the initial moment of the simulation (m/s), the arrows show the wind direction

The calculation grid of the Azov Sea has a resolution of 2 km. Geographical characteristics of the simulated area: the westernmost point: 34.5, the easternmost point: 39.5, the northernmost point: 47.4, southernmost point: 45.2. Data on the current wind speed in the simulated area at an altitude of 10 m is used for calculation.

Prognostic maps of wind wave parameters are constructed based on calculations based on the WW3 spectral model. The predictive modeling of the height of the prevailing waves (Fig. 9), the average length (Fig. 10) and the period of wind waves (Fig. 11) in the Azov Sea at the initial moment of time, and after 3, 6 and 9 hours was carried out. The average wavelength and period of the waves are calculated as weighted averages over the entire frequency-angular spectrum resolved by the model, with weights proportional to the density of the distribution of wave energy over the spectrum.

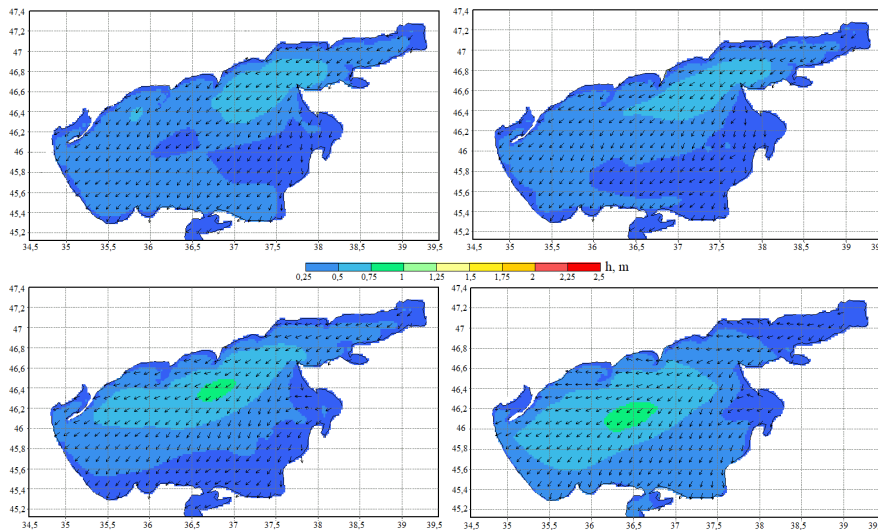


Figure 9. The results of modeling the heights of the prevailing waves (m) at various points in time – at the initial moment, after 3, 6 and 9 hours. The arrows show the average direction of wave propagation

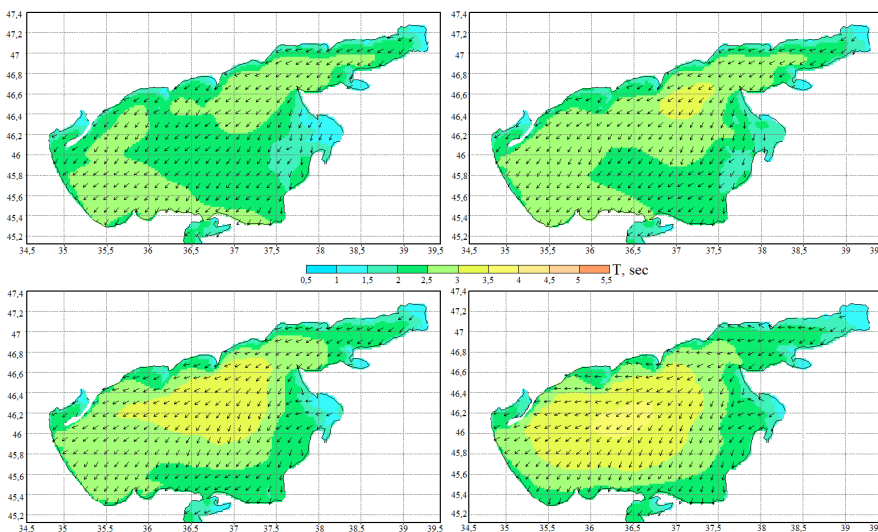


Figure 10. The results of modeling the average wavelength (m) at various points in time – at the initial moment, after 3, 6 and 9 hours. The arrows show the average direction of wave propagation

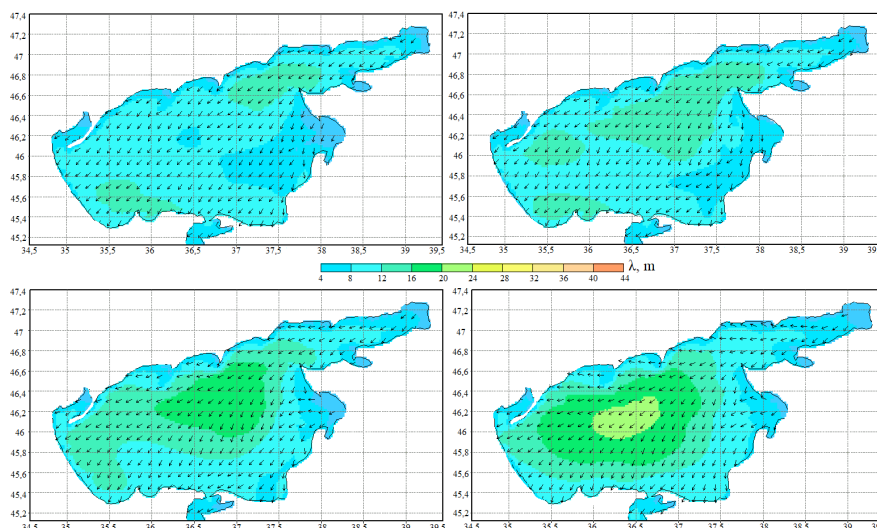


Figure 11. The results of modeling the average wave period (sec) at various points in time – at the initial moment, after 3, 6 and 9 hours. The arrows show the average direction of wave propagation

In the Azov Sea, wave heights predominantly remain below 1 meter, with a recurrence rate of approximately 75%. Waves measuring 1–2 meters occur with a frequency ranging between 20% to 45%, while those exceeding 2 meters repeat no more than 13% of the time. Within the central region of the sea, wave heights typically do not surpass 3.5 meters, with occurrences of heights reaching 4 meters being exceptionally rare [14]. The average discrepancy in the computed wave height values was recorded to be between 5–15 centimeters.

Simulation of wind waves in the Azov Sea makes it possible to predict changes in the height, length and period of waves in a certain marine area, which is important for the safety of navigation and coastal protection. Predictive modeling provides an opportunity to better understand the physical processes underlying wave formation and propagation, climatic and hydrological processes in a given marine area and their impact on the environment and human activities. Thus, wave wave modeling is important both for scientific research and for practical applications in marine and coastal areas.

5. Hybrid Parallelization (MPI-OpenMP) WW3

Parallel launch capabilities have been significantly expanded in WW3 version 5.01, in which multithreading and parallelization using the MPI switch, pure MPI, pure OpenMP and hybrid MPI-OpenMP approaches have become available [8].

In MPI mode, the WW3 model is parallelized using the decomposition of wave spectral components (CD), which imposes a limit on the number of MPI ranks. While hybrid or combined MPI-OpenMP parallelization can enhance node usage, the scalability of OpenMP is limited to multiple threads.

The Domain Decomposition (DD) method serves as a widely utilized tool for parallelization in atmospheric and oceanic models. Presently, most DD applications are based on rectangular subdomains. Despite its simplicity in implementation using standard MPI procedures [12], domain decomposition suffers from potential load imbalances among blocks. Load balancing ensures that each processor carries out a comparable amount of work to prevent overloading on multiple processors. In a parallel environment, the processor with the highest load largely

determines the overall efficiency of the model, as all processors typically await the completion of the last processor's tasks. An effective way to implement domain decomposition involves dividing the x-space into regular rectangular blocks. However, this approach may result in load imbalances in wave models due to variations in the number of sea points across such blocks, influenced by the uneven distribution of land masses. Addressing this issue requires designing the blocks to encompass approximately equal numbers of sea points.

Another method of parallelization in the WW3 model involves decomposing spectral components, abbreviated as CD, and transmitting data via the Message Passing Interface (MPI). Unlike spatial domain division, the CD method addresses the propagation of each spectral component of the wave across the entire region on a single processor, facilitating parallelization with as many processors as there are spectral components. However, the effectiveness of the CD method is limited when two or three spectral components are assigned to the same MPI rank due to variations in computational load across different frequency components. Hybrid parallelization, combining MPI and OpenMP, is a means to extend CD parallelization, theoretically allowing one MPI rank to span multiple processors within a single computing node with shared memory. Nevertheless, in practice, the scalability of the OpenMP method does not exhibit significant improvement beyond multiple OpenMP threads.

Another approach to extending parallelization in WW3 is to combine the CD method with the Domain Decomposition (DD) method. In this configuration, the entire model area is partitioned into subdomains and interconnected using boundary exchange, akin to the standard DD method. However, the combined CD-DD method treats each subdomain as a regional model and additionally parallelizes it using the CD method.

The CD-DD method offers an advantage over the pure DD method by reducing the number of subdomains and, consequently, the number of boundary points. Its scalability is nearly linear when the number of subdomains is small but levels off when the number exceeds the optimal threshold. Another challenge with the model is the memory constraint on a single computer node. Since the MPI method allocates memory across all MPI ranks, data shared by all ranks is duplicated for each rank on a single computing node, even in cases where the node has shared memory capabilities.

CD parallelization is specifically applied to advection and diffusion processes. Wind generation and nonlinear wave interactions are parallelized based on localized sea points. With the CD scheme, land points are excluded from the storage of wave spectra, and the remaining spectra of sea points are evenly distributed across MPI ranks. Terms associated with sea points are computed by iterating through local sea points within each MPI rank. In the MPI-OpenMP hybrid mode, additional parallelization of the local loop at each MPI level is achieved through additional OpenMP directives.

The spectrum in this model is sampled using 30 directions and 36 wavenumbers.

Each individual process in MPI is responsible for storing data on the spectra of wave components for certain points in the sea. The process of calculating wave propagation over the entire region for a specific component of the spectrum is performed in one of the MPI processes. To use resources more efficiently, each MPI (M) process is divided into several OpenMP (N) threads within a single node with shared memory.

In the case of WW3, hybrid parallelization typically comprises the following elements. MPI is employed for parallelization between nodes of a computing cluster or multiple processors. Each process manages its own data region, and MPI facilitates information exchange between

processes and synchronizes task execution across different nodes. OpenMP is utilized for parallelization within a node (i.e., within a processor), enabling more efficient multithreading at the processor level, which is particularly advantageous on modern multicore architectures.

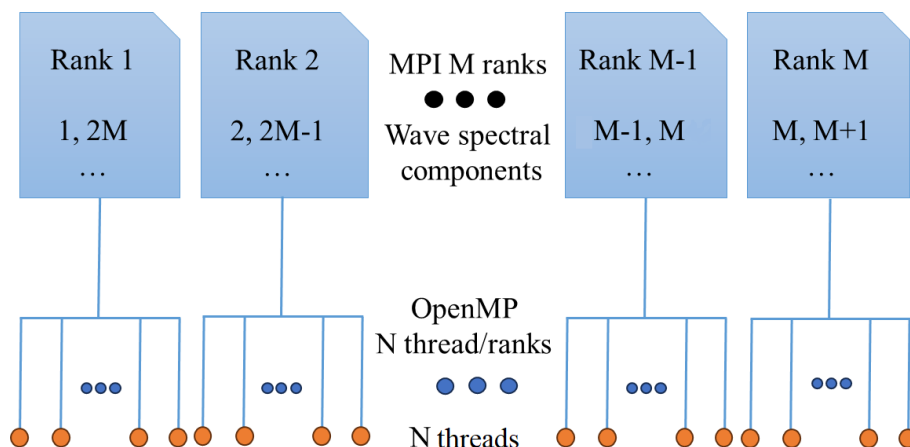


Figure 12. WW3 Hybrid Parallelization

The Sirius NTU cluster, 40 computing nodes (Dell PowerEdge R640 Server) with a total of 1440 computing cores was used. Each computing node has 2 Intel Xeon Gold 6140 processors, 2.3 GHz, 18 cores/36 threads, 10.4 GT/s, 24.75 MB cache, Turbo, HT (140 W), DDR4 2,666 MHz.

Hybrid parallelization can be implemented for $20 \times N$ nodes, but is most effective at $M = 360$ or $10 \times N$ nodes due to different wave velocities in different parts of the computational domain. Hybrid parallelization can use up to 30 computing nodes multiplied by N , but it shows the best efficiency when using about 360 MPI processes or 10 nodes multiplied by N threads (Fig. 13). The total hybrid CPU usage is N per M cores.

Scalability levels off after 360 processes for both MPI and hybrid launches due to load balancing of various spectral components. Component decomposition methods have a limitation that is 1/3 of the spectral components (MPI processes). Hybrid parallelization also has a limit of 9 OpenMP threads.

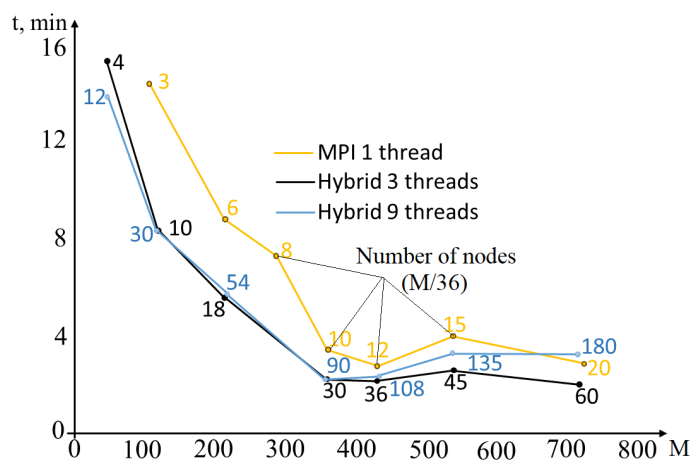


Figure 13. WW3 Parallel Efficiency

The WW3 model incorporates a multiple-cell grid feature, enabling subnets to operate in parallel akin to domain decomposition (DD), while preserving parallelization within each subnet

using the component decomposition (CD) technique. This multiple-cell grid capability not only facilitates the integration of CD-DD parallelization but also allows for higher resolutions.

The multiple-cell grid option has been integrated into the multiple-cell grid module of WW3, enhancing its capabilities.

This update includes enhanced hybrid parallelization in the multiple-cell grid module of WW3, enabling it to effectively handle multiple-cell grids. The improved hybrid parallelization leads to more efficient scaling with increased OpenMP threads and can potentially halve the runtime compared to MPI-only execution.

Conclusions

The study provides a comprehensive analysis of the modern version of the WW3 model, encompassing its fundamental equations, discretization techniques, and software implementation. Additionally, it delves into the construction of a multiple-cell grid for the Azov Sea and elucidates an algorithm for its generation employing two refinement methods.

Furthermore, hybrid parallelization of the wave spectral component (MPI-OpenMP) is introduced, with scalability analysis conducted for both MPI and hybrid launches. It is observed that scalability levels out after 360 ranks for both MPI and hybrid launches due to load balancing considerations across various spectral components with disparate velocities.

The updated multiple-cell grid module in WW3 is presented, boasting improved hybrid parallelization capabilities and the capacity to handle multiple-cell grids. Enhanced hybrid parallelization yields superior scaling with increased OpenMP threads and has the potential to reduce computational time compared to MPI execution.

The findings underscore the significance of the WW3 model in the realms of hydrodynamics and supercomputing. The research highlights the efficacy of hybrid parallelization and the revamped multiple-cell grid module in enhancing model scalability and performance. Further development and optimization of WW3 model parallelization hold promise for delivering even more precise and efficient modeling outcomes in oceanography and marine dynamics.

Acknowledgements

The study was financially supported by the Russian Science Foundation (Project No. 22-11-00295, <https://rscf.ru/project/22-11-00295/>).

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. Alves, J.-H.G.M., Chawla, A., Tolman, H.L., *et al.* The operational implementation of a great lakes wave forecasting system at noaa/ncep. *Weather and Forecasting* 29(6), 1473–1497 (2014). <https://doi.org/10.1175/WAF-D-12-00049.1>
2. Ardhuin, F., Collard, F., Chapron, B., *et al.* Estimates of ocean wave heights and attenuation in sea ice using the SAR wave mode on Sentinel-1A. *Geophys. Res. Lett.* 42(2), 317–325 (2015). <https://doi.org/10.1002/2014GL062940>

3. Ardhuin, F., Rawat, A., Aucan, J. A numerical model for free infragravity waves: definition and validation at regional and global scales. *Ocean Mod.* 77, 20–32 (2014).
4. Gu, J., Li, X., He, Y. A speckle noise suppression method based on surface waves investigation and monitoring data. *Acta Oceanologica Sinica* 42(1), 131–141 (2023). <https://doi.org/10.1007/s13131-022-2103-4>
5. Li, J., Qian, H., Li, H., *et al.* Numerical study of sea waves created by tropical cyclone Jelawat. *Acta Oceanologica Sinica* 5, 64–70 (2011). <https://doi.org/10.1007/s13131-011-0148-x>
6. Li, J., Zhang, Sh., Liu, Q., *et al.* Design and evaluation of an efficient high-precision ocean surface wave model with a multiscale grid system (MSGWav1.0). *Geoscientific Model Development* 16(21), 6393 (2023). <https://doi.org/10.5194/gmd-16-6393-2023>
7. Li, J., Qian, H., Li, H., *et al.* Numerical study of sea waves created by tropical cyclone Jelawat. *Acta Oceanologica Sinica* 5, 64–70 (2011). <https://doi.org/10.1007/s13131-011-0148-x>
8. Li, J.G. Global transport on a spherical multiple-cell grid. *Mon. Weather Rev.* 139, 1536–1555 (2011). <https://doi.org/10.1175/2010MWR3196.1>
9. Li, J.G. Propagation of ocean surface waves on a spherical multiple-cell grid. *J. Comput. Phys.* 231(24), 8262–8277 (2012). <https://doi.org/10.1016/j.jcp.2012.08.007>
10. Li, J.G. Shallow-water equations on a spherical multiple-cell grid. *Q. J. Roy. Meteor. Soc.* 144, 1–12 (2018). <https://doi.org/10.1002/qj.3139>
11. Li, J.G. Upstream Nonoscillatory Advection Schemes. *Monthly Weather Review. Mon. Wea. Rev.* 136, 4709–4729 (2008). <https://doi.org/10.1175/2008MWR2451.1>
12. Voevodin, V.V., Voevodin, V.I.V. *Parallel computing*. BHV-Peterburg. 608 p. (2004).
13. WAVEWATCH III Development Group: Public release version 6.07. <https://github.com/NOAA-EMC/WW3/releases/tag/6.07>.
14. WAVEWATCH III Development Group: User manual and system documentation of WAVEWATCH III version 6.07. Tech. Note 333, NOAA/NWS/NCEP/MMAB, College Park, MD, USA, 465 pp. + Appendices, 2019.

Study of Thin Optical Films Properties Using High-performance Atomistic Simulation

Fedor V. Grigoriev¹, Vladimir B. Sulimov¹ , Alexander V. Tikhonravov¹ 

© The Authors 2024. This paper is published with open access at SuperFri.org

Full-atomistic modeling of the deposition of TiO₂, SiO₂ and TiO₂-SiO₂ films is performed using parallel calculations. The dependence of film density on the deposition angle and deposition energy is studied. Simulation of post-deposition annealing of film structures is also carried out. Mechanical stresses in TiO₂-SiO₂ films, arising due to differences in the properties of silicon dioxide and titanium dioxide, are calculated. It is found that the film density decreases with decreasing deposition energy and increasing deposition angle. The use of surfacing annealing leads to an increase in film thickness. In two-layer TiO₂-SiO₂ films, the stresses are compressive. Particular attention is paid to reducing computational costs when simulating large atomistic clusters, consisting of hundreds of thousands of atoms. Reducing the parameter that determines the calculation of the electrostatic part of interatomic energy significantly reduces the simulation time. At the same time, in this case, the accuracy of determining the electrostatic energy in the reciprocal space decreases, which should be taken into account during modeling.

Keywords: thin films structure, simulation of the deposition process, molecular dynamics, silicon dioxide films, titanium dioxide films, high-performance simulation.

Introduction

Multilayer transparent coatings are widely used in optical and optoelectronic devices for various purposes [23–25]. Typically, these coatings consist of several dozen dielectric layers with different refractive indices. The number of layers and their composition are selected taking into account the required properties of the coating.

There are many different ways to make a clear coat. Physical vapor deposition (PVD) is one of the most widely used methods. In PVD, production process is carried out in a vacuum chamber. Atoms extracted from the target by evaporation or sputtering [10] pass through the chamber and are deposited on the substrate. When the thickness of a certain layer is reached, the target changes and the deposition of the next layer begins. The growth process continues until all layers are deposited.

The deposition process and the properties of the growing layers significantly depend on the conditions in the vacuum chamber, such as the temperature of the substrate, the pressure and composition of the atmosphere in the vacuum chamber, the energy and angular distribution of particles arriving from the target to the substrate, and the flux density of these particles [2, 29, 32]. Experimental study of the corresponding dependencies is a difficult task due to the non-equilibrium nature of the growth process and the large number of parameters affecting deposition. For these reasons, mathematical modeling of the process in a vacuum chamber can make a significant contribution to understanding how exactly process parameters affect film properties.

The most fundamental approach to such modeling is the atomistic one, in which the interactions of the atoms involved in the process are explicitly considered. All atomistic methods can be divided into quantum and classical. Quantum methods are more fundamental than the classical ones, but require large computing resources. For this reason, the characteristic sizes of

¹Research Computing Center, Lomonosov Moscow State University, Moscow, Russia

simulated clusters in quantum simulations are limited to approximately a few nanometers. At the same time, important film parameters require modeling on a scale of tens of nanometers. For this reason, only classical atomistic methods are used to simulate the deposition process.

Currently, these methods are widely used for the calculation of structural, mechanical and some optical parameters of the films [14]. In the present work, we focus on the modeling, requiring a lot of computational resources, in particular:

1. Full-atomistic simulation of thin film deposition. Time of the simulation depends on the number of the atoms N in the films as $\sim N^2 \ln N$;
2. Simulation of the structure of the glancing angle deposited (GLAD) films. These films include nanostructures with characteristic dimensions more than several nanometers. The simulation clusters should include at least several nanostructures. For these reasons, the dimensions of these clusters reach tens of nanometers and consist of millions of atoms.
3. Calculation of stresses in the growing films. The value of main components of stress tensors depends noticeably on the films thickness [13] on a scale of at least several tens of nanometers.
4. Simulation of the post-deposition annealing of the film. The study of changes in film parameters as a result of annealing requires long-term modeling.

The most time-consuming part of atomistic modeling involves calculating the energy of interatomic interaction. The composition of optical film-forming materials includes atoms belonging to various chemical elements. Due to differences in ionization potential and electron affinities of different chemical elements, partial charges appear on atoms. The Coulomb interaction of these charges is long-range and requires special techniques to take it into account. One of these techniques is based on the so-called Ewald summation [9], when the total electrostatic energy is represented by two terms.

First of them quickly converges in real space due to the introduction of a factor that decreases exponentially with the interatomic distance. The second term is long-range, but converges quickly in reciprocal (Fourier) space. The computational cost of this method was significantly reduced by using the Particle Mesh Ewald (PME) method [7]. This method uses approach that simplifies the calculation of inverse electrostatic energy. This approach is justified since this term, as a rule, makes a small contribution to the energy of electrostatic interaction. In this work, in addition to the above-mentioned problems of modeling the characteristics of optical films, we also consider the computational cost of parallel modeling with various parameters used in the calculation of electrostatic interactions.

The simulation was performed using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University [31]. The processors with following characteristics were used: Intel Xeon Gold 6126, 2.6 GHz, 12 cores, 16 GB (queue pascal in the supercomputer Lomonosov-2). The GROMACS program [1] is used for the MD simulation. The VMD [20] program is used for the visual analysis of the atomistic structures of the clusters.

The article is organized as follows. Section 1 is devoted to description of the method of the atomistic simulation of the vapor deposition process. The results of the simulations are discussed in Section 2, including dependence of film density on the simulation parameters (Subsection 2.1), mechanical stresses (Subsection 2.2) and post-deposition annealing (Subsection 2.3). Conclusion summarizes the study.

1. Method of Simulation of Deposition Process

The scheme of the physical vapor deposition (PVD) and simulation area are shown in Fig. 1.

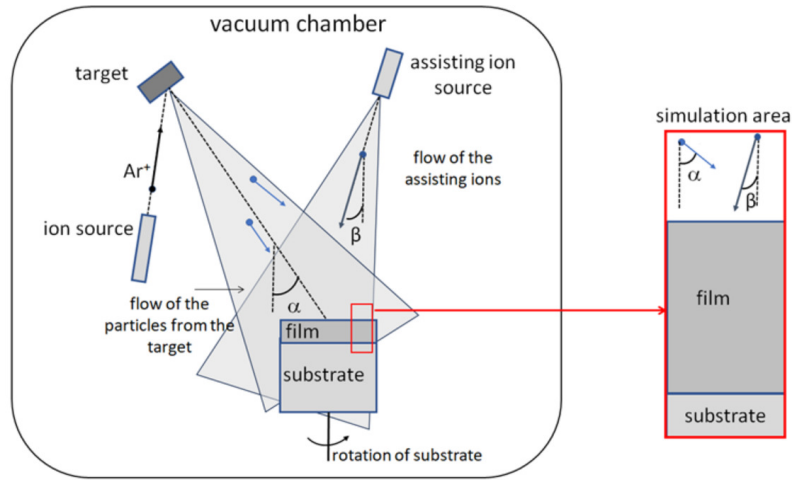


Figure 1. The scheme of the deposition process and simulation area

The particles are sputtered from the target by the high-energy argon ions and are moved to the substrate. The distance between substrate and target is much larger than the dimension of the substrate. Thus, if we neglect the collision of particles with gas molecules, then we can assume that the angles of incidence of the particles on the substrate are the same. The sputtering of the target relate to the high-energy deposition process. In low-energy deposition the particle are emitted due to heating of the target by different methods. In this case, kinetic energy of the particles arriving to the substrate is tenths of an eV. The low-energy deposition, when atoms are extracted from the target by heating, is often accompanied by the ions assisting [19, 34] to improve quality of the growing film (Fig. 1). In this method, the substrate bombard by the high-energy ions, for instance, oxygen ions, that result in the formation of dense and homogeneous film. Substrate rotation is also used to achieve a more homogeneous structure.

The full-atomistic MD simulation is used for the modeling of the final stage of the deposition process, when atoms, extracted from the target, arrive to the substrate, oxidize by the oxygen of the vacuum chamber and form chemical bonds with atoms of film. The macroscopic object – growing film – in the simulation is represented as a microscopic cluster with a characteristic dimension 10–100 nm (Fig. 1, right side). The simulation is organized as a step-by-step procedure [16]. Initially, the substrate structure is prepared using melting-quenching of crystalline SiO_2 cluster [11]. As a result, a cluster of glassy silicon dioxide is formed. The deposition simulation then begins. At each stage, silicon and oxygen atoms are randomly placed at the top of the simulation window. The value of the initial velocities corresponds to the given value of the kinetic energy of the deposited atoms. The directions of the velocities are the same and correspond to the deposition angle. When MD simulation begins, atoms move to the substrate and previously deposited film layers and interact with them, forming chemical bonds. Atoms reflected from the surface are removed after the simulation is completed. The duration of one deposition step is 10 ps. The number of deposition steps depends on the thickness of the film and can reach several thousands.

In the present work, the SiO_2 , TiO_2 and SiO_2 – TiO_2 films are simulated. The potential energy of interatomic interaction for these films is calculated within the DESIL force field [16, 18]:

$$U = \frac{q_i q_j}{r_{ij}} + \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6}, \quad (1)$$

where $q_{i(j)}$ is the charge of the $i(j)$ -th atom, $q_O = -0.5q_{Si} = -0.65e$ for silicon dioxide and $q_O = -0.5q_{Ti} = -1.098e$ for titanium dioxide. The parameters of the Lennard-Jones potential are listed in Tab. 1.

Table 1. Parameters A_{ij} and B_{ij} of the Lennard-Jones potential, Eq. (1)

TiO ₂ [19]		
Types of atoms	A_{ij} (kJ · nm ⁶ /mol)	B_{ij} (kJ · nm ¹² /mol)
Ti-Ti	$5.062 \cdot 10^{-4}$	$3.1195 \cdot 10^{-8}$
O(Ti)-O(Ti)	$3.0009 \cdot 10^{-3}$	$1.735775 \cdot 10^{-6}$
Ti-O(Ti)	$1.4368 \cdot 10^{-3}$	$2.90927 \cdot 10^{-7}$
SiO ₂ [17]		
Types of atoms	A_{ij} (kJ · nm ⁶ /mol)	B_{ij} (kJ · nm ¹² /mol)
Si-Si	$5.0 \cdot 10^{-4}$	$1.5 \cdot 10^{-6}$
O(Si)-O(Si)	$5.0 \cdot 10^{-4}$	$1.5 \cdot 10^{-6}$
Si-O(Si)	$4.2 \cdot 10^{-3}$	$4.6 \cdot 10^{-8}$
Ti-Si1	$5.0 \cdot 10^{-4}$	$2.16 \cdot 10^{-8}$

All the atoms in the simulation clusters are taken into account when potential energy of the interaction is calculated. The Berendsen thermostat [6] is applied to keep the simulation box temperature, $T = 300$ K, constant. The time step in the numerical integration of the equation motion is 0.5 fs. The periodic boundary conditions are applied. The simulation in every step is carried out using the NVT (constant number of particles, volume and temperature) ensemble. After each deposition step, the vertical size of the simulated cluster is increased by 0.01 nm to account for the increase in the thickness of the growing film.

Using this scheme, the following technological parameters of the deposition process can be taken into account: angular and energetic distribution of the arriving to the substrate particles, temperature of the substrate, composition of the flow of the particles, angular and energetic distribution of the assisting ions, relative values of the flows of the deposited particles and assisting ions. The duration of the simulation of the deposition process depends on the previously specified final value of film thickness. Upon completion of the simulation, the calculation of the film parameters begins. These parameters are calculated based on the atomistic structure of the deposited cluster.

2. Results and Discussion

In this section, the results of the simulation of the film density, stresses and post-deposition annealing are presented. As it was mentioned in the Introduction, these tasks require a lot of computational resources due to large dimensions of the atomistic clusters. Also, the ways to reduce the simulation time by varying of the parameters of high-performance calculation are studied.

2.1. Thin Films Density

One of the main structural characteristics of the optical film is density. This is explained by the relationship between density and refractive index: an increase in density is accompanied by an increase in the refractive index [30]. The dependence of the density of silicon dioxide and titanium dioxide films on the film thickness is shown in Figures 2, 3 respectively.

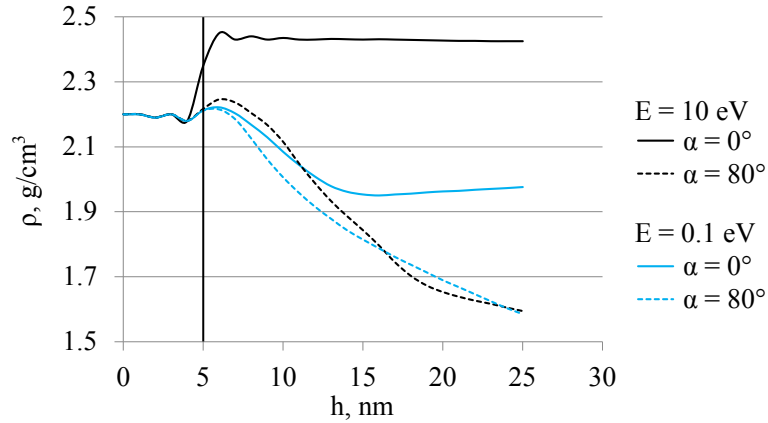


Figure 2. Dependence of the SiO_2 film density (ρ) on the film thickness h , E is the energy of the deposited Si atoms, α is the deposition angle. Vertical solid line denotes the boundary of the substrate

The $\alpha = 0$ corresponds to the normal deposition, when atoms move perpendicular to the substrate. In this case, the film density ρ depends significantly on the energy of the deposition angle. Reduce in the deposition energy results in the noticeable decrease in the film density. The thickness of the transition layer between substrate and film is larger in the case of the low-energy deposition compared to the high-energy deposition. The difference in the ρ at $E = 10$ eV and $E = 0.1$ eV reaches approximately 0.4 g/cm^3 (Fig. 2).

The glancing angle deposition, when the deposited atoms move to the substrate at large deposition angles, results in the formation of the films with low density and high anisotropy [3, 17, 26, 27]. Interestingly, the density of GLAD films depends on the deposition energy much less than the density of normally deposited films (see dotted lines in Fig. 2). The thickness of the transition layer between the substrate and the film is significantly greater than at normal deposition.

Similar trends are observed in the case of titanium dioxide films (Fig. 3). The density shows a significant dependence on the deposition angle and the energy of titanium atoms. Increasing the temperature from 300 K (cold substrate) to 500 K (hot substrate) leads to an increase in the density of low-energy deposited films by approximately 0.1 g/cm^3 .

As it was mentioned in the Introduction section, the time of the simulation of the deposition process is proportional to $t \sim N^2 \ln N$; where N – number of the deposited atoms. The deposition of the clusters, consisting of millions of atoms, requires several days even with using of parallel computations [12]. The parameters PME order and fourierspacing, governing the calculation of the reciprocal part of the electrostatic energy when PME method [7] used in GROMACS program [1], act significantly the simulation time. Reducing the PME order from 8 to 4 reduces the simulation time by more than half. However, such a decrease in time is accompanied by a decrease in the accuracy of the numerical scheme for calculating the inverse part of the electrostatic energy. This decrease acts the calculated parameters of the thin film.

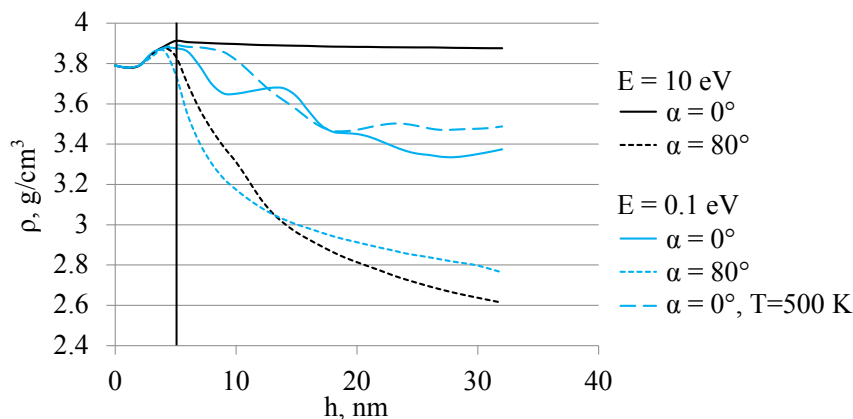


Figure 3. Dependence of the TiO_2 film density (ρ) on the film thickness h , E is the energy of the deposited Ti atoms, α is the deposition angle, T is the substrate temperature, by default $T = 300$ K

The corresponding change in film density is shown in Fig. 4.

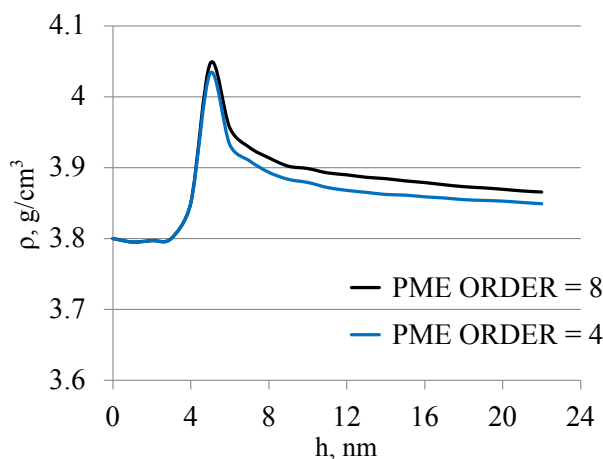


Figure 4. Dependence of the TiO_2 film density (ρ) on the film thickness h at two values of PME order parameter. Energy of the deposited Ti atoms $E(\text{Ti}) = 10$ eV, $T = 300$ K, deposition angle is equal 0

As it can be seen from the graphs, the density calculated for the PME order = 8 exceeds the density for the PME order = 4 by approximately 0.02 g/cm³. This value can be considered as an error in calculating the density when the PME order is reduced from 8 to 4. This error is significantly less than the difference in film density for high-energy and low-energy deposition, as well as for normal and grazing deposition angles (Figures 2, 3). Thus, the PME order = 4 variant can be used when studying the dependence of film density on deposition energy and deposition angle.

2.2. Calculation of Stresses

Stresses in the growing films, arising due to differences in the mechanical parameters of the deposited layers of the different composition act significantly characteristics of the transparent coatings [8, 33]. Large stresses can deform coatings leading to difficulties when using them in devices [5]. For this reason, the calculation of the stresses in the growing films is an actual task for the mathematical modeling.

In frame of the full-atomistic MD approach stresses are defined using the pressure tensor, which is calculated using the kinetic energy tensor and the virial tensor [1, 28]. The procedure of the stress calculation in TiO₂-SiO₂ films includes the following steps:

1. Equilibration of the previously deposited TiO₂ layer in the asymmetric NPT (constant number of the particles, pressure and temperature, $T = 300$ K) ensemble during 20 ps. The asymmetric ensemble is required since the pressure in the horizontal and vertical directions differs significantly due to empty space above the TiO₂ layer surface. The duration of 20 ps is enough for the relaxation of the main components of the pressure tensor.
2. Deposition of the SiO₂ layer to TiO₂ layer in NVT (constant number of the particles, volume and temperature, $T = 300$ K) ensemble. Using of this ensemble models the structural restriction to deposited SiO₂ sublayers by TiO₂ layer. These restrictions results in the occurring of the stresses.
3. Every 100 deposition steps, the deposition procedure is interrupted. The deposited structure is simulated in the NVT ensemble for 100 ps, and the main components of the stress tensor are averaged along the trajectory. The deposition process then begins again.

The several structures of TiO₂-SiO₂ films, obtained in this way, are shown in Fig. 5. The cluster of TiO₂ films, deposited at $T = 300$ K and energy of titanium atoms $E(\text{Ti}) = 10$ eV, was obtained in the previous work [18].

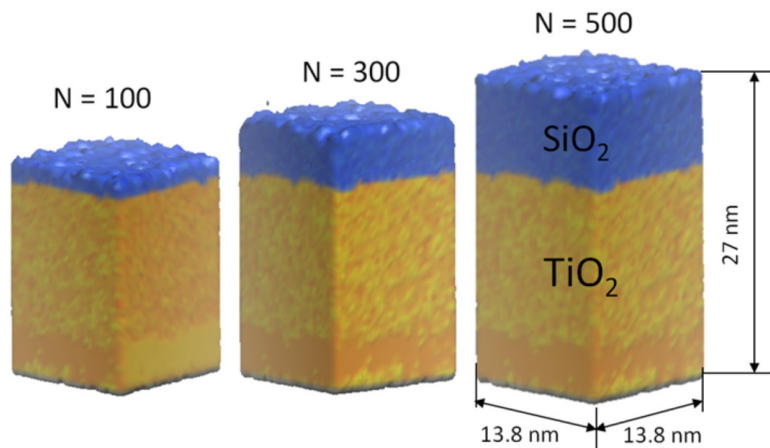


Figure 5. Atomistic clusters of TiO₂-SiO₂ films for different number N of the deposition steps

As it was mentioned in the previous subsection, the value of the PME order parameter significantly decreases the simulation time t (Fig. 6).

In the considered cases, the gain in simulation time is about two and a half times. This gain in t value remains constant over the considered interval of the number of deposition steps N . The second parameter, fourierspacing, also affects the simulation time, but the corresponding effect is less noticeable than for the PME order: increase fourierspacing from 0.12 nm to 0.2 nm leads to a decrease in the t value by approximately 15–20 %.

Results of the calculation of the main component of the stress tensor are shown in Fig. 7.

The all values are negative, that corresponds to the compressive type of the stress. In the case of PME order = 8 the absolute values of σ_{xx} and σ_{yy} are less than in the case of PME order = 4. At PME order = 8 both σ_{xx} and σ_{yy} values decrease at $N > 200$. This tendency is not reproduced at PME order = 4. To summarize, reduce in the PME order value changes significantly the calculated values of the stress tensor components. For this reason, the results

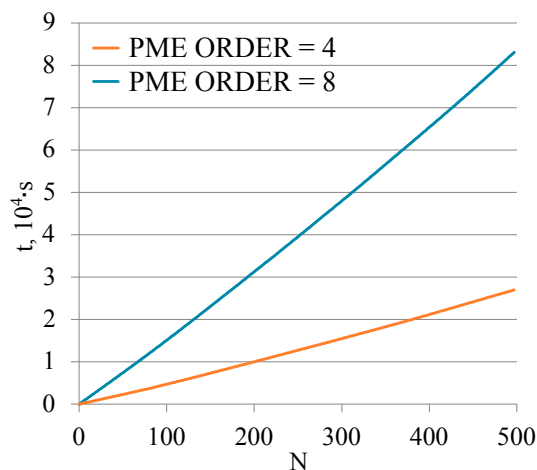


Figure 6. Dependence of the simulation time t on the number of the deposition steps, N at two values of PME order parameters. Number of cores is equal to 32

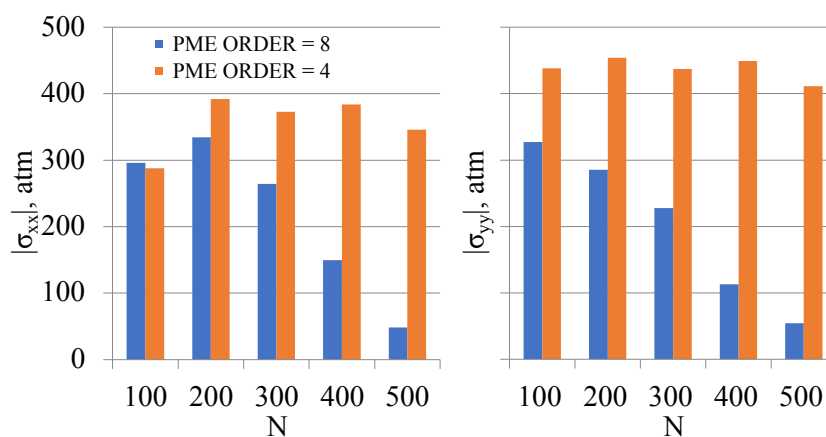


Figure 7. Absolute values of the stress tensor components for two values of the PME order parameter, N is the number of the deposition steps

obtained at PME order = 4 should be considered as preliminary and checked, at least in some points, by calculation at PME order = 8.

2.3. Simulation of Annealing

The post-deposition annealing of the optical coating and film is used to increase their quality and modify the properties [24]. Annealing reduces the refractive index and optical thickness [21], increase the transmittance of the films [22], acts the surface of films, their structural and electrical properties [4].

The annealing can be simulated by full-atomistic simulation. In the agreement with the experiments, the simulation procedure includes the following stages:

1. Heating of the deposited atomistic cluster over time τ from the $T = 300$ K to the annealing temperature T_a .
2. Equilibration of the atomistic cluster at temperature T_a during annealing time τ_a .
3. Cooling from the T_a temperature to the initial temperature $T = 300$ K in time τ .
4. Equilibration of the atomistic cluster at $T = 300$ K during the equilibration time τ_a .

Annealing is performed in the NPT ensemble.

In the present work, the following parameters are chosen: $\tau = 500$ ps, $T_a = 2000$ K, $\tau_a = 1000$ ps. These values are close to those previously used in annealing modeling [15].

The results are shown in Fig. 8.

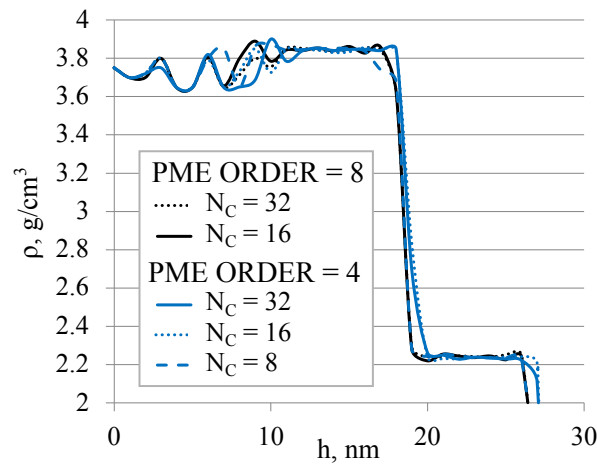


Figure 8. Dependence of the density of the annealed of the in $\text{TiO}_2\text{-SiO}_2$ film on the film thickness at different values of the PME order parameter and different number of the computational cores N_c

Annealing leads to an increase in film thickness. Density fluctuations in the range $h < 12$ nm arise due to the formation of a transition region between the substrate and film. These fluctuations are observed for all values of the PME order parameter and computing cores. The difference in film thickness as the PME order decreases from 8 to 4 reaches approximately 2 nm; see the difference in the position of the film boundary. At the same time, changing the number of computing cores has little effect on the dependence of the film density on h .

The gain in simulation time due to a decrease in the PME order parameter from 8 to 4 times is noticeably greater compared to a similar gain when simulating the deposition process. This can be explained as follows. The simulation of the deposition process is organized as a sequence of steps in which some of the procedures required to run the MD simulation are not parallel. Since the duration of the deposition step is relatively short (10 ps), these procedures have a noticeable impact on the simulation time. In the case of annealing, the procedures are carried out only once and with an annealing simulation duration of 1.5 ns, their impact on the simulation time is negligible.

If the number of computing cores increases by an order of magnitude from 16 to 32, the simulation time is reduced by 1.7 times. A similar gain is obtained in the case of PME order = 4.

Conclusions

Full-atomistic high-performance molecular dynamics simulation of the deposition of TiO_2 , SiO_2 and $\text{TiO}_2\text{-SiO}_2$ films is performed. The films parameters that require the use of large atomistic clusters, consisting of hundreds of thousands of atoms, are calculated.

For both TiO_2 and SiO_2 films, the density decreases with decreasing deposition energy and increasing deposition angle. The stresses in the deposited $\text{TiO}_2\text{-SiO}_2$ films are compressive and reach three hundred bar in absolute value. Annealing leads to an increase in film thickness.

Ways to reduce computational costs are discussed. It is revealed that reducing the PME order parameter, which determines the calculation of electrostatic energy in reciprocal space,

reduces the simulation time. It can be used to calculate the dependence of film density on their thickness. At the same time, when calculating stresses, a decrease in the PME order leads to noticeable changes in the absolute values of the stress tensor components.

Acknowledgements

The work was supported by the Russian Science Foundation (grant No. 23-11-00047).

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, 19–25 (2015). <https://doi.org/10.1016/j.softx.2015.06.001>
2. Amirzada, M.R., Tatzel, A., Viereck, V., Hillmer, H.: Surface roughness analysis of SiO₂ for PECVD, PVD and IBD on different substrates. *Applied Nanoscience* 6, 215–222 (2016). <https://doi.org/10.1007/s13204-015-0432-8>
3. Badorreck, H., Steinecke, M., Jensen, L., *et al*: Correlation of structural and optical properties using virtual materials analysis. *Optics Express* 27(16), 22209–22225 (2019). <https://doi.org/10.1364/OE.27.022209>
4. Bakri, A., Sahdan, M.Z., Adriyanto, F., *et al*: Effect of annealing temperature of titanium dioxide thin films on structural and electrical properties. In: *AIP conference proceedings*. vol. 1788. AIP Publishing (2017). <https://doi.org/10.1063/1.4968283>
5. Begou, T., Lumeau, J.: Accurate analysis of mechanical stress in dielectric multilayers. *Optics Letters* 42(16), 3217–3220 (2017). <https://doi.org/10.1364/OL.42.003217>
6. Berendsen, H.J., Postma, J.v., Van Gunsteren, W.F., *et al*: Molecular dynamics with coupling to an external bath. *The Journal of chemical physics* 81(8), 3684–3690 (1984). <https://doi.org/10.1063/1.448118>
7. Darden, T., York, D., Pedersen, L.: Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems. *The Journal of chemical physics* 98(12), 10089–10092 (1993). <https://doi.org/10.1063/1.464397>
8. Ennos, A.E.: Stresses developed in optical film coatings. *Applied optics* 5(1), 51–61 (1966). <https://doi.org/10.1364/AO.5.000051>
9. Ewald, P.P.: Die berechnung optischer und elektrostatischer gitterpotentiale. *Annalen der physik* 369(3), 253–287 (1921). <https://doi.org/10.1002/andp.19213690304>
10. Greene, J.E.: Tracing the recorded history of thin-film sputter deposition: From the 1800s to 2017. *Journal of Vacuum Science & Technology A* 35(5) (2017). <https://doi.org/10.1116/1.4998940>

11. Grigoriev, F., Sulimov, A., Kochikov, I., *et al*: Computational experiments on atomistic modeling of thin-film deposition. *Applied Optics* 56(4), C87–C90 (2017). <https://doi.org/10.1364/AO.56.000C87>
12. Grigoriev, F., Sulimov, V., Tikhonravov, A.: High-performance full-atomistic simulation of optical thin films. *Supercomputing Frontiers and Innovations* 5(3), 130–133 (2018). <https://doi.org/10.14529/jsfi180325>
13. Grigoriev, F.V., Sulimov, V.B., Tikhonravov, A.V.: Atomistic simulation of stresses in growing silicon dioxide films. *Coatings* 10(3), 220 (2020). <https://doi.org/10.3390/coatings10030220>
14. Grigoriev, F.V., Sulimov, V.B.: Atomistic simulation of physical vapor deposition of optical thin films. *Nanomaterials* 13(11), 1717 (2023). <https://doi.org/10.3390/nano13111717>
15. Grigoriev, F., Katkova, E., Sulimov, A., Sulimov, V., Tikhonravov, A.: Annealing of deposited SiO₂ thin films: full-atomistic simulation results. *Optical Materials Express* 6(12), 3960–3966 (2016). <https://doi.org/10.1364/OME.6.003960>
16. Grigoriev, F., Sulimov, A., Kochikov, I., *et al*: Supercomputer modeling of the ion beam sputtering process: full-atomistic level. In: *Optical Systems Design 2015: Advances in Optical Thin Films V*. vol. 9627, pp. 34–42. SPIE (2015). <https://doi.org/10.1117/12.2190938>
17. Grigoriev, F., Sulimov, V., Tikhonravov, A.: Structure of highly porous silicon dioxide thin film: Results of atomistic simulation. *Coatings* 9(9), 568 (2019). <https://doi.org/10.3390/coatings9090568>
18. Grigoriev, F., Sulimov, V., Tikhonravov, A.: Application of a large-scale molecular dynamics approach to modelling the deposition of TiO₂ thin films. *Computational Materials Science* 188, 110202 (2021). <https://doi.org/10.1016/j.commatsci.2020.110202>
19. Guo, C., Kong, M.: Fabrication of ultralow stress TiO₂/SiO₂ optical coatings by plasma ion-assisted deposition. *Coatings* 10(8), 720 (2020). <https://doi.org/10.3390/coatings10080720>
20. Humphrey, W., Dalke, A., Schulten, K.: Vmd: visual molecular dynamics. *Journal of molecular graphics* 14(1), 33–38 (1996). [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5)
21. Jiang, Y., Ji, Y., Liu, H., *et al*: Insights into effects of thermal annealing on optical properties of SiO₂ films. In: *6th International Symposium on Advanced Optical Manufacturing and Testing Technologies: Advanced Optical Manufacturing Technologies*. vol. 8416, pp. 106–110. SPIE (2012). <https://doi.org/10.1117/12.974980>
22. Li, S., Liu, C., Zhu, T., *et al*: Effects of the thermal treatments on the optical properties of SiO₂ anti-reflective coatings on sapphire windows. *Infrared Physics & Technology* 137, 105151 (2024). <https://doi.org/10.1016/j.infrared.2024.105151>
23. Martinu, L., Poitras, D.: Plasma deposition of optical films and coatings: A review. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* 18(6), 2619–2645 (2000). <https://doi.org/10.1116/1.1314395>

24. Piegari, A.: FF (Ed.) *Optical Thin Films and Coatings* (2018)
25. Schulz, U.: Review of modern techniques to generate antireflective properties on thermoplastic polymers. *Applied optics* 45(7), 1608–1618 (2006). <https://doi.org/10.1364/AO.45.001608>
26. Smy, T., Vick, D., Brett, M., *et al*: Three-dimensional simulation of film microstructure produced by glancing angle deposition. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* 18(5), 2507–2512 (2000). <https://doi.org/10.1116/1.1286394>
27. Tait, R., Smy, T., Brett, M.: Modelling and characterization of columnar growth in evaporated films. *Thin Solid Films* 226(2), 196–201 (1993). [https://doi.org/10.1016/0040-6090\(93\)90378-3](https://doi.org/10.1016/0040-6090(93)90378-3)
28. Thompson, A.P., Plimpton, S.J., Mattson, W.: General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions. *The Journal of chemical physics* 131(15) (2009). <https://doi.org/10.1063/1.3245303>
29. Tolenis, T., Grinevičiūtė, L., Smalakys, L., *et al*: Next generation highly resistant mirrors featuring all-silica layers. *Scientific reports* 7(1), 10898 (2017). <https://doi.org/10.1038/s41598-017-11275-0>
30. Vedam, K., Limsuwan, P.: Piezo-and elasto-optic properties of liquids under high pressure. II. Refractive index vs density. *The Journal of Chemical Physics* 69(11), 4772–4778 (1978). <https://doi.org/10.1063/1.436530>
31. 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. *Supercomputing Frontiers and Innovations* 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>
32. Xu, C., Qiang, Y., Zhu, Y., *et al*: Effects of deposition parameters on laser-induced damage threshold of Ta₂O₅ films. *Optics & Laser Technology* 42(3), 497–502 (2010). <https://doi.org/10.1016/j.optlastec.2009.09.004>
33. Ying, D., Zhong, T.: Effects of thickness and annealing on the residual stress of TiO₂ film. *Optics Continuum* 3(3), 287–295 (2024). <https://doi.org/10.1364/OPTCON.506007>
34. Zoeller, A., Beisswenger, S., Goetzelmann, R., Matl, K.: Plasma-ion-assisted-deposition: a novel technique for the production of optical coatings. In: *Optical Interference Coatings*. vol. 2253, pp. 394–402. SPIE (1994). <https://doi.org/10.1117/12.192112>