CHAPTER 1

Data Analysis and Systems Integration

What all technical systems or applications have in common is data, and for data to serve some purpose there must be analysis of that data to achieve a goal; that goal most generally being System Integration. While you will encounter many systems in this book--another way to say many "Technical Applications", the focus of the book is really on the subject of DATA ANALYSIS. In the history of science and technology the single closest recognized discipline of education and application to what is considered Data Analysis today is that of Numerical Analysis. {Sometimes the discipline is called Numerical Methods, and years ago as large scale computers became the tool of technical and scientific research, the closely allied field of Operations Research developed knowledge and skill to mold the discipline of Numerical Analysis into engineering, computer science, and technology.}

1-1: Simulation of Space Shuttle with MATLAB programming.

One example speaks louder than words, in the following case it is the space shuttle in re-entry, being analyzed by way of simulation. **{By the way, the program coding is in MATLAB.}**

```
%caclulate table for space shuttle re-entry parameters,
%first caclulate Z, 1.8 degrees Rankin equals 1 deg Kelvin
%velocity at reentery, Ve is 36 Kft/sec, beta = m/CD*S, m = 342
```

```
slugs,CD=1.3,S=130 ft2;
clc
clear all
clf
Ve = 36000.;        %velocity at re-entry of 36,000 feet per second
CD = 1.3;           %Coefficient of Drag to be explained
S = 130;            %wing surface area
m = 342;            %mass of space shutte in slugs
g = 32.2            %gravity
beta = 342/(1.3*130);     %beta here is heading angle[1]
format long g
R = 1716;         % r, Ryberg const in Eng sys is 1716 ftXlb/(slug)degR
degR = (288)*(1.8);    %temperature in Rankins
% calculate Z = go/RT; g in eng units 32.2 ft per secsquared
Z = 32.2/(R*degR);
% caclulate velocity deceleration in g's
VEsq = (36000)^2;
MaxDecel = (VEsq*Z*.043)/(2*exp(1));
MaxDecelg = MaxDecel/32;
% calculate density, first at 400,000 feet
h = 150000:10000:400000;  %setup for alt, h, in increments of 10Kfeet
lh = length(h);    %more looping setup with length of h
rhosl = .0023769;        %rho for density, rhos1 is initial density
rho = rhosl*exp(-Z*h);
hkft = h/1000;           %converting feet to Kfeet
%plot(hkft,rho)          %used only in initial debugging
%xlabel('altitude in Kft')
%ylabel('density slub/ft3')
%velocity versus altitude, h and rho
VEL = Ve*exp(-rho/((2*beta)*Z*.043));  %Velocity calculation
subplot(2,1,1)             %top subplot shown below
plotyy(hkft,rho,hkft,VEL)  %plot of rho on left and Vel on right
xlabel('altitude in Kft')
%calculate drag from D = 1/2 rho*VEL^2*S*Cd
const(1:lh) = S * CD;
nrho(1:lh) = const.*rho;
```

[1] Beta for aircraft data analysis is sideslip, and AOA or alpha is angle of attack.

```
nVEL2(1:lh) = VEL.*VEL;
D = (nrho .* nVEL2)/2;
%calculate dV/dt = D/m;
mconst(1:lh) = m;
dVdT = D./mconst;
dVdTg = dVdT./g;
subplot(2,1,2)
plotyy(hkft,D,hkft,dVdTg) %Drag on left axis and dV/dt = a on right
```

NOTE: If you already know MATLAB, go for programming this script if you wish. If not, you see some of the interest and excitement that is in store for you. And the analysis in plots as shown below enhances that excitement. The comments to the right of program code, starting with "%" for comments, like C in another language, will help you see what is happening.

1-2: Numerical Analysis the Anchoring Discipline.

You might consider Numerical Analysis, along with Advanced Engineering Math, the anchor during the digital atomic revelution. The languages of emphasis changed from FORTRAN to MATLAB, the computer hardware changed from large scale computers to mini- and micro-computers; but Numerical Analysis with the additon of only a few methods remained the same. An Engineer of the 1960s in read a book on programming in BASIC or FORTRAN IV would essentially recognize and see the same names for methods like Runge-Kutta, as the Engineeer reading a modern book on Numerical Methods with MATLAB.[1] The evolution of modern numerical analysis **{surely both Excel and MATLAB must represent modern analysis},** is in many ways a paradox; for exmple a book written in 1984 as part of the NATO ASI (Advanced Science Institutes) Series, and entitled SIMULATION[2] AND MODEL-BASED METHODOLOGIES: AN INTEGRATIVE

[1] The one exception that I can think of is: in 1998, only 12 years ago, William J Palm III of the University of Rhode Island, wrote an extremely practical and helpful book on programming with MATLAB specifically for Engineers {it also does not surpass the capabilities of a two year Electronics Engineering Technology student}, in which he makes a startling revelation about the shortcomings of reliable methods in "optimization". MATLAB has tried to overcome that as the "fminsearch" built in function is coming to replace "fmins", and only the future witll tell if fminsearch is sufficient. REFERENCE: page 110 of INTRODUCTION TO MATLAB FOR ENGINEERS.

[2] As in this book where the F-16 and a Transport Aircraft are simulated with MATLAB programming, simulation developed then and quite often now was less hardwired Simulators per se and more languages of programming.

VIEW, and with one chapter on "Optimization in Simulation Studies" at the same time suggests the more modern approach of state-space as used in most current textbooks on Automatic Control, techniques of parameter estimation like Lliff lectured on at NASA in 1987, supposedly for the best of reasearch in Systems ID and linear theory, the cost function of C&S introduced as "computational cost"; yet goes all the way back to the Newton-Raphson method, brings up the famouse simplex method for optimization and numerical methods of Nelder and Mead of 1965,[1] calling it the "polytope method" to separate it from the simplex method of linear programming. Regardless this bit of history and perspective is interesting reading from 26 years ago as the author[2] talks about the increased availability of large scale computers and the need for optimization techniques and specially designed software **{ISIS, ACSEL, GEST, COSY, MACKSIM, and FORTSIM}** to keep up with an increased need for optimization.

> "The determination of optimal values for parameters is often as important aspect in both the formulation of mathematical modesl for systems nd in their subsequent use in simulation studies. Optimization subproblems are therefore intimately associated with model-based studies. The objective of this paper is to explore some festures of the inerface between these two problem calsses and to provide and oveview of some of the numerical procedures taht are available for solving such parameter optimization problems."

We technical workers who lived and worked through technical applications with computers of the last 50 years, the half a century of what will later in the book be called "The Digital Atomic Age" because digital did for technical applications with computers what the atomic bomb did for science, also saw Numerical Analysis grow and develop from applications to large scale vacuum tube and analog computers, through the minicomputers of transistors, and then into the microcomputers of integrated circuits (ICs). This is not to say that some person or group instigated a plan for this Digital and Digital Computer revolution often at the highest technical levels riding on the back of Numerical Analysis, unlike the changes instigated by Bill Gates and his Microsoft Empire {we will not now get into the borrowing Microsoft did from the hard work of the MAC empire}; but it rather just happened as the natural evolution of science and

[1] MATLAB came out with "fminsearch" in XXXX, based on the Nelder and Mead simplex method, or "polytope method" of 1965, a method referenced in modern technical literature today like C&S and Lliff at NASA's "Parameter Estimation in Aerospace".

[2] Louis G. Birta of the Department of Computer Science of the Univsity of Ottawa.

technology of many diversed applications like physics, mathematics, engineering, computer and other electronics hardware and software.

Just the solution of a problem of physics was numerical analysis without being considered as such. The system was analyzed whether it was a simple falling body, a pendulum, a bouncing ball, or the path of a missile; a drawing was made to represent the system with all the known data recorded on the drawing; the applicable physical equations of motion were applied; and then a numerical solution was calculated. Quite often a more detailed problem involved some error analysis between the calculated and the standard as for example when in a physics lab we were using the famous oil drop apparatus of Millikan to determine the charge of the electron as close to the known value of 1.6 X 10^-19 as possible. **{Note we will use in this text the notation of the MATLAB language.}** So the error analysis gave us a percent comparison between our experiemental calculations and the known standard. And what we were really doing without any such real noble goal as what Lord Kelvin said **{"It is not scientific until you attach a number."}**, was to understand and integrate a technical problem by attaching a number that had more significance to us than for example E = m X c^2 or Newston's second law of Force is equal to mass times acceleration (F = ma). In fact, we might say in looking back on the early history of science and classical physics, that Newton when the apple under the acceleration or force of gravity fell from the tree on his head, or when he formulated with much thought, application, and numbers, the second law of motion, was the process of numerical analysis, or Data Analysis.

1-3: Excel and MATLAB for Data Analysis.

Most people are at home with Excel, so some comparisons along the way will be made with using Excel or MATLAB for Data Analysis. **{Yes, the very same Microsoft Excel of Microsoft Works, and on your computer, can do Data Analysis.}** MATLAB was originally developed by Cleve Moler in the 1970s while he was chairman of the Computer Science Department at UNM, then with Jack Little, an engineer that specialized in control system design, and Steve Bangert, they founded MathWorks in 1984, after rewriting MATLAB in the C language. It is a little harder to date Excel, especially for you younger generations who think it existed before computers. It has been a widely used spreadsheet since version 5 in 1993. Microsoft first issued a Windows version **{2.05}** in November of 1987, and after 1993 when Microsoft included Visual Basic for Application {VBA}, Excel was well on its way of extensive use in Data Analysis. Although somewhat limited even in technical applications, when in 2004 Robert de Levie wrote ADVANCED EXCEL FOR SCIENTIFIC DATA ANALYSIS, since the emphasis and professional speciality of this author for 34 years was as professor of analytical chemistry and electrochemistry at Georgetown. If the copyright credits Levie gives in his excellent book on Data Analysis are any indication of how long Excel has been used in Data Analyis, your at first would think all the way back to 1974[1]; however since this is impossible, we might think without extensive research on Levie's references, that VBA was used for Data Analysis. MINITAB, a statiscal software once more for teachers, is also used some for analysis in this book. It was developed by Penn State in 1972, although rarely known or used until recently as it has become popular for quality control work in Six Sigma. **{What I found useful was with histograms in Flight Test Reports and presentations.}** We will want to compare analysis of data between Minitab, Excel, and MATLAB.

1-4: Large Modern Systems in the Evolution of the Digital Atomic Age.

You will find in the course of the reading and study of this book that two systems dominate the material--airplanes and missiles; but it is hoped that you will see up front, and foremost, that these two now very complex systems as seen in the Technical Applications to the Boeing 787, the General Dynamics F-16, and the Space Shuttle which is now a Space Airplane robotically controlled, illustrate techniques and tools of DATA ANALYSIS. **{Also we must write about what we know from experience and having retired from General Dynamics as a Flight Test Engineer on the F-16 and from Raytheo as a Principal Systems Engineer, testing the missiles that carried the KW and EKV into the exo-atmosphere to shoot down incoming ICBMs, the material naturally evolved into a focus on missiles and airplanes.}** Yet you will find telemetry, flight test, communiations, and other modern systems, once agains as illustrations of DATA ANALYSIS.

Quite often modern and complex electro-mechanical systems consists of many systems; for example the F-16 operates centered around over 20 distinct computers and systems like the weapons control systems, the fire control system, the flight control system **{each having its own quad-redundant computer or in the case of the FCS a system or 4 computers--the CADC, the FCC, the**

[1] Y. Bard wrote NONLINEAR PARAMETER ESTIMATION in 1974, and Levie gained permission to use some of the data. "Parameter Estimation", like the footnotes, becomes very important in this book, so 3 standard methods of parameter estimation are introducted right now up front: (1) probability plotting; (2) Least Squares, also called Regression Analysis; and (3) Maximum Liklihood Estimation {MLE}, which is the favorite of some of the flight test aerospace experts utilized in Flight Test Reports of this book. Right now you should become accustomed to the Engineering usage of "estimation" as much stronger than a guess, sort of in the same category as a theory in physics; and really the MLE is the best choice of a system model with parameters that can be made.

ECA, and the PSA[1]}, the engine warning systems, the engine control system, and on and on. And almost as often the modern in design, system integration, and flight test is that well designed partial systems work quite well independent of the aircraft {that is, in the integration lab}, but do not work as a whole in the total system of the aircraft. This was not so much a problem in the F-16 as it was after design of the first block 1 system was further developed through the years in a block system, going from block 1, to 10, all the way up to 40 and beyond; so that any subsequent total systems integration and development problems such as engine warning, direct battery power to the flight controls and an auxiliary generator just for the flight controls, secure voice, and even newer ECM and weapons were wisely programmed through the years as large scale modifications. This was not the way the Lockheed Martin C-130J, an all computer controlled aircraft, program was designed and developed, so that for approximately one year after exit from production the bugs of total systems integration **{the well designed parts working together as a whole}** were still be worked out. One obvious case in point since LMC had previously sold the wind tunnel at Marietta to Ford Motor Company and no tunnel model of the C-130J was tested, it was a surprise to all at the total system flight test when the props, the composite airframe, the engines, and aerodynamics of the J model departed from the flight history of many years of other models of the C-130; in fact, departed during a stall with a slip right of approximately 5,000 feet. That is a hard way to learn about a further need of systems integration based on data analysis of wind tunnel data.

As you can see our concept of "system" goes far beyond the fundamental definition of system as used in the Linear System Theory of an engineering course, like that of THE LINEAR SYSTEM THEORY, THE STATE SPACE APPROACH[2] written in 1963 by two professors of the Electrical Engineering Department at UC Berkeley. It goes something like this: an absract system[3] **{they do not apologize but support the abstract concepts and proofs for them}**; but then say,

[1] This was in the A and B models before the C and D models which started at block 40, and before the analog Flight Control Computer was converted to a digital flight control computer.

[2] Written by Lotfi A. Zadeh & Charles A. Desoer, McGraw-Hill, 1963.

[3] Do not let this smaller defintion of system scare you, nor the matter of proofs and details and theory; for the real contributions of Linear System Theory, especially back in those days just before the Digital Atomic Revolution, was in proving the feasibility of the state-space approach to problem solution and the transfer funtion. They did it so well, running graduate and junior level engineering students through a whole semester of laborious details and proofs, that we now can just use such tools of analysis as the state-space aproach and the transfer function with great confidence. Not only do they work, and have been proved to work time and time again; but they have also by Zadeh and Desoer {and others by the ways since 1963) been justified as valid.

"or system" **{enginnering held them down to practical appliations, and they do refer to many typical systems like electrical networks and even the famous mass-spring system}** which is a partially interconnected set of abstract objects termed the components. Of course, in something like in circuit analysis the components can be resistors and capacitors. While you will encounter many of these so-called systems on a smaller scale such as for circuit analysis, even sub-systems of the Flight Control Computer, and yes, the famous mass-spring system, you have already noted above that the modern systems of this book and technical applications of computers are much larger like airplanes, missiles, and flight control computers. In fact the expertise of this author for writing and previously for technial work was to rapidly "come-up-on" **{a phrase used to Aerospace to get smart on a system}** on new systems. These, if taken in the order of a career in Aerospace, momentarily forgetting the 10 years of teaching electronics engineering technology interspersed in the Aerospace years, would roughly go like this: an Air Early Warning Radar System; the Atlas Missile System; the Minuteman Missile System; the Athena Missile System fired in Green River Utah and which impacted on White Sands Missile Range from which was obtained re-entry data; Telemetry Systems for data acquisition and intelligence; the C-130 Gunship Forward Looking Infrared System; the F-16 which was really a system of many systems; and the KW and EKV payloads of the missile shield around this nation.

1-5: Differential Equations and Physics Have Taken a Bad Rap.

Like Physics, calculus has taken a bad rap for being difficult in the initial learning, and this bad reputation has kept many from getting into the twin disciplines, even more engineerting tehnoclogy students from delaying the learning benefits for the second and much later year. We will show right now how easy differential equations are to get into **{large part of the applications of calculus to technical programming}**, especially with the ease of accessibility of MATLAB to your finger tips.

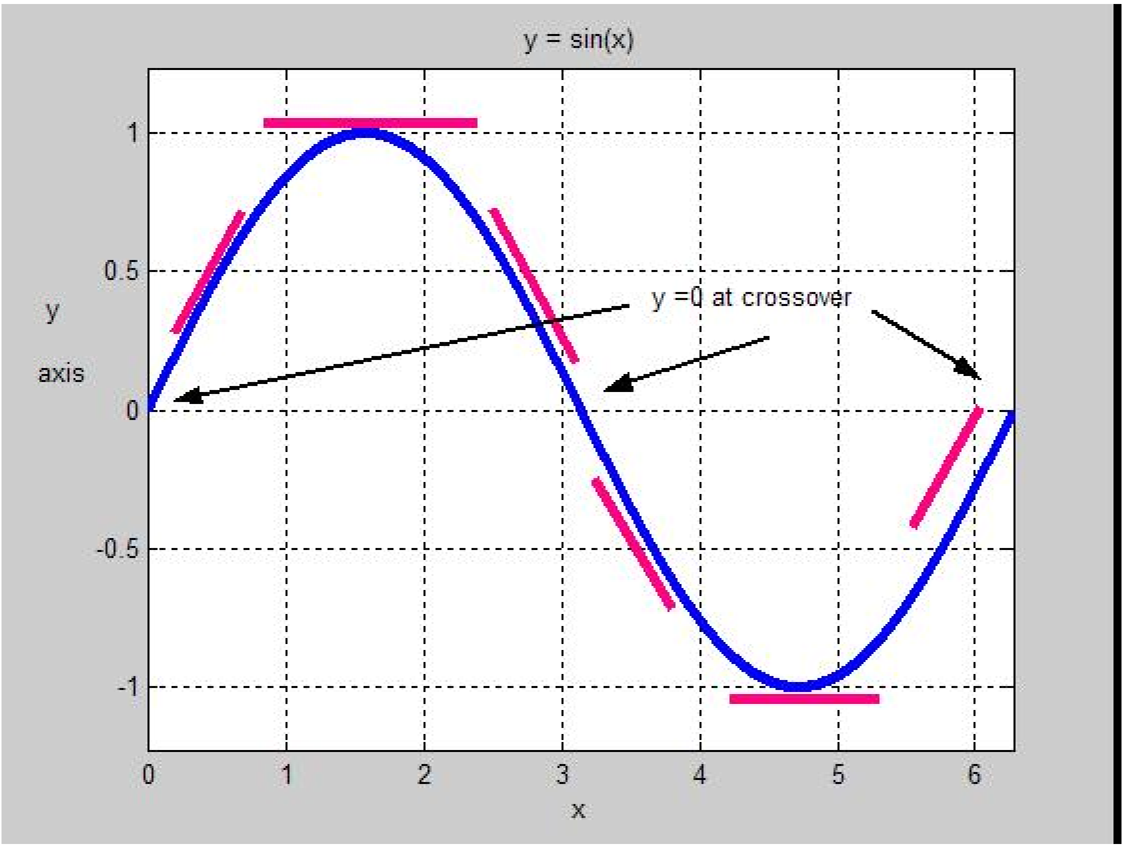
Somethig must definitely be said about the foundation of mathematics in the history and evolution over the last 50 years of numerical analyis for data analysis. You will read more about this in later chapters, especially the one on "The Digital Atomic Revolution", but it can be simplied by considering math as passing from an emphais on differential equations **{which of course like all motion and physical phenomenon itself are analog}** to the matrix of linear algebra; of course with mathematical techniques like Taylor series and the LaPlace transform to convert from an analog to a digital format, or from differential to linear format. That is where MATLAB in technical computer applications, much like Microsoft in total computer applications, has ridden the back of the Digital Atomic Revolution: MATLAB has provided the tools in a single concise language {many statements called commands like “diff” to differentiate and “FF” for the fast fourier transform of the LaPlace equation} to easily provide calculations in analog, digital, and both. Then you throw in the simulation part of MATLAB with “Simulink”, the built in simulator of MATLAB, and you have a language and software, albeit quite expensive except for large companies except for the student version used primarily in this book, and MATLAB can do just about anything for you, including your calculus and linear algebra homework, except cook your meals. Even as Microsoft developed into the more popular software than MAC while not being at all any more functional, so other software today like FORTRAN, Mathematica, or BASIC depending on the technical application can be as fucntional as MATLAB, we just have in MATLAB and Microsoft the more popular means of communicating with our technical, social and work environment.

Let us begin right now, to take some of the mystery of differential equations with introduction to a very simple one of y’ = f(x) = sin(pi*x), where y’ is an easy and standard way to write dy/dt. What this equation means then is the first derivative of y, which is a function of x, that is f(x), which in turn is the speific function of x which is sin(pi*x).

NOTE: Get use right at this beginning chapter, that all equations will be written in the language of MATLAB; that is * for multiplication, and pi for pie. And what a way to start feeling at ease with differential equations in terms of what you know so well for fundamental mathematics, the most common functions of sine and cosine. These by the way, written in the language of MATLAB, are among the many built-in functions of MATLAB.

If you back to the basic definition of the derivative, y’ = dy/dt, the basic concept of which comes to mind as the rate of change, the rate of change of y, or delta y, as compared to the change in time, delta time. What is different about the calculus of derivations is that the delta change is extremely small; in fact, a small change that approach infitesimal. **{It is about the smallest change that you can conceive of.}** When you operate on a function to get it out of its derivative back into the original function, like to integrate y’ = -sin(pi*x) to get y **{notice we are passing in integration from y’ to y}**, or y = -1/pi * cos(pi*x) + c.

Even as it use to be the custom to take semesters of “analytical geometry and calculus”, since so many of the concepts of the calculus of differential equations comes from geometrical visualization and foundations. For example in the MAT-



LAB plot above {edited over in the figure windown to show slopes and places where y is equal to zero}, you can visualize y' as the slope at different x points on the y = f(x) = sin(x) curve. Left to right, and without numbers, you can see that between 0 and 1/2 pi the slope {delta y/delta x, or so much rise over run} is max; then at pi, zero slope; then through the sequence of minus max, zero slope, and max postive. And intuitively you can see that these slope values in turn plot a cosine function. You can also think of differentiation and integration as the opposites: diff takes the rate of change of the curve, and int is the antiderivative.

MATLAB makes it easy for you to switch between functions like y and y' with the "int" and "diff" commands. As we spell out the process of gong back to y from y' with the proper symbology of differential equations, we get:

MATLAB makes it so easy to pass from y' to y by use of the "int" command with the following code {check it out in the command window}.

```
%drill on integration and differentiation
%since int and diff work with the symbolic sin and cos, or explicit, use %s
for function
s = '-sin(pi*x)'
y = int(s)
yy = numeric(int(s,0,pi))
yyy = numeric(int(s,0,2*pi))
s1 = '-sin(2*pi*x)'
y1 = int(s1)
s2 = '1/2/pi*cos(2*pi*x)'
y2 = diff(s2)
%now to get some intuitive feel for integration, try
s3 = 'abs(x)'
y3 = int(s3,.5,.6)
s4 = 'x'
y4= numeric(int(s4,.5,.6))
y5 = numeric(int(s4,0,pi))
y6 = numeric(int(s4,0,2*pi))
```

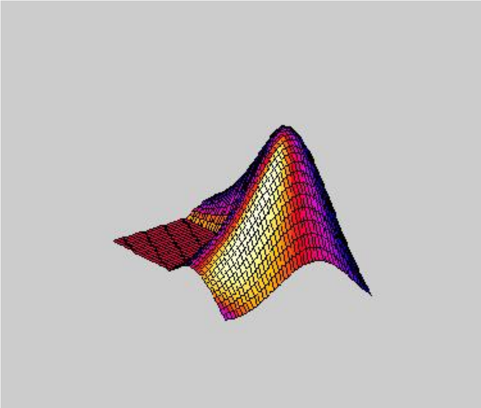
And you can match up the answers in the command window:

```
s = -sin(pi*x)
y = 1/pi*cos(pi*x)
yy = -0.6056
yyy = -0.1179
s1 = -sin(2*pi*x)
y1 = 1/2/pi*cos(2*pi*x)
s2 = 1/2/pi*cos(2*pi*x)
y2 = -sin(2*pi*x)
s3 = abs(x)
y3 = 11/200
s4 = x
y4 = 0.0550
y5 = 4.9348
y6 = 19.7392
```

1-6: Many Threads of Modern Technology Made the Technical Revolution.

You will find many threads that run consistently through the book which will not be in the chapter titles as at the highest level this is a book of Technical Applications; however you will see them at the Section Heading Level, like 1-6 above: numerical methods, Excel, Mathematica, MINITAB, optimization, sytems ID, linear theory, and advanced engineering math, even data analysis itself and systems integration. You might call some of those threads of book specialization, the "continuity in the book" like: optimization, simulation of a transport aircraft and F-16 in MATLAB, data analysis, the programming of advanced engineering math, systems integration {closely related to the process}, parameter estimation and modeling.[1] These threads of speciality, along with the engineering and teaching experience of the author, help to make this book unqiquely what it is. Many more specialized, and much more difficult, books have been written on the specialized and other threads; but it is the unique selection with applications that makes TECHNICAL APPLICATIONS OF COMPUTERS. **{Therefore each thread, especially the specialized threads, will be introduced as early as possible in the chapters of the book; and then hung onto and progressively applied.}**

And yes, along the way I will share with you some of the intuitive excitement of 50 years of working in technology, teaching and engineering. It takes one who has experienced, and still remembers often, the excitements of learning physics, different equations, computers, electronics, and sophisticated systems to boot, to have those memories; and it is hoped that some of that emotion, with the technical content, is communicated in this book. One might call it true learning or scholarship as contrasted to a priority for gaining a piece of paper to earn money, and might have its roots in thinking like of Ralph Waldo Emerson in "The American Scolar", perhaps you would like to call it "The American Technical Scholar". This helps to make the subject of TECHNICAL APPLICATIONS OF COMPUTERS more interesting; and you will enjoy simultaneously technically working with your computer while you make graphs and plots like the MATLAB logo.



```
%MATLAB logo
```
[2]

[1] While today model aided design most often has to be with the use of the modeling blocks of Simulink, modeling has a more general and persistent concept of the math and parameters used to describe a system.

[2] MATLAB code for programs, routines, and functions will be shown in GREEN, also the displays from the command window: comments will be in BLUE; and equations will be in bold BLACK.

```
load logo
surf(L,R),colormap(M),n=length(L(:,1));
axis off, axis([1 n 1 n-.2,8]), view(-37.5,30)
```

1-7: Sharing of Learning Theory.

Another consistent thread throughout this book will be sharing on learning theory. Finally, it is apparent that the "you are picking on me" attitude of the boomer generation just did not work, as the cover-up in education had been blown, even bigger than the cover-ups in financial institutions and government related to the economy, American now recognizing that over one/third of high school students do not graduate. The proof in is in the pudding, and how can you claim to teach and the system is working when the results are so miserable. Poor education of teachers in colleges and university, and the practicing of poor teaching in high schools and other levels of education, has sacrificed generations of student drop-outs on the altar of "you are picking on us".

As one successful teacher in California recently said, "The education system is broken. Teachers can't teach, and students can't learn."

There are several obvious reasons for this, but more of the responsibility should fall on the education departments of universities across the nation who have so poorly trained teachers. It is a well known fact among college students and alumnus that the easies route to go in college is to major in eduction, and likewise that when a student has taken approximately 3 education courses for then on it is a repeat.

Of course, the real problem, a weakness that has been known for years by these money-making eduation organizations like Sylvan, is that of the four levels of learning, teachers normally take the path of least resistance and use only the lower two levels. Of the four levels of learning--rote memory, example, trial and error, and problem solving--memory work and example learning are the easiest, require less time, and take less preparation.

An extension of this problem is in continuing education. The average knowledge and skill of the average college degree now has a half-life of approximately 8 years. What that means is simple: in 8 years after graduation over half of what you have learned is outdated. Increasing the average worker, expecially in the more challenging skills of Aerospace, must find practical ways of continue learning in what is generally called adult education; and yet at the same time, he must avoid the same systems of education that have made teachers what they are not today! And the solution is obviously not in more and higher pieces of paper.

In this book, all four levels of learning will be encouraged and promoted, especially the highest levels of trial and error, also of the creative learning of problem solving.

1-8 Ups and Downs of UAV Testing by John Del Frate of NASA.[1]

You will enjoy for perspective on UAV testing the very personable report of Del Frate in newspaper/magazine format in February 2008, the "Abstract" of which is:

> If you could see the road ahead, you might just pass up a fantastic opportunity because you're blinded by the potential pitfalls. In my case, I was testing the project management waters at the NASA Dryden Flight Research Center after ten years of being a research engineer. I was an eager (but ignorant) rookie project manager (PM) and I was willing to engage in just about any project without knowing what it would entail. The assignment I accepted was to help NASA's Environment Research Aircraft and Sensor Technology (ERAST) Project, a partnership with a fledgling Uninhabited Aerial Vehicle (UAV) industry, to tackle stratospheric flight. I remember one of our industrial partners querying me about whether or not I understood what I was getting into. Like one of those bobble-head toys that have become quite popular, I nodded. But in reality, I didn't have a clue. His response was, "Hang on, it's going to be a wild ride." He was right. In retrospect, if I had clearly understood the ten years of pitfalls that were coming, I might not have "hung on." Now I can look back and say that I would not trade the experience for anything. The lows included the destruction of a number of UAVs on my watch. Later someone told me that we should not be surprised if we lost one UAV for every ten flights. We wrote many chapters in the book on what can go wrong with UAVs-and we are still writing. As you can imagine, each mishap was accompanied by an investigation. What an education!

Jerry V. McMichael
May 8, 2010
Portales, New Mexico

1-9: Drilling with MATLAB Basics.

For some these drills to follow, and at the end of many chapters, may be a review, and for others they may be a first time; however without the memori-

[1] Entitled "Old Journey, New Heights" in the NASA ASK Magazine.

zation, example of knowledge and skill in MATLAB and other program languages and algorithms, there can be no trial and error and creativity.

1. Some Basic Built In Data Analysis Functions in MATLAB.

MATLAB, among many other things, is made for basic data analysis with built-in functions like mean, median, std {standard deviation}, sum, trapz {trapezoidal integration}, and others[1]. You will begin to think, rightly so, that MATLAB is almost magic, but there is also a built-in function, not so much for data analysis but will allow us to quickly set up a matrix of data for analysis, which is v = magic(3), where the 3 in parenthesis stands for a 3x3 matrix of 3 columns and 3 rows.

Perhaps you know by now that MATLAB stands for MATrix LABoratory, and as such you will find much of the math, like in the basic and built-in functions of data analysis and also for the magic function are columns oriented. For example if you used the plot function on a magic function, it will automatically plot a line for each column as shown below.

2. Files, Functions, and Data Structure.

{INTRO p 79}

3. Histograms and Cumulative Curves.

{DATA p 9}

[1] You can find a more complete list of basic data analysis built-in functions on pg 21 of Andrew Knight’s BASICS OF MATLAB AND BEYOND, which by the way is also recommended as a good first course textbook in MATLAG for classroom or individual study. It is a publication of Chapman & Hall/CRC.