Present and Future Scientific Computation Environments

Steve Hague and Uwe Naumann

ABSTRACT Problem Solving Environments and Environments for Scientific Computing play an increasingly important role in the modelling, simulation, and optimization of real-world applications. In this chapter we discuss some of the experiences of NAG Ltd. in several European projects aiming towards the development of such tools. It will also note some of the encouraging developments, particularly in the area of interface standards that might make PSE construction a more effective and realistic prospect in the future.

5.1 The Big Picture

Mathematical models of real-world applications have grown more complex over the past decades. The availability of high speed information technology has not only lead to a speedup in the solution of most problems, it has also caused a remarkable scale-up of the problems that scientists and engineers are willing and able to consider. Large numbers of often highly specialized software applications have been developed, each application seeking to address a major computational problem-solving task, or class of tasks. For most of these applications, the underlying philosophies, platforms and interfaces vary drastically. These variations make it very difficult to achieve desirable properties such as portability, extensibility, and ease-of-use. The

FIGURE 5.1. Applications
intention of this chapter is to present a (subjective) view on how software should be developed to support both application programmers and end-users from various fields in their attempts to understand, to simulate, and to optimize real-world processes.

![Diagram of software development process]

FIGURE 5.2. Software

The typical stages of the modern applications development process are illustrated by Figure 5.1 and Figure 5.2. Usually, this process requires contributions from experts from several fields. Due to their structure and complexity, most computational problems have to be approached using numerical methods which result in discretizations and approximations of the underlying continuous model. For example, algorithms are developed to generate meshes, for solving systems of linear or nonlinear equations, or for interpolating intermediate function values. These algorithms are usually implemented using some high-level programming language leading to highly complex software packages running on increasingly powerful computers. The development of mathematical and visualization software represents one of the major challenges faced by both industrial and academic experts. Furthermore, application code enhancement and transformation tools, such as compilers for scientific computing, automatic differentiation packages, and software for verified computation based on interval arithmetic, are gaining importance. Considerable effort has been put into the development of Problem Solving Environments / Scientific Computation Environments (PSE/SCE) with the aim to formalize and simplify both the modeling and the algorithmic aspects of the application development process. Simulation and optimization as well as visualization are addressed by various existing PSEs/SCEs. Different approaches lead to software packages typically based on some combination of symbolic and numeric computation, such as Maple, Mathematica, Matlab and its Toolboxes, or with a more statistical emphasis such as S-Plus and SPSS. These could be described as general or multi-purpose computational packages that can solve a variety of problems within their scope, sometimes with the aid of a supplementary