REFLECTIONS ON TEACHING COMPUTATIONAL PHYSICS AND APPLIED MATHEMATICS

Paul Abbott (paul@physics.uwa.edu.au)

School of Physics, M013,The University of Western Australia, Crawley, WA 6009, Australia

KEYWORDS: applied mathematics; computational physics; procedural programming; integrated computational environment; *Mathematica*

ABSTRACT

In most applied mathematics and computational physics courses, simulation and modelling are taught by stressing numerical techniques, while visualisation often requires a range of specialised software tools. One approach is to use a procedural programming language such as Fortran or C. Although learning procedural programming is very useful it can detract from the desired goal of teaching computation. A second approach is to develop “black-box” applications for illustrating physical concepts. When well done this approach requires little instruction, and the focus is entirely on the situation under investigation. A disadvantage is that the student may not learn any computational techniques. A third way is to use an integrated computational environment, for example Mathematica, which couples an excellent graphical user interface to a high-level programming language. In this talk I will demonstrate my approach, developed over the last 20 years, by working through the solutions to selected problems from assignments and exams.

Proceedings of the 16th UniServe Science Annual Conference, University of Sydney, Sept 29th to Oct 1st, 2010, page 116, ISBN Number 978-0-9808597-1-3