MATLAB Story

MATLAB is a full-featured technical computing environment, but it started as a simple "Matrix Laboratory." Three men, J. H. Wilkinson, George Forsythe, and John Todd, played important roles in the origins of MATLAB. This story began more than 50 years ago.

Wilkinson was a British mathematician who spent his entire career at the National Physical Laboratory (NPL) in Teddington, outside London. Working on a simplified version of a sophisticated design by Alan Turing, Wilkinson and colleagues at NPL built the pilot Automatic Computing Engine (ACE), one of Britain's first stored-program digital computers. The Pilot ACE ran its first program in May 1950. Wilkinson did matrix computations on the machine and went on to become the world's leading authority on numerical linear algebra.

At about the same time, mathematicians at the Institute for Numerical Analysis (INA), a branch of the National Bureau of Standards, located at UCLA, were working with the Standards Western Automatic Computer (SWAC), one of the USA's first computers. Researchers at INA included George Forsythe, John Todd, and Olga Taussky-Todd. When the INA dissolved in 1957, Forsythe joined the faculty at Stanford and the Todds joined the faculty at Caltech.

Alston Householder from Oak Ridge National Laboratory and the University of Tennessee began a series of research conferences on numerical algebra in the late 1950s. These are now held every three or four years and are called the Householder Conferences. The organizing committee for the 1964 Gatlinburg/Householder meeting on Numerical Algebra, all six members of the committee – J. H. Wilkinson, Wallace Givens, George Forsythe, Alston Householder, Peter Henrici, and F. L. Bauer – have influenced MATLAB.

Cleve Moler's 1965 Ph.D. thesis under Forsythe's direction was entitled "Finite Difference Methods for the Eigenvalues of Laplace's Operator." The primary example, on which both Forsythe and Wilkinson had worked earlier, was the L-shaped membrane, now the MathWorks logo.

Forsythe and Cleve Moler published a textbook about matrix computation in 1967 that was later listed by the Association for Computing Machinery as an important early text in computer science because it contained working software: programs in Algol, Fortran, and PL/I for solving systems of simultaneous linear equations.

Over several years in the late 1960s, Wilkinson and a number of colleagues published papers in *Numerische Mathematik* that included algorithms in Algol for various aspects of matrix computation. These algorithms were eventually collected in a 1971 book by Wilkinson and Reinsch, edited by Cleve Moler.

Even today, more than 30 years after its publication, this collection of algorithms for matrix computation is an important reference.

Every summer for 15 years, Wilkinson lectured in a short course at the University of Michigan and then visited Argonne National Laboratory for a week or two. Researchers at Argonne translated the Algol code for matrix eigenvalue computation from the Wilkinson and Reinsch handbook into Fortran to produce EISPACK. This was followed by LINPACK, a package of Fortran programs for solving linear equations. The EISPACK manual was published in 1976 and

the LINPACK manual in 1979. At that time, Cleve Moler was a math professor at the University of New Mexico, teaching numerical analysis and matrix theory. With the aim of use of by LINPACK package without writing Fortran programs, he studied a book by Niklaus Wirth to learn about parsing computer languages. Later, Niklaus Wirth developed PASCAL language.

In the late 1970s, following Wirth's methodology, Cleve Moler used Fortran and portions of LINPACK and EISPACK to develop the first version of MATLAB. The only data type was "matrix." The HELP command listed all of the available functions, with their names abbreviated.

This first Fortran MATLAB was portable and could be compiled to run on many of the time-shared computers that were available in the late 1970s and early 1980s. The first "personal computer" for this purpose was the Tektronix 4081, which Argonne acquired in 1978. The machine was the size of a desk and consisted of a Tektronix graphics display attached to an Interdata 7/32, the first 32-bit minicomputer. There was only 64K, that's 64 *kilobytes* of memory. But there was a Fortran compiler, and so, by using memory overlay, it was possible to run MATLAB.

In 1979 Cleve Moler had his students use MATLAB in CS237 graduate numerical analysis course taught at Stanford. Half of the students in the class were from math and computer science, and they were not impressed by the new program. It was based on Fortran, it was not a particularly powerful programming language, and it did not represent current research work in numerical analysis. The other half of the students were from engineering, and they liked MATLAB for applications in control analysis and signal processing , and the emphasis on matrices in MATLAB proved to be very useful to them. A few of the Stanford engineering students from this class joined two consulting companies in Palo Alto. These companies extended MATLAB to have more capability in control analysis and signal processing and, in the early 1980s, offered the resulting software as commercial products.

Jack Little, a Stanford- and MIT-trained control engineer, was the principal developer of one of the first commercial products based on Fortran MATLAB. When IBM announced their first PC in August, 1981, Jack quickly anticipated the possibility of using MATLAB and the PC for technical computing. He and colleague Steve Bangert reprogrammed MATLAB in C and added M-files, toolboxes, and more powerful graphics.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in

a scalar non-interactive language such as C or Fortran. The name MATLAB stands for matrix laboratory. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system consists of five main parts.

I. The MATLAB language.

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

II. The MATLAB working environment.

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

III. Handle Graphics.

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

IV. The MATLAB mathematical function library.

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

V. The MATLAB Application Program Interface (API).

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

What about the use of MATLAB for business applications? Many of the businesses use data analytics and visualization extensively. MATLAB accelerates the development of data analytics. Some of the other areas of MATLAB applications in business are energy trading, risk modeling, and portfolio optimization.

Reference Links:
http://www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html
http://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm
https://www.mathworks.com/company/events/conferences/matlab-computational-finance-conference-nyc/2015/proceedings/integrate-matlab-analytics-into-business-critical-applications.pdf