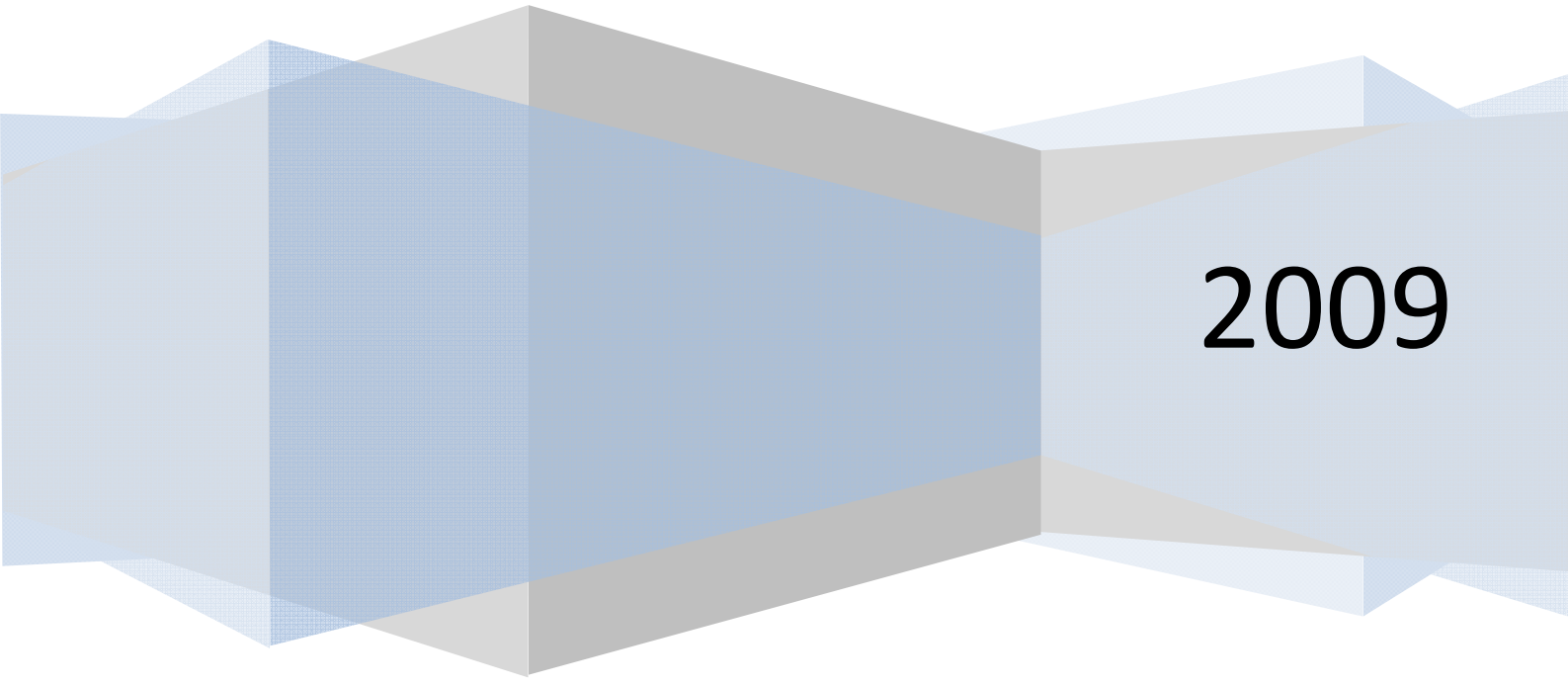
European Science Foundation

European Computational Science Forum: The “Lincei Initiative”: from computers to scientific excellence



2009

Organising Committee

Doctor Vassilis Pontikis, *Chair***,**
Commissariat à l'Énergie Atomique , Saclay , Gif-sur-Yvette, France

Professor Carmen N. Afonso, *PESC rapporteur*,
Consejo Superior de Investigaciones Cientifica, Instituto de Optica, Madrid, Spain

Professor Isabel Ambar, *LESC rapporteur*,
Directora Instituto de Oceanografia Faculdade de Ciências da Universidade de Lisboa

Professor Kenneth Badcock,
Dept. of Engineering, The University of Liverpool, Liverpool, United Kingdom

Professor Giovanni Ciccotti,
Dept. of Physics, Università "La Sapienza", Roma, Italy

Professor Peter H. Dederichs,
Institut für Festkörperforschung, Jülich Research Centre, Jülich, Germany

Professor Paul Durham,
Daresbury Laboratory, Warrington, United Kingdom

Professor Franco Antonio Gianturco,
Dept. of Chemistry, Università "La Sapienza", Roma, Italy

Professor Volker Heine,
Cavendish Laboratory (TCM), Cambridge University, Cambridge, United Kingdom

Professor Ralf Klessen,
Institut für Theoretische Astrophysik, Zentrum für Astronomie der Universität Heidelberg, Germany

Professor Peter Nielaba,
Lehrstuhl für Theoretische Physik, Fachbereich Physik, Universität Konstanz, Konstanz, Germany

Doctor Simone Meloni, *Scientific Secretary*,
Consorzio per le Applicazioni del Supercalcolo per Università e Ricerca - CASPUR, Roma, Italy

Further contributors

Professor Jonathan Essex,
Department of Chemistry, University of Southampton, Southampton, United Kingdom

Professor George Evangelakis,
Physics Department, University of Ioannina, Ioannina, Greece

Professor Peter J. Knowles,
School of Chemistry, University of Cardiff, Cardiff, United Kingdom

Professor Richard Lavery,
Institut de Biologie et Chimie des Protéines, Lyon, France

Doctor Marie-Christine Sawley,
CERN, Genève, Switzerland

Doctor Rodolphe Vuilleumier,
Département de Chimie ,Ecole Normale Supérieure, Paris, France

Table of Content

Executive summary

The growing importance of numerical studies in science is a major fact attested by the increasing use of computers as an heuristic research tool in parallel with the traditional experimental and theoretical approaches. Problems which looked untreatable years ago, like *ab-initio* modeling of the structure and dynamics of complex materials, are nowadays within reach. The same could be said about bridging space- and time-scales in simulations of biological problems or in modeling in engineering. These possibilities open new perspectives on a number of scientific challenges, and it is expected that much of our understanding in the coming years will rely on progress in the computational sciences.

This trend was recognised and integrated at an early stage into European initiatives aimed at the improvement of computational tools by means of an infrastructure providing researchers access to High Performance Computing.

However the effort in Europe is mainly focused on computer hardware, while much less structured support is given to the writing, maintenance and dissemination of scientific codes of increasing complexity; to the dissemination of new advances among the scientific community to speed innovation; and to enable other diverse forms of cooperation, including training. This approach produces also a considerable waste of resources in the development of similar codes in several centres. This unequal treatment between hardware and software leads to a long time lag between a new conceptual development by one researcher and its availability to the wider community or, similarly, between the appearance of a new hardware architecture and the adaptation of existing codes to it; all facts which are threatening to compromise Europe's leading role in scientific computing.

Some structuring of research organisations for supporting software is a prerequisite for keeping Europe ahead of the international competition in science and its industrial applications. This includes, the production, updating, maintenance, shared use, sustainability and dissemination of codes, as well as the dissemination of ideas, and training in complex codes and their methodology. Such needs are well recognised and dealt with in the USA (Appendix 1).

The European Science Foundation Forward Look “European Computational Science Forum: The Lincei Initiative - from Computers to Scientific Excellence” has collected viewpoints, needs and suggestions from several different scientific communities that are presented together in the main text of this report, and more individually in the appendices. These have led to the following findings and recommendations:

Principal findings

Computational Sciences have become a major tool in the theoretical understanding of experiments

in many different fields. It is expected that this trend will become even more important while progress is achieved in algorithms and methods.

The most important bottleneck is the support to software. Effort in Europe is mainly focused on hardware, while less structured support is given to the writing, maintenance and dissemination of scientific codes although their complexity is constantly increasing. Moreover, sustainability issues are not in general explicitly considered and, very often, scientific computer programs do not comply with best practices in programming.

European expertise in applied computer science needs to be brought into the development of robust and widely usable codes, and to sustain them under continuous development and the arrival of new computer architectures. The situation in Europe does not yet fit this ideal configuration. Indeed, developers of scientific software are often acting on an individual basis, though limited support is sometimes obtained in the context of high performance computing initiatives.

There is an absence of good training of researchers in using today's sophisticated codes, due to the lack of a long-term organization in scientific software development.

Developing several areas of European cooperation would greatly enhance Europe's competitive excellence in scientific computing, including dissemination and training in code use with 'help desk' assistance, dissemination of the latest advances in research and code development, education in the methodology of scientific computing and dissemination of training.

The solution to the above problems has to be on a European cooperative basis because one country is too small a unit for all the expertise. To maximise first class work, researchers need good access to the ideas, expertise and codes of others across Europe forming a dynamic cooperating community. Varying degrees of cooperation already exist among researchers in most fields of scientific computing. It has developed best where there is some structure that brings people together.

Recommendations

It is recommended that National science funding agencies in Europe undertake a coordinated and sustained effort in scientific software development, including documentation, updating, maintenance and dissemination. This necessarily implies the means for training and cooperation. Restructure and federate, within an European-scale infrastructure, existing and expanded activities on scientific software and other forms of cooperation and dissemination in Europe through European Computational Collaborations specific to each scientific area. This would be guided by active research scientists and deliver the infrastructural services to the working scientists.

To achieve those goals, it is proposed to set up a Computational Sciences Expert Committee (CSEC) attached to ESF which would speak for the whole community of computational sciences. Its purpose

would be to start setting up a durable plan for European cooperation in each of the fields of science using computers. It would address the policy issues involved, and work with national and European organisations to optimize the development of scientific computing in Europe.

Only successful efforts in all the technical areas required to support scientific computing, namely hardware, system and application software, as well as greater cooperation in dissemination of codes and state-of-the-art developments, will allow Europe to maintain and extend its competitive leadership in the future.

Chapter 1. Introduction: origins, purpose and outline of this Forward Look

Over the last decade or so, computing has made a qualitative change in the kind of scientific understanding of the world that can be achieved. It has turned qualitative suggestions into quantitatively proven understanding, e.g. about how to recognise black holes or the origin of the solar system. In biology, computing allows one to extract meaning from the large amount of data in the genetic code, and similarly computers allow one to visualise, and thus extract meaning from, the vast data streams produced by satellites. In physics, chemistry, surface science and materials science, it has allowed quantum mechanical calculations to understand phenomena at the basic atomic level, which is feeding into technology and modern industry.

In some areas such as the examples above, the computing (while very sophisticated and complex) is a component contributing to an experimental or theoretical research programme; but in other cases, giant computer simulations are themselves the cutting edge research. For example one can study in many ways the circulation of the atmosphere and that of the oceans and their interactions, but only a computer simulation incorporating all that information can predict how global warming may affect future climate. Similarly a biological cell includes many individual processes from the interaction between two biomolecules to the passage of various substances in and out through the cell wall, but only a large computer simulation can show the full complexity of their interactions and hence allow one to study how this is affected by some agent causing disease.

In the last few years, scientific computing has really come of age. Although one can point to earlier important contributions, scientific computing took off from the advent of powerful computers at the end of the 1970s, not only central supercomputers but also powerful individual workstations. Since then computers, of course, have become fantastically more powerful both in speed and extent.

However even greater progress has come from discovering how to do the calculations and do them better. One talks about 'codes', 'programmes' or 'applications software' running on the computers (the 'hardware'), and the 'algorithms' in them for doing or organising the actual calculations.

Progress of hardware also induces evolution of codes. In fact, every time a new computer is available, codes need to be adapted to it, in order to take advantage of its characteristics. Continuous growth in the computing capacities (doubling every 13 months) mainly relies nowadays on the evolution of new processor conception and computer architecture. It is also expected that new systems will be needed in the coming years to keep up with present power increase. Those changes will affect all the levels of the computational pyramid.

On top of this, computational sciences are “sciences in the making”, with new approaches and methodologies being constantly discussed inside the respective scientific communities. As a matter of fact there is a continual development and innovation of algorithms at the cutting edge of scientific computing.

Unfortunately the extent and speed of that change in computer codes has resulted in some bottlenecks now holding back scientific computing in Europe. They relate to the computer codes themselves and, more widely, to the essential scientific cooperation, pointing to the need for a structured approach. In USA similar strains have been felt and are being tackled (see Appendix 1).

Similarly to large-scale equipment infrastructures (e.g. telescopes, synchrotrons), code development is very interdisciplinary as it combines scientific modeling, mathematics, and computer science to serve a broadly-based research community. However, unlike the relative stability of other large facilities, computer architecture changes every two years or so, and major computer codes are under continuous development lasting, typically, ten years or more.

The infrastructural nature of computer hardware facilities has been recognised for quite some time in Europe, with GÉANT[1], DEISA[2] and recently PRACE[3] being examples of European efforts in this direction. However the codes to run on the computers, and various aspects of dissemination, cooperation and training, are equally vital parts of the necessary infrastructure. This is the area where the rapid pace of change requires improvement and restructuring. Although hardware evolution is one of the drivers of this process, hardware support itself is outside the scope of this Forward Look as it is already covered by several initiatives at national and European level.

The aim of this Forward Look is to develop a vision of how scientific computing will evolve in the coming 10 to 20 years. Based on this scenario, and on the needs of the scientific computing community, a strategy will be presented aiming at structuring software support and development at the European level. This strategy will also have to take into account the evolution of hardware. Implementation of the recommendations of this Forward Look should lead to an advanced cyber-infrastructure that allows Europe to maintain its leading position in this field.

Scientists from several fields, Engineering, Astrophysics, Material Science and Nanotechnology, Chemistry, Life Sciences, and Meteorology and Climatology met at the “Accademia dei Lincei” in Rome on July 12th-13th 2005 with the objective of analysing the present state of scientific computing and identifying priorities, needs and weaknesses in this field.

It was decided to carry out a detailed survey on this topic. The Centre Européen de Calcul Atomique and Moleculaire (CECAM) coordinated the writing of a proposal asking the European Science Foundation (ESF) to support a Forward Look activity entitled 'European Computational Science – The

[1] The GÉANT project, http://www.geant.net/, was a collaboration between 26 National Research and Education Networks (NRENs) across Europe, the European Commission, and DANTE (the project's coordinating partner). The project ran 2000–2005 and was succeeded by GÉANT2 http://www.geant2.net/.

[2] DEISA, Distributed European Infrastructure for Supercomputing Applications, http://www.deisa.eu/, is a consortium of leading national supercomputing centres that currently deploys and operates a persistent, production quality, distributed supercomputing environment with continental scope.

[3] PRACE, Partnership for Advanced Computing in Europe, http://www.prace-project.eu/, prepares the creation of a persistent pan-European HPC service, consisting several tier-0 centres providing European researchers with access to capability computers and forming the top level of the European HPC ecosystem. PRACE is a project funded in part by the EUs 7th Framework Programme.

LINCEI Initiative: From Computers to Scientific Excellence'. Favourably considered by the ESF, the Forward Look[4] has also been supported financially by other national institutions and European networks[5]. The Forward Look started on May 25 2006, by establishing a Scientific Committee representing some of the main disciplines involved and coming from several European countries.

The Organising Committee of this Forward Look has organised six workshops in the thematic areas of Astrophysics (Institut für Theoretische Astrophysik, Heidelberg (DE), Dec. 1st-2nd 2006), Fluid Dynamics (Daresbury Lab., Warrington (UK), Nov. 29th-30th 2006), Meteorology and Climatology (Swiss Supercomputing Centre, Manno (CH), Jan. 27th 2007), Life sciences (Chilworth Manor, Southampton (UK), Nov. 19th-21st 2006), Material Science and Nanotechnology (Jülich Research Centre, Jülich (DE), Nov. 13th-14th 2006) and Quantum Molecular Sciences ("Accademia dei Lincei", Rome, Nov. 25th-26th 2006). The participants were asked to describe the actual state of infrastructure for scientific computing in their field, commenting on strengths and weaknesses. They were also asked to identify the needs in relation to future scientific challenges, in a 10-20 year timeframe.

A word of caution is in order about the choice of disciplines for thematic workshops. To report on the complete set of the fields where computing has become an important tool for solving problems would be a very difficult task, going well beyond the human and technical capacities of the Organising Committee. As an example, members of the particle physics community, who were first users of High Performance Computing (HPC) in Europe, or researchers in bio-informatics or social sciences may feel that problems specific to their practice are not addressed in the report. They would be partly right. However, we think that the fields chosen represent a fair sample of the whole picture. The analysis presented here may be applied, with little adaptation to the fields not represented. Moreover, we see this Forward Look as a first step, which has to be followed by actions, one of which would be some sort of community building, where each community would be identified and its specificity recognized.

The ensemble of conclusions of these scientific meetings were presented and discussed in a workshop attended by representatives of national supercomputer centres, hardware manufacturers and software vendors (IBM, HP, Cray, NEC, Intel and Microsoft) and the organising committee members of the Forward Look ("Accademia dei Lincei", Rome (IT), April 23rd-24th 2007).

All the information collected in these workshops and the analysis performed by the participants were elaborated by the Organising Committee of this Forward Look and are presented in this report.

[4] http://cyberinfrastructure.caspur.it and http://www.esf.org/activities/forward-looks/physical-and-engineering-sciences-pesc/current-forward-looks-in-physical-and-engineering-sciences/european-computational-science-the-lincei-initiative-from-computers-to-scientific-excellence.html

[5] The European network Psi-k, The Marie Curie MOLSIMU and the Molecular Simulations in Biosystems and Material Science European programmes (SimBioMa), the Jülich Research Centre (GE), the research centre at Daresbury (UK), the Swiss National Supercomputing Centre (CSCS), and the Italian Consorzio interuniversitario per le Applicazioni di Supercalcolo Per Università e Ricerca (CASPUR).

The general conclusion is that in spite of specific problems in different areas of scientific computing, the needs for software development, sharing, standardization, user support, training, education and dissemination of the latest advances are common to all of them. These needs can be satisfied by providing long-term financial support for code development and sustainability and user support, and careers for people providing these services. These services, and the means of providing them, are very similar to those offered by synchrotrons, telescopes and other large research facilities. This is why the need for creating a European software 'code and cooperation' infrastructure has been approved unanimously by the Organising Committee and is considered as the optimal way forward.

In the following chapters, all these issues will be addressed in detail and recommendations will be suggested to make codes available not only to the scientific and engineering communities represented in this Forward Look, but also to many others which will be very likely interested, thus enhancing synergy among groups that usually do not communicate with each other, and preventing duplication of effort. The outline of this report is the following: chapter 2 describes general aspects of scientific computing emerging from the analysis of the topical workshops. Common trends of the present shortcommings and needs are summarized. In chapter 3, steps for improving the situation are listed and a strategy is proposed, before the conclusions and recommendations are provided in the end. Background information, workshop reports and more detailed analyses are shifted to the appendices.

Chapter 2. Scientific computing: present, shortcomings and future evolution

Computational scientific software is written on purpose to satisfy the needs of research and applications in different environments such as universities and research or industrial centres. Whereas ten years ago every research group could develop its own research codes, nowadays the complexity and sophistication of scientific codes in use have grown to such an extent that the research work of many groups relies on the availability of free or licensed software, and the number of users is steadily increasing. The latter are not only theoreticians solving numerically models for which analytical solutions are not available, or computer scientists developing improved algorithms and new types of applications, but also experimentalists turning to numerical simulations for de-convoluting and interpreting complex data.

In the following we first examine the present situation of scientific computing in Europe and identify a list of common needs and difficulties which have been collected during the area-specific workshops convened by the organizing committee of this Forward Look. Then we list the conclusions drawn from discussions held during the workshops about the future of scientific computing in Europe and the obstacles hindering their development.

The present situation

Software for scientific computing (atomic and molecular simulations, astrophysics, life sciences, …) can be classified into two main categories: homemade software and community packages. These are written in different environments, correspond to a large variety of coding styles, and may not incorporate best programming practices. As a consequence, such packages are not reusable[6] without partial or total rewriting (software duplication); porting and optimizing such codes on new architectures is a difficult or impossible task. Moreover, it is worth noting that both reusability and portability issues are heavily affected by the complexity of codes, partly accounted for by their size in terms of the number of lines, usually ranging from few thousands to millions of lines.

User support, that is activities covering code documentation, user training and education and "help-desk" services for solving problems, are important ingredients for efficiently using scientific software. In spite of the serious efforts undertaken to implement these items for modern codes (cf. Appendix 9), researchers still complain that there is a lack of detailed documentation. At the present time, documentation is a side activity of code developers so that it is difficult to increase effort on it without adding new staff members to existing teams. User training, education and help-desk activities are undertaken by several national or European institutions (CCP's[7], CECAM[8], EMBO[9],

[6] Reusable software is a software module or product designed to provide a function or facility that other designers may include in their own codes.

[7] Computational Collaborative Projects (UK).

FEBS[10], ESO[11] …) and by code developers. However, the lack of a common general approach underlying these activities is responsible for leaving important issues, such as training in software design and education, unaddressed. In consequence, scientific codes are often not reusable and the risk of misuse of computational tools is increasing with the increasing trend of using scientific codes as "black boxes", i.e. without having a sufficient understanding of the underlying science. Finally, since scientific computing is by essence inter-disciplinary, training and support of researchers requires expertise from different fields (e.g environmental sciences and climatology, engineering, computer science). Thus, education and training of new researchers implies strong networking and interaction with computer scientists, hardware vendors and other experts, external to the given scientific community.

A scientific computer code is a hierarchical combination of a scientific model with numerical methods and approximations inherent to the model, or related to the precision of the numerical scheme and the finite precision of digital computers (rounding). Due to the aforementioned deficiencies affecting code development, maintenance and user support, detailed knowledge about a code is available only within development teams. However, with few exceptions, writing codes is mainly an activity of non-permanent staff, postgraduate or postdoctoral researchers. Therefore expertise and knowledge are fluctuating in time at the rate of replacement of members of development teams and can even vanish when overlap between new and departing members is not sufficient. As a consequence, resources are wasted in the process of appropriate knowledge transfer or re-writing code when recovering was not possible. It should also be noted that postgraduate or postdoctoral researchers do not generally have extended skills in numerical mathematics and programming, whereas computer scientists are not systematically members of code-development teams. Exceptions however exist in supercomputer centres and/or computer industries supporting porting, parallelizing and optimizing of selected academic packages[12].

In addition to software development and user support, scientific software needs also code management, including merging, versioning and validation of new or modified code components. Code management leads to user-support requirements: documentation, training and "help-desk" must be aligned with the new versions. Such activities are continuous, time-consuming and require the expertise of computer scientists which is not always present in developer teams (see above). Thus, code management often lags behind, resulting in incompatibilities, conflicting components and less reliable codes.

The overall picture of scientific computing that emerged from the subject-workshops is however not homogeneous. Notable exceptions are observed in the astrophysics community, in which

[8] Centre Européen de Calcul Atomique et Moléculaire
[9] European Molecular Biology Organization
[10] Federation of European Biochemical Society
[11] European Southern Observatory, http://www.eso.org
[12] In most European countries no discipline-specific organizations exist coordinating the development or supporting the writing of codes. This is in contrast with the situation in the UK, where the Collaborative Computational Projects (CCPs) partially fill the need.

considerable attention has been given to the data management problem, and in the community of environmental sciences and climatology within which drawbacks still hindering development of other computational areas, have been considered and solved for a long time.

Simulations can produce a large amount of data (from terabytes up to petabytes). In environmental sciences and climatology, data are maintained and archived by national institutes and high-performance computing centres for post-processing purposes. However, a lack of data integration (data originated from simulations and experiments) prevents scientists from accessing information available outside their own institutions. Much progress has been made on data management in astrophysics where the "International Virtual Observatory Alliance" is an infrastructure coping with this problem, but much less in other scientific computing communities.

Another exception is in environmental sciences and climatology where service organizations, research institutes and universities take care of code development (e.g. codes Alpine3D, CCSM, ECHAM, LM, NEMO and their derivatives). Similarly, these organizations and community forums work on the reusability of codes via a European cooperation aiming at the development, evaluation and maintenance of state-of-the-art codes and software tools while supporting standardization.

The future

An important issue in scientific computing is that of future developments requiring the writing of new codes and their implementation on existing and forthcoming hardware architectures. It is likely that the aforementioned limitations and drawbacks will lead to an increasing waste of resources if appropriate measures are not rapidly adopted. More specifically, foreseen difficulties are expected to relate to: (i) the development of codes to solve new scientific problems; (ii) new applications using combinations of existing codes e.g. multiscale approaches; and (iii) the evolution of the architecture (beyond-petascale computers, GPUs for scientific computing[13], cell computing[14]) and of the basic functional principles of hardware (quantum computers[15]). Indeed, from past experience it

[13] Graphic Processing Units are emerging as a competitive architecture for general purpose computing, and especially scientific computing, thanks to their low cost/ power consumption with respect their performance. See http://www.scientific-computing.com/hpcforscience/feature-gpu.html for a presentation of GPU for scientific computing and http://www.khronos.org/opencl and http://www.nvidia.com/object/cuda_home.html about languages for programming such architectures.

[14] The aim of cell architecture is to emphasize the energetic efficiency and the cost effectiveness of processors. The architecture prioritizes bandwidth over latency, and favours peak computational throughput over simplicity of programming. IBM's latest supercomputer, Roadrunner, the first petaFlops-class machine,, is a hybrid general purpose computer (based on AMD Opteron/Cell processors). See http://www.research.ibm.com/cell for a description of such an architecture

[15] A quantum computer is a device for computation that makes direct use of quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. The basic principle of quantum computation is that quantum properties can be used to represent data and perform operations on these data. See http://www.cs.caltech.edu/~westside/quantum-intro.html for an introduction and

is well known that adapting existing software to new architectures is sometimes more difficult than an extensive or total re-writing of the codes. Similarly, in multiscale approaches, the coupling between codes operating at different space/time scales is made through appropriate interfaces, which is a difficult or impossible task if the programs to be coupled are heterogeneous. Finally, codes written for solving new problems in scientific computing must address portability and sustainability aspects (object oriented programming, archiving, documentation) if they are to be used efficiently over a long period.

Here we give few examples of future scientific challenges that require the actions described above.

Aerospace

The scientific challenges in aerospace research are driven by the green agenda for civil aircraft. High fidelity simulations are needed to allow for the opportunities and risks associated with nonlinearity to be accounted for in the assessment of novel configurations. For example, the use of active structural elements in an aircraft wing has the potential to optimise fuel efficiency over several flight conditions. The development of a credible realisation of this concept requires a simulation that accounts for nonlinear fluid-structure interaction. This can only be done with a properly validated code.

The maintenance of such codes for research purposes requires a community approach, where established methods are robustly available and validation from widespread sources is exploited. A code which is open rather than commercial or tied to one company would be the strongest in the medium term, and would have the largest chance of impact on the aerospace community.

Biological systems.

Another example of scientific challenges that require some action along the direction drawn above is in the simulation of biological systems. In this field, the range of modelling methods applied is very large, with a high degree of specialisation appropriate to each level of detail. At the smallest scale, quantum mechanical methods may be used to describe reaction mechanisms within an enzyme (100 $Å^3$ system sizes, 10 ps timescale), molecular mechanics may be used to model protein-drug interactions (100 nm^3, 100 ns), while coarse-grained modelling can capture the interactions between protein assemblies and lipid membranes (100 nm^3, 1 ms). Further models exist to capture larger-scale behaviour. For example, continuum models based on reaction-diffusion equations have been used to capture the response of a synapse to changes in concentration of acetylcholine, biochemical reaction networks in cells have been simulated using numerical modelling of the associated rate equations, while the electrical behaviour of the heart has also been described using coupled

http://qist.lanl.gov for a roadmap on quantum computing

differential equations.

For a biological model to be truly complete, it is essential that each level is able to feed into the next. This would allow, for example, for the binding of an irreversible ligand to a particular enzyme to be predicted using quantum mechanics, the result of this binding event to cascade through the hierarchy of models, to finally yield a predicted impact at the whole organ level. This interaction between various levels is not to be understood in a reductionist perspective: as recent progresses in integrative biology have shown, the connection of successive levels needs refined analyses, where systems theory provide ways to determine what are the essential elementary steps, and eventually predict new pathways. Successful approaches along those lines require close cooperations between physicists, chemists, engineers, mathematicians, biologists and physicians. Matching all those expertises, together with the corresponding modeling and software tools, is a major challenge of the computational life sciences.

Astrophysics

Yet another example of computational challenges requiring the building of a computational infrastructure for software is understanding the formation and evolution of our Milky Way Galaxy. New survey missions will provide a wealth of high precision observational data of the Milky Way. For example, the Gaia[16] satellite will yield the complete phase space information of 1 billion stars in the Galaxy. To understand and interpret these data is an enormous task. A major goal of Galactic theoretical astrophysics therefore is to model formation and evolution of the Milky Way *ab initio* with very high precision and with all relevant physical processes taken into account consistently. The goal is to build a one-to-one model of our Galaxy on the computer. This requires validation of codes, management of and interoperability of scientific software with theoretical and experimental data as well as interoperable scientific software for simulating the various physical and chemical processes that occur in this process.

Materials Science

One of the next big step in computational Materials Science will be multiscale modeling of materials and this requires some restructuring of computational infrastructures as described before. The advantage of this approach is the possibility of combining the predictive power of *ab-initio* simulations with the ability of "coarser grained" approaches, like classical molecular dynamics, to simulate larger samples of materials. In fact, many phenomena occur at length scales that cannot be accessed in *ab-initio* simulations. Crack propagation in brittle materials is an example of such a phenomenon. In this field, the aim is the design of new materials able to resist high external stress

[16] Global Astrometric Interferometer for Astrophysics. www.rssd.esa.int/index.php?project=GAIA&page=index

before the formation and propagation of cracks.

As mentionned in the case of life sciences (biomaterials), further progress will result from an interaction of modeling expertise at several different scales. Another aspect which is worth mentionning in this context is that, even at the most elementary scale, difficult problems remain unsolved as yet. This is the case, for example, for the effect of strong electron correlations in Density Functionnal Theories, as well as many aspects of the quantum dynamics modeling. The scientific community should continuously discuss and decide which approaches are more likely to be successful, in order to concentrate efforts, in particular for code development. Several ESF networks (psi-k, intelbiomat) have been and still are concerned with those issues.

Before closing this chapter, let us stress that for the six communities inspected, organizations of various status and importance have been created by the scientists which organize the discussions on the science as well as on the more practical aspects of code development and validation. Those discussions are for the most organized inside the respective domains, as very little interaction with other fields is taking place, showing an overall fragmented scientific community. This Forward Look has been the opportunity to have a first encounter and comparison between those different disciplines.

Chapter 3. Scientific computing: Ways for improvements

The synthesis of the conclusions of the subject-workshops has led the organizing committee of the Forward Look to the following conclusions:

What should be achieved

Cooperation and networking at the European scale are essential for the development of excellent, efficient and competitive software, dissemination of the latest advances and preventing the loss of effort/knowledge. Development, software maintenance and user support activities should appropriately combine technical aspects aiming at the efficient use of existing codes with tight and durable interactions between scientists and applied computer scientists for new directions to emerge in scientific computing.

Documentation, education, training and 'help desk' services are necessary on a permanent basis and at an appropriate level for efficiently using scientific software.

Definition of programming rules, best practices and standards for inter-operability is required for the development of sustainable software.

Long-term support is needed for development of new codes and maintenance of existing software. Concerning the first aspect, it is important that good cooperation is organized in the respective scientific communities so that researchers will be able to set priorities for what they decide to be their community "flagship" activities. As for the second aspect, career perspectives should be offered to the technicians and scientists involved in development, maintenance and associated services (documentation, training, …).

Those requirements are partly met at the present time, but in an indirect way, through funding of proposals essentially on the basis of their scientific goals. It would be extremely beneficial for the whole scientific community to have those activities directly and explicitly funded, and to have it organized at the European scale.

How to implement

There are several possible ways to deliver the services just listed. Possibilities for the structure required have been considered by the Forward Look OC and the one chosen is reproduced in appendix 10. It is named the ESCRINS model and it is meant to be used as a reference helping to clarify ideas.

The principles on which this construction has been based are the following:

(a) the existence of field-specific community organizations which we call European Computational Collaborations (ECC) which are led by scientists. In most of the fields, the community fora already exist in a variety of forms and status: they could be invited to join and form ECC.

(b) to link those collaborations to respective existing permanent institutions for each given field (EMBO, CERN, CECAM..), this for the sake of durability, funding and control.

In order to initiate the process, it is proposed to create rapidly a high-level policy forum in the form of an Expert Committee attached to ESF (Computational Sciences Expert Committee, CSEC). It would be an advisory body, speaking for computational sciences and addressing policy issues with the national funding agencies and the European Institutions. Its role would be to start with two or three subject areas, gather the researchers and try to persuade the funding agencies, national and European, to provide support in a scheme which would have to be negotiated. It should also be charged to connect to the whole of computational sciences, and to make an in-depth study of the particular needs in each area of science.

Conclusion and Recommendations

The European Science Foundation Forward Look "European Computational Science Forum: The Lincei Initiative - from computers to scientific excellence" has collected viewpoints, needs and suggestions from several different scientific and technical communities about scientific software development that have been presented in the foregoing sections. This Forward Look has identified common needs and drawbacks in current methods of developing, disseminating and using scientific software in Europe.

Computational Science has become a major tool in the theoretical understanding of experiments in many different fields. It is expected that this trend will become even more important as progress continues to be made in algorithms and methods.

The most important bottleneck is the support to software. Effort in Europe is mainly focused on hardware, while less structured support is given to the writing, maintenance and dissemination of scientific codes, although their complexity is constantly increasing. Moreover, sustainability issues are not in general explicitly considered and, very often, scientific computer programs do not comply with best practices in programming.

European expertise in applied computer science needs to be channelled towards the development of robust and widely usable codes, which can be sustained under continuous development and the arrival of new computer architectures. The situation in Europe does not yet fit this ideal configuration. Indeed, developers of scientific software are often acting on an individual basis, though limited support is sometimes obtained in the context of high performance computing initiatives.

There is an absence of good training of researchers in using modern sophisticated codes, due to the lack of a long-term organization in scientific software development. Training in code use with 'help desk' assistance, dissemination of the latest advances in research and code development, and education in the methodology of scientific computing, are all most effectively addressed at the European level.

Developing several areas of European cooperation to address the above issues would greatly enhance Europe's competitive excellence in scientific computing.

To answer these needs and eliminate the obstacles hindering the development of scientific computing in Europe, the organizing committee of the Forward Look proposes that a European Scientific Computing Research Infrastructure be created. A model (ESCRINS, see Appendix 10) is being suggested which would undertake a durable effort in scientific software and other cooperation. Only through such an effort to support hardware, system and application software, and other cooperation, can Europe maintain and extend its competitive leadership in scientific

computing in the future[17]. Such an effort require the definition of a new policy in Europe: it is proposed that computational researchers at the highest national and European level state this policy in an advisory committee, the Computational Sciences Expert Committee (CSEC), hosted by the ESF.

In summary the organizing committee of this Forward Look recommends:

- To undertake a sustained effort in scientific software development, including documentation, updating, maintenance and dissemination. This necessarily implies the means for training and cooperation. Restructure and federate, within a European-scale infrastructure, existing and expanded activities on scientific software and other forms of cooperation and dissemination in Europe through European Computational Collaborations specific to each scientific area. This would be guided by active research scientists and deliver the infrastructural services to the working scientists.
- To this effect, set up a high level Computational Science Expert Committee (CSEC) which will consider policy issues, including those required for the initial stage of implementation of the present recommendations and will represent scientific computing at the highest level of policy-making, both nationally and European.

[17] In the USA the SciDAC programme with similar objectives is following on from earlier coordinated action, and is already in its second round of funding. See Appendix 1.

Appendices

Appendix 1. Initiatives in the USA

This appendix presents a summary of some of the American initiatives in the field of scientific computing developed over the last fifteen years.

An early, and still active, relevant initiative was taken in the early 90s (1992) to develop a facility to study environmental chemistry. In this context, a parallel software package for quantum chemistry simulation was developed. The software, still under development and available free of charge, was named NWChem[18]. NWChem is now much broader in scope, allowing users to perform classical and *ab-initio* simulations both in vacuum and in condensed phase.

Several aspects of this project are pertinent to the present report. First of all, at that time parallel computing was just starting and parallel algorithms, as well as tools for parallel programming, were not available. In order to overcome these limitations an interdisciplinary team, including scientists of different backgrounds, was set up. This team included 'applied scientists' (physicists, chemists, etc.), applied mathematicians and applied computer scientists. In the first part of the project (1992-1997) this team consisted of 15 people. The NWChem project was supported by the US Department of Energy (DoE) with ~ $2.5 M/year. NWChem turned out to be very successful and modern both from the point of view of the physics implemented in the code (several new algorithms were developed under this project) and the code structure. As a matter of fact, the current version of NWChem (V5.1) still preserve the original structure.

Later, in 2005 the 'President's Information Technology Advisory Committee' (PITAC) focused its attention on computational science and its needs[19]).

Their main conclusion is that "Among the obstacles to progress there are rigid disciplinary silos in academia that are mirrored in the organizational structures of the Federal research and development agency. These silos impede the development of multidisciplinary research and educational approaches that are essential to computational science."

In order to overcome these obstacles the committee recommends that "both universities and Federal R&D agencies should fundamentally change these organizational structures to promote and reward collaborative research."

[18] http://www.emsl.pnl.gov/docs/nwchem/nwchem.html see also “NWChem: Development of a Modern Quantum Chemistry Program", T. H. Dunning, R. J. Harrison, and J. A. Nichols, CTWatch Quarterly, 2 (2006), Vol 2 (http://www.ctwatch.org/quarterly/articles/2006/05/nwchem-development-of-a-modern-quantum-chemistry-program)

[19] PITAC Report "Computational Science: Ensuring America's Competitiveness’, http://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf

In practice, they suggest to have hardware, scientific and system software, and education, evolve in a harmonized way so that at any given time all these aspects can be at a homogeneous level of development.

They introduce the key concept of "Sustained Infrastructure for Discovery and Competitiveness", which includes the "Software Sustainability Centres", closely related to the aim of ESCRINS we propose in this report. Their analysis and consequent recommendation related to this infrastructure are crucial: "Like any complex ecosystem, the whole flourishes only when all its components thrive. Only sustained, coordinated investment in people, software, hardware, and data, based on strategic planning, will enable the United States to realize the promise of computational science to revolutionize scientific discovery, increase economic competitiveness, and enhance national security. The Federal government must implement coordinated, long-term computational science programs ... forming a balanced, coherent system that also includes regional and local resources."

Two major initiatives are actually being implemented or developed following the recommendations of the PITAC report: the DoE's Scientific Discovery through Advanced Computing (SciDAC) programme and the National Science Foundation's (NFS's) Office of Cyberinfrastructure (OCI).

The aim of the DoE's SCIDAC programme[20] is to develop and implement new algorithms able to fully exploit the computational power of modern computer architectures. This programme was started in 2001 with $60 M/year funding, and is now in its second 5-year term (2006-2010), funded at $70 M/year. "Most of the funds were invested in software research and development, recognizing that the greatest need was to begin work in the Scientific Computing Software Infrastructure" said T. H. Dunning Jr. during the Keynote speech at the SciDAC 2005 conference[21]. In practice, a set of Integrated Software Infrastructure Centres (ISICs) was established. The aim of these centres is to provide software of broad interest for SciDAC Scientific projects. Together with the ISICs, SciDAC has also founded a series of SciDAC Institutes. Their main goal is to train students and researchers on scientific and high-performance computing topics.

The aim of the NSF's OCI[22] is to "Provide the science and engineering communities with access to world-class CI tools and services, including those focused on: high performance computing; data, data analysis and visualization; networked resources and virtual organizations; and learning and workforce development;"[23]. Actions are planned for the "development and maintenance of portable, scalable application software". In fact, it was recognised that "The creation of well-engineered, easy-to-use software will reduce the complexity and time-to-solution of today's challenging scientific applications." The organizational aspects were also considered. Among the guiding principles

[20] http://www.scidac.gov

[21] "The once and future SciDAC", T H Dunning Jr, *J. Phys.: Conf. Ser*. **16** (2005) http://dx.doi.org/10.1088/1742-6596/16/1/E03 .

[22] www.nsf.gov/oci

[23] NSF 07-28, "Cyberinfrastructure Vision for 21st Century Discovery" (March 2007) http://www.nsf.gov/pubs/2007/nsf0728/index.jsp

established for implementing the organisational objective for the 2006-2010 time-frame, "NFS investments will promote equitable provision of and access to all types of physical, digital, and human resources to ensure the broadest participation of individuals with interest in science and engineering enquiry and learning."

In conclusion, several actions have been carried out in the US over the last 20 years to strengthen the computational science community and develop a software infrastructure. A very distinctive characteristic of these initiatives is that all the components are considered on the same footing: hardware, system and application software, human and organisational aspects. The level of the investments planned for the software aspects (including development of new algorithms for modern computer architectures) is also relevant. Considering only the SciDAC programme, the US are actually investing $70 M/year in the field.

Appendix 2. Questions put to the six Subject Workshops

The Forward Look organized six community level workshops. At that stage, what is now called the ESCRINS infrastructure was referred to as the CyberInfrastructure, and this term appears throughout the following six reports, but it has since then been dropped to avoid confusion with a somewhat different meaning in the USA. Also in what follows the expression 'computational science' must be interpreted as mainline science using computers: it is not a separate branch of science.

Each of the six communities was asked to address the following points:

Item 1 Presentation of your community and of the current and future level of importance of computational science in your field, with examples of applications.

Item 2 Current status of CyberInfrastructure in your field, describing strengths and weaknesses of computational science applications.

- How is the lack of a CyberInfrastructure hindering progress in your field?
- What are the strengths of the elements of CyberInfrastructure already implemented in your field?
- Among the elements missing in the CyberInfrastructure identified in first point, which are to be implemented most urgently?

Item 3 Requests for the CyberInfrastructure of the future, in a 10-15 year timeframe.

- Describe in details the elements of the future CyberInfrastructure in your field.
- Technological challenges in implementing such a CyberInfrastructure:
  - in the field of data intensive applications: management, mining, interrogation and validation of data produced by observatories and laboratories;
  - in the field of intensive computing applications: novel paradigms in serial and parallel programming, novel languages, novel hardware architectures;
  - in the field of management: deployment, maintenance and operation of the CyberInfrastructure.
- Scientific progress to be expected after implementing the CyberInfrstructure, with prototypical examples.
- Opportunities in advancement and foundation of novel multidisciplinary sciences.

Item 4 The path from the current status to the envisioned CyberInfrastructure.

- Analysis of successive steps, and milestones, to develop the current CyberInfrastructure into the future one:
  - describe the components of the CyberInfrastructure in successive steps along the 10-15 year time-frame;
  - cost analysis of implementing such a CyberInfrastructure;
  - describe the organizational structure required to run the future CyberInfrastructure.
- What expertise and technologies (i.e. GRID, software engineering) apart from the ones available in your field are needed to build the CyberInfrastructure?
- How is the CyberInfrastructure related to current e-science initiatives in your field?
- Should hardware manufacturers be involved in projecting and developing the CyberInfrastructure? Should we try to influence their plan in developing new computing systems?
- What is the role of the research community in developing, deploying and maintaining the CyberInfrastructure?

Appendix 3. Astrophysics

Participants

R. Klessen (Heidelberg University, DE) - chair , G. Mellema (Stockholm Observatory,SE), G. Hensler (University of Wien, AU), G. Lodato (University of Leicester, UK), G. Yepes (Universidad Autónoma de Madrid, ES), R. Teyssier (Commissariat à l'Énergie Atomique, FR), L. Meyer (University of California, USA), M. Bruggen (Jacobs university, DE), R. Spurzem (Heidelberg University, DE), M. Steinmetz (Astrophysical Institute of Potsdam, DE), H. Wozniak (Centre de Recherche Astronomique de Lyon, FR), G. Murante (INAF - Osservatorio Astronomico di Torino, IT), S. Meloni (CASPUR, IT)

Presentation of the Astrophysics community

Astrophysics has always been a major driver of technology development ranging from optical sciences to the design of novel numerical algorithms. This will continue in the future. New generations of astrophysical instruments and satellites will deliver a wealth of observational data in the next decade (VLT[24], GAIA[25], ALMA[26], LOFAR[27]). Theory has to be prepared for that. Addressing astrophysical problems requires the capability of the fastest supercomputers available at any point in time (we indeed have a capability problem). Considering the current trends in hardware development, this requires algorithms that are capable of running on tens of thousands of processing units. The development of new algorithms and the incorporation of new physical processes in our numerical schemes is a major challenge and demands a new interdisciplinary approach. It requires a more modular approach to programming and the close collaboration between different fields of science. Progress in theoretical star-formation studies for example needs input from hydrodynamicists, chemists and experts in radiation transfer.

In what follows, we will present three major issues in astronomy and astrophysics that are both huge challenges and significant of many others we will be dealing with. We point out the necessarily close interrelation between the theoretical approach and the observations that will become possible with new instruments and technology in the near future. On large scales, the Planck satellite[28] and the

[24] Very Large Telescope. http:// www.eso.org/projects/vlt
[25] Global Astrometric Interferometer for Astrophysics. http://www.rssd.esa.int/index.php?project=GAIA&page=index
[26] Atacama Large Millimeter/submillimeter Array. http://www.eso.org/public/astronomy/projects/alma.html
[27] LOFAR is the first telescope using an array of simple omni-directional antennas instead of mechanical signal processing with a dish antenna. http://www.lofar.org
[28] Planck satellite is designed to image the anisotropies of the Cosmic Background Radiation Field over the whole sky, with unprecedented sensitivity and angular resolution. http://www.rssd.esa.int/index.php?project=planck

LOFAR interferometer will give us unprecedented insight into the early phases of cosmic evolution. Interpreting the expected wealth of new data requires new computer simulations of cosmic structure formation treating dark matter, gas physics and radiation consistently with high resolution. On the scale of our Milky Way, the GAIA satellite mission will yield information about 1 billion stars of our Galaxy. We need to engage in modeling the formation and evolution of the Milky Way *ab initio* with very high precision, all the relevant physical processes being taken into account consistently. The goal is to build a one-to-one model of our Galaxy on the computer. On small scales, understanding the formation of stars and the origin of planetary systems such as our Solar System constitutes an enormous challenge. New telescopes such as ALMA or Herschel will provide unprecedented insight into the cradles of new-born stars and planets. On the theory side In terms of theory, we need to combine magnetohydrodynamics with turbulence theory, chemistry, radiation transfer and possibly. nuclear fusion processes. What is common to all three grand challenges is the huge dynamic range in time scales and spatial scales in three dimensions, and the large number of physical processes that need to be taken into account.

Current state of Cyberinfrastructure serving Astrophysical Sciences in Europe

Within the past decade, computational astrophysics has undergone a transitional phase - from a patchwork of individual research groups writing and maintaining their own "private" software tools, to a few community-wide "workhorses" that are being used by a growing number of users. The development and maintenance of these code packages has outgrown the capabilities of individual researchers or research groups. These tasks require the development of a new infrastructure to ensure that the growing demands are being met.

We all agree there is a need for a (high-performance) sustained software initiative, focusing on the human aspects of "CyberInfrastructure". This requires new funding schemes on all levels, European as well as national.

The needs for CyberInfrastructure in the future

We need an infrastructure that fosters the development, documentation, maintenance and validation of open source software packages for the various communities. This infrastructure has to be flexible to swiftly adapt to the diverse and changing needs of the community. It must ensure that existing software is well maintained, documented and freely accessible, but at the same time it must make sure that the search for new and better algorithms and new numerical approaches is supported and encouraged (allowing for "old codes to fade away, and new codes to come up"). We explicitly point out, that this is not only confined to simulation software, but also includes tools for data mining or data reduction pipelines for astrophysical instruments.

Path for realization, deployment and operation of the future CyberInfrastructure

We envision an infrastructure with a small central organizational body on the top devoted to the following tasks: (1) to oversee and coordinate new and existing software developments; (2) to define and ensure common as well as community-specific software standards; (3) to deploy the software products to the community (i.e. provide a single access point); and (4) to actively promote training and dissemination OU distribution? of the software products (including technical training). This will be complemented by a large network of (research) teams that develop, maintain, document, and validate individual software packages. This is the main aspect of the decentralized structure as we foresee it.

We propose a two-stage funding scheme which both supports early code development as well as maintenance, validation, and dissemination of mature code packages. In the first stage individual researchers or research groups can present an application for the development of a new code to the central organizational body ; this application is then reviewed by a board of experts who assess the feasibility of the project. If funding is granted (say for an initial period of five years) the central organizational body oversees that the project milestones and deliverables are being reached. After delivery of the final software product (which is required to be highly modular, well documented, and to meet the standards for sustainable software) the central organization provides the community with the central access point for the new product. In the second stage, funding is granted for maintaining, expanding and disseminating the software product. We see two funding schemes. First, research groups responsible for a software package can apply for funding for people who provide user support and training. We think it is important to keep software support and software development as closely connected as possible. Support should thus come from the groups responsible for a software package. This community service needs funding. Second, researchers who engage in code development within the framework of an existing code package, who provide new modules for instance, need to be rewarded for that service as well. Postdoctoral fellows and PhD students may spend years of their time on code development and often have little time left to publish research papers. As astronomy and astrophysics is highly research oriented, they often have difficulties finding a position later on, and consequently it is hard to motivate young people in our field to engage in software development. We therefore propose that researchers who contribute software modules to the central code repository be rewarded with a personal grant (for a period of one or two years) to enable them to go to any research institute they want within Europe and do the research they want to do. This funding scheme should be open to anybody. (A similar set-up has been adopted by ESO[29], granting their fellows funding for one year at an institute of the fellow's choice after three years of support service in Chile.)

[29] European Southern Observatory. http://www.eso.org

We expect a high degree of interdisciplinary collaboration in the form of working groups or special interest groups (for example, on special numerical schemes, such as particle-based methods or adaptive-mesh refinement schemes, etc.). There is also need of sufficient funding for workshops, mutual visits and other networking activities.

The envisioned new "CyberInfrastructure" needs of improved high-speed data links within Europe (especially to the new member states), a more homogeneous user interface to computing centres, equal access to computing resources for all researchers within Europe and the availability of super-computing facilities at the European level.

Appendix 4. Computational Material Science and Nanotechnology

Participants

Berend Smit (CECAM, FR) - Chair, Emilio Artacho (University of Cambridge, UK), Stefano Baroni (SISSA, IT), Peter Blaha (Technical Universityof Wien, AU), Peter Bolhuis (University of Amsterdam, NL), Salvatore Cannistraro (Universita "La Tuscia", IT), Giovanni Ciccotti (Università "La Sapienza", IT), Burkhard Duenweg (Max-Planck-Institut for Polymer Research, DE), Mauro Ferrario (Università di Modena e Reggio Emilia, IT), Claudia Filippi (Universiteit Leiden, NL), Xavier Gonze (Université Catholique de Louvain, BE), Volker Heine (University of Cambridge, UK), Juerg Hutter (University of Zurich, CH), Georg Kresse (University of Wien, AU), Erik Lindhal (Stockholm University, SE), Michel Mareschal (Université Libre de Bruxelles, BE), Peter Nielaba (University of Konstanz, DE), Risto Nieminen (Helsinki University of Technology, FI), Matthias Scheffler (Fritz Haber Institute , DE), Christof Schuette (Freie Univeriteat, DE), Ari Seitsonen (CNRS & Université Pierre et Marie Curie, FR), Ivano Tavernelli (Ecole Polytechnique Fédérale de Lausanne, CH), Walter Temmerman (Daresbury Laboratory, UK), Peter Dederichs (Forschungszentrum Jülich, DE), Rodolphe Vuilleumier (Université Pierre et Marie Curie, FR), Simone Meloni (CASPUR, IT).

Presentation of the Computational Material Science and Nanotechnology community

Europe is probably the leader in the world at the moment for atomistic and molecular simulations. These simulations cover a large range of topics: from crystals and semiconductors to liquids and chemical reactions to very large biological systems. One should also include in this list nanoscale problems and colloidal particle assemblies. The community thus has an interdisciplinary aspect and collaborations exist with many other communities.

Atomistic and molecular simulations, aiming at modeling the dynamics of a system of ions and electrons can be used for more than one purpose: calculation of energetics, structure and dynamics but also experimentally accessible quantities such as infrared or electronic spectra, NMR[30], EPR[31], x-ray absorption. Simulation has thus become an essential tool for the interpretation of many experiments. Over the last decade or so simulations have been increasingly used by experimental groups who have hired people for this purpose.

Atomistic and molecular simulations are also a fundamental tool for Materials Science, that is the study and design of materials for technological applications. Atomistic and molecular simulations

[30] Nuclear Magnetic Resonance
[31] Electron Paramagnetic Resonance

have been used so far primarily to understand experimental results, but their predictive power is being increasingly used to design new materials with selected properties. An example of the predictive power of atomistic simulations, and their use in practical applications, is the prediction and, successively, the experimental discovery of an ultra-hard phase of TiO_2 (L. S. Dubrovinsky et al, *Nature* **410** 653 (2001)). The predictive power of atomistic and molecular simulation is now recognized also by high-tech industries. For example, Fujitsu is using *ab-initio* molecular dynamics for the development of new dielectric materials for their next generation of CMOS[32] devices (C. Kaneta et al, *Fujitsu Scientific & Technological Journal – Advanced Silicon Device Technologies*, **39** 106 (2003)).

The next big step in Materials Science will be multi-scale modeling of materials. The advantage of this approach is the possibility of combining the predictive power of *ab-initio* simulations with the ability of "coarser grained" approaches, like classical molecular dynamics, to simulate larger samples of materials. In fact, many phenomena occur at length scales that cannot be accessed in *ab-initio* simulations. Crack propagation in brittle materials is an example of such a phenomenon. In this field, the aim is the design of new materials able to resist high external stress before the formation and propagation of cracks. While it is necessary to describe the propagation of the crack tip, which is due to the breaking of inter-atomic bonds, at an *ab-initio* level, the rest of the sample, which plays the important role of transmitting the stress, can be described using classical molecular dynamics and methods for simulating continuum media. This approach, which combines different models for describing a sample at different length scales, is referred to as multi-scale modeling. A similar approach will be used in the future for simulating entire devices, starting from the microscopic level up to the macro scale.

In short, there are two main challenges that Computational Material Science and Nanotechnology are now facing. One is to obtain energies with chemical accuracy (of the order of 1 kcal.mole^{-1}). The use of correlated methods to describe the electronic structure is then crucial and Quantum Monte Carlo (QMC) will probably lead the way. The second challenge is to be able to bridge the length and time scales so as to simulate the most chemically relevant situations: while the microsecond timescale is already short for many chemical reactions or biological processes, such as protein folding and enzymatic reactions, it is still enormous at the atomic and even molecular scale. Addressing this challenge will require the simultaneous use of several approaches, and a higher interoperability of today's solutions for atomistic simulations (*ab-initio*, empirical, molecular dynamics, continuum theory...).

Current state of Cyberinfrastructure serving the Atomic and Molecular Simulations community in Europe

Software for atomic and molecular simulations can be classified into two main categories:

[32] Complementary Metal Oxide Semiconductors

homemade software and community packages. Most of these codes are written in Fortran, but this conceals a variety of coding styles ranging from the very poor (in most cases), making these codes barely reusable, highly complex and hard to build upon, to a much more professional quality (from a software engineering point of view).

It was felt that, to some extent, help to porting codes to new platforms, and making them parallelisable etc. can be obtained from High Performance Computer centres (HPC centres), but to date, codes are being maintained by their own developers in research groups. On many occasions "homemade" codes (sometimes with deep physical or mathematical insight) have been successfully ported to new local or national machines, but in an unsystematic way so that, as the complexity of the code increases, a crisis emerges/is triggered, and the code is abandoned. When this happens, a significant amount of work, insight and understanding is lost.

Another problem, common to both "homemade" and community programs, is duplication of sometimes highly complex modules.

Community packages, with thousands of users, demand a great amount of non-scientific work for maintenance, user support, web site management etc. It was evaluated during the round table that the ten main community codes in Europe each require about five full time equivalent posts (FTEs) for maintenance at the current level of service. Most of the time, this work is performed by research scientists (as opposed to software engineers), and represents a very significant and sometimes damaging diversion from their research duties. Ultimately, many "home codes" never become community packages, due to inadequate user support, and lack of code management. Part of the problem here is due to the fact that the physicists and chemists lack a proper training in programming techniques.

The issue about user support can lead to a highly limited distribution of codes, where only personal contacts of the developers are granted access. This is typically the case for Quantum Monte Carlo packages. A wider distribution of codes such as this one would require the provision of professional user support.

A different, but important issue is that code developers increasingly have to write packages which implement very different paradigms, such as electronic structure, molecular dynamics, and continuum mechanics. This requires a specialized knowledge in widely different fields, and can only be achieved through networking activities of, for example, the Psi-k[33] and SimBioMa[34] communities, through workshops and advanced tutorials.

[33] Psi-k "Ab initio (from electronic structure) calculation of complex processes in materials" http://www.psi-k.org

[34] SimBioMa "Molecular Simulations in Biosystems and Material Science" http://www.esf.org/activities/research-networking-programmes/physical-and-engineering-sciences-pesc/current-research-networking-programmes/molecular-simulations-in-biosystems-and-material-science-simbioma.html

The requirements of CyberInfrastructure in the future

Maintenance, porting (including porting to Grid structures), optimization and user support should be externalized outside the research groups. As software and its maintenance etc., at this level is a service, it should be funded accordingly via infrastructural funding.

From a technical perspective, the main requirement is a better software development paradigm. Open source codes should be based on well established common modules. Indeed the availability of common modules allows more specialized and diverse codes focusing on often different physical issues to be developed. This fact would also facilitate the development and consolidation of "homemade"/minority codes, from which most of the breakthroughs arise. While it is obvious that common standards should be adopted, the effort that this entails should not be underestimated. This would be a major improvement, as in current packages data flow is often obscure: retrieving the information needed to implement a new feature can be a very difficult task. Using modules with automatic management of dependencies should allow for much simpler data-management. The importance of data standardization for visualization and data management was also underlined in our discussions.

For all of this, and to ensure that the community at large has access to simulation packages, training is absolutely crucial. In particular, training is needed both for users, and students and post-docs involved in code writing. Three levels of training have been identified:

- HPC orientated, such as parallelization, optimization;
- software engineering, to ensure quality and re-usability;
- user training, a mix of science and specific code usage training.

Finally, in view of the growing importance of computation in many areas, it is desirable that computer simulation be a more important part of graduate studies.

Training should be provided not only for simulation-related software development, but also for teaching courses (e.g. statistical mechanics) by simulation.

Path for realization, deployment and operation of the future CyberInfrastructure

We envisage a strategy for the implementation of the CyberInfrastructure consisting of short- and long-term goals:

Short-term goals

- Web-based platforms (such as sourceforge or berlios) as a common repository for scientific software. Such platforms should also provide tools for both developers and users: a concurrent version system for keeping an up-to-date version of the software in one place, for bug reporting, mailing lists, etc.
- Interoperability of codes. This can be achieved through the development of tools allowing packages to communicate with each other.
- User support, maintenance and porting provided by HPC centres.
- Financial support for network activities of the kind provided by Psi-k and SimBioMa.

Long-term goals

- Development of common modules.
- Having a farm of different computers for testing and porting purposes.
- Establishing a training structure for:
  - software engineering for reusability, quality of the code, software optimization, parallelisation;
  - end user training.

Further proposals include:

- Developping data management. This could be done by building a database of simulation results. It could help in verifying the reproducibility of a calculation and also help to avoid the loss of information and the re-calculation of the same data (by another group).
- Building a common access point where the user can find a web-based GUI form to submit a calculation on any remote platform.
- Moving towards e-training, for example by providing online, interactive tutorials.
- The realisation of this roadmap requires some organizational structure. Such a structure should have three levels:
- National agencies. These should provide staff to research groups (PhDs, post-docs, and staff) for development of new functionalities. Since it is considered as a genuine research activity,

it should be funded from research grants. The current Forward Look and the future CyberInfrastructure should help in making the national agencies aware of the needs for code development.

- HPC centres. The link between the community and the HPC centres should be reinforced. Probably, members of the research community (developers and users) should be present in the HPC centre steering committees. The HPC centres could provide user support for their users and part of the training: optimization and parallelization aimed at an efficient use of their computers.

- At the European level. A group of about 20 people should be established in a kind of European institute. Their goal could consist in building up software interoperability, provide documentation, maintenance and user support for codes. This institute should also aim at building the module libraries suggested above. Being a sort of extension of CECAM it should also provide training, workshop and network activities. At least a few people in this group should have experience in research in the field. The question was raised, but not fully answered, as to whether this institute should be centralized at one place or relocated close to the research groups. A compromise could be a central unit with satellites (at HPC centres?) close to research groups but managed by the central institute.

Appendix 5. Computational Fluid Dynamics (CFD)

Participants

Ken Badcock (University of Liverpool, UK) – Chair, Martin O. Laver Hansen (Technical University of Denmark, DK), Jean-Yves Berthou (Electricité de France, FR), Dominique Laurence (University of Manchster, UK), Jan Vos (CFS Engineering, CH), Edouard Canot (Université Rennes-1 ,FR), David Emerson (Daresbury Laboratory , UK), Chris Greenshields (OpenFOAM, UK), Simone Meloni (CASPUR, IT)

Presentation of the CFD community

This report is a distillation of the views of the attendees who represented four countries (UK, France, Denmark and Switzerland), three different sectors (energy, aerospace, environment) and three types of organization (industry, academic and research centres).

The importance of CFD in modern engineering was reviewed in a series of workshops held in late 2005 and early 2006 to develop a scientific case for European funded supercomputing centres (the initiative being called HPCEur[35] at that time and now known as PRACE[36]). This material is focused on High Performance Computing in engineering, but the case applies more generally to computational engineering

Collaboration within the community is organized through a number of fora. First, ERCOFTAC[37] is a well established organization, funded through member subscriptions, in the area of fluid mechanics, turbulence and combustion. ERCOFTAC members receive a quarterly newsletter, and ERCOFTAC organizes several workshops and summer schools every year. However, ERCOFTAC has limited resources, its influence is not felt as strongly as it might be, and it largely relies on the (unfunded) goodwill of members to advance its activities. Industry sees ERCOFTAC mainly as a club of academics, and the benefits of being or becoming an ERCOFTAC member are not clear. GARTEUR[38] runs working groups which bring organizations together to pursue their existing interests within a collaborative framework. A typical exercise in CFD would involve several partners pooling solutions and experimental data to learn from the comparisons. Each organization brings in their own funding. A similar structure is adopted by NATO's research and technology organization. ECCOMAS[39] organizes a number of conferences (with one being focused on CFD, which is organized every 2

[35] High Performance Computing for Europe
[36] Partnership for Advanced Computing in Europe. http://www.prace-project.eu
[37] European Research Community on Flow, Turbulence and Combustion. http://www.ercoftac.org
[38] Group for Aeronautical Research and Technology in Europe
[39] European Community on Computational Methods in Applied Sciences

years).

At a European level the major collaborative activities are carried out within the EU-funded framework programmes. STREP and Integrated Projects mostly bring together partners with a focus on improved capabilities, and have a limited duration. As such, they do not provide lasting collaborative structures. The same can be said of (thematic) networks. One successful network was QNET-CFD[40], which at present will be continued via ERCOFTAC.

At a national level, learned societies organize workshops and conference series. In the UK there are four consortia which access the national supercomputer and organize networking activities (turbulence, large eddy simulation, combustion and applied aerodynamics). In Germany, a national CFD programme is organized by DLR[41], which maintains and distributes common multiblock and unstructured codes to industry and academia, mainly involved in aerospace.

The workshop has identified three key themes, which will be developed in the remaining sections of this report:

- Codes (including Multi-Disciplinary Frameworks)
- Archiving of Knowledge (including Data Exchange Standards)
- Education

Current state of Cyberinfrastructure serving CFD community in Europe

Codes (including Multi-Disciplinary Frameworks)

In the research community there is a strong tradition for groups to develop their own codes and use these as the basis for research. At the workshop there were at least three codes being developed by participants which serve as examples of this. The features relevant here of each of these codes can be summarized as: i) substantial ventures with tens to hundreds of thousands of lines of code, ii) developed and used over a period of a decade or more, iii) very complex applications undertaken by teams of varying experience, both internal and external to the home group, iv) lack of detailed documentation, v) support relying on a small number of people who were the main developers of the code, vi) small number of people who understand the whole code and would be able to plan and support further developments.

It is also relevant to note that while these three codes are different in detail, the basic data structures and requirements for input/output, documentation and version control are identical. In

[40] repository of structured knowledge and advice designed to underpin quality and trust in the industrial application of CFD. http://eddie.mech.surrey.ac.uk
[41] German Aerospace centre. http://www.dlr.de/en

addition, it was agreed that each of these codes relies on the continuing interest of a small number of people.

A similar situation can be seen in the industry where many companies have regarded having an in-house CFD code as part of their competitive edge. When it is not possible to sustain an in-house effort, the natural development is to switch to commercial codes (this also applies in the research community). Interestingly, the participants from EdF[42] presented a different approach, which is characterized by the following features: the value of CFD is in the generation of data to support decision making and to back up cases made to regulators in this scenario. Emphasis is placed on the realism of the results and the confidence decision makers have in these results. This requires the company to have detailed knowledge of the basis of the results, which in turn requires access to source code. The computational framework provides a means of encapsulating the expertise of the company engineers. To quickly transfer this expertise from research to engineering, opening up the framework to people outside the company extends the user and developer base of the code, which in turn is likely to help to increase confidence in the code.

Archiving of Knowledge (including Data Exchange Standards)

Large investments and effort have been put into CFD research over the last 20 to 30 years. As an example for the Aerospace sector, the EU has funded via the framework programmes since 1989 many STREP projects related to CFD research, and this effort continues today. Similar examples can be found in other industry sectors. A major issue discussed at the workshop was the curation of knowledge and data from these efforts. A CFD-oriented project will generate one or more CAD[43] models, grids, models (good and bad), source code, experimental data, cross-comparisons and reports. The only generally available outcome is normally the reports. The lasting impact could be increased by archiving and making the other aspects available. A very poor job is done on this at present.

A recent thematic network funded by the EU, QNET-CFD (2000-2004), established quality procedures for CFD and experimental data, and created a database of test cases which included these data, together with the results of the quality review and a synthesis for each test case with Best Practice Advice. ERCOFTAC has taken over this data base, but it is today not clear how the data base will be accessible to the CFD community. A number of databases are kept under the ERCOFTAC organization, but these depend on individuals for maintenance and updating. NATO RTO[44] has published several compendiums of data which have proved invaluable to researchers.

Finally, an impediment to the sharing of knowledge in CFD is the lack of accepted formats for

[42] Electricité de France
[43] Computer Assisted Design
[44] NATO Research and Technology Organisation.http:// www.rta.nato.int

exchanging grids and solutions. The CGNS[45] standard is not established, but if this or a similar effort succeeded, then this could have an great influence on the community.

Education

CFD has developed to the point where codes and methods are complex, requiring practitioners to bring with a range of skills. People enter the field from several different directions, including engineering, physics, mathematics and computer science. Big organizations can establish teams which include skills from all these directions. For smaller research groups it can be very difficult to find people who have a suitable mix of skills. In particular, programming was thought to be a weakness.

The needs for CyberInfrastructure in the future

Codes(including Multi-Disciplinary Frameworks)

The workshop has concluded that the era of multiple codes which are sub-critical in terms of expertise for support and development is coming to an end. The complexity of the codes and the difficulty of finding people with the right skills to pursue development largely contribute to this. Perhaps more fundamentally, as simulation is used to drive increasingly complex and critical decisions, the emphasis on the code quality (in all senses) increases. In addition, the complexity of the code will inevitably increase as simulations become more multi-disciplinary.

There are two competing frameworks that could meet such needs in the next decade. First, commercial codes provide an option that removes the need for keeping large teams in place for development and research. There was concern expressed from an industry perspective since they would like to have access to the source, to allow knowledge from the company to be maintained in the framework, and also to be able to continually assess the basis for the CFD results. The concern from the research community was focused on the influence on advances in the field. With limited opportunities to develop the framework, the question was raised of how easily new people could be educated and innovations formulated and tested. Commercial grid generation packages gained predominance a decade ago, and it could be argued that this field has not, in the research community, received the volume of attention that it still merits.

The alternative is given by open source. In this framework a number of different groups focus their efforts on a shared code. The crucial strength of this framework is that a large user and developer base provides increased confidence in the basic code, a critical mass for documentation and the potential to harvest the innovation from a large pool of people, and to make this widely available.

[45] CFD General Notation System. http://cgns.sourceforge.net

There are problems with this framework also. The OpenCFD[46] experience shows that a significant central organization is required to obtain such benefits.

Archiving of Knowledge (including Data Exchange Standards)

The establishment of a database of CFD knowledge covering all the aspects listed above is crucial. This requires a long-term effort which will gain the support of the community. Publishers may fill part of this requirement by enabling the publishing via the internet of supporting data for papers.

The basis for a community database exists. Efforts such as QNET-CFD and within ERCOFTAC could provide the starting point. In addition, projects funded under the Framework programme all feature databases. What is required is an effort which is funded for the long term and which has a suitably large scale to establish credibility.

Education

The provision of skilled people is crucial for the future of research in CFD. There are two aspects to this. First, in large organizations CFD will make a contribution within a large multi-disciplinary team, placing the emphasis on specialization, communication and team working. Secondly, in smaller organizations researchers will have to acquire key skills, which will not have been present in their undergraduate degrees. These skills could be in programming or in disciplines outside fluid mechanics. It was not felt that suitable modules and programmes were available to meet the latter demand. At present these aspects can be covered in an ad-hoc manner within PhD programmes.

One possible model which might provide a pointer for the future is the High End Computing studentship scheme run by the EPSRC[47] in the UK. Quoting from their web site:

In the "Strategic Framework for High End Computing", published by the High End Computing Strategy Committee in June 2003, the need was identified to increase investment in training for computational science and engineering. It recommended that 15 to 20 four-year high end computing (HEC) studentships should be awarded each year to sustain a healthy skill-base. Each studentship would include at least 25% of the four-year period committed to training modules at one of the UK's leading computation centres; the balance of time would be spent working on a research project at their host institution. To make the studentship attractive to young people, two qualifications would be offered: a Doctorate and a Master's degree in computational science and engineering. In response to this recommendation, EPSRC issued a call in 2004 for centres of

[46] OpenCFD produce the OpenFOAM® open source CFD toolbox and documentation and distribute it through this web site and Sourceforge. http://www.opencfd.co.uk
[47] Engineering and Physical Sciences Research Council. http://www.epsrc.ac.uk

expertise to provide Masters level training for students in computational science and engineering."

In this model a specialist masters degree is taken in parallel with a traditional PhD, each of which benefit from the other.

Another option is to establish a network of high quality specialist modules which are available to PhD students of institutions participating in the network. The taught aspect of the PhD could then be tailored to meet the particular needs of the student (which will depend on his/her background and the plans for his/her project and career).

One point that was emphasized in the discussions was that the need for extra skills should not detract from the depth and specialization? of the PhD itself. Any schemes should be designed to produce specialists with other crucial skills, rather than generalists.

Finally, it was noted that if taught programmes and modules made use of large open source CFD codes then this would have a significant organizing effect on the community for the future, since the students would be likely to adopt these frameworks as they progressed through their career.

Path for realization, deployment and operation of the future CyberInfrastructure

The following conclusions and suggestions are made to realize a cyber-infrastructure suitable for the CFD area.

Codes (including Multi-Disciplinary Frameworks)

It is important that a number of open source codes (noting that it is not possible for one code to cover all flow regimes) with a critical mass emerge to form the basis for future CFD research. Commercial CFD codes are likely to be popular, but do not cover all aspects required for a healthy cyber-infrastructure.

The key issue for open-source codes is to have an organizational structure that allows the experience and developments of the user base to feed back into releases of the code. This implies a centralized support structure which can scale with the size of the user base. The only long term approach to funding this is through subscriptions from users and organizations. When compared with the cost of maintaining a large number of codes, the level of the subscription is likely to represent good value.

The organization of the community around a small number of open source codes is likely to take a long time. However, as people responsible for the current generation codes move to retirement, it will probably be very difficult for the next generation to maintain existing codes or establish new ones. In addition, with CFD sitting in a complex framework for multi-disciplinary analysis, and the requirement for increasing certainty on the outcomes of simulation, there are factors which will

push the infra-structure in this direction.

The development of multi-disciplinary and multi-scale simulation environments will undoubtedly make the emergence of standards necessary. Examples are standards for exchanging grids and solutions between codes, and between codes and pre/post tools, and standardized API for codes interoperability. These efforts cannot be handled by one discipline but must be shared by all. These standards must be Open source in order to be widely accepted/recognized.

Progress towards a small number of open source codes is likely to be driven by industry, with codes being released as open source. However, the following developments would encourage progress.

- Funding agencies (including industry) should establish clarity in the expected exploitation (or at least availability) of the research they fund. This would tend to encourage the adoption of larger community-effort codes.

- Universities should base their PhD and Masters modules on open source codes, to establish a positive attitude to these codes within the next generation of engineers.

Archiving of Knowledge (including Data Exchange Standards)

As a matter of urgency an initiative is needed to establish a database of knowledge as described above.

An organization such as ERCOFTAC could be in a position to take the lead on this. Initial public funding could facilitate the setting-up of a database (but a clear plan for longer-term funding would be needed from the outset).

Funding agencies for research projects should ask for a detailed plan at the proposal stage as to how knowledge will be archived at the end of the funding period.

The archive needs to be seen as a long-term effort, which requires long-term funding, likely only to be available through subscriptions.

It was the strong feeling of the workshop that (well-funded) initiatives in e-Science and the grid had focused on facilitating computation. This was felt to be completely useless in CFD, and instead, that emphasis on remote access to data may have some impact.

Education

There is a need to rethink how researchers in the future will acquire the mix of skills needed. There is a chronic shortage of suitable people.

An initiative at European level under the Marie Curie banner is appropriate. This could have the following components.

- Development of a network of modules in areas such as programming and disciplines outside fluids that students could take as the taught component of PhD projects.

- Development of a range of innovative PhD+Masters courses that retain the depth of the current PhD, but which allow the student to gain expertise in an appropriate field (such as high performance computing, computing science, adjacent disciplines).

- Adopting open source codes for undergraduate and postgraduate courses could provide a good framework for meeting several education objectives, and have a good impact on shaping the community in the future.

Appendix 6. Climatology and Meteorology

Participants

Marie-Christine Sawley (CSCS, CH) – Chair, Ulrich Schaettler (German Meteorological Service - DWD, DE), Sylvie.joussaume (Commissariat à l'Énergie Atomique - CEA, FR), James Penman (Met Office, UK), Michael Lehning (Institute for Snow and Avalanche Research SLF, CH), Reinhard Budich (Centre for Marine and Atmospheric Sciences - ZMAW, DE), Joachim Biercamp (Climate research Centre – DKRZ, DE), Christoph Raible (University of Bern, CH), Simone Meloni (CASPUR, IT)

Current state of Cyberinfrastructure serving the environmental sciences and climatology community in Europe

Code development

In the environmental sciences and climatology community, there are strong traditions in code development. The community, including service organizations, research institutes and universities, develops community codes such as Alpine3D[48], CCSM[49], ECHAM[50], COSMO-CLM[51] and NEMO[52] (and their derivatives). The development activities are mainly carried out by the community itself.

The codes have long life-cycles, and they are applied in operational (e.g., weather forecasts, natural hazard warnings) and research (e.g., climate change) activities where high quality requirements and standards for code development have to be met. In order to share the development, evaluation, maintenance and support of standards and state-of-the-art codes and software tools, fora like COSMOS[53], ENES[54] and PRISM[55] have been created by the community.

[48] Hydro-meteorological model with a detailed description of surface processes, especially the snowpack development in open and forested (sub)alpine areas. http://www.wsl.ch/staff/manfred.staehli/projects/alpine3d.ehtml

[49] Community Climate System Model. http://www.ccsm.ucar.edu

[50] ECHAM is a comprehensive general circulation model of the atmosphere. http://www.mpimet.mpg.de/en/wissenschaft/modelle/echam.html

[51] Lokal Modell. http://clm.gkss.de/index.php?menuid=47&reporeid=63

[52] Nucleus for European Modelling of the Ocean is a state-of-the-art modeling framework for oceanographic research, operational oceanography seasonal forecast and climate studies. http://www.nemo-ocean.eu

[53] COmmunity earth System MOdelS is major enterprise conducted at the international level, and specifically in Europe (Max Planck Society in Germany, Hadley Centre in the UK, Institut Pierre Simon Laplace in France), in the US (NSF/NCAR, NASA, DOE, NOAA) and in Japan (Frontier Program), for the development of complex Earth System Models. http://cosmos.enes.org

[54] European Network for Earth System modelling. http://www.enes.org

[55] Partnership for Research Infrastructures in earth System Modelling. PRISM provides the Earth System

Highly complex phenomena in environmental sciences and climatology require complex and coupled models where couplings between atmospheric, ice, ocean, chemical and biochemical models, and local and global models are made. In the code development, this leads to the coupling of sub-models and the usage of couplers (e.g. OASIS[56]) making the shared software infrastructure feasible.

Weather forecasts and local models are high-resolution models (smaller element/cell sizes in computational meshes) thus requiring more memory capacity. Climate change models do not need (at the moment) high resolutions. However, climate change models do require longer runs for simulating longer time. This means that, while weather forecasts can benefit from massively parallel computers (adopting a domain decomposition parallel scheme) climate change simulations are still run on vector computers with a (relatively) small number of CPUs. Ensemble simulations could also benefit from massively parallel computers, distributing "realizations" across nodes.

Simulations produce a large amount of data (from terabytes up to petabytes). The data, maintained and archived by the institutes within the community and national high-performance computing centres, is post-processed through data analysis and visualization tools. The data is typically presented in metadata formats. Across organizations and sub-fields in environmental sciences and climatology, there is a lack of data integration (data originating from simulations and experiments). In fact, the lack of a standard for describing archived data, as well as models used for producing data, make it difficult for scientists to access the data available outside their own community.

Training and support of researchers

Training and support of researchers is organized by the institutes within the community. In addition, the European network activities contribute to information dissemination and transfer on focal areas like code coupling and I/O, integration and modeling environments, data management and high-performance computing.

In training and support of researchers, the following challenges are encountered:

- Due to the inter-disciplinary nature of the field, it is essential to find expertise from different fields (environmental sciences and climatology, engineering, computer science), and to combine them.

- Since professional (scientific) programming expertise is required, it is challenging to find new researchers having a proper background in computational science and programming (lack of courses in curricula supporting computational environmental sciences and climatology).

Modelling community with a forum to promote shared software infrastructure tools.
http://www.prism.enes.org

[56] http://www.prism.enes.org/PAEs/coupling_IO/software_OASIS4.php

Interaction with computer scientists, hardware vendors, policy makers and other experts outside the community

The community is interacting with high-performance computing centres and hardware vendors mainly in the testing, optimizing, porting and benchmarking of codes. With hardware vendors, the RAPS[57] forum has been created to share knowledge and information. The partners of the RAPS Consortium develop portable parallel versions of their production codes which are made available to a working group of hardware vendors for benchmarking and testing.

The needs for CyberInfrastructure in the future

The needs for Cyberinfrastructure in the future were expressed by the workshop members as follows:

- Environmental sciences and climatology need available (sustainable) capacity cycles. The community needs to be closely linked to the future European high-performance computing initiatives (e.g. PRACE[58]) and activities.

- The competence development framework has to be further reinforced. This includes the balance of competencies (connections and interactions between environmental scientists, applied mathematicians, computer scientists and software engineers), training and increased networking activities (due to the increased complexity of models).

Technical development:

- Efficient use of high-performance computing resources: optimization of actual codes, new codes for the future high-performance computing, optimization of work flows. The last issue refers to the process of bringing a lab-level code to a level of maturity adequate for operational activities.

- Accessibility to a large amount of data originating from observations and simulations. Creation of a collaborative European data centre.

- Standard interfaces and data formats (metadata).

- Code development

Within the community codes, the diversified trend will continue, i.e., there is no convergence

[57] Real Applications on Parallel Systems is a European Software Initiative. It comprises the RAPS Consortium (founded in the early 1990's) and a Working group of Hardware Vendors.
http://www.cnrm.meteo.fr/aladin/meetings/RAPS.html
[58] Partnership for Advanced Computing in Europe. http://www.prace-project.eu

towards one single ultimate code. Instead, the variety of codes will form platforms benefiting from continuous code development, code integration and couplings for operational and research activities. The quality of the existing palette of community codes is ensured by high requirements in operational tasks (e.g. weather forecasts), and community reviews. For instance, in the meetings of the Intergovernmental Panel for Climate Change[59] (IPCC, meeting every six years), code developers are expected to deliver model outputs for model cases for comparison.

Future applications in environmental sciences and climatology will be even more data and computationally intensive than today. Efficient data management remains an issue. This includes challenges (and still unsolvable problems) in federated databases and parallel I/O. In data integration (and tools, code development and training), ECMWF[60] could be consulted. Also, the Data Grid achievements and benefits in EGEE[61] and EGEE-II projects (EGEE data catalogue) could be further investigated. In data visualization, there is room for improvements since no common tools and standards exist.

The trend towards larger-scale computing (capability computing) is evident because of the increasing complexity of the models, longer time-scales, higher resolution simulations and a larger number of ensemble simulations.

The usage of Grid environments (distributed environments) for data intensive and, with minor emphasis, for computationally intensive applications will be investigated. This would facilitate optimal usage of European high-performance computing resources and data accessibility using international service standards. As a communication intensive field JE NE COMPRENDS PAS : As it is a field where communication is intense?, efficient research networks (GÉANT[62] and national research networks) are required.

Training and support of researchers

Due to high scientific and technological requirements and standards, highly qualified and professional experts and researchers in the field are required. To support high professionalism in computations, it is expected that curricula in environmental sciences and climatology contain more courses in mathematical modeling and software development. In supporting this proposal, a survey on “Identification of mathematical and computer science competencies in environmental sciences

[59] http://www.ipcc.ch

[60] European Centre for Medium-Range Weather Forecasts is an international organisation supported by 31 States. It provides operational medium- and extended-range forecasts and a state-of-the-art super-computing facility for scientific research. http://www.ecmwf.int

[61] Enabling Grid for E-science. http://public.eu-egee.org

[62] The GÉANT project was a collaboration between 26 National Research and Education Networks representing 30 countries across Europe, the European Commission, and DANTE. Its principal purpose was to develop the GÉANT network - a multi-gigabit pan-European data communications network, reserved specifically for research and education use. http://www.geant.net

and climatology" among the above organizations is proposed.

Interaction with computer scientists, hardware vendors, policy makers and other experts outside the community

The interaction with other communities follows two different paths:

- It is expected that the interaction with high-performance computing centres, hardware vendors and experts outside the community will increase. High-performance computing centres act as support organizations providing expertise in computer science related issues like code optimization, tuning, porting and parallelization. RAPS activities with hardware vendors are expected to continue.

- Missions of service organizations (Met offices) within the community will be (have been) slightly modified. Instead of providing just weather forecasts, it is expected that these organizations will yield further outputs for the general public, such as information on the effects of weather and climate on life, human and economic activities. This creates a basis for interdisciplinary dialogue with biologists and economists, among others.

Path for realization, deployment and operation of the future CyberInfrastructure

Coarse estimates, as concluded in the workshop, indicate that there are thousands of end-users and tens of institutes developing and making use of the community codes. The main efforts are directed at research and development activities, and there is more room (nor need) for service activities. However, already at the present stage, community codes are developed, maintained and supported by teams of scientists and software engineers (~10-20 people). The community is starting to recognize this kind of work as a worthwhile activity.

On organizational issues, close links to the future European high-performance computing activities have to be established. Also, the needs and paths for Cyberinfrastructure listed in this document should be considered on national levels (synergy and interaction between national and European levels).

Appendix 7. Life Sciences

Participants

Jonathan Essex (Southampton University, UK) - Chair, Rebecca Wade (EML Research, DE), Johan Aqvist (Uppsala University, SE), Richard Lavery (Institut de Biologie et Chimie des Protéines , FR), Maria Ramos (University of Porto , PT), Richard Enchman (University of Manchester , UK), Chris Reynolds (University of Essex, UK), Simone Meloni (CASPUR, IT)

Presentation of the Life Sciences community

Computation is an essential component for modeling biological structure (protein, DNA, membranes) and underpins a huge amount of experiment (e.g. structure refinement in X-ray crystallography, nuclear magnetic resonance/structural biology). The resultant structure and dynamics are essential for the understanding of function and therefore has a strong interaction with genome initiatives. Computation is also essential for bioinformatics initiatives - the assembly, analysis and interpretation of sequence information requires an extensive use of computers.

The understanding of biological systems is an essential element of modern science. Computation is directly involved in much of this effort, with the areas it impinges ranging from the molecular to large-scale biological organisations. Examples include:

- Drug discovery
- Enzyme chemistry
- Synthetic pathways
- Sequencing, functional analysis and annotation
- Prediction of biomolecular structure (protein, DNA and membrane)
- Dynamics and thermodynamics
- Molecular recognition and assembly
- Mutagenesis

Computational results are now used as input for designing ad-hoc experiments, and as the final step of experimental protocols for the interpretation of experimental results.

At a more fundamental level, the computational approach is changing the way biological functions are viewed through molecular level insight.

Notable successes of Computational Life Sciences include genome sequencing, the biophysics of molecular systems, with, for example, understanding protein folding, and the involvement of computation in all stages of drug development.

The Computational Life Sciences community consists in a large number of closely related interacting multidisciplinary/interdisciplinary groups (chemistry, mathematics, physics, biology, pharmacy), with very diverse interests ranging from sequencing through to quantum chemical biology. In the field of molecular modeling, on average, every large European country counts about 20 Principal Investigators in academia and research institutions, and about 10 researchers per large industrial site in the field of pharmaceuticals, biotechnology, food industry and environment. In the field of bioinformatics, numbers are about five times larger. Cheminformatics is very important in industry, particularly the pharmaceutical sector. Many experimental groups working on biological issues use computational methods as part of their experiments. In academia, the maintenance and development of the CyberInfrastructure is often ad hoc, with work being paid from other research funds and performed during workers' spare time.

The increasing importance of computational methods in Life Sciences is easily predictable, both in academic and industrial endeavours. Massive DNA sequencing efforts, structural genomics projects, systems biology, high-throughput ligand design and structure prediction will all require computational underpinning in terms of calculations and data handling. The key challenges for computer simulations in the next 10-15 years will be the simulation of longer time scales to study the dynamics of slow events, and multi-scale modeling for studying large, complex systems such as the complete cell. Modern software with user-friendly interfaces is required not only to reach these goals, but to bring these tools to experimentalists.

Current state of Cyberinfrastructure serving The Life Sciences community in Europe

Much of the simulation software currently used by the Life Sciences Community is rather archaic, difficult to port to new architectures, difficult to use, poorly documented and often not supported. Moreover, this software is mainly written in old-style Fortran, which introduces extra difficulties due to the lack of programming expertise in this language. The situation for bioinformatics is better; there is greater penetration of industrial and experimental labs, and the software relies on widely used tools, such as databases. Indeed, the algorithms, computational techniques and programming languages adopted in bioinformatics are largely used in other fields, and this makes it possible to adopt their best practice. The open source paradigm is also well established in the bioinformatics community.

In the field of biomolecular computer simulation, groups often devise their workflows relying on old-style legacy codes, complemented by the software produced by each group. This is partly due to the large diversity within the community and partly to the lack of oraganization of the community at international level. The result of this situation is that there is no one code that may clearly be taken forward as a flagship code. Competition between different codes is valuable, but it should be carried out at a high level, rather than duplicating well established algorithms. In any case, competing software should agree on standards for input and output files, which will enable code interoperability and the design of automatic workflows. Commercial codes are not the solution: they are expensive, their use is restricted, and source code is usually unavailable.

For research training and support, the available resource is patchy. A number of organizations (for example, the CCPs[63], CECAM[64], EMBO[65], and FEBS[66]) offer training at graduate level, and universities offer different levels of computing education, as do some of the national supercomputing services. Yet, there is a need for general courses on software design. Similarly, access to coding expertise and relevant libraries is also patchy – computer scientists and software developers have trouble understanding computational biologists, and vice versa.

The current lack of proper CyberInfrastructure is hindering our progress towards grand scientific challenges such as, for example, the simulation of a whole cell, which would require multi-scale modeling and, therefore, software interoperability. A lot of current effort is spent on porting code, dealing with compiler issues and systems administration. All is not gloom however. The CCPs and CECAM, for example, show that good community organizations exist, and many bioinformatics tools are readily available and widely used by non-experts.

Although the current situation is far from being satisfactory, the development and maintenance of the current software portfolio to a level of maturity adequate for general application currently requires about 50 FTEs (Full Time Equivalent), according to our estimate, with additional maintenance requiring about 50 FTE as well. Money for these enterprises is currently taken, in many cases, from existing research budgets.

The needs for CyberInfrastructure in the future

The main issue for the future software infrastructure is the availability of Open Source software based on toolkits. In fact, although Open Source software fitting some of the needs of the Computational Life Sciences community are already available (e.g. Gromacs[67], NAMD[68]), these codes

[63] Computational Collaborative Projects (UK).
[64] Centre Européen de Calcul Atomique et Moléculaire
[65] European Molecular Biology Organization
[66] Federation of European Biochemical Society
[67] http://www.gromacs.org
[68] http://www.ks.uiuc.edu/Research/namd

are too complex to allow researchers to add new features without the support from the developers. The research community is so diverse that the development of a single code to meet all their interests is not practicable. A software infrastructure based on libraries of self-contained modules, implementing well established algorithms, would therefore simplify the task of developing codes fitting the specific needs of our diverse community. In fact, we imagine that this infrastructure should also contain one or more simulation packages based on these toolkits; research groups which need some features not yet implemented in a package can easily extend it and eventually implement this new method in a general form which then might be included in the toolkit. Moreover, because of the duplication of the same routines in many codes, the effort required to maintain the whole set of codes serving the community, which includes porting these codes to new architectures, is many times more expensive than maintaining the single routine contained in the toolkit. The issue of maintaining and porting codes is going to be even more important in the coming years because of the appearance of new computer architectures such as acceleration boards (e.g. ClearSpeed[69]), cell computing and technical computing on GPUs (Graphical Processing Units).

There is another issue concerning toolkits - they can be the building blocks of multi-scale multi-disciplinary software. Many old-style codes, which are procedural in nature and based on an obscure data structure, are simply inadequate to incorporate very different models representing biological systems at various length scales. The situation becomes even more complicated if one considers that different models might require a completely different approach to parallelization. A modular toolkit would, in principle, allow the easier construction of a multi-scale program, with a well-defined data structure.

There is currently a proliferation of file types in computational biology. In general, each code adopts its own standard (ASCII or binary) with its own style and semantics. This makes it very difficult for scientists to interoperate among codes and even to develop analysis software which runs the same analysis on the output of different codes. Moreover, this community feels it is of primary importance to store and access previous simulations for further analysis in the future. However, the unnecessary proliferation of input and output formats makes it very difficult to devise a tool for automatic archival and retrieval of such simulations.

As to the issue of training, it is very important for many aspects of Computational Life Sciences. There is a shortcoming of professional programming expertise in our community. Specific courses for training computational scientists in programming should be devised at post-graduate level. Another topic is the training on specific research issues in conjunction with the use of software through hands-on sessions. Although there are several international institutions organizingsuch tutorials (e.g. CCPs, CECAM, EMBO, FEBS), there is no European institute coordinating this activity. There is also a need for Gordon-type[70] conferences within Europe. Our community needs training targeted to both computational scientists and experimentalists; the latter are increasingly using computational methods.

[69] http://www.clearspeed.com
[70] Gordon Research Conferences. http://www.grc.org

Finally, there are significant issues relating to accessibility of computing power. Many algorithms used in our community are not well suited for massively parallel machines, and the lack of modern software, running efficiently on large parallel machines, hinders our access to international and national SuperComputing facilities. This community requires a balanced portfolio of computational resources, including capacity computing facilities, serving researchers interested in grand challenge calculations, as well as capability computing, serving the broader community of scientists. Without such a balanced hardware infrastructure, many in our community are forced to resort to in-house computing. Using local resources is quite inefficient in that each group must spend a significant fraction of their time on system administration. This additional capacity computing may be provided through academic institutions, or national resources dedicated to a particular application area. For those groups with their own facilities, access to central expertise and system management software would help to reduce the management overhead.

Path for realization, deployment and operation of the future CyberInfrastructure

We envisage a strategy for the implementation of the CyberInfrastructure in terms of short- and long-term goals:

Short term goals

- Consolidation and maintenance of current major software. Porting codes to new architectures is required, as well as documentation and support. We think that porting and optimization could be assigned to computing centres. It will be the responsibility of the owner of the codes to include the ported versions with their official release.

- Documentation and training for current major software. This task may be assigned to the code developers or, more generally, to the community of users. However, this activity must be funded on a specific budget through the European Software Cyberinfrastructure, rather than relying on existing research budgets, as is currently the practice.

- Standardization of input/output. This task is in fact two-fold: standardization of file format, and standardization of the semantics of the content of input/output files. The first must precede the second and will consist in choosing a file format that is portable across different computer architectures and flexible enough to accommodate the very different needs of our community and, possibly, other scientific communities with whom we collaborate (material sciences, chemistry,…). Our suggestion in this respect is some form of mark-up format i.e. the adoption of XML. The second task has to do with the content of input/output files and consists in defining names and data types for all (common) data that can be contained in these files. This second step is more challenging because it requires some taskforce for defining a list of likely data.

- Software dissemination list. Nowadays most free or licensed software is published on the web. However, there is no central repository that stores, or even lists, these codes, which makes it difficult for workers to find what is available. We suggest making a single access point where all the software in our area can be published or, at least, where links lead to the appropriate web page.

Long term goals

- Toolkit. The realization of a series of libraries or modules implementing most of the available algorithms is the main goal in the long term. A first attempt in this direction has been made by Konrad Hinsen with his "Molecular Modeling Toolkit"[71]. The realization of a set of libraries will require a strong level of coordination for the definition of the API and programming language interoperability. This is a difficult, but achievable task. This toolkit will allow the community to tackle the grand challenge of multi-scale modeling of very large biological systems.

- Training and dissemination. The realiszation of a more coordinated series of tutorials, workshops and symposia is mandatory to complement the advancement of the software infrastructure and of Computational Life Sciences in general. At the moment, such kind of training is partly offered by Gordon Research Conferences.

- The realization of this roadmap requires some organizational structure. Our primary choice would be the founding of a European BioSimulation Institute (EBS), which would be responsible for the realization of the entire roadmap. EBS should have its own team of developers for the development of toolkit and common software, documentation maintenance and support. This Institute would also be the reference institution for training and dissemination, hosting many of the tutorials and symposia. A possibility is to build EBS sharing some common facilities with the EBI[72]. An alternative to this central Institute might be a light, networked, organization. In this scheme, software development, maintenance and support as well as training would be performed at a national or regional level, with the central institute acting only for coordination.

[71] http://dirac.cnrs-orleans.fr/MMTK/
[72] European Bioinformatics Institute. http://www.ebi.ac.uk/

Appendix 8. Quantum Molecular Sciences

Participants

Franco A. Gianturco (University "La Sapienza", IT) – Chair, Peter J. Knowles (University of Cardiff, UK) – Chair, G. A. Worth (University of Birmingham , UK), Nico Sanna (CASPUR, IT), Stefano Baroni (SISSA, IT), Wim Klopper (University of Karlsruhe, GE), Massimo Mella (University of Cardiff, UK), Simone Meloni (CASPUR, IT)

Presentation of the Quantum Molecular Sciences community

Theory supported by significant computation has been, and continues to be, a central and indispensable component of research into the properties of molecular matter. The impact of computations on fields that are traditionally experimental (e.g. spectroscopy, structural bioorganic and inorganic chemistry, rational drug design, radiation damage of biomaterials, combustion chemistry, chemical vapour deposition modeling for the semiconductor industry) has grown with the advances in both methodology and accessible computer power. Simulations in areas such as ultracold chemistry allow the probing of properties that are difficult or impossible to access with experiments. In more detail, problem areas of interest for Computational Quantum Molecular Sciences are: molecular structure and chemical reactivity; energy flows in molecular systems; time-dependent phenomena in molecular systems; electron transport; positron annihilation; nanoscience; quantum molecular biology; radiation-matter interactions, including control of molecular processes. We perform our research applying the following methods: computational molecular electronic structure; computational molecular spectroscopy; quantum reactive dynamics; quantum energy transfer dynamics; quantum stochastic methods;.

The growing impact of molecular simulation will surely keep on increasing. This is set against a landscape of some forms of experimental chemistry becoming ever more expensive and difficult, which provides an increasing imperative to model before measurement; in some cases, the experiment is impossible without high-quality computed data. Thus the demand for cooperative working between computational and experimental scientists is becoming stronger and more widespread. Examples may include the design of materials that have specific magnetic and/or optical properties, the rational design of functionally selective catalysts, and the prediction of the future chemical and climatic properties of the earth's atmosphere.

Current state of Cyberinfrastructure serving Quantum Molecular Sciences community in Europe

It is difficult to provide a generalization of the whole field in terms of existing practice, since there is a wide variety of codes. Some of them are very large (1 million lines, 1000 users), whereas others are intended for the use of the sole group that writes them. There is a corresponding diversity in practice of code development, with some groups having a well-developed infrastructure that embodies many aspects of good software-engineering practice, whilst in others there is no possibility for this to take place at a perceivable level. Most of the code development is done within the context of specific requirements generated by the local scientific agenda, rather than the desire to produce a complete and comprehensive software package; this pattern is a direct consequence of the normal methods of funding scientific research in European countries. There are some individual and national initiatives that support a limited amount of code sharing in small communities, but no general substantial central support.

In general, the majority of the codes are written by postgraduate and postdoctoral researchers with primary background in chemistry or physics, but without specific training in numerical mathematics or programming. At present most of the necessary additional training is done in an ad-hoc fashion, locally. There are some nascent national initiatives focused on providing education in these missing aspects through MSc and similar programmes, but these are as yet on a small scale.

In a number of countries there exist national frameworks, often delivered through the supercomputer centres, for providing support for researchers who wish to port, parallelize or optimize their code. In some cases, this support is applicable to particular subject areas, and elsewhere it is more generic. Our codes take extensive advantage from the efforts that have been made to produce generic components such as BLAS[73], (sca)LAPACK[74], MPI[75], Global Arrays[76], automatic code generators.

In most European countries, the interaction between computer scientists and computational scientists is underdeveloped, and there is little evidence of cooperative efforts between computational scientists and those who carry out research in numerical mathematics or parallel programming. Computer vendors often engage helpfully with the developers of some of the larger academic packages, although this aspect is obviously less developed with codes that have a small user base. Engagement of research funding on the question of properly supporting and protecting long-term infrastructure development is incomplete.

In most countries, or across Europe as a whole, there are no existing discipline-specific organizations

[73] Basic Linear Algebra Subprogrammes. http://www.netlib.org/blas
[74] Linear Algebra Package. http://www.netlib.org/lapack. Scalapack: Scalable LAPACK. http://www.netlib.org/scalapack
[75] Message Passing Interface. http://www-unix.mcs.anl.gov/mpi
[76] The Global Arrays (GA) toolkit provides an efficient and portable "shared-memory" programming interface for distributed-memory computers. http://www.emsl.pnl.gov/docs/global

that coordinate or support code infrastructure; in the UK, the Computational Collaborative Projects fulfill part of the role. This can be contrasted with the rather well developed programmes to provide this support in the USA.

It is difficult for a code to progress from its beginnings as a group-private code to something with the software-engineering strength necessary for wide distribution. This in some cases results in duplication of effort, and also has the effect of hindering collaborative and cross-disciplinary research. In the worst cases, knowledge and experience is lost when key personnel move on.

Computer codes in our field become increasingly complex and voluminous. Many of the quantum chemistry codes contain a million lines or more, and aspire to a comprehensiveness of coverage of all methods, old and new. This state of affairs, where no human can manually read and verify correctness of a code, has arisen with the help of advanced programming tools, for example automatic code generators, and this trend will surely increase; there is a risk that the codes become unmaintainable and unverifiable. The community needs both tools and expert support in order to meet this challenge and continue to develop software that embodies the latest advances in theory and that is internationally competitive.

Computational chemists are amongst the heaviest users of both high-end and mid-range computing facilities, and they were at the forefront of algorithmic developments to exploit vector machines in the 1980s and parallel computers from the 1990s. It is probably fair to say that most of the large-scale electronic structure codes have not kept up in this way in recent years, and no longer scale out to hundreds of processors even though they did on the machines existing 10 years ago; some of the dynamics codes have been able to exploit developments in generic components such as parallel eigensolvers in order to solve ground-breaking problems on high-end machines, but for the most part the codes in our community are not at present well positioned to use high-end machines. A different story is taking place with the current generation of mid-range machines, where codes do in general embody the modest degree of parallelism necessary to effectively exploit commodity clusters. Recent and emerging developments in computer architecture, however, are threatening this position. The mid-range machines of the immediate future will consist of nodes with increasing numbers of processor cores without matching increases in memory bandwidth, and will additionally start to include novel architecture features such as attached processor arrays. High-end machines will contain ever-increasing processor counts, again, probably, without matching increases in bandwidths at the various levels of memory hierarchy. It is unlikely that the majority of our codes can be engineered to take full advantage of these developments without significant investment of effort.

There is an increasing dearth of researchers in chemistry who also possess knowledge, skills and aspiration in theory, numerical mathematics and programming. The absence of these topics in most undergraduate chemistry curricula means that beginning PhD students are both underequipped for, and largely unaware of, the numerical and computational technicalities of the project they are about to embark on.

Despite the problems outlined above, the electronic structure community has been successful in bringing forward big general-purpose codes, some of which have reached the commercial market.

The inability to fully coordinate program development efforts between groups has enforced a diversity of approaches, which has produced innovation in the field.

In spite of the lack of infrastructure, there is great demand from experimentalists for collaboration, and with a proper infrastructure there would be more chance of satisfying this demand.

The lack of manpower support at the level of individual sub-disciplines to enable code engineering standards to be raised, and current and emerging machine architectures to be fully exploited is the most urgent problem to tackle. Education of young researchers in computational science is the other most important issue to be dealt with.

The needs for CyberInfrastructure in the future

Code development:

- Coordination of data structures for interoperability of codes.
- Support for collaborative code development (travel, people).
- Assistance to owners of small codes to strengthen the software engineering of their code to the point where it could be shared.

Training:

- Education in those aspects of computer science essential for numerical programming.
- Support for deploying existing codes on emerging computing systems.
- Engagement of numerical analysts and computer scientists by supporting and promoting collaborative research.

Access:

- Development of generic tools and libraries supporting code development and the exploitation of existing and future computer architectures.
- Evaluation and recommendation of 3rd-party tools and libraries.

Organization:

- A visible centre at a single physical location that offers a focus for all the activities, facilities for meetings and collaboration, and hosts the staff.
- Support for workshops focused on collaborative development of codes and methodologies. This might follow the CECAM[77] model.
- A Sourceforge facility for European computational science[78], with continued development of tools to support code provenance.
- A way for potential users to find codes – some kind of web-hosted catalogue.
- A programme that is guaranteed for the long term and that is sustainable.

The major technical challenge is related to interoperability issues. In the field of data handling, there is at least one endeavour: computation of molecular spectra, where very large computed (rather than experimentally-derived) datasets are produced, and there are significant issues associated with curation and distribution.

New architectures present challenges because most of our codes will not immediately exploit their features. Quite often, this challenge is at the level of the underlying algorithm and is therefore not amenable to solution by tuning or replacement of generic components and libraries.

Scientific achievements to be expected from a European CyberInfrastructure were already presented in the introduction. Here we describe in more detail the impact of such an infrastructure on our research tools. Most of the existing mid-range numerical codes operating in our community in Europe would greatly benefit from the sort of modernization/optimization that could be provided by IT personnel at a European centre dedicated to such performance-enhancing activities for scientific codes. This would be especially true for the variety of codes which deal with quantum dynamics in molecular systems (e.g. energy transfer processes and reactive processes). It therefore follows that our community will be greatly helped by the presence of a permanent Personnel infrastructure at the European level.

Moreover, opportunities for multidisciplinary simulations are restricted unless individual codes are engineered to a high standard, and protocols for data interchange are developed beyond the current status.

[77] Centre Européen de Calcul Atomique et Moléculaire

[78] see http://www.softeng.cse.clrc.ac.uk/CCPForge/workshop/requirements.pdf for discussion of this at a national level.

Path for realization, deployment and operation of the future CyberInfrastructure

We have considered 4 possible models for organization and location of the European CyberInfrastructure:

I. Distribute people around the community of scientific laboratories.

II. Central facility where all people are gathered, and with facilities for meetings and collaboration.

III. Central facility with some aspects contracted out.

IV. Different aspects of the requirement to be contracted to organizations such as computer centres, national labs. They could bid in a competition to provide the service.

Option I. would efficiently and directly help the development of codes, but only a small number of groups would be likely to benefit from it. IV. has the advantage that it fosters a sense of ownership by the participating countries. But we consider that II. is better, since there would be less fragmentation of expertise, and the existence of a central site would have significant advantages: (i) visibility; (ii) workshops which benefit from the presence of staff; (iii) stronger management control.

It will be important to construct the programme with robust governance so that to prevent from domination by special-interest groups.

It is difficult for this community to estimate the total cost of such an infrastructure. But the structure should be on a 5-year or longer time scale, and should include personnel focused on the various areas of generic component development, application support, data interfaces, etc. In other words, a reasonably significant team of software experts needs to be assembled for this endeavour to make sense. In addition, funds will be needed to support community-wide activities such as meetings, workshops and short collaborative visits.

Some of the activities we considered in this report are already being carried out, although still at an unsatisfactory level:

- There are existing activities in developing data-interface standards (CECAM/Psi-k, UK e-Science[79]) but these need to be extended and developed in order to assure wide adoption of standards.

[79] http://www.nesc.ac.uk/esi/events/394/

- Software engineering in our field is probably not yet developed to the highest possible standards, but this should be seen in the context of increasingly complex codes, and increasing need to couple codes together. Most code developers are not software engineering experts, and therefore some expert support is somehow needed. We note the existence of initiatives such as ALPS[80] which could perhaps be extended in both breadth and depth with appropriate support.

- At present, Grid technology does not have a strong impact on this field; relatively few of our applications are latency-tolerant parallel. However there is certainly activity in exploring this (eg COST action D37), and it is of particular relevance in the emerging importance of multi-scale modeling.

[80] Algorithms and Libraries for Physics Simulations http://alps.comp-phys.org/wiki/index.php/Main_Page

Appendix 9. Specific discussion of the development and maintenance of high quality software and data

Credible, accessible and sustainable software and data

Software is the main instrument of computing for scientists. They use software for performing *in-silico* experiments and testing the new methods and algorithms they develop. This, in turn, produces larger and larger amounts of data: the possibility of archiving and retrieving this data through archives will be soon crucial. Software has to satisfy a precise list of requirements to enable scientists to do their work. Software and archived data must be credible, accessible and sustainable. This is especially true for the scientific codes to be used in industrial environments.

By **credible** we mean that software has to produce correct results, within the limit of numerical accuracy. Correct, in this context, means that software must produce results in agreement with what is expected for a given method or model in certain specific conditions.

Software must also be **accessible** for use. The concept of accessibility touches several aspects, some being rather obvious, others less so. For example, it is rather obvious that there should be no limitations on using software due to affiliation. In this respect, the type of licensing, free (open source) or based on a fee, is an important factor. Also the method for downloading software might become a severe limitation to accessibility, although in the last few years this problem has been relieved by the tendency of publishing scientific software, especially open source and free software, on the web. In addition to these evident factors there are less obvious ones. One major factor improving accessibility is the availability of software on modern computers, environments and operating systems, including high-end supercomputers as well as commodity hardware. Moreover, software must be documented and users should receive support about how to use it properly. Software should be easy to use, including the possibility of controlling software via a graphical interface. Moreover, a major challenge that faces modern scientific computing is the growth of multi-disciplinary multi-scale simulations. In this respect, it is important that software adhere to some standard in order that it may be possible to use it in a workflow or concurrently (in parallel). In this sense, software is accessible for people who study phenomena that involve more than a single temporal and/or spatial scale. In a broader sense, computational science should also aim at software accessible beyond the computational domains: in experimental labs, in education and in industry.

Software must also be **sustainable** - otherwise the effort to build it will be lost. First, software should be continuously maintained. But its architecture should also be flexible enough and it should be programmed according to modern methodologies that favour its evolution from the point of view of both scientific features and performance. If a piece of software cannot evolve to include up-to-date algorithms, it will simply be discarded. At the other extreme, extensibility of software means that it should be possible (and easy) to extract core algorithms from one code to produce brand new software implementing new scientific ideas. This is particularly important for multi-disciplinary multi-scale simulations. In these simulations, a piece of matter is described at various levels of

theory: one part may be simulated including electrons with higher or lower level of accuracy, another part may be simulated by classical description of atoms and effective inter-atomic potentials, another part may be simulated using continuum models. Finally, the output of this multi-scale model can be used as input data for a mathematical model describing the evolution of the system. The multi-scale multi-disciplinary approach requires the combination of the best software developed within each field of science. That is, it should be possible to plug together pieces of software developed by each scientific community.

Simulations require input files and produce huge amounts of output data. These input and output data are rather precious as they are an important source of information that can be used again for further computation or analysis; they can also be a good source of examples for users and test-beds for new algorithms or new implementations of established algorithms. The same requirements we ask for software hold true for data: it must be **credible**, **accessible** and **sustainable**.

Credibility of data is strictly connected with credibility of software, as it is software that produces data. Some mechanism of data validation, for example peer reviewing, is necessary to make data credible.

To make data **accessible** it must be 'published' in the form of searchable archives. It must be possible to read and write data from/to archives independently from the software and platform which generated them and/or will make use of the data. The independence of archives from software/platform generating data is very important:this makes data reusable even outside the specific field and software/platform that it was produced from.

Data must be **sustainable** as the validity and usefulness of data last for a very long time. Sustainability requires that the infrastructure containing data be a long-term (possibly an ongoing) infrastructure. This, in turn, requires that data format and metadata describing the content of archives remain valid over a long period.

Finally, we want to underline the fact that there are two different kinds of users: end-users and algorithm developers (hereafter referred to as developers). Both kinds of users are interested in credibility, accessibility and sustainability but from different points of view. For example, accessibility means a user-friendly interface for end-users but modern software architecture for developers, as the latter need to access the internals of software for implementing new algorithms. However, the boundary between the two communities is very permeable. In fact, usually, after the initial stage of implementing a new algorithm, developers become the first end-users of the improved software. Some of the concepts presented here should make this distinction even deeper as accessibility, for example, aims at facilitating interaction with the code itself.

In the following paragraphs, we shall describe the structural features of software and data, and the services that can improve credibility, accessibility and sustainability.

Structural features and services for credible, accessible and sustainable software and data

In what follows we shall present, heuristically, some of the characteristics that software and data should feature to meet the needs described in the previous paragraph. A series of services should also be provided together with software and data. Such services are also presented here.

- **Modular software.** Modular software is software composed of strictly independent modules. A module is a group of opaque routines. It is unnecessary to know the internals of these routines to use them. This approach is already used in scientific software but it is mostly restricted to mathematical routines (BLAS[81], LAPACK[82]). An important factor that makes modules easy to use is that data flow is explicit. It is important to stress here that it is possible to implement such a modular approach in all the standard languages, including Fortran77 and ANSI C, but a modern approach to programming, together with the expertise of computer scientists, is needed. The optimal solution would be to build scientific software on archives of modules (scientific software libraries). This will make software reusable, thus making the investment in software development and maintenance more cost-effective.

- **Porting on a wide range of computers.** Community software should be useable on all modern computer architectures. Since scientists often use local resources for their research activity, commodity hardware must be included in this list. Porting includes optimisation and tuning.

- **Documentation.** Documentation for end-users as well as developers should be provided together with software. Documentation should refer to theoretical methods and numerical tricks related to the software. Documentation for both end-users and developers should adhere to standards.

- **Archive of knowledge.** An archive of knowledge is an archive of input and output simulation files. Contributions to this archive should be provided in some standard format and on a voluntary basis.

Data also requires a series of structural features and services:

I. **Standard metadata.** Users need to search, compare and judge the validity of archived data. This requires metadata describing the content of data and their origin (including software, the researcher who ran the simulation, etc.). These metadata should conform to some (extendible) standard; that is, the standard for metadata must represent the current needs but must be flexible enough to accommodate future extensions.

[81] Basic Linear Algebra Subprograms. http://www.netlib.org/blas/
[82] Linear Algebra Package. http://www.netlib.org/lapack/

II. **User and software interface to archives.** Different software (users), running on various hardware and operating system environments, need to import/export data. This requires tools for extracting data from archives and translating it from the general standard to the native standard of a given code. This also implies that data is independent from the binary representation of numbers on the given computer architecture, as otherwise it would be difficult to import/export data from/to different environments.

III. **Validation.** Data requires validation. We then run into the circular argument that data is used for validating software and validity of data is due to validity of software (and other simulation conditions). However, in practice, the fact that data is used to validate software implies a process of peer reviewing.

Relation between structural features and services, and software requirements

The structural features of software and data archives just described have different implications for their credibility, accessibility and sustainability. The relationship between the structural features and services reported above and the requirements described in the first paragraph is depicted in Fig. A.9.1.

Credibility. Archiving knowledge greatly contributes to credibility. A large archive of simulations from a given code would allow extensive verification of results obtained with it. Such an archive would allow a benchmarking of the software in terms of accuracy as well as performance. Although there exist examples of such archives, the main problem with them is that their time span is usually rather short, often only of the length of a project, i.e. a few years.

We would also like to stress that modularity is important for credibility. Indeed, it is far easier to debug a modular code. Moreover, if software is based on a library of well-tested modules the developer is left with the single task of validating the new modules.

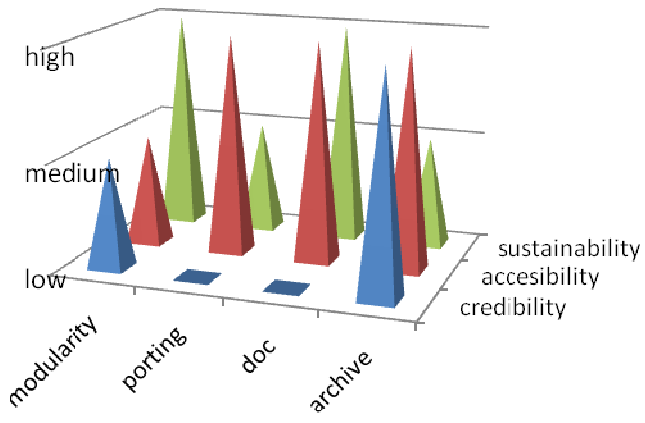


Figure A.9.1 – impact of variuos activities on credibility, accessibility and sustainability of scientific software

Accessibility. Three components have a large impact on accessibility: porting, documentation and the archive of simulations. The availability of a code on a computer with some standard architecture is an obvious precondition for accessibility. Availability implies being able to port the

software onto (possibly) all current computer architectures, operating systems and execution environments (compilers, mathematical libraries, etc.). Actually, most of the work on porting is done by the code developers. The problem with this is that usually code developers are scientists with personal agendas driven by different aims than software maintenance. It is well understood that a different, more expandable model must be put in place. This model could include close collaborations with computer scientists (including those working at HPC centres and/or hardware and software companies).

Setting up simulations with current software is usually a very complex task. Good documentation is a key ingredient in this respect. The documentation should explain the meaning of input parameters and possible correlations among them. However, documentation is usually not sufficient. An extensive archive of sample simulations, of the type described above, would be very useful. Moreover, many kinds of simulation are dependent on an extensive pre-processing phase. For example, in fluid dynamics it is necessary to build the grid on which differential equations are solved. Making such data available to the scientific community would reduce pre-processing work and, in particular, would reduce the duplication of work of little scientific impact. Documentation and archiving of simulations should be complemented by tutorials and help-desk type of support.

Modularity also contributes to the accessibility of software as it makes it possible to implement interoperable codes. In fact, it is far easier to combine modules adhering to some standard (or complete codes based on such modules) than codes produced just for solving problems in a specific field, which (usually) only satisfy the requirements of their particular purpose without foreseeing any possible future use outside their original limited scope.

Sustainability. The main issue for sustainability is again modularity. That would make it easy for any developer to introduce new features, focusing only on the synopses of the routines the new module depends on. On the contrary, with non-modular software, a developer must know the internals of a large part of the code to follow the rather obscure flow of data of interest to him. Modularity is also helpful for debugging, optimisation and tuning as, in this case, errors and factors influencing performance tend to be rather localised. Documentation for developers, with description of routines and their synopses and pre- and post-requisites, is the necessary complement to modular software. Software must be validated all along its life cycle. Validation of software requires a large set of testbeds ‘stressing’ the software under various conditions. An ideal set of simulations could be provided by an Archive. Since the evolution of computer architectures is usually continuous (with some step changes from time to time), a continuous activity of porting would make software sustainable, as the optimisation and tuning on an evolution of a previous architecture would be a relatively minor step.

Despite the fact that the benefit of credibility, availability and sustainability has been acknowledged for a long time, the lack of an infrastructure for maintaining software has prevented the adoption of programming guidelines and provision of services related to software of the type we depicted above.

Table A.9.1: codes included on the survey on the state of European scientific software. DFT stands for Density Functional Theory, PW for planewaves basis set and gaussian for gaussian basis set.

Name	Scientific Area	Brief Description	Licensing
ABINIT	Condensed Matter Phys	DFT+PW+Pseudopotentials	Free
ESPResSo	Condensed Matter Phys	coarse-grained off-lattice simulations	Free
VASP	Condensed Matter Phys	DFT+PW+Pseudopotentials	Licensed
CP2K	Condensed Matter Phys	DFT+(gaussian+PW)+classical simulations	Free
CPMD	Condensed Matter Phys	DFT+PW+Pseudopotentials	Free for academic use
Wien2K	Condensed Matter Phys	Full-electrons Augmented PW	Licensed
Quantum-Espresso	Condensed Matter Phys	DFT+PW+Pseudopotentials	Free
Code_Aster	Engineering	mechanical and thermal analysis	Free
Code_Saturne	Fluid Dynamics	Incompressible + expandible fluids + heat transfer + combustion	Free
OpenFOAM	Fluid Dynamics + structural mechanics	Finite volume on an unstructures grid	Free
Salome	framework for multi-physics simulations	Engineering	Free
COSMO-Model	Climatology and Meteorology	Weather forecasting and climate	specific agreement

The Status of European Scientific Software

In order to establish In order to establish the situation of scientific software, we conducted a survey of the most important European community codes in several fields of science.

Table A.9.1 lists the software considered in this survey. We considered both free and licensed software. Data were provided by code maintainers on a voluntary basis.

Europe is at the forefront of software development and these tables illustrate the large amount of activity in this field. However, there is not yet a structure that would ensure long term credibility, accessibility, and sustainability of European software.

In this regard our first remark is that support to end-users is offered mainly on a voluntary basis. In Table A.9.2 we report data relative to documentation and user support. The number of people involved in the support is rather small compared to the number of users. This makes user support quite unsatisfactory, even though huge efforts are made by volunteers. In some cases, supporters maintain an archive of previously asked questions and the mailing list of questioners directing first to

them any incoming request for support. However, searching the desired information among thousands of questions is rather fastidious. A more efficient approach would consist in providing lists of Frequently Asked Questions (FAQ). However, in this case also, the effort required for compiling a FAQ is huge and there is no economic incentive for doing it. The few exceptions to this situation are the codes developed by Electricité de France (EdF), which demonstrate that a budget devoted to software maintenance and support is worthwhile.

Table A.9.2: data on documentation and user support. FTE stand for Full Time Equivalent

Name	# users	# keywords	Manual Size	Support for End Users
ABINIT	~1000	439	10MB	voluntary, mailing list
ESPResSo	~20 groups	N.A.	100 pages	voluntary, mailing list (~10 FTE)
VASP	800 licenses	150	200 pages	1 FTE
CP2K	~100	1300	20000 lines	voluntary, mailing list
CPMD	1000-6000	320	150 pages	voluntary, mailing list (5 people involved)
Wien2K	~1100	100 (typical simulation)	200 pages	mailing list, (4 people involved)
Quantum-Espresso	~700	400	70 pages	voluntary, mailing list
Code_Aster	300 (EDF) + 22000 download since 2001	2000	12000 pages (including theory)	3 FTE for helpdesk + 20 experts for EDF, three companies offer services
Code_Saturne	80 (EDF) + 20 industrial/Academic partners + 5 for teaching	400	250 pages	2 FTE
OpenFOAM	~2000	N. A.	200 pages	4 FTE
Salome	50 (EDF) + 21 institutions	N. A.	504 HTML files	6 FTE
COSMO-Model	7 Natl. Centres, 80 res. groups	N. A.	200 (400 including scientific doc)	Voluntary

The situation regarding documentation is rather heterogeneous, with some codes being distributed with only a few dozen pages of documentation, while others are distributed with exhaustive documentation discussing theoretical and technical aspects. Also in the latter case, the limiting factor for producing adequate documentation is the lack of economic support and technical personnel devoted to this non-scientific task.

Table A.9.3: data describing aspects correlated to software sustainability. The effort in the scientific (Sci.) and technical (tech.) (porting, parallelization, etc.) is expressed in Full Time Equivalent.

Name	Developers sci./tech.	# lines of code	# routines or classes	# pervasive variables	Programming languages	Trend
ABINIT	4-6/4-6	405000	1077	NA	Fortran90, C, scripting	
ESPResSo	8/2	50000	500	NA	C ,Tcl	Doubled in 2 years
VASP	3/1	165000	2000	NA	Fortran90	
CP2K	10/N.A.	420000	4297	NA	Fortran95	Quadrupled in 5 years
CPMD	3-5/N.A.	225000	2400	17000	Fortran77 + Cray Pointers	10* in 14 years, doubled in 5 years
Wien2K	2.5/2.5	200000	2500	NA	Fortran90, C, scripting	
Quantum-Espresso	7.5/2.5	265000	2500	1700	Fortran95	
Code_Aster	50/15	1200000	6000		80% Fortran 77, 20% Python	10* in 16 years, +50% in 5 years
Code_Saturne	4.5/5	500000	500	~100	49% Fortran, 41% C, 10% Python	
OpenFOAM	2/2	500000	1300	NA	95% C++, 5% Java	
Salome	0/8-10	500000	4076		mainly C/C++, Python, shell scripts, Fortran	
COSMO_Model	10/1	250000	140 F90 mod.	220	Fortran, C	

Moving to sustainability, we first note that current software is very complex (see Table A.9.3). A typical size is 400000 lines of code and 2500 routines/classes. On top of this, one must consider that usually a piece of software makes use of a large number of 'pervasive' variables which pass through the code in an obscure data flow. Here again there are exceptions such as OpenFOAM (written in C++) and CP2K (written in FORTRAN95) which adopted a strictly object-oriented programming practice. These exceptions demonstrate that developing scientific software with modern programming techniques is possible. If we combine these observations with the trend of growing complexity, it is rather clear that very soon we will be confronted to a software sustainability crisis. Another factor must be considered: it will be very difficult to adapt most existing complex codes to the coming massively parallel computers. In fact, the structure of many of the codes considered in this survey is strongly dependent on the parallel programming paradigm adopted in the early stage

of the development. The current shift from hundreds to tens of thousands of CPUs will require a change in the parallelisation scheme, which will be very difficult to implement in such very complex community codes.

Conclusions

Scientific software requires a new improved model for development, dissemination, maintenance and support. Achieving credibility, accessibility and sustainability requires that some features be built into the structure. The main elements we have identified are: modularity, documentation, porting and an archive of simulations. There are also other activities needed in connection with software development, namely standardisation, training in software use, and a help-desk for end-users and developers of new extensions.

Similarly data also require a different, more sustainable, model of archiving. The main characteristics needed for improving data credibility, accessibility and sustainability are: standard metadata, user/software interface for accessing archived data, and validation.

The effort, the human effort in particular, needed to implement and maintain these requirements for software and data does not constitute research per se, but we strongly believe it to be a necessary part of the research infrastructure. Such a part of the cyber-infrastructure has to be largely managed by the scientific community itself, and this is the goal of standardization of modules/data. There is also an important human aspect for achieving these goals of credibility, accessibility and sustainability, as we need both trained developers in modern, professional, programming, and efficient training of users. Collaborations and networking are also necessary for implementing modularity and interoperability. They are also often the first level of "peer reviewing".

Appendix 10. ESCRINS: a model for a permanent infrastructure for scientific software

As has been recognized in chapter 3, the need exists for a European initiative to be undertaken on scientific software. There is a global demand of scientific computing for setting up a permanent technical infrastructure to federate and competitively enhance existing individual and national initiatives in this matter. This infrastructure should be put into existence through a convention between National Research Councils. The organizing committee of this Forward Look has examined structural and other aspects of a "European Scientific Computational Research Infrastructure (ESCRINS[83])" by considering the kind of services it should provide to the scientific communities, its possible structure and governance, and a reasonable funding scheme providing long-term support. These aspects are presented in the following paragraphs. The following is then a model of how the ideal situation would look like: we believe it is useful to present this scheme since it provides explicitly concrete examples of typical problems to be solved and of what sort of organization to build, in order to tackle the issues raised.

Scope and services offered by ESCRINS

Scope

The infrastructure aims at code development, documentation, updating, maintenance, user support and education for scientific computing activities. For these objectives to be successfully achieved, the infrastructure should offer training to code users and developers and promote a large and open cooperation between scientists, programming specialists, and applied computer scientists working in universities, research and/or industrial centres. Moreover, ESCRINS should adopt a federal structure, integrating existing national activities on scientific software, computer architecture and system operation.

A non-exhaustive list of the envisaged tasks and activities of ESCRINS is given below:

Tasks

- Fund and support the development, documentation and maintenance of a set of community codes. Establish the coding standards which one ought realistically aim to attain in scientific software.
- Professionalize the activity of code writing in the computational sciences, including:
- involving 'applied computer scientists' in code development;
- training more of them if there are not enough;
- establishing employment and career paths for them, because they may only be involved part-time with a particular code, and that will certainly not extend over their whole working

[83] **écrin** [ekYR] **n. m.** French for 'Casket', from Old French *escrin* apr. 1050; lat. *scrinium*

♦ Boîte ou coffret où l'on range les bijoux, les objets précieux. ⇒ baguier, étui. Offrir un collier dans un écrin. Écrin à petites cuillères. Ranger l'argenterie dans les écrins.

life;

- educating scientists in relevant aspects of computer science, which is essential in the long run for those initiating new scientific applications of computing and might include seconding some new PhD students to do a graduate diploma in Computer Science;
- Collaborate with High Performance Computing Centres in supporting all the levels of the pyramid of computer resources (Fig. A.10.1) and the whole community of scientific computing[84];

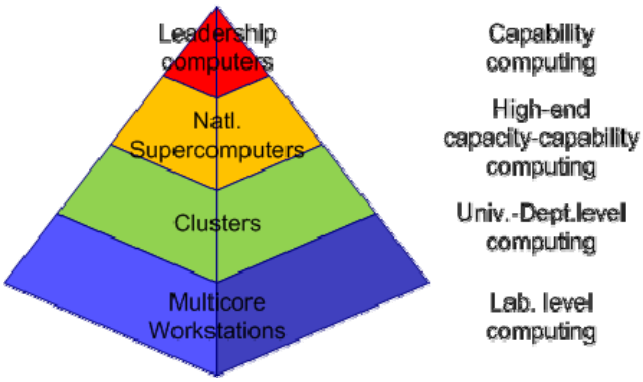


Figure A.10.1 – pyramid of the computer resources

Services offered

- Disseminate the latest advances through pre-research workshops where new ideas get nucleated that result in international collaborations for specific research projects being formed.
- Develop code and re-write existing scientific software in modular form to facilitate new additions and portability.
- Maintain existing codes (including standard test suites to ensure preservation of functionalities through successive versions).
- Provide user support for new code users (especially with complex codes).
- Provide training in code use and education.
- Write and disseminate documentation and training materials for new users.
- Provide education in the methodologies of computer simulation for young researchers.
- Provide training in aspects of applied computer science relevant to good code writing.
- Establish contacts with industry.
- Establish procedures for continuously adding new types of applications and capabilities.
- Produce newsletter and internet communication tools for announcement of conferences and workshops.

Structure and governance (see Fig. A.10.2)

The Management Committee

The management committee is responsible for setting, approving, directing and monitoring the overall strategy direction of ESCRINS. It is chaired by the President and is composed of experts in

[84]Although lip service is often paid to this principle, it is largely ignored in practice with computing centers focusing predominantly on the high-end computing, as is perhaps inevitable. However today's calculations on a local workstation become tomorrow's work on supercomputers, and a better balance has to be established for nourishing the whole community and all layers of the pyramid.

scientific computing agreed with the funding agencies. The implementation of this strategy is the responsibility of the ESCRINS director.

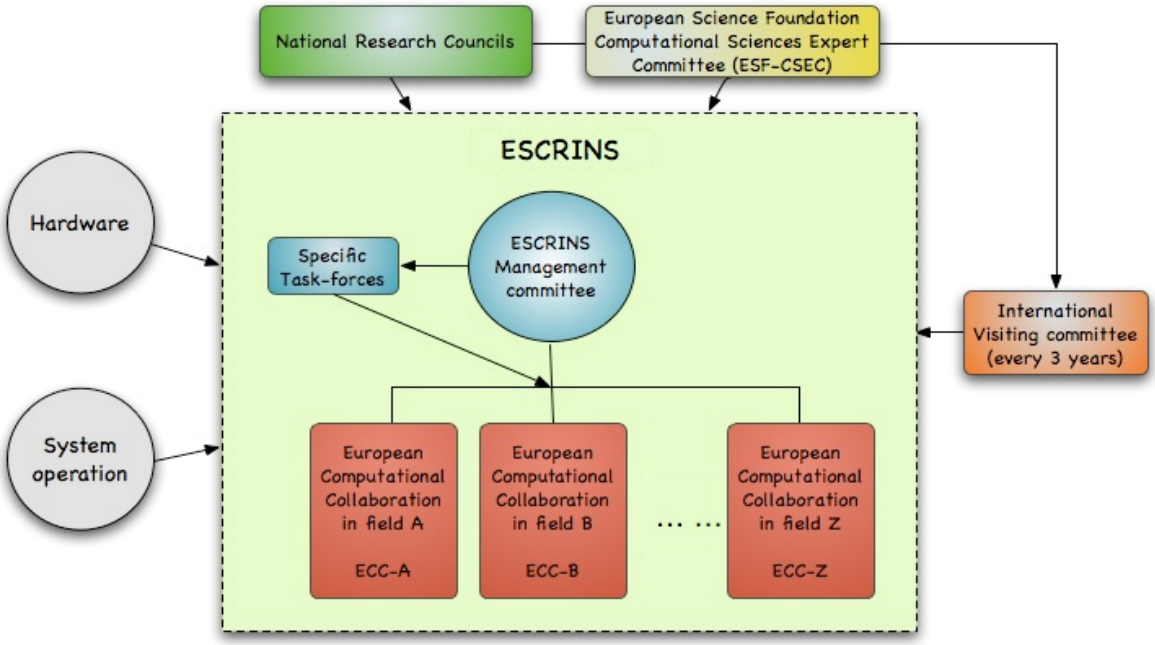


Figure A.10.2 – structure of the ESCRINS and its relation with other institutions and bodies

The specific task forces

The management committee can set up specific task-forces composed of experts (researchers, computer scientists, …) to tackle specific problems. For example, a specific task force might be needed for developing coding standards.

European Computational Collaborations

The European Computational Collaborations (ECC's) are the main pillars of the infrastructure. These are responsible for code development, code management, maintenance and user support in the specific scientific area envisaged. ECC's should report annually to the ESCRINS Management Committee and, through it, to supporting agencies. At the start of ESCRINS the scientific communities participating in the Forward Look exercise could form the first ECC's in their respective areas:

- Astrophysics,
- Material Sciences and Nanotechnology,
- Fluid Dynamics,
- Climatology and Meteorology,
- Life Sciences,
- Quantum Molecular Sciences.

ESCRINS in relation to external bodies

In setting up and developing its activities, ESCRINS would have tight and diverse interactions with existing organizations such as research and computer centres, national research councils and scientific policy makers.

Interactions with high-performance computer centres (HPC) and the computer industry are of crucial importance for code development, maintenance, code management and user support, which are the main activities of ESCRINS. Thus, the management committee of ESCRINS should establish, maintain and promote interactions between ECC's on the one hand and with HPC's and the computer industry on the other hand.

As is usual for infrastructures, the periodic evaluation of their activities and achievements would condition the financial support for the next period. An International Visiting Committee,would evaluate the work of ESCRINS as a whole, every three years, and report to bodies and organizations supporting the infrastructure.

Funding Model

Principles of Funding Infrastructure

In considering the options or possible models for the funding of ESCRINS, the following key principles should be satisfied:

ESCRINS is an infrastructure needed to support, enable and federate the large number of individual current and future research projects in the European scientific and engineering community, in the framework of ESCRINS objectives within each ECC. These projects may be quite small or extremely large, involving many collaborating research groups. However, ESCRINS itself is not a project, even a large one. It is a research infrastructure requiring long-term and stable funding so that to reach successfully its objectives.

The principal beneficiaries or their sponsors should, ultimately, be the principal sources of funding for the infrastructure that supports their work. There are certainly many beneficiaries of ESCRINS, but it is essential "raison d'être" is to support, and make more productive, the scientists actually doing the computational research and encourage those complying with ESCRINS objectives for the given ECC scientific area. Their main sponsors would be their national research councils or science foundations.

Following from the previous point, we expect the funding model for ESCRINS to be responsive to the research community it exists to support. This means that the appropriate funding model must put in place a continuous assessment of the needs of the community for infrastructural support and of the value for the money it delivers.

Scale of Funding

In order to estimate the scale of funding required for ESCRINS, the following assumptions were made:

- Each major code used and under development in the European computational research communities should have 2 expert staff devoted to it (scientist/programmer and applied computer scientist);
- Cost human resources at roughly 100 k€/year;
- Support for collaborative working with the relevant research groups in the community is evaluated at 50 k€/year (travel, meetings, …).

This leads to a rough costing of 250 k€/year for each major code. Additionally, it is assumed that ten (10) broad scientific and engineering communities will be serviced by ESCRINS and that each of these communities would nominate 5-10 key codes. Thus, a rough estimate for the total number of codes to be supported by ESCRINS amounts to 50-100, leading to an overall cost estimate in the range of 12.5 M€/year - 25M€/year.

The above numbers give a rough but realistic estimates of the overall financial scale of ESCRINS.

Possible Funding Sources

ESCRINS would have many potential beneficiaries and therefore possible funding sources, including:

- Inter-governmental agreements
- national governments
- regional governments
- National academic research agencies
- National Laboratories
- Computational Science centres
- Supercomputer centres
- Commercial and industrial sources
- software companies
- hardware companies
- user industries
- Charities
- Universities
- European agencies
- European Commission (EC)
- European Research Council (ERC)

- European Institute of Technology (EIT)
- European Science Foundation (ESF)

In considering these possible sources of funding, the essential point is the need for long-term stable funding. There have been many attempts to fund such infrastructures (similar to ESCRINS but within individual fields and thus of a smaller scale) as finite, fixed-term projects. These attempts have always proved unsatisfactory, usually leading to a dilution or loss of the mission of the infrastructure in question as it struggles to fulfill the specific project demands of the agency funding it. Infrastructures simply are not projects, and they should not be funded as if they were.

Proposed Model

The foregoing analysis leads to the conclusion that the optimum funding model for ESCRINS must be based on core funding from national sources, supplemented by a portfolio of additional project-based sources (grants, contracts and so on) each of which may be more short-term and intermittent but which, taken together, could amount to substantial leverage on the core funding.

It is thought that initially the core funding should be provided via an agreement or convention between the key national research councils or science foundations. However, the scale of ESCRINS, and the value it delivers to the European research community as a whole, are expected to grow over time. We can thus envisage that at some later time, perhaps 5 years or so from the commencement of ESCRINS, it may be more appropriate to replace this inter-agency convention by a direct agreement or treaty between Governments, as is the case for a number of other large-scale scientific infrastructures in Europe. This perspective is depicted in Fig. A.10.3 which is deliberately and inevitably impressionistic at this stage.

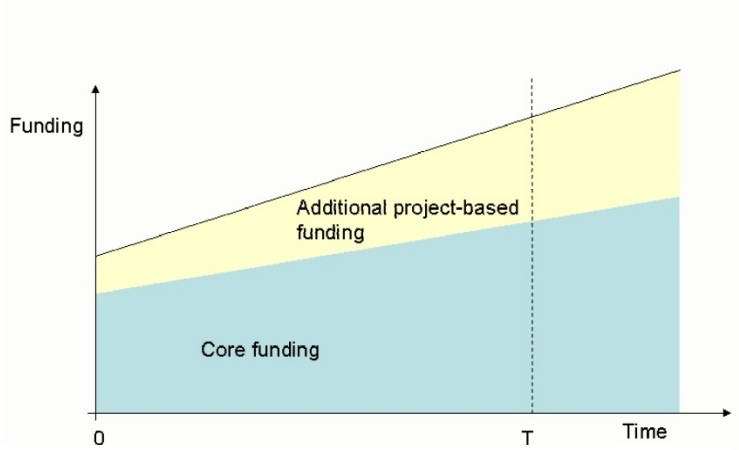


Figure A.10.3 - Schematic representation of the evolution with time of ESCRINS funding.

Many important questions remain to be settled. For example, it may be possible to assemble a proportion of the initial core funding by bringing together the funding of existing collaborations on which the collection of ECC's might be based. Moreover, the precise nature of the additional leveraged funding is unspecified at this stage: it is likely to include a contribution from the European Commission but specific suggestions for this cannot yet be offered. However it is an essential part of the conception that the infrastructure provided by ESCRINS will always require core funding since it is simply not appropriate to fund such operations as one would a finite project.