ACADÉMIE
DES SCIENCES
INSTITUT DE FRANCE

Comptes Rendus

Mécanique

Nicolas Bertier, Denis Gueyffier and Eric Quémerais

Introduction: from pioneering works to nowadays cutting-edge multiphysics simulations

Volume 353 (2025), p. 1157-1159

Online since: 5 November 2025

https://doi.org/10.5802/crmeca.326

The Comptes Rendus. Mécanique *are a member of the Mersenne Center for open scientific publishing*
www.centre-mersenne.org — e-ISSN : 1873-7234

ACADÉMIE DES SCIENCES
INSTITUT DE FRANCE

Comptes Rendus. Mécanique
2025, Vol. 353, p. 1157-1159
https://doi.org/10.5802/crmeca.326

Foreword / *Avant-propos*

Introduction: from pioneering works to nowadays cutting-edge multiphysics simulations

Introduction : des travaux pionniers aux simulations multiphysiques de pointe actuelles

Nicolas Bertier [a], Denis Gueyffier [b] and Eric Quémerais [c]

[a] DMPE, ONERA, Université de Toulouse, 31000 Toulouse, France

[b] DSG, ONERA, Université Paris-Saclay, 91120 Palaiseau, France

[c] DMPE, ONERA, Université Paris-Saclay, 92320 Châtillon, France

E-mails: nicolas.bertier@onera.fr, denis.gueyffier@onera.fr, eric.quemerais@onera.fr

This issue explores the shift in the field of numerical simulation towards developing increasingly complex software platforms and frameworks. Among the reasons which led to this trend is the need to simulate complex industrial systems (e.g. airplanes, rockets, power plant…), systems in extreme conditions involving a large number of physics (e.g. atmospheric reentry, plasma assisted ignition, nuclear fusion…) or the need to create simulation software leveraging the potential and power of next generation supercomputers. We have gathered in this issue scientific articles showing the rich interplay between the sciences and techniques of numerical simulation, high-performance computing and software engineering necessary to build simulation frameworks for solving the multiphysics problems which are still intractable today. Advanced applications achieved with these recent tools are also presented in fields such as aerodynamics, energy, and in domains requiring the resolution of coupled multiphysical problems.

Before introducing the issue's papers, we describe, starting from the early days of numerical simulation, how the scientific community has made use of increasing computer power along with advanced simulation software to solve more and more complex problems. New algorithms and software architecture developed in the meantime has fueled the creation of such cutting-edge simulation software.

Although many mathematical foundations already existed, modern numerical simulation is often said to begin with the pioneering work of John Von Neumann at Los Alamos during the Manhattan Project (from 1942 to 1946). The objective was to simulate nuclear explosions, leading to numerous foundational works in numerical fluid mechanics, particularly on compressible fluid dynamics. The wartime effort also led to the invention of early computers in several countries (Z3 in Germany, Colossus in England, Harvard Mark I in USA…) and under the impulsion of Von Neumann, the US Army's ENIAC became the first computer with memory storage capacity. At that time, models were extremely simplified, often associated with one-dimensional approaches, and programs were written in machine languages (binary, octal, or decimal) using punched cards as physical support.

The 1950s saw the establishment of several foundational elements of numerical simulation, including the development of finite element and finite volume methods, the emergence of computers with more advanced memory storage (IBM 701 in 1952 with a Von Neumann's architecture), and the advent of the Fortran language (Formula Translation, 1957). Several specialized calculation codes (in thermal, structural, and fluid mechanics, acoustics...) were developed in the 1960s–80s, preceding the rise of multiphysical numerical simulation as we know it today and the development of commercial software. In parallel, computing power increased considerably, roughly following Moore's Law ("The number of transistors in an integrated circuit doubles approximately every 18 to 24 months at constant cost"). The late 70s and early 80s saw the rise of the Cray family of supercomputers (Cray 1, Cray X-MP, Cray 2...) using vector processors and shared memory. The release of the first Silicon Graphics SGI station (SGI IRIS 1000) in 1984 enabled significant advancements in visualization and post-processing of simulation results. Researchers started tackling three-dimensional problems using these computers.

The 2000s marked the entry into the era of distributed memory clusters which enabled multiphysics calculations (notably using MPI protocol) on an increasingly large number of cores. This era led to better exploitation of processor chip architecture (memory levels and vectorized floating-point operation registers). But, during this period, the trend of increasing processor speeds was replaced with increasing the number of processor cores. With the development of GPU, today's supercomputers are hybrid clusters with multicore processors and GPU accelerators. All these scientific and technical advances have led to a complexification of numerical tool development, profoundly modifying the way numerical simulation tools are designed. While, until the 1990s, each numerical simulation software could still be developed by one person, teams with specialized profiles (experts of physical modeling, numerical method specialists, computer scientists, HPC gurus...) were gradually formed. As application complexity and tool development increased, it became increasingly difficult for a single team to possess the critical mass and expertise required to develop a tool with all the needed functionalities. This led to the development of more elaborate simulation platforms with modular design, where each software component can be developed by a different, specialized team. Each component can benefit from algorithms, programming techniques, and languages best adapted to both usage and the developing team's culture. The complexity is partially shifted to inter-component communication, raising coupling and data model problems. New architectures simplify code maintenance and addition of new functionalities. In 2025, we are at a true turning point in the discipline of numerical simulation. As several articles in this special issue attest, great maturity has been achieved in multiphysics platform development and in related coupling challenges.

In this issue, de Chaisemartin et al. present "ArcNum", an Arcane-based numerical framework, and its application to porous media flow simulation. In a different field, Antoine et al. focus on aeroelasticity with a strong emphasis on coupling challenges. Simon et al. discuss high-order adaptive multistep coupling algorithms, with an application to partial differential equations. In addition, a particularly important challenge for coupling is to provide efficient geometric tools on parallel hardware architectures. This aspect is highlighted by Andrieu et al., with a work on dynamic load-balanced point location algorithm for data mapping. Finally, considering the full complexity of most advanced applications while limiting CPU cost requires the use of advanced features such as dynamic mesh adaptation. Lugrin et al.'s work on a canonical dual mode scramjet isolator is a fine illustration of this aspect.

We can now seriously consider the advent of genuine "digital twins", using current tools. However, since 2012 and the deep learning revolution, artificial intelligence and its applications have experienced exponential growth. Its incursion into numerical simulation is likely to revolutionize the discipline. We believe that machine learning will become pervasive in the field of numerical simulation, as can be attested by the rise of state of the art machine learning models like Aurora, GraphCast and GenCast in the field of weather prediction, challenging classical atmosphere simulation models. Substantial advances can be imagined in modeling (turbulence, combustion, liquid atomization...), in augmenting and replacing space and time discretization methods or in the way code is generated (using copilot-like assistants, on-the-fly documentation, automatic verification and optimization...).

Another consequence of the massive training of large language models and multimodal generative AI is the exploding demand for GPUs together with the multiplication of chips vendors. The rise of GPUs makes it essential for numerical simulation tools to adopt versatile architectures and to enable efficient use on both CPUs and GPUs. This has been performed elegantly using code generation among other techniques as shown in the the paper of Lienhardt et al., “SONICS: design and workflow of a new CFD software”.

Two important trends in the field of numerical simulation are not addressed in this issue. The first one is cloud computing, as we believe it will greatly democratize access to HPC resources (e.g., small businesses performing high-fidelity simulations without supercomputer infrastructure) and improve user experience. The second one is the development of quantum computers. In the long term, they will likely become the backbone for solving numerical simulation problems, but this entails a completely different approach to numerical methods for solving PDEs than the ones which have persisted since the first CPUs, making it a significant challenge. These two trends will certainly be the subject of intense scientific and technical efforts in the community, and may become the subject of a future issue of *Comptes Rendus de l’Académie des Sciences*.