"the subprograms form a software package which is readily applicable to a broad range of engineering problems."

Applying the Computer in the Solution of Electrical Engineering Problems*

by

Miguel T. Escoto, Jr. **

ABSTRACT

This paper lists some of the potential areas where computers might be applied in the solution of electrical engineering problems.

Specifically, it briefly describes the goals that were used as guides in the development of a computer oriented circuit analysis course (COCA) offered at the University of the Philippines, Department of Electrical Engineering. An example program demonstrates some points of interest pertinent to the applications of microcomputers for digital computations in basic circuit theory.

INTRODUCTION

The decreasing cost of computers and wide availability of software make the computer today a powerful computing resource within the reach of many. Once upon a time, only research scientists could understand these machines, crude as they were. Today, better technology and the emergence of high level languages and application software have brought the computer out into public view and widespread use.

Computers find use as simple aids in many areas some of which include statistics, finance, business, communications, technical uses, office use, industrial uses, space vehicle, and school use.

In most of these areas, computers are used either for their processing powers, bookkeeping powers, computing powers or an application mix. One enters data, does processing with rules prescribed in some manner beforehand, then derives information for immediate consumption or storage. The computer can provide timely and accurate information and interfaces that convey these information to the human being in understandable form.

And yet another possible use of the computer may stem in its use as the object under study and investigation rather than as the aid or tool for study. The computer may be dissected for its architecture, hardware aspects, software, microprogram, interfacing, instruction set, peripheral support, real time applications, memory system, etc. as might the case be in computer-related electrical engineering courses.

---

*A paper presented in a Seminar-Workshop on Computers in Engineering Education, September 19-20, 1986, Development Academy of the Philippines.

** Electrical Engineering Department, College of Engineering, University of the Philippines.

In spite however of its potential uses, barriers still prevent some schools from fully utilizing the computers as tools or objects for learning and instruction. Its underutilization as a tool or as an aid may well be attributed to the lack of software learning packages in the various engineering disciplines. Thus far, most software available locally deal with either games or business packages. Very little include hard core technical or engineering use.

Another possible explanation is that there is gap between mere knowledge of programing and the knowledge of finding a solution via programing. Both these aspects are equally important if one has to apply microcomputers in the solution of problems successfully. The effort falls short with the lack of one or the other.

In the application of computers, a third problem arises with respect to data structures, i.e., data representation for network topology and parameters. The problem is therefore three-sided: knowing the theory for the particular area of concern which suggests the step-by-step procedure towards a solution, knowing programing and an associated language for expression, and data representation. In short, successful programs equal data structure + algorithm + language.

SOME ELECTRICAL ENGINEERING APPLICATION AREAS

Listed below are some potential application areas of computers.

Electronics and Communications:
- Digital Filtering
- Spectral Analysis
- Non-linear Distortion Analysis
- Active and Passive Filter design

Electronic Measurements and Instrumentation:
- Loading Error Analysis
- Statistics of Measurements
- Regression Analysis
- Linearization Problems

Electronic Control and Simulation:
- Time Response Analysis
- Frequency Response Analysis
- State Variable Analysis
- Discrete Time Studies
- Root Locus Analysis

Network and Circuit Theory:
- Convolutional Methods
- Sensitivity Analysis
- AC and DC Analysis
- Transient Analysis
- Non-linear Analysis
- Network Analysis

Computer Engineering:
- Algorithmic State Machine Design
- State Chart Analysis
- Minimization Problems
- Digital Logic Simulation
- High-level Language Constructs
- Performance Simulations on Local Area Networks

SOME GUIDELINES FOR MATERIAL DEVELOPMENT

In view of the aforementioned gaps, the following goals were used by the author in the development of the necessary materials for a course in computer oriented circuit analysis (COCA) at the University of the Philippines. Most of the guidelines are also quite suitable for the other application areas.

1. The emphasis of the presentation should be on the application of numerical techniques to engineering problems rather than on the techniques themselves. For example, numerical integration and numerical differentiation equation solving techniques are presented as a means of relating voltage and current variables in inductors and capacitors.

2. The student should be motivated by being shown how numerical techniques make it possible to solve problems which cannot be solved by non-numerical analysis techniques. For example, numerical-differential equation solution techniques are shown to give solutions for variables of networks with time varying and non-linear elements as well as for those containing only linear time-invariant components.

3. The student should be provided with tools that he can apply all through his undergraduate curriculum. This goal has been implemented by developing all the numerical techniques as subprograms. Taken as a group, these subprograms form a software package which is readily applicable to a broad range of engineering problems.

4. The programs and subprograms that are developed should be kept as simple as possible. This was done so that emphasis could be placed on theory rather than on sophisticated programing. This makes absorption of the material an easier task for the average student. The exceptional student, meanwhile, is easily motivated to undertake his own improvement of the programs to any desired level of sophistication.

5. The problems and examples should be used to develop the student's confidence in results obtained from numerical techniques and to emphasize the need to substantiate his results. To implement this, many of the problems have been chosen so that the answers can be verified by direct, non-numerical analysis.

6. The problems should be used to supplement the student's knowledge of the basic course material, i.e., circuit theory. As one example, there are parallel problems showing how both time response and frequency response of various networks are changed when the network parameters are varied.

7. Attention should be given to the verification of the accuracy of the programs and the workability of the assigned problems. To achieve this, the instructor has to fully work out the problems.

8. Since most students of the college have taken a course in Fortran Programing, the course does not begin with a formal presentation of the basic details of the Fortran language. Instead some of the more advanced Fortran topics such as the use of subroutines and subprograms, common statement, etc., which are usually treated superficially in basic texts, are used to illustrate top-down, bottom-up structure programing. This eases the assembly of larger programs later on in debugging.

9. The material should be usable in as many different types of computing installations or resources as possible, both large and small. To help insure this, the programs and subroutines have been made compatible with FORTRAN IV compilers. These are readily available on personal computers as well.

Other languages may also be used such as PASCAL or Basic. The choice has to be made before actually presenting the material in the classroom. Showing the structure or flow charts of the program may facilitate translating from one language to another.

10. The programing techniques should be illustrative of those required in developing large programs. Again, the extensive use of subroutines helps accomplish this goal. One, however, has to pay attention to the limited memory capacity of personal computers.

EXAMPLE PROGRAM

The following program demonstrates how an integral may be evaluated using the trapezoidal technique. The last section shows how one might use the computer to evaluate a convolution integral using FORTRAN.

The Subroutine ITRPZ

This subroutine is used to integrate a time-domain function whose mathematical description is implemented in a Fortran function with the identifying statement FUNCTION Y (X). The ITRPZ subroutine is called by the Convolution demo program.

Identifying Statement: SUBROUTINE ITRPZ (XA, XB, ITER E)

Purpose: To find the definite integral of a specified function y (x), that is, to find E, where

$$E = \int_{XA}^{X_B} y\,(x)\,dx$$

using trapezoidal integration.

Additional Subprograms Required:
This subroutine calls the function identified by the statement

FUNCTION Y (X)

This function must be used to define y (x) i.e., the integrand.
Input Arguments:

- XA the lower limit of the variable of integration
- XB the upper limit of the variable of integration
- ITER the number of trapezoidal segments used to approximate the area.

Output Argument:

- E the value of the integral

```
C
C TRAPEZOIDAL INTEGRATION
      SUBROUTINE ITRPZ (XA, XB, ITER, E)
      HITER=ITER
      DX=(XB-XA)/HITER
      E=Y (XA) / 2.
      X=XA
      DO 100  I= 2, ITER
      X=X+DX
100   E=E+Y (X)
      E=(E+Y (XB)/2.)  DX
      RETURN
      END
```

Example Main Program: Solving a Convolution Integral

Problem:

A sinusoidal pulse is applied to the input terminals of a series RC network whose time constant is one second. Find the resulting voltage across the capacitor. The input pulse is Vi (t) = sin (t) for $0 < t < 3.14159$ and Vi (t) = 0 otherwise.

```
C CONVOLUTION PROGRAM
      DIMENSION A (5,100)
      COMMON T
      T=0.0
      DT=0.1
      A (1, 1) = 0.0
      DO 100  I = 2,100
      T=T+DT
      XA=0.0
      IF (T. GT. 3.14159) XA = T – 3.14159
      ITER=25
 4    XB=T
      CALL ITRPZ (XA, XB, ITER, E)
      A(1, 1)=E
      WRITE (3,105) I, T, E
 105  FORMAT (14, 2E15.8)
 100  CONTINUE
      STOP
      END

C IMPULSE RESPONSE FUNCTION
      FUNCTION H (T)
      H=EXP (–T)
      RETURN
      END

C FORCING FUNCTION F(T)
      FUNCTION F (T)
      F=SIN (T)
      RETURN
      END

C INTEGRAND FUNCTION Y (X)
      FUNCTION Y (TAU)
      COMMON T
      Y=H (TAU) * F (T-TAU)
      RETURN
      END

C TRAPEZOIDAL INTEGRATION
      SUBROUTINE ITRPZ (XA, XB, ITER, E)
      HITER=ITER
      DX=(XB-XA)/HITER
      E=Y (XA)/2.
      X=XA
      DO 100   I=2, ITER
      X=X+DX
 100  E=E+Y (X)
      E=(E+Y (XB) /2.) * DX
      RETURN
      END
```

The use of subroutines readily extends the range of the circuit analysis techniques developed. All that is required is to add relatively simple main programs. In general, the subroutines have been designed with the minimum level of sophistication necessary to effectively perform their function. The obvious advantage of such an approach is the minimization of computer memory requirements as well as the reduction of compilation time and execution times. The student or professor who wishes a higher level of performance will find that embellishments may be readily added.