

# Preface

Simulation has gradually become the preferred method for designing complex systems. The aerospace industry provided the impetus for this design approach because most aerospace systems are at best difficult or at the worst impossible to test. I started my career working on the Lunar Module flight control system. Testing this system in space was extremely expensive and very difficult. In the entire development cycle, in fact, there was only one unmanned test in earth orbit. All of the designs, by necessity, required simulations. Over time, this approach has become pervasive in many industries, particularly the automotive industry. Even in product developments as mundane (seemingly) as disk drives for computers, simulation has become the dominant method of design.

A digital simulation of the environment that a system “lives” in is crucial in this modern simulation based design process. The accurate representation of this environment allows the designer to see how well the system he is designing performs. It allows the designer to verify that the design meets all of the performance specifications. Moreover, as the simulation develops the designer always can see how the performance requirements pass through to the various subsystems in the design. This refinement of the specification allows the model to, in essence, become the specifications. The simulation environment can also allow the designer of embedded digital signal processing and control algorithms directly to test the code, as it is developed, to insure that it satisfies the specification. The last, and newest, step in this process is the ability to use the simulation model automatically to generate the computer code for computers in the system. This can eliminate most of the “hand coding” that is required by the current practice (in most companies this process consists of handing a stack of papers with the written specification to a team of computer programmers for coding). This approach eliminates two potential sources of error: the errors introduced by the conversion of the design into a written specification, and the errors introduced by the manual coding.

Simulink<sup>®</sup> is a remarkable tool that fills the niche for a robust, accurate, and easily used simulation tool. It provides a visual way of constructing a simulation of a complex system that evolves in time (the designer can optimize the system by optimizing the mathematical model’s response). It allows rapid and accurate simulation as various numerical constants in the model are changed. It allows the user to tune these constants so that the simulation accurately portrays the real world. It then allows sharing of the model among many different users so they can begin to design various components or parts of the system. At the same time, it provides a way for understanding design problems whose solutions will make the system operate in a desired way. Finally, the resulting design’s embedded software is rapidly convertible into embedded code that is directly usable for testing in a stand-alone computer.

All of this can be accomplished using one easily understood graphical environment. A major thesis of this book is that Simulink can form the basis for an integrated design environment. As your use of the tool evolves, the way this can happen should be more apparent.

Because simulation is such an important adjunct to numerical analysis, the book can be a supplement for a course on numerical computing. It fits very well with the previous book in this series, “Numerical Computing with MATLAB” by Cleve Moler, [29] (we refer to this book so frequently that its title is abbreviated NCM in the text) and it has been written with the same eclectic mix of practice and mathematics that makes Cleve’s book so much fun to read. I hope that this work favorably compares with – and comes close to matching the quality of – Cleve’s work. As is the case with “Numerical Computing with MATLAB”, this book is based on a collection of 88 model files that are in the library Numerical Computing with Simulink (we abbreviate this as NCS) that can be downloaded from The MathWorks web page ([www.mathworks.com](http://www.mathworks.com)). This library of SIMULINK<sup>®</sup> models and MATLAB<sup>®</sup> M-files forms an essential part of the book, and through extensions that the reader is encouraged to peruse, an essential part of the exercises.

The book will take you on a tour of the Simulink environment, showing you how to develop a system model and then how to execute the design steps to make the model into a functioning design laboratory. Along the way, you will be introduced to the mathematics of systems, including difference equations and z transforms, ordinary differential equations (both linear and nonlinear) and Laplace transforms, numerical methods for solving differential equations, and methods for simulating complex systems from several different disciplines.

The mathematics of simulation is not complete without a discussion of random variables and random processes for doing Monte Carlo simulations. Toward this end, we introduce and develop the techniques for modeling random processes with predetermined statistical properties. The mathematics for this type of simulation begins with “white noise”, which is a very difficult entity to pin down. We introduce the concept of Brownian motion, and show the connection between this process and the fictitious, but useful, white noise process. The simulation of the Brownian process in Simulink is developed; it is then used to form more complex (statistically) processes. We review and show how to simulate and analyze random processes, and we formulate and use the “power spectral density” of a process.

We introduce other tools, in addition to Simulink, from The MathWorks. Each of the tools is an expansion into a different “domain of knowledge”. The first tool is the Signal Processing Blockset that extends Simulink into the domain of signal processing (both analog and digital). The second tool, Stateflow<sup>®</sup>, expands Simulink to include state charts and signal flow for modeling event driven systems (i.e. systems where the actions start at times that are asynchronous with the computers clock). This tool naturally extends Simulink into the traditional realm of “Finite State Machines”.

The third tool, “SimPowerSystems”, extends Simulink into the realm of “Physical Modeling”, and in particular into the realm of electrical circuits including Power Systems, motor drives, power generation equipment including power electronics, and 3-phase power transmission lines.

The last tool, “SimMechanics”, is also a Physical Modeling tool; it develops models of mechanical systems such as robots, linkages, etc. These tools can all work together in

the common Simulink environment, but they all have their own method for displaying a picture of the underlying system.

Both MATLAB and Simulink are available in a student version. All of the examples in this book use the tools in the student version supplemented with the “Control System Toolbox”, the “Signal Processing Blockset”, “SimMechanics”, and “SimPowerSystems”, that can be purchased separately. If you are not a student, but wish to learn how to use Simulink using this book, the MathWorks can provide a demonstration copy of MATLAB, Simulink and the add-ons – contact them at <http://www.mathworks.com>.

This book not only introduces Simulink, and discusses many useful “tricks” to help develop models, but it also shows how it is possible for Simulink to be the basis of a complete design process.

I hope the material is not too theoretical and that you find the models we will work with are fun, instructive, and useful in the future.

As is always the case, innumerable people have contributed to the genesis and development of this book. I thank Cleve Moler for sharing an early manuscript of his wonderful book and I thank The MathWorks for many years of interesting and stimulating work. In particular, I wish to thank the MathWorks staff with whom I have had many long and interesting discussions: Rob Aberg, Paul Barnard, Jason Ghidella, Ned Gully, Loren Shure, Mark Ullman, and The MathWorks CEO, Jack Little. These interesting and productive discussions helped me create the various examples and models described in this book.

Words are never enough to thank ones family. My wife, and soul mate, Jean Frova Gran, provides support, joy, affection and love. My daughter Kathleen keeps my thoughts focused on the future. Her hard work and dedication to the achievement of her goals, despite many small business frustrations, is laudable.

My extended family includes two step-grandchildren, Ian and Ila, who provide me with insight into the joy brought by the process of learning about this unbelievable world we live in. My stepdaughter Elizabeth has a sense of humor and a joie de vivre that are inspirational.

Lastly, I need to say “thank you” to my sister, Cora and her wonderful husband Arthur who have always been loving and supportive in more ways than can be stated; Cora is, as the self-proclaimed “matriarch” of the Gran’s, much more to me than a mere sister.

Dr. Richard J. Gran  
Mathematical Analysis Co.  
[www.math-analysis.com](http://www.math-analysis.com)  
Norfolk, Massachusetts, August 2006