FOURTH ANNUAL LECTURE

Life Experiences in MATHEMATICS

March 21, 2013 • 3:30 p.m.
Sid Richardson Building, SR 344

MICHAEL HOSEA

MathWorks

Dr. Michael Hosea is a numerical specialist with The MathWorks, the company that produces, among other important products, the software packages Matlab and Simulink. Mike graduated with his BS and MS degrees in mathematics from Baylor in, respectively, 1986 and 1988. In 1993, he earned his PhD in Mathematical Sciences from Southern Methodist University under the supervision of well-known numerical analyst Larry Shampine. Mike also earned another MS degree in Operations Research from SMU in 2004.

Prior to his present position at The MathWorks, where he has been since 2004, Mike had considerable experience in both academic and business positions. He was an actuarial assistant at Lewis & Ellis, Inc. From 1988-1992, Mike worked at Texas Instruments developing scheduling algorithms for semiconductor wafer fabs. He was an Assistant Professor of Mathematics at Northern Illinois University from 1993-1996 before returning to Texas Instruments from 1996-2004 as a Software Development Engineer, working on TI calculators and educational software. Mike was awarded two US Patents in connection with this work. He and his wife live in Natick, Massachusetts with their three children.

Closing the Mathematical Software Deployment Gap with Automatic Code Generation

Over three decades ago, Cleve Moler developed the first version of MATLAB to make numerical linear algebra software available to his students without writing FORTRAN driver programs. Since then it has expanded and acquired its own, easy-to-use programming language and is used by scientists and engineers the world over for analysis, design, prototyping, and simulation of systems that rely on sophisticated mathematical software. However, when the ultimate goal is to deploy software developed this way, the properties that make MATLAB so easy to use work against the software developer. There is a divide between the design and simulation environment in which the software has been developed and the target hardware on which it must ultimately run, which could be anything from a small glucose meter to a jumbo jet. Not only must developers convert their code to a more restrictive programming language like C, they must write or obtain source code for sophisticated mathematical functions that MATLAB makes available from its standard library. Rewriting in another programming language is an expensive and error-prone step, often costing hundreds of thousands or even millions of dollars. Automatic code generation is an alternative to rewriting that can take the source code developed in the design and simulation environment and generate C code automatically for the target hardware. We'll look at the mathematician's role in making this possible.

1845

BAYLOR
UNIVERSITY

COLLEGE OF ARTS & SCIENCES