On Computational Electromagnetic Code Testing and Benchmarking
AP-S/USNC-URSI Joint Symposium

Andrew D. Greenwood
Air Force Research Laboratory, Directed Energy Directorate, Kirtland AFB, NM 87117 USA

Code testing and benchmarking is an important subject that is often neglected. High quality benchmarks are difficult to define and generate. Often there are policy, security, and data release issues. Close collaboration between experimental efforts and numerical simulation efforts is needed. In one case, an antenna with existing experimental data was defined as a potential benchmark. Care was taken to correctly define the feed region, but multiple simulations failed to reproduce the experimental return loss. Unfortunately, there was no funding for further experimentation, which limited the utility of this benchmark. Thus, quality benchmarks require teamwork and commitment from experimental and computation efforts.

To ensure quality simulation results, computational codes also require continuous testing. Multiple levels of software testing are defined (see http://istqbexamcertification.com/what-are-software-testing-levels). Electromagnetic benchmarks are useful for the top levels of testing (system testing, acceptance testing) where the complete application is exercised. Lower levels of testing (unit testing, component testing, integration testing) are also important. Robust testing at all levels helps to pinpoint errors within a large simulation code. Further, at the top levels of testing, analytic results are valuable. The claim is sometimes made that geometries with analytic results are too simple for robust code testing. However, computations on smooth geometries such as spheres ensure that boundary conditions and code interfaces are working properly. If the code accurately matches analytic results but fails to produce acceptable results for more complex geometries, then many basic code functions are eliminated as the cause.

The presentation will discuss electromagnetic benchmarks and challenges in the development of new benchmarks. Robust software testing at multiple levels will also be discussed, as well as use of continuous, automated testing.