

USE OF ENGINEERING PACKAGES

1.0 INTRODUCTION

Use of engineering packages involves the application of computer and the appropriate software packages for solving mathematical equations related to the physics of engineering systems or processes to provide the desired numerical results and also supply the outputs in a graphical form for impressive presentations. The data or information can then be interpreted by the engineer for sound judgments in designing, analyzing and optimization of engineering systems or processes. Some of the engineering software packages readily available in market today include MATLAB, EES, EXCEL, MATCHAD, FEMLAB, FLUENT, etc.

The use of engineering packages enables engineers to solve problems easily and quickly. However, inadequate understanding of the software packages can lead to abuse and misinformation as well as colossal damage in engineering systems. Therefore, there is the need for students to understand the fundamentals of engineering packages and how to apply them in solving engineering problems.

1.1 Importance of engineering packages and programming to engineers

1. They enable engineers to solve problems easily and quickly as possible.
2. They reduce drudgery in computation.
3. They reduce the computation time thereby allowing engineers more time for interpretation of results.
4. They provide graphics for impressive presentation of information.
5. They enhance engineer's decision making concerning the design, construction, operation and maintenance of engineering systems.
6. Some software packages can be used as hand calculator e.g MATLAB or EES.
7. Packages like MATLAB and EES offered engineers the opportunity to extend the capability by writing computer programs.

1.2 Engineering packages

Engineering packages are application software used by scientists and engineers to solve numerical problems to facilitate good decision making in developing systems. Packages are built to simplify interaction with the users by using a menu to initiate operation or entering the data through templates.

Engineering software packages are limited by capabilities found in their standard mode of operation. For example, a system of linear equations or plot of x - y values can be solved with either Excel or MATLAB or EES software. This is because of the little effort involved, hence most users tend to adopt this mode of operation.

However, when problems arise that is beyond the standard capability of the tool, users are advice to choose from the following alternatives:

1. Search for a different package and see if it is capable of solving the problem.
2. Learning how to write computer programs that allow the user to extend the capabilities of the tool.
3. Learning the basic skills required for linking one software package with another.

1.3 Computer programs

Computer programs are set of instructions that direct the computer to perform a certain task. The grammatical structures are syntax and semantics, which give the symbols and rules used to compose statements and the meaning associated with the statements. For a program to be executed, it must be translated into a sequence of basic machine instruction.

User's statements are typed on a keyboard for entry into machine-readable medium. The translator (compiler) program accepts these statements as input and translates (compiles) them into a sequence of basic machine instructions which form the executable version of the program. After that, the translated (compiled) program can then be run. The process whereby the user supplies information to a computer via command or with menu option is referred to programming.

Typically, a computer program contains the following:

1. Simple information representation (constants, variables, and type declarations).
2. Advanced information representation (data structure, arrays, and records).
3. Mathematical formulas (assignment, priority rules, and intrinsic functions).
4. Input/output.
5. Logical representation (sequence, selection, and repetition).
6. Modular programming (functions and subroutines).

1.4 Structured programming

Structured programming is set of rules that prescribe good style for writing programs in order to ensure that the programs are clear and easy to understand. A well-structured program is

easier to debug and test. It is also flexible enough to allow considerable creativity and personal expression. Its rules impose enough constraints to render the resulting codes far superior.

The key idea behind structured programming is that any numerical algorithm can be composed using the three fundamental control structures: sequence, selection, and repetition. By these structures, the resulting computer code will be clearer and easier to follow.

1.5 Logical representation

1.5.1 Sequence.

The sequence structure expresses the idea that unless the user directs it otherwise, the computer code is to implement one instruction at a time. The structure can be expressed as a flowchart or pseudocode.

1.5.2 Selection

Contrary to the step-by-step sequence structure, selection provides a means of splitting the program's flow into branches based on the outcome of a logical condition. The two fundamental ways for doing this are *IF/THEN* and *IF/THEN/ELSE* structures.

The *IF/THEN* structure also called single alternative decision allows for a detour in the program flow if a logical condition is true. If it is false, nothing happens and the program moves directly to the next statement following the *ENDIF*.

The *IF/THEN/ELSE* structure also known as double-alternative decision behaves in the same manner for a true condition. However, if the condition is false, the program implements the code between the *ELSE* and the *ENDIF*.

1.5.3 Repetition

Repetition can also be referred to as looping. It provides a means to implement instructions repeatedly. Loop is divided into two based on how they are terminated.

The first and most fundamental type is called a decision loop because it terminates based on the result of a logical condition.

The other is *DOEXIT* construct, also called a break loop. This structure repeats until a logical condition is true.

1.6 Modular programming

It involves the breaking down of computer programs into small or subprograms or modules that can be developed and tested separately.

Important feature of modules are:

1. They can be as independent and self-contained as possible.
2. They are typically designed to perform a specific and well-defined function.
3. They have one entry and one exit point.
4. They are usually short and highly focused.

Advantages of modular programming

1. Program maintenance and modification are facilitated. This is primarily due to the fact that new modules can be developed to perform additional tasks and then easily incorporated into the already coherent and organized scheme.
2. The use of small, self-contained units makes the underlying logic easier to devise and to understand for both the developer and the user.
3. Development is facilitated because each module can be perfected in isolation.
4. Modular design also increases the ease with which a program can be debugged and tested because errors can be more easily isolated.

2.0 Engineering packages

Excel

Excel is the spreadsheet produced by Microsoft, Inc. Spreadsheets are a special type of mathematical software that allows the user to enter and carried calculations on rows and columns of data. Excel has some built-in numerical capabilities including equation solving, curve fitting, and optimization. It also has several visualization tools, such as graphs and three-dimensional surface plots that serve as valuable tools for numerical analysis.

Illustrative example:

We will show how these capabilities can be used to solve the parachutist problem given as

$$V = \frac{gm}{C_d} \left(1 - e^{-\frac{C_d t}{m}} \right)$$

Where: g = acceleration due to gravity (9.81 m/s), m = mass of parachute (68.1 kg), C_d = drag coefficient (12.5) and t = time (0 to 3 s).

Solution

The first step involves entering labels and numbers into the spreadsheet cells as shown below

	A	B	C	D
1	Parachutist Problem			
2				
3	m	60.1 kg		
4	cd	12.5 kg/s		
5	dt	0.1 s		
6				
7	t	vnum (m/s)	vanal (m/s)	
8		0		
9		2		

Move to cell C8 and enter the analytical solution

$$=9.8*m/cd*(1-exp(-cd/m*A8))$$

When this formula is entered, the value 0 should appear in cell C8. Then copy the formula down to cell C9 to give a value of 16.405 m/s. All the above is typical of the standard use of Excel. For example, at this point you could change parameter values and see how the analytical solution changes.