GoldSim:
Using Simulation to Move Beyond the Limitations of Spreadsheet Models

White Paper

Abstract

While spreadsheets are appropriate for many types of applications, due to a number of inherent limitations they are poor choices for some important applications (e.g., building realistic dynamic models). Despite this fact, they continue to be used for modeling applications where much better options are available. This white paper discusses the limitations of spreadsheets for complex modeling applications, and introduces GoldSim, a graphical dynamic simulation program that moves beyond spreadsheets and makes it easy to build and maintain dynamic models of complex business and engineering systems.

Contents

Introduction

After email, the spreadsheet is probably the most widely used business software application in the world. Originally developed for accounting applications and for creating simple databases of information, spreadsheets are now used for a wide range of tasks, including quantitative business modeling and engineering analysis. To facilitate this, a variety of spreadsheet add-ins have been developed over the years to augment the capabilities of spreadsheets (e.g., to facilitate Monte Carlo simulation).

While spreadsheets and their add-ins are appropriate for some types of applications (e.g. managing data, building relatively simple models), due to a number of inherent limitations, they are quite poor choices for other applications (e.g., building realistic dynamic models). This white paper discusses the limitations of spreadsheets for complex modeling applications, and introduces GoldSim, a stand-alone graphical dynamic simulation program that moves beyond spreadsheets and makes it easy to build and maintain dynamic models of complex business and engineering systems.

The paper discusses how GoldSim addresses the major weaknesses inherent in spreadsheet modeling, and describes how you can take advantage of the best attributes of spreadsheets while using GoldSim as a quantitative modeling platform.

The Advantages and Limitations of Spreadsheets

Spreadsheets are very powerful when the calculations being carried out lend themselves to the electronic grid format (rows and columns represented by numbers and letters). This includes, for example, calculations where several "output" spreadsheet cells are updated based on a number of "input" cells, or where you are managing a list of entries (effectively using the spreadsheet as a database). For these types of applications, spreadsheets are advantageous because they are easy to learn to use, and you can quickly create, edit and view your data.

Another advantage of spreadsheets (as well as the key reason that they are often applied to problems where better solutions exist) is that they are ubiquitous, and from the point of view of many users, freely available. That is, because most business computers come equipped with at least a basic version of a spreadsheet tool, from the point of view of the user there is no perceived cost. However, as will be discussed below, for complex modeling applications this is an erroneous conclusion. The cost of the extra time required to create, debug and modify complex models, as well as the costs associated with making decisions based on models that are poor representations of reality, can make the inappropriate use of spreadsheets costly indeed.

So what are the limitations of spreadsheets for complex modeling applications? In broad terms, the major limitations are outlined below:

Complex spreadsheet models are generally not transparent and can be very difficult to explain to others. Because of the row and column paradigm used by

spreadsheets, the fact that equations are written in terms of cell references, the invisibility of the dependencies between cells, and the lack of a graphical means to explain the model, most spreadsheet models have a low level of transparency when viewed by anyone other than the author of the model. It can even be difficult to understand your own model, particularly if you have not looked at it for some period of time.

Complex spreadsheets are prone to errors. Due to the lack of transparency of complex spreadsheet models, not only can it be difficult to communicate your model to others, but it can also be difficult to check for errors. As a result, several studies have shown that complex spreadsheets have a very high incidence of errors (e.g., Panko, 2005[1]).

Spreadsheets are not well suited to representing dynamic systems. In a spreadsheet, you typically deal with dynamics by adding a row (or column) for each timestep (i.e., each day, each quarter, each month) that you want to forecast a value for. In addition to being an awkward and error-prone way to represent dynamics, this has a number of disadvantages: 1) it is difficult to represent dynamic feedback loops and delays, where a change made to one part of the system has a delayed impact; 2) sudden events (e.g., a bank deposit or withdrawal, an interest rate change, a storm) are difficult to represent accurately; 3) changes in the system's structure with time are hard to represent (e.g., taking out a loan when required), and 4) the length of the timestep cannot be dynamically adjusted during a simulation (e.g., in response to changing conditions).

Spreadsheets have no ability to handle dimensions and units. Because spreadsheets deal only in numbers, and cannot represent units, great care must be taken when building models to handle unit conversions. Unit conversions and inconsistencies are a common source of errors in spreadsheet models.

Spreadsheets do not inherently handle uncertainty and randomness. For most real-world systems, at least some of the controlling parameters, processes and events are often random, uncertain and/or poorly understood. The objective of many models is to identify and quantify the risks associated with a particular option, plan or design. Modeling a system in the face of such uncertainty and computing such risks requires that the uncertainties be quantitatively included in the calculations. Due to the importance of this particular spreadsheet limitation, a number of third-party add-ins have been developed to address this issue.

In the following sections, we introduce the GoldSim simulation framework, and then describe how GoldSim moves beyond spreadsheets to address these issues.

What is GoldSim?

GoldSim is a powerful and flexible platform for visualizing and numerically simulating nearly any kind of physical, financial or organizational system. In a sense, GoldSim is like a "visual spreadsheet" that allows you to visually create and manipulate data and equations. Unlike spreadsheets, however, GoldSim allows you to readily evaluate how systems evolve over time, and predict their future behavior.

[1] Panko, Raymond, "What We Know About Spreadsheet Errors", University of Hawai'i, January 2005. (http://panko.shidler.hawaii.edu/SSR/Mypapers/whatknow.htm)

That is, rather than using rows or columns of items to represent time-varying parameters, the concept of time is directly built into GoldSim. Hence, to simulate a system that changes with time, you simply describe the equations and rules that control the system's behavior, and GoldSim then automatically "steps through time" to *simulate* (predict) the future behavior of the system.

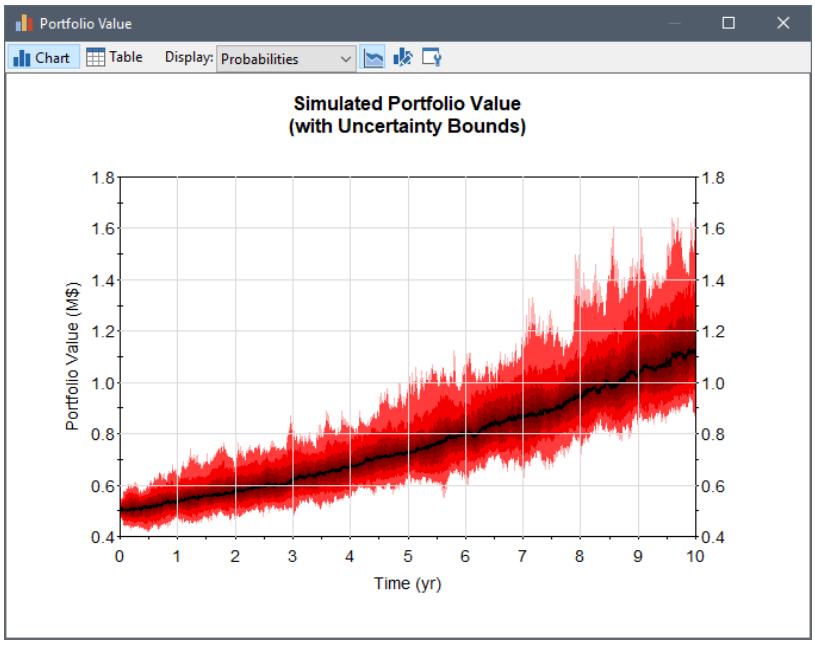


Figure 1. GoldSim output showing the evolution of a variable as a function of time.

The GoldSim simulation environment is highly graphical and completely ***object-oriented***.

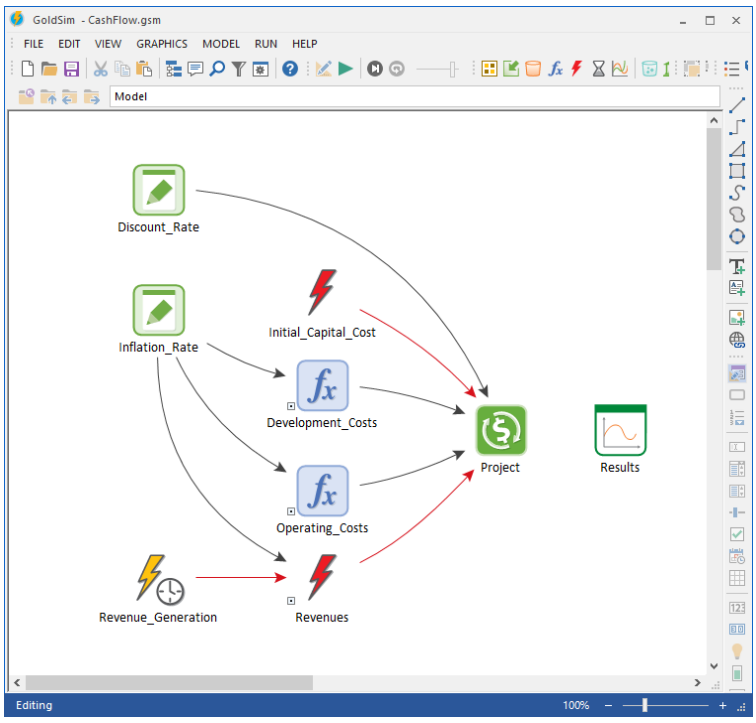


Figure 2. GoldSim models consist of graphical depictions of the equations.

That is, you create, document, and present models by creating and manipulating graphical objects representing data and relationships between the data. Based on how the various objects in your model are related, GoldSim automatically indicates their influences and interdependencies by visually connecting them in an appropriate manner.

In addition to the graphical depiction of the model structure, the following combination of features makes the GoldSim approach particularly unique and powerful:

GoldSim is very flexible, allowing it to be applied to nearly any kind of system. Like a spreadsheet, GoldSim is a generic simulation framework. As such, it can be (and has been) applied to a wide variety of systems, ranging from engineering to manufacturing to finance. As a result, the same tool can be used to simultaneously model nearly any aspect of a system (e.g., engineering, financial, organizational).

GoldSim supports creation of hierarchical, modular models, and this facilitates the reuse and sharing of models across an organization. GoldSim models are built in a hierarchical and modular manner, by creating and linking subsystems together.

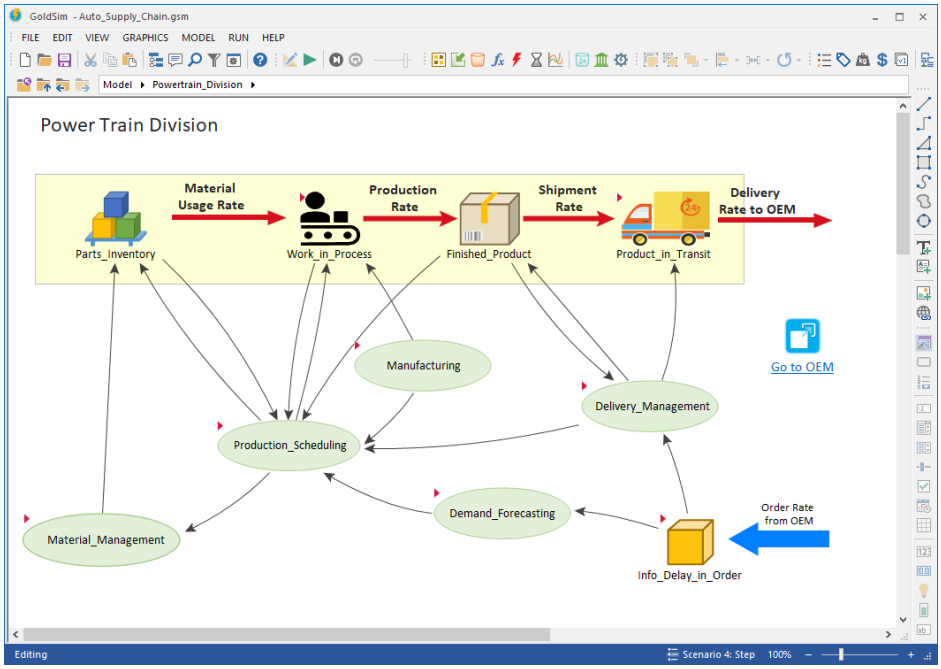


Figure 3. A typical GoldSim model with multiple subsystems.

These subsystems, after being built for one application or project, are often readily transferable with only minor modifications to another application. Sharing and re-using submodels in this manner can result in significant cost savings by eliminating the need to "reinvent the wheel". In effect, GoldSim acts as a framework to share knowledge and experience across the organization. Not only does this reduce redundant efforts, it promotes consistency in the assumptions and approach to quantitative modeling within an organization.

Uncertainty in processes, parameters and future events can be explicitly represented. Uncertainty in processes and parameters can be represented by specifying model inputs as probability distributions. This capability also makes it easy to represent stochastic processes (e.g., precipitation, demand, interest rates) that vary with time but can only be described statistically. The impact of sudden, uncertain events (e.g., accidents, storms, political changes) can also be represented by specifying the occurrence rates and consequences of such "disruptive events". GoldSim uses Monte Carlo simulation to propagate uncertainty through the model.

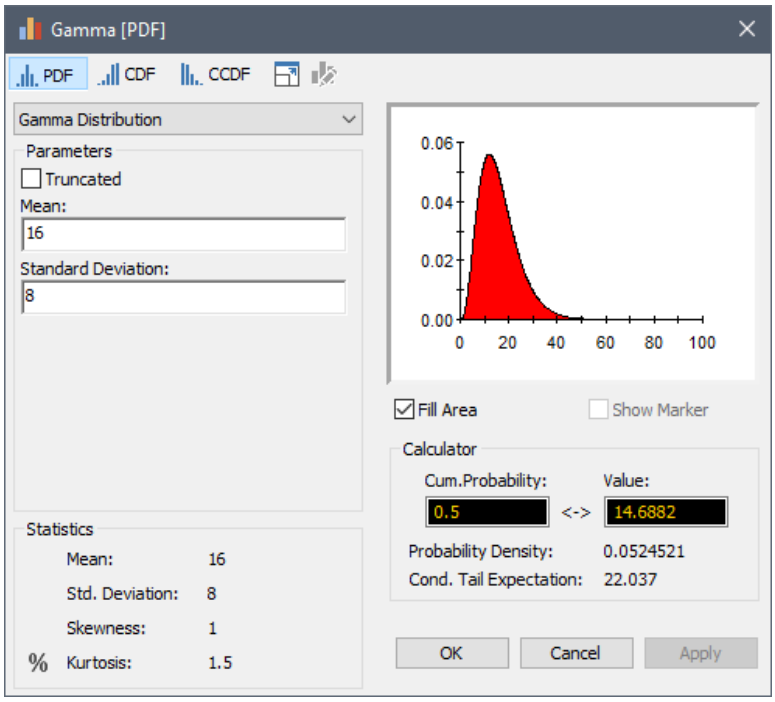
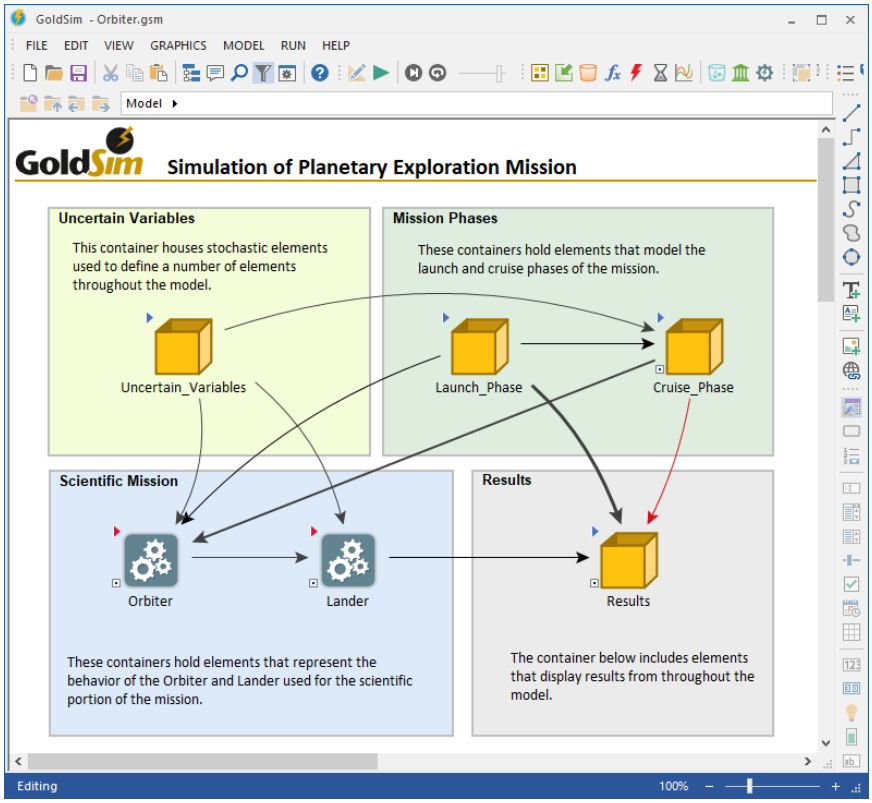


Figure 4. Input dialog for defining a probability distribution.

GoldSim is highly extensible. GoldSim provides a wide variety of built-in objects ("elements") from which you can construct your models, and, if desired, you can program your own custom objects, and link them seamlessly into the GoldSim framework. In addition, GoldSim was specifically designed to support the addition of customized modules (program extensions) to address specialized applications (e.g., financial modeling, risk and reliability analysis).

GoldSim allows you to create compelling presentations of your model, and therefore facilitates effective interaction with stakeholders. A model that cannot be easily explained is a model that will not be used or believed. GoldSim was specifically designed to allow you to effectively document, explain and present your model. You can add graphics, explanatory text, notes and hyperlinks to your model, and organize it in a hierarchical manner such that it can be presented at an appropriate level of detail to multiple target audiences.

Figure 5. Graphics, notes and other documentation make GoldSim models highly transparent.



The ability to create hierarchical, top-down models, coupled with GoldSim's powerful documentation features, allows you to design transparent, highly-graphical models that can be effectively explained to any audience at an appropriate level of detail.

GoldSim provides a specialized set of authoring tools that allow you to create custom graphical user interfaces, or "dashboards" for your models to make them accessible to non-technical users. Models created using the GoldSim authoring tools can be saved and subsequently viewed and run using the free GoldSim Player. The interfaces can be designed to include buttons, input fields, sliders and result displays, and the author can embed text, tool-tips and graphics to provide instructions on the use of the model. Such an interface allows a model to be easily used by someone without requiring them to be familiar with either the GoldSim modeling environment or the details of the specific model. In effect, this allows you to use GoldSim as a high-level programming language to create custom applications for distribution to end users who may not necessarily be modelers.

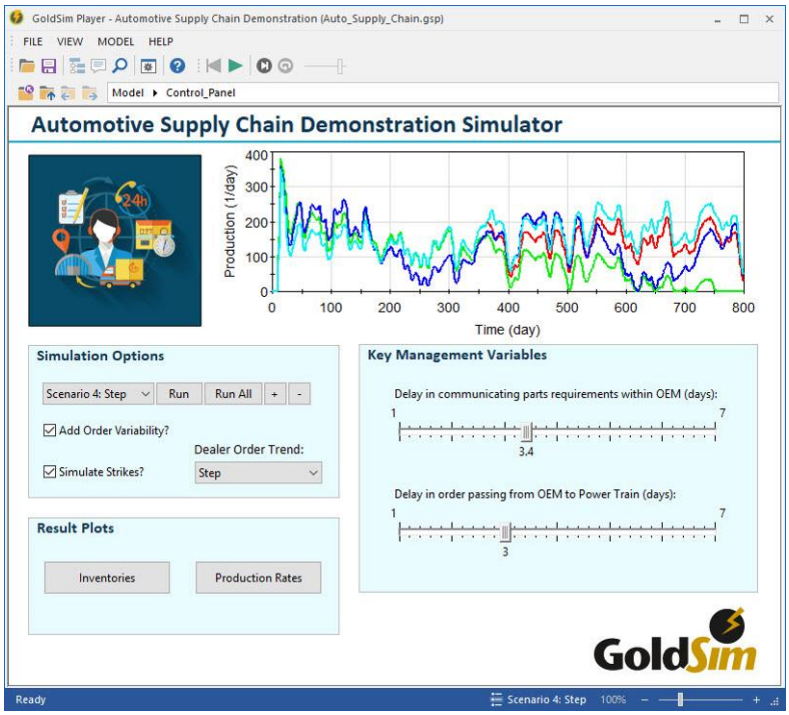


Figure 6. GoldSim dashboards make your models accessible to non-technical users.

How Does GoldSim Address the Limitations of Spreadsheets?

Given the brief overview presented in the previous section, let's now revisit the primary limitations of spreadsheets, and discuss how GoldSim addresses them.

Complex spreadsheets are generally not transparent.

Because of the row and column paradigm used by spreadsheets, the fact that equations are written in terms of cell references, the invisibility of the dependencies between cells, and the lack of a graphical means to document the model, most spreadsheet models have a low level of transparency.

Building hierarchical models

GoldSim's interface, which allows you to build hierarchical, graphical representations of your system (in terms of influence diagrams) was specifically designed to facilitate the construction of transparent, well-documented models that can be easily explained to others. The manner in which GoldSim can be used to create hierarchical, graphical, well-documented models is illustrated in Figures 2, 3 and 5. You build a model in an intuitive manner by literally drawing a picture (an influence diagram) of your system. In a sense, GoldSim is like a "visual spreadsheet" that allows you to graphically create and manipulate data and equations.

Referencing meaningful names instead of row and column labels

One of the major weaknesses of spreadsheets is that equations are written in terms of cell addresses (e.g., B1 + AA5). Even if you use named ranges in a spreadsheet, inherent design limitations make it virtually impossible to use this feature for all cells.

In GoldSim, all objects are, by definition, named and referenced as such in equations.

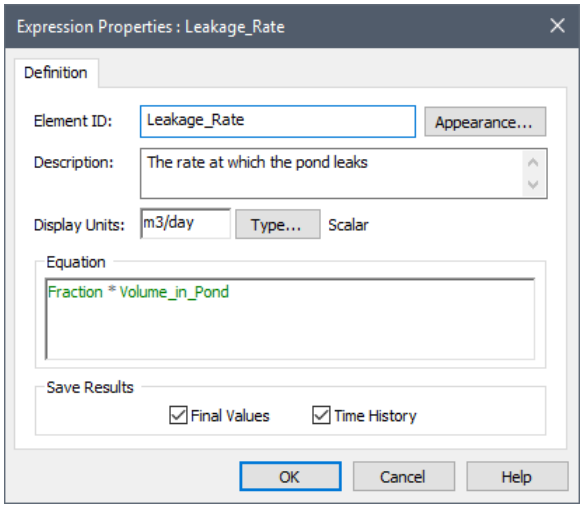


Figure 7. GoldSim object representing a simple function of other model variables.

Referencing another variable in the model is as simple as right-clicking within an edit field, which results in the display of a hierarchical browser for selecting the variable:

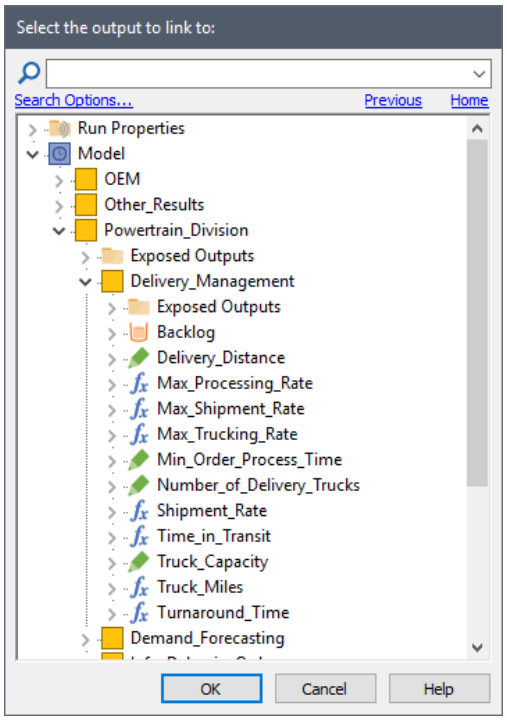


Figure 8. Linking to other GoldSim variables can be done via a hierarchical browser.

Of course, if you decide to change the name of a variable, GoldSim automatically updates all the other objects that reference that variable.

GoldSim objects have specific functions

In a spreadsheet, one cell is indistinguishable from another, even if one represents input data, and another represents a formula or equation. In contrast,

within a GoldSim model the various objects all have specific functions or roles. For example, GoldSim provides one object type to define input data, another to define probability distributions, and another to define quantities that accumulate (such as a bank account). In fact, GoldSim provides over 40 such objects:

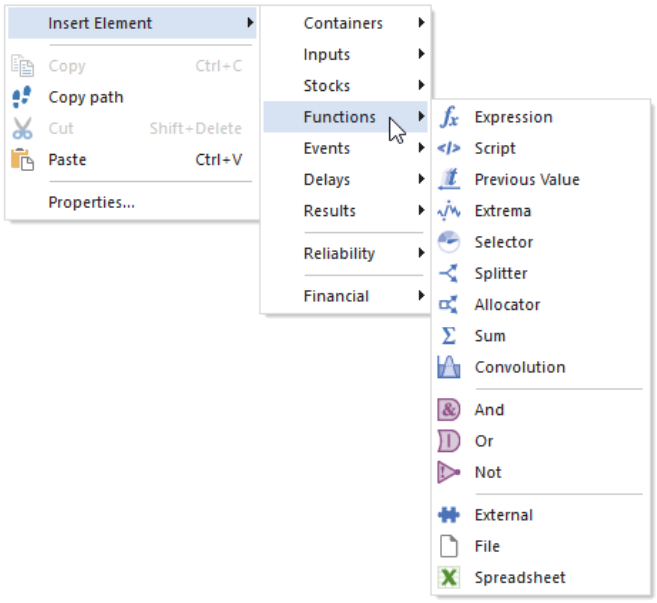


Figure 9. Menu showing one category of objects that can be inserted into a GoldSim model.

This makes models more transparent because the objects themselves provide visual cues as to their function and role in the model.

Spreadsheets are prone to errors.

The general lack of transparency of complex spreadsheet models not only makes it difficult to communicate your model to others, but can also make it difficult to detect errors. In addition, the inherent row and column structure upon which the spreadsheet is based makes spreadsheet models highly susceptible to errors.

Viewing relationships between variables

GoldSim's hierarchical, graphical approach makes identifying and correcting errors much easier than in spreadsheet. Not only are dependencies between variables explicitly illustrated (in the form of influence diagrams), but GoldSim also provides powerful tools for exploring how each variable impacts others.

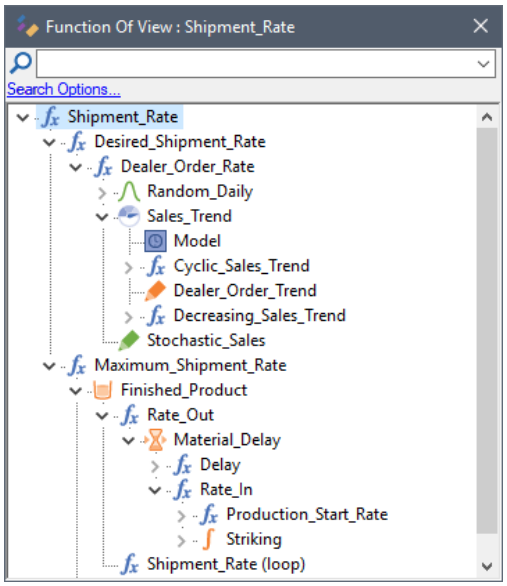


Figure 10. Dialog showing the variables that a selected variable depends upon.

In this case, the dialog displays all of the variables that *Shipment_Rate* is directly or indirectly a function of. A similar dialog can show what variables are affected (directly and indirectly) by *Shipment_Rate*.

Handling arrays of variables consistently

Spreadsheet models consist of rows or columns of cells, where typically all cells in a particular row or column have the same formula (e.g., referencing a cell in an adjacent column). This is probably one of the single greatest sources of errors in spreadsheet models. If the formula changes, you must remember to copy the new formula to all cells. If someone inadvertently changes one cell (out of hundreds in a column), it is likely that the error will go undiscovered. Auditing and reviewing a model to ensure that hundreds or thousands of cells all have the same formula is difficult.

This problem arises because each row or column in the spreadsheet is intended to represent what is actually one item in an array of data (e.g., the profit for each of 20 different stores). In a spreadsheet, the equation must exist separately for each item. In GoldSim, the equation would be written once, and would act on all items of the array.

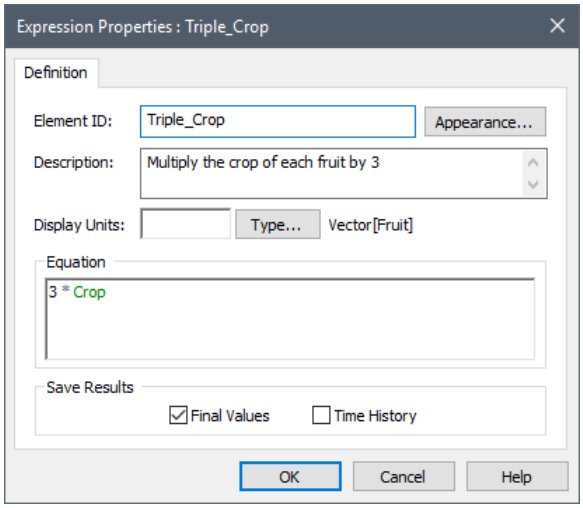


Figure 11. *Triple_Crop* and *Crop* represent arrays (of fruit). That is, they consist of multiple items. In this simple example, every item of *Crop* is multiplied by 3 to produce a new array called *Triple_Crop*.

Spreadsheets are not well suited to representing dynamic systems.

In a spreadsheet, you typically deal with dynamics (i.e., systems that change with time) by adding a row (or column) for each timestep (i.e., each day, each quarter, each month) that you want to forecast a value for.

There are a number of shortcomings to this approach. The most important is directly related to the issue discussed in the previous section. Spreadsheets represent different timesteps in a model using multiple cells. Hence, if you have an equation that is a function of time, it must be copied into the spreadsheet separately for each timestep. If you have 1000 timesteps, you need 1000 cells (any of which could inadvertently be modified and made inconsistent with the others).

GoldSim is a dynamic simulator. As such, you directly specify the duration of your simulation, as well as the length of the timestep.

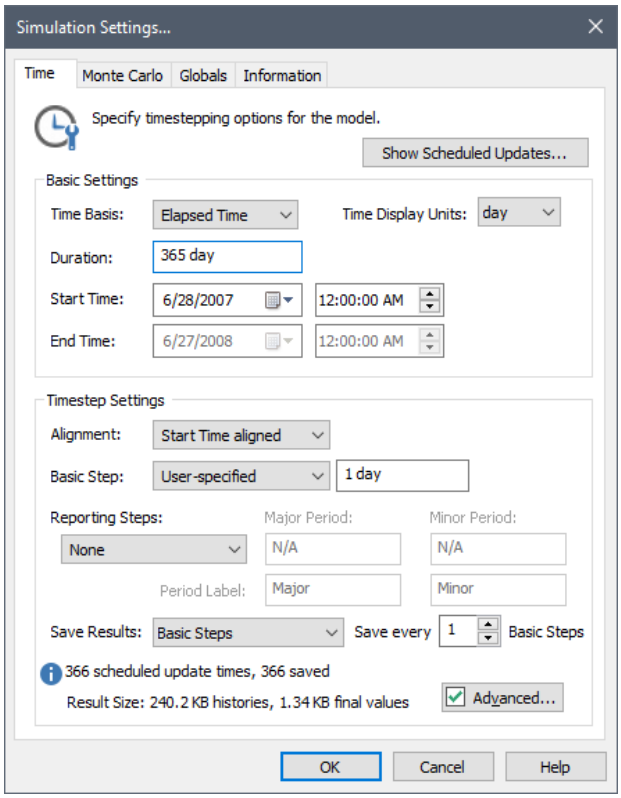


Figure 12. In GoldSim, the duration and timestep length for your model are specified directly.

As a result, you can directly specify variables by referencing the Elapsed Time or the simulated time (i.e., date) in a simulation.

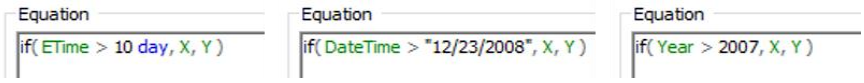


Figure 13. Expressions can directly reference the simulation time in multiple ways.

Moreover, a number of objects in GoldSim inherently incorporate time into their calculations. For example, a built-in object called an Integrator numerically integrates a specified rate of change over the duration of the simulation.

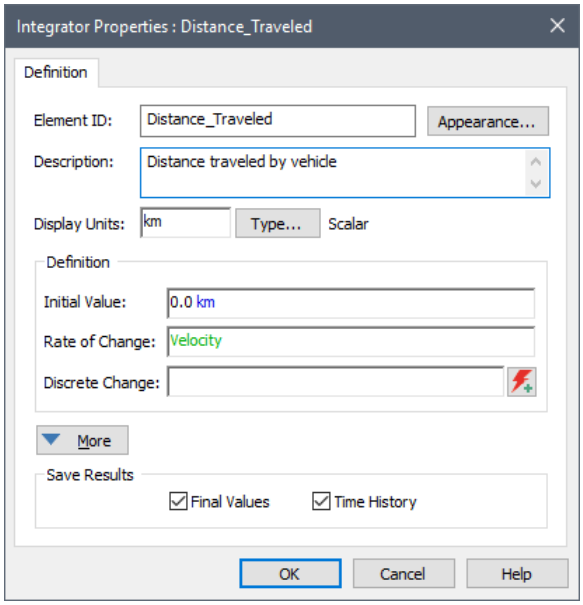


Figure 14. The Integrator requires an Initial Value and a Rate of Change, and outputs the integral as a function of time.

Because GoldSim explicitly represents time in this way, it can represent complex dynamic systems that are difficult or impossible to accurately represent in a spreadsheet. For example,

- In a spreadsheet, it is difficult to represent dynamic feedback loops and delays, where a change made to one part of the system has a delayed impact and/or may recursively impact itself. GoldSim was specifically designed to handle such dynamics.

- Sudden events (e.g., a bank deposit or withdrawal, an interest rate change, storms) are difficult to accurately represent in a spreadsheet. GoldSim provides a suite of specialized objects for representing discrete dynamics and stochastic processes.

- In a spreadsheet, the length of the timestep cannot be dynamically adjusted during a simulation (e.g., in response to changing conditions). GoldSim allows you to dynamically adjust the timestep in response to changing conditions.

Spreadsheets have no ability to handle dimensions and units.

Because spreadsheets deal only in numbers, and cannot represent units, great care must be taken when building models to handle unit conversions. In fact, this is probably one of the most common causes of (undetected) errors in spreadsheet models.

GoldSim understands dimensions and units, preventing you from constructing dimensionally-inconsistent models. All variables are assigned units (and hence dimensions), and GoldSim automatically converts between units having the

same dimension, and ensures dimensional consistency (and displays an error message if there are dimensional inconsistencies).

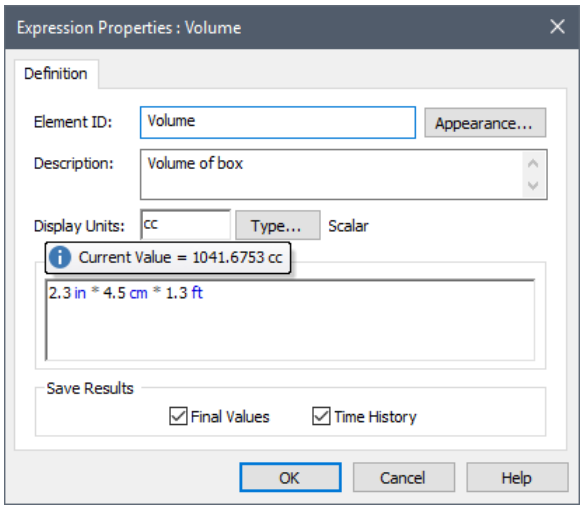


Figure 15. GoldSim understands units and ensures dimensional consistency.

In order to carry out unit conversions and ensure dimensional consistency, GoldSim has a large built-in database of units and conversion factors. You can even define your own custom units.

Spreadsheets do not inherently handle uncertainty and randomness.

For most real-world systems, at least some of the controlling parameters, processes and events are often random, uncertain and/or poorly understood. Modeling a system in the face of such uncertainty and computing the ensuing risks requires that the uncertainties be quantitatively included in the calculations.

Even though this particular limitation can be largely addressed via third-party add-ins that add this functionality to spreadsheets, because spreadsheets cannot easily represent dynamic systems, it is difficult to represent random and/or stochastic processes (processes that vary in time randomly, but can be described statistically). GoldSim was specifically designed to simulate such systems.

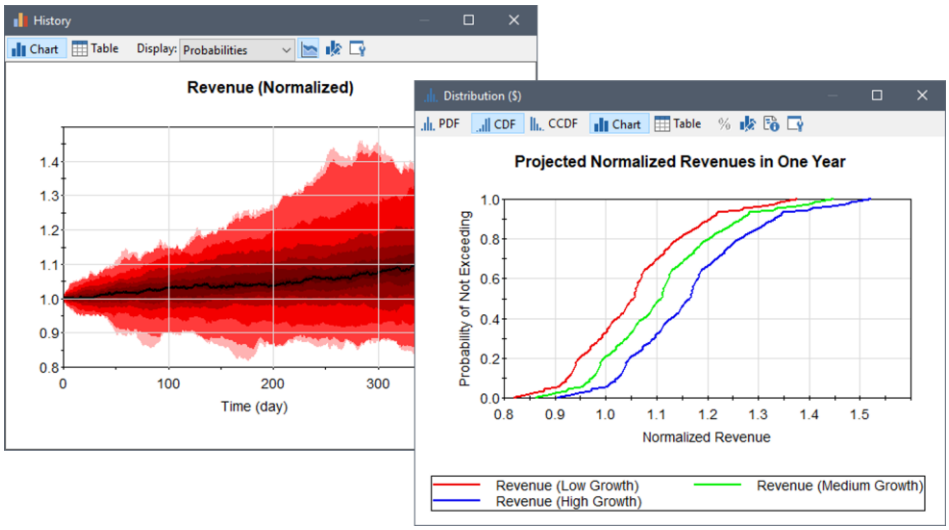


Figure 16. GoldSim displays probabilistic outputs both in terms of probability distributions and by displaying multiple possible "futures".

In addition to allowing you to represent any input parameter as a probability distribution, GoldSim allows you to easily specify stochastic variables, as well as random events (e.g., accidents, storms), that could suddenly alter the behavior of the system.

Linking Spreadsheets to GoldSim

As pointed out previously, although spreadsheets are not well-suited for complex modeling tasks, they are well-suited for some tasks, particularly managing a list of entries (effectively using the spreadsheet as a database). As a result, although it may be inappropriate to do all your calculations in a spreadsheet, a spreadsheet may in fact be an excellent place to store your input data (e.g., a long time series of historical data). To facilitate this, GoldSim allows you to dynamically link a spreadsheet directly into your model. For example, you can import individual cells, or a list of time series data from a spreadsheet into GoldSim. You can also readily export GoldSim results to a spreadsheet (to facilitate post processing or importing into another program).

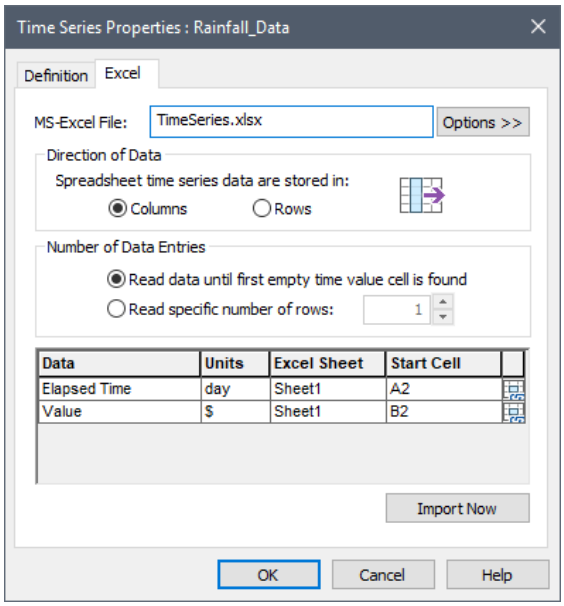


Figure 17. This Time Series in GoldSim imports its data from a spreadsheet.

You can even dynamically send data from GoldSim to a spreadsheet, force the spreadsheet to recalculate, and then retrieve (updated) data from the spreadsheet back into GoldSim during a simulation (e.g., every timestep).

These capabilities allow you to take advantage of the best features of spreadsheets while using GoldSim as your modeling framework.

Summary

After email, the spreadsheet is probably the most widely used business application in the world. While spreadsheets and their add-ins are appropriate for some types of applications (e.g. managing lists, building relatively simple static models), they have a number of inherent limitations:

- Spreadsheet models are not transparent;
- Spreadsheet models are error-prone;
- Spreadsheets are not well-suite to representing dynamic systems;
- Spreadsheets cannot handle units or dimensions; and
- Spreadsheets do not inherently handle uncertainty and randomness.

GoldSim, a graphical dynamic simulation program, was specifically designed to address these weaknesses. It moves beyond spreadsheets and makes it easy to build and maintain dynamic models of complex business and engineering systems.

About the GoldSim Technology Group

The GoldSim Technology Group is dedicated to delivering software and services to help people understand complex systems and make better decisions. The GoldSim Technology Group focuses on building great simulation software and supporting the technical aspects of building effective GoldSim models. To provide other dimensions of complete solutions, we maintain close relationships with partners around the world, including consulting firms with specific industry expertise.

For additional information, please contact:

GoldSim Technology Group LLC
Email: software@goldsim.com
http://www.goldsim.com/