

variable length arrays for numerical computing, and support for 64-bit computing. All variations of C that conform to the C99 standard contain these features.

As a language, C continues to evolve but remains backward compatible. As long as it conforms to the C99 standard, a compiler (which translates high-level C programs into executable code) will work with any program written in C89. And C is unlikely to disappear.

Matlab, on the other hand, is a proprietary mathematical programming language controlled by a single company, MathWorks Inc. This makes it harder to share information with colleagues not running the same software.

Academic institutions have a mission to teach technologies that are widely used in the real world so that students have the skills that employers need. More than 90 percent of the programs running on our desktops, from operating systems and e-mail clients to Web browsers and word processors, are written in C or its relative, C++.

Just how widespread is C? There are many measures of programming language popularity.

The Programming Language Popularity Web site (www.langpop.com) and Programming Community Index (www.tiobe.com) are two of the best known and most authoritative statistical sources.

The Programming Community Index has been published monthly since 2001 by Tiobe Co., a software consultant. The June 2009 index indicates that Java, a language used to program Web sites, holds the top spot with a 20.1 percent share of the market. Next comes C at 16.8 percent, followed by C++ with 10.6 percent. Matlab ranks 23rd, with a 0.5 percent share, and Fortran 24th, with a 0.4 percent share. Since Java is based on simplified C language syntax, students with C programming experience can compete for a broad range of jobs. In the Programming Language Popularity Website, C tops the list, while C++ is fourth. Fortran is No. 21 and Matlab is nowhere to be seen.

C, originally invented to write the Unix operating system, runs on all platforms, from microcontrollers and hand-held devices to PCs and supercomputers. It is the language most often used to write operating systems and major applications, as well as modify hardware and software. Users can also translate most other languages into C.

C is especially useful for mechanical engineers because it is the language of choice for hardware interfaces. Often, the only other alternative is tedious, low-level programming using an assembly language. As a hardware interface language, C is used commonly for data acquisition and real-time control of robots and other mechatronic systems. Many hardware and software vendors even provide libraries of C code for hardware interfaces. C is also the most widely used language for programming embedded processors, which control everything from automobiles, industrial machinery, medical equipment, and household appliances to traffic lights, vending machines, cell phones, and toys. Of the 9 billion microprocessors manufactured in 2005, 8.8 billion were embedded into products.

C excels as a model programming language. It does an excellent job of illustrating the underlying working principles of scientific computing and disciplined software development. Students gain valuable knowledge of such fundamental programming concepts as data types, compiling, linking, optimization, and static and dynamically shared libraries. Studying C provides a solid foundation for students who want to learn advanced programming skills, such as object-oriented programming in C++.

Equally important, users can either compile or interpret a C program. Compiling translates a C program into binary code and saves it as an executable file that runs on a specific processor. Interpreting executes C statements one at a time. C interpreters let students execute a single line of statement or a script file composed of several statements without compilation. This provides immediate feedback, and makes them powerful tools for interactive classroom presentations and for learning basic programming concepts. Some C interpreters also contain graphical plotting and advanced numerical computing capabilities typically found in mathematical programming languages, for conveniently solving and visualizing engineering and science problems.

Granted, C has a reputation for being difficult to learn and even more difficult to master. Yet the language is small enough to teach in a one-term course. Once students have learned C, they can pick up other languages and mathematical programming languages without much difficulty because all other modern languages and mathematical programming languages borrowed heavily from C. On the other hand, students who have learned Fortran or Matlab will still find it difficult to learn C by themselves, especially if they want to write serious software programs.

// BILINGUAL ADVANTAGES

Teaching C by itself, however, may not be truly sufficient for today's mechanical engineers. Just as learning foreign languages helps students understand their own language better, learning C with other languages can shed light on the fundamentals of computer programming. Besides, some languages are more suitable for certain applications.

At the University of California, Davis, we teach C with Matlab and provide optional introductory materials for Fortran so students can learn Fortran on their own.

Matlab is an interpreted language, with all the flexibility that implies. It finds many applications in scientific and engineering analysis. Like other commercial mathematical programming packages, such as Mathematica and Maple, Matlab has graphical plotting and advanced numerical features that make it a powerful system for solving and visualizing engineering and science problems.

Matlab's strength is its numerical computing and array processing capability. Most students, however, take introductory programming as freshmen, before they learn linear algebra, so they cannot fully appreciate the language's numerical power. Still, we believe that if we expose them

```
while(fabs(fmid) > tolerance) { /* convergence criterion to terminate loop */ fmid = fmid(xmid);  
/* update the left point */  
/* f(xleft)*f(mid) to determine the updating of left or right point */ while(fabs(fmid) > tolerance) { /* convergen  
in the mid point */ fleft = fmid; /* update f(xleft) */ f(xleft)*f(mid) to determine the upd  
if (fleft*fmid <= 0.0) { /* mid point becomes the right point */
```

to Matlab or similar mathematical computing environments early, students will be able to apply the knowledge to solve complicated engineering problems when they take advanced math and engineering courses.

Although Fortran dates back to the 1950s, it remains one of the primary languages used by professionals in both academia and industry, especially for such computationally intensive programs as computational fluid dynamics. It is one of the best candidates for mechanical engineering students to compare with C.

C99 has features, such as restricted pointers, that enable it to be optimized as efficiently as the equivalent Fortran programs. C99 also supports the complex numbers and variable length arrays found in Fortran that are so useful in engineering and science applications. These changes have led to growing use of C for numerical intensive supercomputing. The latest Fortran 2005 standard added many new features, some of them borrowed from C, though these broader capabilities make the language even more difficult to cover in a single course.

// TEACHING C AND MATLAB

An introductory computer programming course should focus on using computer programs to solve problems. Our goal is to teach students the fundamentals of doing this in C because it is widely used in engineering and because it provides an excellent foundation for solving problems with other programming languages.

Our introductory computer programming course at Davis runs for one quarter, or 10 weeks. The weekly schedule includes three one-hour lectures and a one-hour discussion session. We offer three optional laboratory hours with a teaching assistant in the lab to answer students' questions. Although we provide open lab hours (without a teaching assistant), most students do homework assignments on their own computers using free C/C++ compilers and interpreters with graphical plotting and advanced numerical capabilities running in Windows, Mac OS X, or Linux.

This demands a lot from students. In order to cover adequate material, we rely heavily on information technology for teaching. We use a combination of PowerPoint slides and live interactive code executed in a C/C++ interpreter with graphical plotting and advanced numerical capabilities for classroom presentations. Students learn the fundamentals of disciplined software development in C from examples that range from algorithm development, flow charts, different data types, and structured and modular programming to pseudocode, code reuse, coding style, on-line documentation, and debugging.

Matlab was implemented in C and borrowed heavily from C. Like most modern languages, it shares similar operators, expressions, repetition statements, control structures, arrays, input and output, and functions. Because they start with a solid foundation in C, students quickly learn to use Matlab to solve engineering problems. We demonstrate

the strength and some unique features of Matlab by having students use it to re-solve many of the same problems that they solved earlier in the course in C.

We cover a lot of ground in 10 weeks. That means we must make compromises. Due to the time constraints, we teach students both C and Matlab, and give them handout materials to learn Fortran as a second programming language on their own. Academic institutions that use a 15-week semester system should have enough time to introduce Fortran in class for comparison study with C. They may also have time to cover some of the more difficult C/C++ features.

Since 2004, I have been teaching freshmen how to use C and Matlab to solve engineering problems. Our department expects students who complete the 10-week course not only to solve problems in C, but also to achieve greater proficiency in Matlab than those students who learned only Matlab. We also expect students who would like to work on Fortran projects to pick up that language quickly on their own.

After five years teaching, we believe that we have achieved this goal. Students demonstrate proficiency by applying C and Matlab to projects and homework in subsequent courses. They are also able to port Fortran programs used in an upper division applied aircraft aerodynamics course to C.

They also approach the programming and mechatronics aspects of mechanical engineering differently. Many developed a keen interest in the application of information technology to mechanical devices. Others performed very well in the advanced C++ object-oriented programming course offered primarily to computer science and electrical engineering students.

C is one of the most popular programming languages used in engineering and science. It is the only language for many programming tasks. C is one of the core foundations of information technology. As the pace of integration of electronic controls with mechanical devices increases, it is more important than ever for mechanical engineers to master this fundamental technology.

With a solid foundation in C, mechanical engineering students are well prepared for today's projects, which increasingly integrate mechanical hardware with control software. They have the foundation to learn more advanced and mathematical programming languages, as well as to take advantage of new and emerging computing paradigms. ■

To Learn More

For more detailed information about UC Davis' introductory computer programming course and a review of our teaching resources, visit our Web site at <http://iel.ucdavis.edu/cfores>. The site includes a textbook, detailed lecture notes, PowerPoint lecture and discussion slides, homework and solutions, and handouts for learning Fortran and comparing it with C.