GPS MATLAB Toolbox Review

Avram K. Tetewsky
Arnold Soltz

Draper Laboratory

is coordinated by Richard Langley of the Department of Geodesy and Geomatics Engineering at the University of New Brunswick, who appreciates receiving your comments as well as topic suggestions for future columns. To contact him, see the "Columnists" section on page 4 of this issue.

Simulate, as defined by the Concise Oxford Dictionary of Current English*, means to "imitate conditions of (situation etc.) with model, for convenience or training." Very often in the fields of science and engineering, we need to simulate a situation — just as the definition indicates — before it occurs to help us design or understand a system or its components. So it is with GPS.*

We can carry out GPS simulations using either hardware — which we briefly examined in a previous column — or software, which we'll take a look at this month. Our discussion will take the form of a review of four GPS simulation packages that use the popular and versatile MATLAB programming language. Our reviewers are Avram K. Tetewsky and Arnold Soltz from the Draper Laboratory in Cambridge, Massachusetts.

Tetewsky is a staff analyst in Draper Laboratory's Electronic Design and Sensor Development Directorate. His current work involves GPS receivers and GPS simulators. He holds a B.S. from Rensselaer Polytechnic Institute and S.M. and E.E. degrees from the Massachusetts Institute of Technology.

Soltz received a B.A. in mathematics from Johns Hopkins University in 1964 and an M.S. in mathematics from Northeastern University in 1969. He has designed, implemented, and tested real-time navigation and guidance systems at Draper Laboratory for more than 30 years. His most recent work includes a detailed MATLAB simulation of inertial and GPS sensors on a rapidly spinning vehicle.

"Innovation" is a regular column featuring discussions about recent advances in GPS technology and its applications as well as the fundamentals of GPS positioning. The column

Suppose your boss asked you: "Will there be enough satellites visible on the date of our big test to ensure a successful demo?" or "Are the receiver autonomous integrity monitoring (RAIM) thresholds going to work?" or "Can you integrate this GPS receiver into our ship navigation system and tune the latency parameters to handle 'short' outages in differential GPS (DGPS) service?" or "How will interference from our new paging system affect GPS reception?" or "Can you simulate how our new GPS product is going to perform?" What do you tell her?

As engineers design GPS into more products, the demand for detailed GPS performance and integrity simulation will likely increase. Precise and repeatable "hardware" navigation simulations use a GPS simulator — a tool that generates actual radio-frequency (RF) signals that can fool a GPS receiver into thinking it is moving with a particular velocity at any location in the world. Such a system typically costs from $50,000 to $300,000. With a good GPS simulator, you can control the truth models, inject known tropospheric, ionospheric, and even selective availability/antispoofing (SA/AS) errors, and then assess their impact.

If you can't afford a GPS simulator, or if that resource is already overcommitted, or if your receiver does not already exist, you could either use "canned" data collected from known locations or write your own simulation software. Unless you need to perform jamming or intricate multipath analysis, the detailed signal processing of code correlation and Doppler removal (also called Doppler wipe off or phase rotation) is not important; you need only create measurement data with known errors and additive thermal noise, figure out which signals are visible at your loca-

tion, and then prototype the back-end navigation algorithms. After that, you merely pump measurements into your new navigation algorithms and compare the navigation fixes with the truth models.

The hurdles along this route are choosing a programming language, and writing, testing, validating, and maintaining your own software library. But just as the revolution in microcomputing hardware and new paradigms in such software as spreadsheets and Visual BASIC has made it possible for more people to create business applications, a similar revolution has been taking place with software targeted at scientists and engineers.

Because GPS simulation requires a broad spectrum of tools covering matrix algebra, digital signal processing, control theory, and navigation algorithms, a high-level programming language is desirable. The MathWorks, Inc.'s MATLAB provides an easy-to-use matrix programming language that is both portable and third-party extendible (see the "What is MATLAB?" sidebar on page 55). Although other simulation tools and matrix languages exist, industry and academia routinely use MATLAB for wide ranging analysis tasks in a variety of fields, including GPS. Software modules for carrying out a variety of related tasks are typically grouped together in "toolboxes."

A number of good GPS MATLAB toolboxes for GPS and/or GLONASS simulation — complete with source code — have appeared. Their authors recently provided the following products for us to review (others may also be available):

GNSS Toolbox. Orion Dynamics and Control Corporation developed its GNSS (Global Navigation Satellite System) Toolbox as an integrated collection of utilities designed to model, simulate, and analyze GPS, GLONASS, and Wide Area Augmentation Systems (WAAS) for various navigation applications. The software includes masking for earth-fixed or body-fixed satellites or rockets and models satellite orbits and signals, dilution of precision (DOP) parameters, and receiver noise.

Constellation Toolbox. Constell Inc. offers Constellation Toolbox for analyzing and simulating GPS, GLONASS, and WAAS as well as for other navigation and visibility applications. The package is designed as a highly integrated and vectorized collection of utilities that enables users to model visibility masking in either fixed or mobile environments. Utilities range from basic visibility analysis to simulation of DGPS, SA, tropospheric and ionospheric effects, and receiver clock errors.

SatNav Toolbox. Designed to simulate satellites, the propagation environment, receiver measurements, and data processing, SatNav Toolbox from GPSoft supports navigation applications for both standalone and DGPS positioning. The package can emulate a pair of receivers with proper decorrelation of data — one as reference, one (or more) as a rover. Differential carrier phase is also explicitly supported through ambiguity resolution for differential carrier-phase applications. By simulating the GPS navigation system, this package is said to cut down on the number of iterations needed to carry out a GPS navigation experiment.

GPS Signal Simulation Toolbox. Navsys Corporation has developed the GPS Signal Simulation Toolbox as part of its GPS Data Analysis Tools to simulate the GPS signals and the effects of signal degradation and dynamics on a receiver. The software simulates the receiver's code and carrier tracking loops, and users can observe the tracking noise, loss of lock, and cycle slippage. This tool can assist in testing and evaluating GPS receivers, surpassing the abilities of conventional RF simulators, which fail to provide insight into why various receivers perform differently.

OUR APPROACH

Although no single package should be expected to solve all GPS problems for all people (at a finite cost), we quickly ascertained each package's efficacy by analyzing its quality and breadth of coverage. To begin our review, we provide a brief overview of each toolbox in Table 1, which covers the platforms and the number of library, tutorial, and demo functions each product delivers. In the next section and in Table 2, we examine the software's organization and the supplied documentation, which includes the user's manual, organizational aids (indices), and on-line help. Table 3 summarizes how well each package is structured, including how well the MATLAB programming language is used to implement the various functions and whether the core coordinate routines are World Geodetic System of 1984 (WGS 84) compatible.

Our fourth set of tables addresses our evaluation of each package's performance. To more easily compare these products, we decided to evaluate the toolboxes using three criteria: how well they executed their GPS functions, how adaptable and efficient the individual tools were at simulating arbitrary GPS problems, and the correctness of select core routines with respect to the Interface Control Document, ICD-GPS-200. To exam-

ine these aspects, we selected several generic problems that users might wish to study. Solving each of these scenarios requires a set of GPS toolbox features, which we will briefly introduce in the article text and have listed in the summary feature table corresponding with each particular problem.

Table Abbreviations. In all of the tables, we indicate whether the product provides a given function with a "Y" for yes. If the particular functions have special fidelity for details of geometry, physics, or data processing, then we noted that as "Y-WD" for "yes, well done." No entry indicates the package did not supply that function. To further clarify which

functions truly addressed the specified simulation problem, we list an "S" for "solvable" next to those features that contribute to the solution. We also indicate whether we could solve the problem with only "trivial modifications" to the supplied demo programs (S-TM). Any functions we felt were important to solve the given problem, but that no vendor addressed, we have listed in a footnote to the appropriate table.

In the final section of our review, we describe our experiences, further address how well the packages lived up to ICD-GPS-200 in select core areas, and rationalize our subjective recommendations.

Table 1. Toolbox overview

	Orion GNSS Toolbox 2.05	Constell Constellation Toolbox 3.02	GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox
MATLAB version	4.2c, 5.x graphics	4.2c, 5.x graphics	4.2c, 5.x	4.2C
Platforms	PC, can coerce for Mac/Unix	PC, can coerce for Mac/Unix	PC, can coerce for Mac/Unix	PC, can coerce for Mac/Unix
Library m-files	68	68	30 m files	118
Tutorial m-files	7	7	26 m files	9
Demo m-files	34	34	x	11
Space (32kB blocksize)	2.88 MB	2.88 MB	266 kB	1.29 MB
Support	1 year, free, e-mail	1 year, free, e-mail, web	1 year, free, email	1 year, free, e-mail
Copyright	repay each copy	repay each copy	repay each copy	unknown
Upgrades	call	call	call	call
	(tutorial and demos were in separate directories)	(tutorial and demos were in separate directories)	(all in one directory, examples started with *ex*)	(three subdirectories: *geo* for coordinates, *sat* for satellites, and *signal* for the rest)

Table 2. Organization and documentation

	Orion GNSS Toolbox 2.05	Constell Constellation Toolbox 3.02	GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox
Purpose/problems	Y	Y	Y	Y
Programming tutorial	Y-WD	Y-WD	Y-WD	Y
Conventions	Y	Y		
GPS theory			Y	
Calling tree summary	Y	Y		
On-line help				
Input/outputs	Y	Y	Y	Y
Algorithm reference	Y	Y	Y	
Contents.m	Y	Y-WD		Y
Look for 1st H-line			Y	
See-also	Y	Y		
Reference organization				
Alphabetical	Y	Y	Y	
Subject				
Indexes			Y	

Table 3. Implementation

	Orion GNSS Toolbox 2.05	Constell Constellation Toolbox 3.02	GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox
Vectorized	Y	Y-WD		Y
Error checks	Y	Y-WD		
Outages and gaps	Y	Y		
Document tricks	Y-WD	Y-WD	Y	
Chunking/memory		Y *		
Correct ECEF,WGS 84	Y	Y	Y	Y
Platform+ver test		Y		

Note: Subcategories that no vendors currently support but are of interest to the community include global controls and self version test.

* At the time of this review, a chunking/memory demo was not shipped with the package but was available on the web site.

THE REVIEW

First, we would like to thank the vendors for participating in this review. This article would not have been possible without their cooperation. We are also happy to report that each of the four packages touched on almost every possible GPS modeling requirement, providing users with the necessary breadth of coverage for today's varied GPS application areas.

We also discovered, however, that the wide variety of simulation topics led to one downside — most toolboxes had similar organizational flaws. Most of the software programmers dumped roughly 50 MATLAB functions in their main GPS toolbox subdirectories to which the products' documentation only has an alphabetical reference section and/or no index or topical index.

Thus, if you don't know the arcane eight-letter filename of the function you need, you have to wade through the forest to find it. Despite this shortcoming, though, these toolboxes are truly useful and actually make mastering their learning curves a good investment.

SIMULATION CHALLENGES

As mentioned, we chose several potential simulation activities or problems (P1–6) for the packages to address: P1) GPS scenario simulation; P2) GPS measurement selection, navigation accuracy, and algorithm development; P3) fault monitoring; P4) reading recorded data; P5) total GPS receiver RF, digital signal processing (DSP), navigation, and modeling simulation test tools; and P6) presentation quality graphics. Tables 4.1

through 4.6 detail the functions necessary to achieve the solutions described in the following paragraphs. They also indicate which products supply the capability using the previously explained abbreviations.

First Things First. To solve problem 1 (see Table 4.1), the package must generate all of the data necessary to evaluate navigation accuracy in P2. Thus, the product must be able to generate the line-of-sight pseudo-range and carrier-phase truth and intentional error data as well as their rates for each satellite or space vehicle (SV) visible to a moving user; the received power levels and interference levels (caused by thermal noise or intentional jamming); these same two types of data for one or more differential or wide-area differential reference stations; and the scalar or vector differential corrections that the differential base station broadcasts.

The product must also ensure that the temporal and spatial correlations are correct. If they are not, it should be documented as to what they don't do. By carefully keeping the data separated into truth and error terms, and with good control options, the user can then explore the effects of different error sources. Ultimately, the package should faithfully generate the input data so that single- and double-differenced pseudorange and carrier-phase data for multi-antenna systems can be fed into kinematic differential carrier-phase navigation algorithms. Because many of the packages simulate ephemeris, SA, ionosphere, troposphere, and user clock models, several of these packages actually achieve most of these goals, at least for earth-based geometries.

Problem Two. A package that can solve P2 should supply several algorithms for selecting which satellite measurements to process, several types of typical navigation algorithms, and the tools to manage and build up new algorithms. (As reviewers, we didn't expect to find a product with all possible algorithms, but the more a toolbox supplies, the better the value.)

Typical starting points for identifying desirable features in a simulation package are selection of satellites based on DOP, weighted DOP, and robust DOP (*robust* indicates that the package can select the best number of satellites greater than four such that the loss of a satellite does not drastically alter DOP); point position and time fix; point velocity and drift fix; and local-area DGPS (LADGPS) fixes using scalar corrections broadcast from a single reference station. Enhancements include carrier-phase smoothing of pseudo-range, various Kalman filter components, some complete PVT (position,

Table 4.1. GPS scenario simulation — Problem 1

	Orion GNSS Toolbox 2.05		Constell Constellation Toolbox 3.02		GPSoft SatNav Toolbox v. Feb 98		Navsys GPS Signal Simulation Toolbox
GPS constants							
Hardwired	Y		Y		Y		Y
GPS coordinates							
ECEF,ECI,LLA	Y	S-TM	Y	S-TM	Y	S-TM	Y
Local-level wander azimuth	Y	S-TM	Y	S-TM	Y	S-TM	Y
Body-frames	Y-WD	S-TM	Y-WD	S-TM			Y
Azi & elevation angles	Y-WD	S-TM	Y-WD	S-TM	Y		Y
Time							
GPS Time Week, TOW	Y-WD	S-TM	Y-WD	S-TM			Y
Julian Day GPS		S-TM				S-TM	Y
UTC GPS	Y	S-TM	Y	S-TM			Y
Y2K compliant	Y	S-TM	Y	S-TM			
GPS Time sim time	Y-WD	S-TM	Y-WD	S-TM			
SV orbit tools							
Approx. circular	Y-WD	S-TM	Y-WD	S-TM	Y	S-TM	Y
Alm-based GPS	Y-WD	S-TM	Y-WD	S-TM			Y
Ephm-based GPS	Y-WD	S-TM	Y-WD	S-TM			Y
GLONASS circular	Y-WD	S-TM	Y-WD	S-TM	Y	S-TM	Y
GLONASS actual	Y-WD	S-TM	Y-WD	S-TM			
Trajectory tools							
File P, V, BodyAtt & acc, AttRates	Y-WD	S-TM	Y-WD	S-TM	Y		
Motion generation					Y		
Visibility, masks							
Earth block	Y-WD	S-TM	Y-WD	S-TM	Y	S-TM	Y
Vehicle block	Y-WD	S-TM	Y-WD	S-TM			
Space capable			Y-WD				
Measurement Simulation							
LOS range	Y	S-TM	Y	S-TM	Y	S-TM	Y
User Clock & drift	Y-WD	S-TM	Y-WD	S-TM			
Doppler (Velocity)	Y	S-TM	Y	S-TM		S-TM	Y
Carrier Phase	Y	S-TM	Y	S-TM	Y	S-TM	
SA Dither	Y-WD	S-TM	Y-WD	S-TM	Y-WD	S-TM	
SA Eps	Y-WD	S-TM	Y-WD	S-TM	Y-WD	S-TM	
Ionosphere	Y	S-TM	Y	S-TM	Y	S-TM	
Troposphere	Y	S-TM	Y	S-TM	Y	S-TM	
Multipath					Y		
Scalar LADGPS	Y-WD	S-TM	Y-WD	S-TM	Y	S-TM	
Single-diff CDGPS	Y	S-TM	Y	S-TM	Y	S-TM	
Double-diff CDGPS	Y	S-TM	Y	S-TM	Y	S-TM	
GLONASS versions	Y		Y		Y		

Note: Subcategories that no vendors currently support but are of interest to the community include lever arm(t), cg(t); antenna frames; SV rotation; profgen formats; simulator formats; user antenna; terrain masking; SV clock and drift; light speed transit; LOS acceleration; LOS jerk; vector WADGPS; attitude, n ants; C/No dB Hz; phase wrap up; and L2 versions.

Table 4.2. GPS measurement selection, navigation accuracy, and algorithm development — Problem 2

	Orion GNSS Toolbox 2.05		Constell Constellation Toolbox 3.02		GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox
Meas Select Tools						
DOPs	Y-WD	S-TM	Y-WD	S-TM	Y	Y
Nav Fix Tools						
Point pos & time	Y WD	S-TM	Y-WD	S-TM	Y	S-TM
Point vel & drift	Y WD	S-TM	Y-WD	S-TM	Y	S-TM
Scalar LADGPS	Y	S-TM	Y	S-TM	Y	S-TM
Carrier-phase smooth					Y	
Single-diff CDGPS	Y		Y		Y	
Doubl-diff CDGPS	Y		Y		Y	S
K.F. PV PVT model					Y	S-TM*

Note: Subcategories that no vendors currently support but are of interest to the community include weighted DOP; robust DOP; vector WADGPS; attitude fix; carrier phase K.F. PV; and K.F.+sensors.
* available on request

Table 4.3. Fault monitoring — Problem 3

	Orion GNSS Toolbox 2.05	Constell Constellation Toolbox 3.02	GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox
Post Fix				
Point RAIM	Y-WD		Y	
WADGPS Coverage		Y-WD		
CEP, drms, . . .				

Table 4.4. Reading recorded data — Problem 4

	Orion GNSS Toolbox 2.05	Constell Constellation Toolbox 3.02	GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox
Data Interchange				
NMEA	Y-WD S-TM	Y-WD S-TM		

Note: Neither RINEX nor any other data interchange format functions were supplied with any of the packages.

Table 4.5. Total GPS receiver RF, digital signal processing, navigation, and modeling simulation — Problem 5

	Orion GNSS Toolbox 2.05	Constell Constellation Toolbox 3.02	GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox	
Receiver Simulation					
Exact C/A I/Q				Y-WD	S
Tracking loops					
DLL				Y	S
PLL,FLL				Y	S
SNR				Y	S
Lock detector				Y	S
Generate C/A			Y	Y	
Generate C/A corr			Y	Y	
Predict ROC				Y	

Note: Subcategories that no vendors currently support but are of interest to the community include rule of thumb; linear 2kHz I/Q; analytical quant; analytical AGC; exact P I/Q; tracking loops — bit sync, frame sync; variable I/D; tracking moding; search; receiver management; and n channels.

Table 4.6. Presentation quality graphics — Problem 6

	Orion GNSS Toolbox 2.05		Constell Constellation Toolbox 3.02		GPSoft SatNav Toolbox v. Feb 98	Navsys GPS Signal Simulation Toolbox
Presentation	Y-WD	S	Y-WD	S	Y	Y
Basic	Y		Y			
Mercator	Y WD	S	Y-WD	S		
Quality	Y WD		Y-WD			Y
Interactive scale						

Note: Subcategories that no vendors currently support but are of interest to the community include gaps, chunking, and prune points. Other categories of interest include simulator utilities (subframe databits and trajectory), pseudolites, and jamming models.

velocity, and time) and/or carrier-phase–assisted Kalman filter models, vector differential GPS algorithms for wide-area DGPS (WADGPS), and single- and double-differenced kinematic navigation solutions (see Table 4.2).

Challenge Three. A package should deliver a few basic RAIM routines that can detect and isolate bad measurement data to solve P3 (Table 4.3). RAIM routines with tuning parameters and thresholds to detect different types of bad data — biases, noise-like errors, and so forth — enhance a package's value. You will note in Table 4.3, although some vendors supplied functions that could help solve this problem, none contributed significantly to such a solution.

Pinning Down P4. The reading-of-recorded-data problem, P4, can be solved by a package that at least supplies Receiver Independent Exchange (RINEX) and National Marine Electronics Association (NMEA) data parsing routines so users can easily process actual receiver data (see Table 4.4). Because receivers tend to store interesting auxiliary data in the manufacturer's own binary formats, a toolbox vendor would be justified in charging additional money for receiver-specific translators.

The Key to Five. To solve P5 (Table 4.5), detailed simulation of an entire GPS receiver, a package should generate base-band in-phase (I) and quadrature (Q) error signals at 2-kHz sample rates. These data mimic the integrate and dump outputs of the DSP chip found in all modern GPS receivers. The package should also provide some sample tracking-loop models. If it generates the actual sample values using a 4-MHz bandwidth for C/A-code and 40 MHz for P(Y) simulation, which includes the front-end automatic gain control (AGC) and quantizer, this would be very valuable. Simulink, a product for block-diagram design of dynamic systems for use with MATLAB, is probably a good implementation option when combined with MEX files for speed.

Last But Not Least. Finally, to solve P6, the package should supply some routines that allow the user to visually display results. It is desirable for the software package to do some interactive graphic scaling and to intelligently handle large data sets with scattered outages (see Table 4.6).

We had also considered a seventh problem regarding geodesy, which could have been solved by software that provides several datum conversion routines such that map data or geoid models could be handled. Unfortunately, none of the software offered functions that addressed this particular task.

OUR EXPERIENCES

During our review, we developed some engineering intuition about each of these products. All of the vendors knew that we were reviewers, so we can't tell you what type of support a nonreviewer is going to get. We can, however, convey our general impressions and specific experiences regarding each product.

All of the packages had a minor installation flaw when it came to modifying the MATLAB file path to include the new toolbox. They had the wrong advice, that of modifying the *Matlabrc.m* file, a file that changes from version to version of the program. All four should have recommended having a GPS *startup.m* file for each GPS folder. But once we got past that minor hurdle, the programs ran as documented.

Comments and Suggestions. The following sections will provide a few summary comments regarding our evaluations of the various packages. Please note, all comments took the packages' prices into account. For higher-end products, therefore, we made more suggestions. Before we start this final part of our review, though, we also need to note something about the companies Orion and Constell. If their products look almost identical, that is because, until recently, they were one company and one product. The products are starting to diverge, but at the time of this review, there were only subtle variations. We have, therefore, grouped our comments about their packages into one section.

Orion and Constell. Each of these products is presented as a highly integrated set of utilities for modeling, simulating, and analyzing GPS, GLONASS, and WAAS, and we believe that they met most of their claims, with the exception of WAAS. We liked the documentation, coding style, input variable checking, and the demos. We liked Constell's improved centralized error handling and the "chunking" demo with DGPS versus standalone GPS. We also felt the Constell staff knew its subject matter and the MATLAB language quite well and provided outstanding technical support.

Both packages could benefit from several additional items: an antenna frame and some antenna pattern routines; changes to the routine *utc2gps* to handle four-digit years; changes to the online *utc2gps* help to document how two-digit years of 50 and below are wrapped to years above 1999; and some improvements to the simulation of pseudo-range and carrier-phase data. The specific recommended improvements for the pseudorange and carrier phase are suggested to enable the addition of satellite

clock errors, specifications of receiver and antenna identification (ID), and to allow selected line biases as a function of receiver and antenna ID.

What is MATLAB?

Cleve Moler, chairman and chief scientist of The MathWorks Inc., and a former professor of mathematics and computer science, conceived of MATLAB, short for Matrix Laboratory. He aimed to create a computer language that would give scientists and engineers a way to naturally transform equations with complex matrix algebra into working computer programs. He also wanted it to have the appeal and simplicity of BASIC (Beginners All-purpose Symbolic Instructional Code — an easy-to-learn, highly flexible computer language invented at Dartmouth College in 1964 and still in wide use in a variety of offshoots) so that engineers and scientists with only novice programming ability could use it. The program also needed to support graphics and encourage users to write reusable function libraries.

Moler created the MATLAB programming language such that the mathematical operators

$$+ \quad - \quad * \quad \wedge$$

obeyed the rules of complex matrix algebra. He even included both left, \, and right, /, matrix inverse operators. But, he also recognized the need for traditional "do loop" actions that operated on matrices element by element. Thus, Moler defined the mathematical operator symbols

$$.+ \quad .- \quad .* \quad ./ \quad .\wedge$$

to carry out element by element addition, subtraction, multiplication, division, and powers. If x and y are matrices, possibly with complex entries, $z=x*y$ performs the appropriate matrix multiplication assuming compatible dimensions, while $z=x.*y$ performs an element by element multiplication assuming x and y have the same dimensions. The symbol ´ performs the complex transpose of a matrix.

Because MATLAB is interpreted rather than compiled and executed, the output variable z need not be defined; it is created on-the-fly. Although interpreters can be slow, MATLAB supports vectorization. If users structure the code to employ intrinsic vector operations such as the dot product or element by element operations while avoiding explicit do-loops, the program can execute very quickly.

Moler published the first user's manual for a FORTRAN-based MATLAB implementation in August 1980. During the next 15 years, several companies implemented the MATLAB language for mainframes, minicomputers, and microcomputers. MathWorks, however, was the first company to port the MATLAB interpreter to the C programming language and field versions on DOS, Windows, MAC-OS, UNIX, and DEC VMS computer operating systems. Moler ultimately joined MathWorks.

Today, MathWorks's implementation of MATLAB delivers a powerful interpreter that understands complex arithmetic, matrix algebra, element by element operations, hosts of primitive math functions, and traditional for, while, if-then-else, global, and case constructs. In addition, the MATLAB language has powerful indexing functions so that one can find all values in a matrix satisfying certain conditions and then use those indices to directly manipulate the desired elements.

MATLAB version 5.x recently introduced objects with more than two dimensions, C-like structures, and C++-like classes. When necessary, one can also create links from the MATLAB interpreter to compiled C and FORTRAN functions using MATLAB's *MEX* external interface programming convention. Novice programmers can still work with a simplified subset of the language and quickly write small programs that solve complex equations.

The basic MATLAB package comes with several built in toolboxes (libraries) that cover topics including linear algebra, cubic-splines, special functions, indexing, and basic signal processing. Engineers and scientists can quickly write scripts, functions, or combinations of both and then build up more complex programs. Because MATLAB comes with a powerful set of graphic functions, one can rapidly create professional looking plots and movies. There is even a real-time Vx-Works extension for digital signal processing boards as well as Simulink, a block diagram GUI (graphical user interface) driven simulation package.

MathWorks also has a formal partner program, MATLAB Connections, to support third-party vendors, which MathWorks lists on its web site <http://www.mathworks.com/>. MATLAB also offers source code portability and binary file interoperability across PC, UNIX, and Macintosh systems (only up to Version 5.2 for the Mac because of Apple's small market presence, although this could change). Finally, there is a low-cost university version of MATLAB with restrictions on matrix size. The combination of this edition and the program's 18-year history has led to a recent spate of textbooks that offer companion MATLAB code (see the "Further Reading" sidebar for some GPS-relevant examples.)

If you can afford them and don't need more than standalone PVT and LADGPS navigation algorithm options (or you can write them yourself), these two packages are

probably the top choice. At this time, the Constell package has the definitive edge because of the chunking and superior technical support as well as the recent 4.0 release that appears to have addressed these previously mentioned concerns. (The release of version 4.0 fell outside our review period, which ended in May.)

GPSoft. This simulation package promises to cut down on the number of iterations it takes to carry out a GPS navigation experiment. It meets this goal with the exception of restrictive use of circular orbit approximations and a single representative constellation. This package had the best variety of back-end navigation algorithms, with the exception of RAIM, and the most discussion of GPS theory. It was the only package to have a double-difference baseline navigation fix algorithm. In addition, we felt this vendor also knew its subject matter and offered outstanding technical support.

We do, however, have several recommendations. The package needs real orbit and GPS time utilities and could use some more examples that call the navigation routines and show how the Kalman filter routine handles outages during constant and nonconstant velocities. The Kalman filter should also be rewritten to incorporate measurements sequentially, one at a time. This would solve the problem of what to do when there are fewer than four visible (or in-track) satellites.

The *prange* function could be improved to better handle both receiver and antenna IDs with appropriate line bias terms and to account for satellite clock errors. The current RAIM algorithm is not complete and needs to have a geometry check, prescaling, and RAIM thresholds. The demos that use double-differencing techniques should at least contain references that document the algorithms being used, or there should be a separate "read me" text file. Overall, this package offers a useful product for its price range, if you are willing to work with only the circular orbits or are willing to combine this package with your own orbital routines.

Navsys. Offered as the only package that can explain why a receiver failed to track, this product meets that goal whenever a single-channel analysis suffices. The Navsys package is the only one that does a one-channel Monte Carlo receiver simulation (in which the software runs a large number of trials with randomly added errors). In addition, it is the only program that faithfully models the front-end digital signal correlation and carrier conversion processing. However, you need a really fast computer to run this package (for example, a 400-MHz Pentium II) and you have to be prepared to really work with this package in its current form. If a receiver is failing because of bad decisions on which satellites to track, or failing to navigate well, you would have to clone multiple versions of this package and add your own back-end receiver code complete with satellite selection logic and position fix algorithms.

Our recommended improvements include a comprehensive user manual with more flow charts for each of the major subsystems within the receiver simulation and adding an option to use a linear 2-kHz DSP model as a substitute for the 4-MHz model. This second option would be useful when the linear front-end model works (usually when the total noise plus jamming is Gaussian). If you absolutely need the 4-MHz model and can make do with one channel modeling, there is no MATLAB competition at this time. In terms of the other geographical and satellite orbit routines and compared to the other packages, however, what this package really needs is more demos to show how to use the toolboxes. It does have one really unique capability — an exact, high-fidelity Monte Carlo model of a GPS DSP chipset — but it also costs an order of magnitude more than the other packages.

OVERALL SUGGESTIONS

The one technical recommendation we would give to all vendors is to think about adding an option to the *prange* generation function so that the truth data contains the effect of finite GPS signal propagation speed and earth rotation such that users are generating data that a receiver would actually output. This enhancement would allow one to drive stand-alone GPS data processing software with data generated by a MATLAB software simulation.

All of the vendors are promising new versions (Constell's 4.0 has already been released) and prices change rapidly, so we strongly advise contacting each vendor to find out what has changed since this review was written. Finally, we would like to thank all of the GPS vendors for being forthright with information, providing us with review copies, and providing us with technical support. Please note that the opinions expressed in this article are those of the authors and do not necessarily represent the views of *GPS World* or Draper Laboratory.

Interested readers can access this review's tables and a detailed explanation of the entries at <http://gauss.gge.unb.ca/gpsworld/gpsmatlab.html>. ∎

Further Reading

For the official GPS signal description, see
■ Interface Control Document, *Navstar GPS Space Segment / Navigation User Interfaces,* ICD-GPS-200, Revision C (IRN-200C-002), published on behalf of the Department of Defense by ARINC Research Corp., El Segundo, California, 1997. This document is available as a PDF file from the U.S. Coast Guard's Web site at <http://www.navcen.uscg.mil/gps/geninfo/gpsdocuments/icd200/icd200c.pdf>.

For details about GPS receiver operation, observables, and coordinate frames, see
■ *Global Positioning System: Theory and Applications,* edited by B.W. Parkinson and James J. Spilker Jr., published by the American Institute of Aeronautics and Astronautics, Inc., Washington, D.C., as Vol. 163 and 164 of *Progress in Astronautics and Aeronautics,* 1996.
■ *Understanding GPS: Principles and Applications*, edited by E.D. Kaplan, published by Artech House, Inc., Norwood, Massachusetts, 1996.

For discussions about GPS observable analysis with sample MATLAB codes, see
■ *Linear Algebra, Geodesy, and GPS,* by G. Strang and K. Borre, published by Wellesley-Cambridge Press, Wellesley, Massachusetts, 1997.
■ *Introduction to Random Signals and Applied Kalman Filtering: With Matlab Exercises and Solutions,* 3rd edition, by R. G. Brown and P. Y. C. Hwang, published by John Wiley & Sons, Inc., New York, 1996.

For more information about the companies and products mentioned in this review, consult their web sites at
■ Constell, Inc. <http://www.constell.org/>
■ GPSoft LLC <http://www.gpsoftnav.com/>
■ Navsys Corporation <http://www.navsys.com/>
■ Orion Dynamics and Control Corporation <http://www.sni.net/oriondc/>

MANUFACTURERS

The companies participating in this review included **Constell, Inc.** (Longmont, Colorado); **GPSoft LLC** (Athens, Ohio); **Navsys Corporation** (Colorado Springs, Colorado); and **Orion Dynamics and Control Corporation** (Boulder, Colorado).