

# Preface

Climate modelling in former times mostly covered the physical processes in the Earth's atmosphere. Nowadays, there is a general agreement that not only physical, but also chemical, biological and, in the near future, economical and socio-logical—the so-called anthropogenic—processes have to be taken into account on the way towards comprehensive Earth system models. Furthermore these models include the oceans, the land surfaces and, so far to a lesser extent, the Earth's mantle. Between all these components feedback processes have to be described and simulated.

Today, a hierarchy of models exist for Earth system modelling. The spectrum reaches from conceptual models—back of the envelope calculations—over box-, process- or column-models, further to Earth system models of intermediate complexity and finally to comprehensive global circulation models of high resolution in space and time. Since the underlying mathematical equations in most cases do not have an analytical solution, they have to be solved numerically. This is only possible by applying sophisticated software tools, which increase in complexity from the simple to the more comprehensive models.

With this series of briefs on “Earth System Modelling” at hand we focus on Earth system models of high complexity. These models need to be designed, assembled, executed, evaluated, and described, both in the processes they depict as well as in the results the experiments carried out with them produce. These models are conceptually assembled in a hierarchy of sub-models, where process models are linked together to form one component of the Earth system (Atmosphere, Ocean, ...), and these components are then coupled together to Earth system models in different levels of completeness. The software packages of many process models comprise a few to many thousand lines of code, which results in a high complexity of the task to develop, optimise, maintain and apply these packages, when assembled to more or less complete Earth system models.

Running these models is an expensive business. Due to their complexity and the requirements with respect to the ratios of resolution versus extent in time and space, most of these models can only be executed on high performance computers, commonly called supercomputers. Even on today's supercomputers, typical model

experiments take months to conclude. This makes it highly attractive to increase the efficiency of the codes. On the other hand the lifetime of the codes exceeds the typical lifetime of computing systems and architectures roughly by a factor of 3. This means that the codes need not only be portable, but also constantly adapted to emerging computing technology. While in former times computing power of single processors—and that of clustered computers—was resulting mainly from increasing clock speeds of the CPUs, today's increases are only exploitable when the application programmer can make best use of the increasing parallelism off-core, on-core and in threads per core. This adds complexity to areas like IO performance, communication between cores or load balancing to the assignment at hand.

All these requirements put high demands on the programmers to apply software development techniques to the code, making it readable, flexible, well structured, portable and reusable, but most of all capable in terms of performance. Fortunately, these requirements match very well an observation from many research centres: due to the typical structure of the staff of the research centres, code development oftentimes has to be done by scientific experts, who typically are not computing or software development experts. Therefore, the code they deliver needs a certain amount of quality control to assure fulfilment of the requirements mentioned above. This quality assurance has to be carried out by staff with profound knowledge and experience in scientific software development and a mixed background from computing and science.

Since such experts are rare, an approach to ensure high code quality is the introduction of common software infrastructures or frameworks. These entities attempt to deal with the problem by providing certain standards in terms of coding and interfaces, data formats and source management structures, that enable the code developers as much as the experimenters to deal with their Earth system models in a well acquainted, efficient way. The frameworks foster the exchange of codes between research institutions, the model inter-comparison projects so valuable for model development, and the flexibility of the scientists when moving from one institution to another, which is commonplace behaviour these days.

With an increasing awareness about the complexity of these various aspects, scientific programming has emerged as a rather new discipline in the field of Earth system modelling. At the same time, new journals are launched providing platforms to exchange new ideas and concepts in this field. Up to now we are not aware of any text book addressing this field, tailored to the specific problems the researcher is confronted with. To start a first initiative in this direction, we have compiled a series of six volumes, each dedicated to a specific topic the researcher has to face when approaching Earth System Modelling:

Volume 1. Recent Developments and Projects

Volume 2. Algorithms, Code Infrastructure and Optimisation

Volume 3. Coupling Software and Strategies

Volume 4. IO and Postprocessing

Volume 5. Tools for Configuring, Building and Running Models

Volume 6. ESM Data Archives in the Times of the Grid

This series aims at bridging the gap between IT solutions and Earth system science. The topics covered provide insight into state-of-the-art software solutions and in particular address coupling software and strategies in regional and global models, coupling infrastructure and data management, strategies and tools for pre- and post-processing, and techniques to improve the model performance.

Volume 1 familiarizes the reader with the general frameworks and different approaches for assembling Earth system models. Volume 2 at hand highlights major aspects a researcher is confronted with when it comes to the real work. The starting point of the whole process chain is the programming of the physical component models, and the adaptation of the software to make it run efficiently on a given computer system. As there is already a large amount of literature available that deals with the physical and numerical problems to solve, our focus here is on design issues that are related to the software development, its maintenance and performance. Volume 3 describes different technical attempts from the software point of view to solve the coupled problem. Once the coupled model is running, data are produced and postprocessed (Volume 4). The whole process of creating the software, running the model and processing the output is assembled into a workflow (Volume 5). Volume 6 describes coordinated approaches to archive and retrieve data.

Hamburg, June 2011

Reinhard Budich  
René Redler

Earth System Modelling - Volume 2  
Algorithms, Code Infrastructure and Optimisation  
Bonaventura, L.; Redler, R.; Budich, R.  
2012, XI, 96 p. 20 illus., Softcover  
ISBN: 978-3-642-23830-7