



Preface¹

This special volume is devoted to computation in problem solving environments (PSEs) with emphasis on differential equations. Exploiting the amazing developments in computer hardware and graphical interfaces, PSEs try to make it as easy as possible to solve mathematical problems. They are so convenient in the classroom that it is unusual nowadays for a student in engineering, science, or mathematics to graduate without having used at least one PSE. The contribution of R.L. Lipsman, J.E. Osborn, and J.M. Rosenberg relates their extensive experience with teaching a first course in ordinary differential equations using the “big three” PSEs, MATLAB, Maple, and Mathematica.

The convenience of a PSE is not free—programs are interpreted and so run slower than compiled programs written in Fortran or C. Nevertheless, even small computers are now so fast that the convenience of working in a PSE and access to a wide range of quality mathematical software have resulted in many scientists doing all their numerical computing in this way. A number of the papers of this issue are concerned with how to exploit capabilities of a PSE to make mathematical software easy to use. Others consider how to make the software efficient, either by using compiled functions or programming practices that take full advantage of the PSE. Several contributions consider how to make software in other libraries available in a PSE of choice. In this connection we quote from M. Dewar’s discussion of how to make available in two PSEs the vast resource of mathematical software that is the NAG library:

Users come from disciplines that span many areas of science, engineering and finance, in both industrial and academic settings. However modern users are less keen on programming (at least in the traditional sense) than their predecessors, not least because they have access to excellent interactive tools such as MATLAB and Maple which, combined with modern hardware, allow them to formulate and solve problems much more quickly than by writing a Fortran program. Indeed many NAG users actually start out in such packages and develop a prototype of their program there before re-coding it in C or Fortran.

The PSEs we have cited are large software projects, but they have inspired smaller, more specialized PSEs, some of which are discussed in contributions to this volume.

L.F. Shampine
Guest Editor

¹Published electronically March 31, 2008