

Scientific Computing and Numerical Software

Scientific computing has become a major branch in many scientific, engineering and technological disciplines. The main output of scientific computing is numerical software, which is central to our industrialized, technological society. Numerical software lies at the heart of computer simulation and computer aided design technology.

For example, starting from differential equations based on basic laws such as Ohm's law and Kirchhoff's law, numerical software is used to design larger and faster electronic circuits for consumer goods like mobile phones, compact-disc players and personal computers. Starting from differential equations from structural mechanics, numerical software is used to design new infrastructure such as bridges, harbours, tunnels, dikes and waterways. Numerical software is used in the financial market to analyse future options, in the chemical industry to optimize plants and refineries, in the shipbuilding industry to design faster sailing ships. It is used around the world for weather forecasting, climate modelling and for analysing the threat of global air pollution. In more recent years scientific computing has entered research in biology and medicine. There is a common expectation that scientific computing and numerical software will play an important role to provide the understanding necessary for solving great challenges in medicine, like the treatment of cancer and heart diseases.

The development of user-oriented numerical software has begun end of the 1960s, at the time when the first time-sharing mainframe computers were installed in academic and research laboratories. Since then, research in leading-edge mathematical modelling, numerical algorithms and software, have brought us faster and faster algorithms. Research in computer hardware has brought us more and more processor speed and central memory. With this performance at the desktop, mathematicians, computational scientists, engineers and software developers have more computer power on hand than offered by the huge mainframe supercomputers of the Cray line of just a few years ago.

With respect to applications the trend is towards really novel, large-scale, highly complex problems. An example of such a novel application is a model for the whole human cardiovascular system. Another example, from cell biology, is the Silicon Cell, which is to become a computer model that 'computes life at the cellular level': a project of such computational complexity that a consortium of researchers in the Amsterdam area from cell biology, mathematics and computer science is preparing a decade-long research program on this subject. To keep up with demands from these challenging life-science applications, a substantial amount of novel mathematics and computer science must be developed.

With respect to software design, the future trend is towards making available more diagnostic tools for testing and evaluation, validation and verification. Numerical software packages have reached a certain level of maturity, but as a rule they are still mainly workable for experts with a broad numerical experience. This delays their use and distribution into new areas. Therefore, improvements are necessary to relieve users from technical intricacies. Encouraging new tools for this will come, among others, from software engineering, computational steering and visualization. These tools will further enhance the widespread use of numerical software.

There is no doubt that for industrialized countries like the Netherlands, the dependence on scientific computing of technological and scientific development is considerable and will further increase for many

years to come. This dependence calls for a strong and long lasting research effort. Because scientific computing can be seen as part of applied and numerical mathematics, of the computational sciences and computer science in general, developments in all these areas are relevant. In particular the complete spectrum from mathematical modelling, algorithm design and analysis to software implementation will be extremely important.