CHAPTER III

SOFTWARE DEVELOPMENT

Because this software is developed by using an Object-Oriented Programming concept, this chapter will discuss about this technique. The tools and flow diagrams in developing process are also mentioned here.

3.1 Object-Oriented Programming, OOP

The Object-Oriented Programming is a new programming technique. It is a structured design technique, but it also offers the possibility of reusing components. Objects, polymorphism, and inheritance are the common terms using in this technique.

Object :

Object in Object-Oriented Programming technique has basic properties as follows:

- It is an abstraction of something in the system.
- It has certain attributes and provides certain services.
- It can occur any number of times in the system.
- It can be classified as classes, and a number of classes can belong to a superclass (for example the **car**, **bus**, and **aircraft** objects might belong to a **transport** superclass).
- It may inherit attributes and services from its parents in the classification structure as shown in figure 3.1-1.

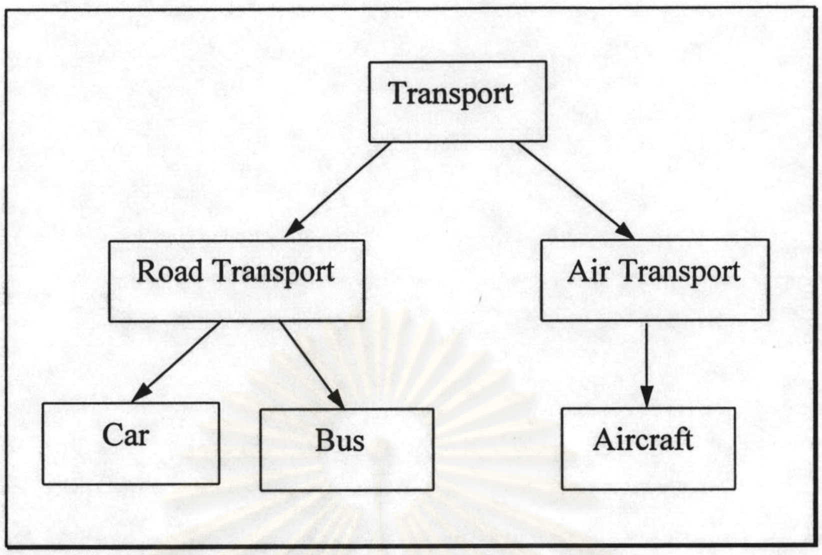


Figure 3.3-1 Example of an inheritance hierarchy.

In programming languages, objects are the collections of operations that share a state. Each object is an instance of a "class". Variables representing the internal state of an object are called "instance variables" and its operations are called "methods". After receiving a message, the object will respond by invoking the appropriate method.

Polymorphism:

This term means that one name can be used for several relations but slightly different purposes. For example, calling any executed programs in Microsoft Windows from program manager uses the same method by double-clicking on the icon. But it has different purposes depend on which program a user wants to run. In programming, polymorphism enables the instance of different classes to response to the same function in ways that are appropriate to each class.

Inheritance:

Inheritance is the process by which one object can acquire the properties of another object. You can derive a class from another in Object-Oriented languages. This property helps the programmers to develop application software more conveniently.

3.2 Tools for Development

The tools used for developing this CAL consist of two major parts: hardware and software.

3.2.1 Hardware

The hardware used in this work is

1) An IBM compatible personal computer with Pentium processors with 16 MB RAM, 520 MB hard disk and color VGA monitor display.
2) A high resolution scanner was used for scan pictures required by this project.

3.2.2 Software

Two major software that used in this work are Turbo C++ for Windows package and Help Compiler. The former was used for developing the calculation section and constructing Graphic User Interfaces. The later was used for creating keywords, hot spot areas and related topic jump, which used hypertext technique, in theory section.

Furthermore, this research used various packages such as PhotoShop, PaintBrush, and PC PaintBrush to generate graphics. Word Processors, for example, Microsoft Word and AmiPro, were used for creating text files for theory section. Hot Spot Editor was used for building hot spot areas in pictures and Notepad Editor was used for creating help project files.

3.3 Software Design

3.3.1 Major Concept

The procedures in developing this software is shown in figure 3.3-2.

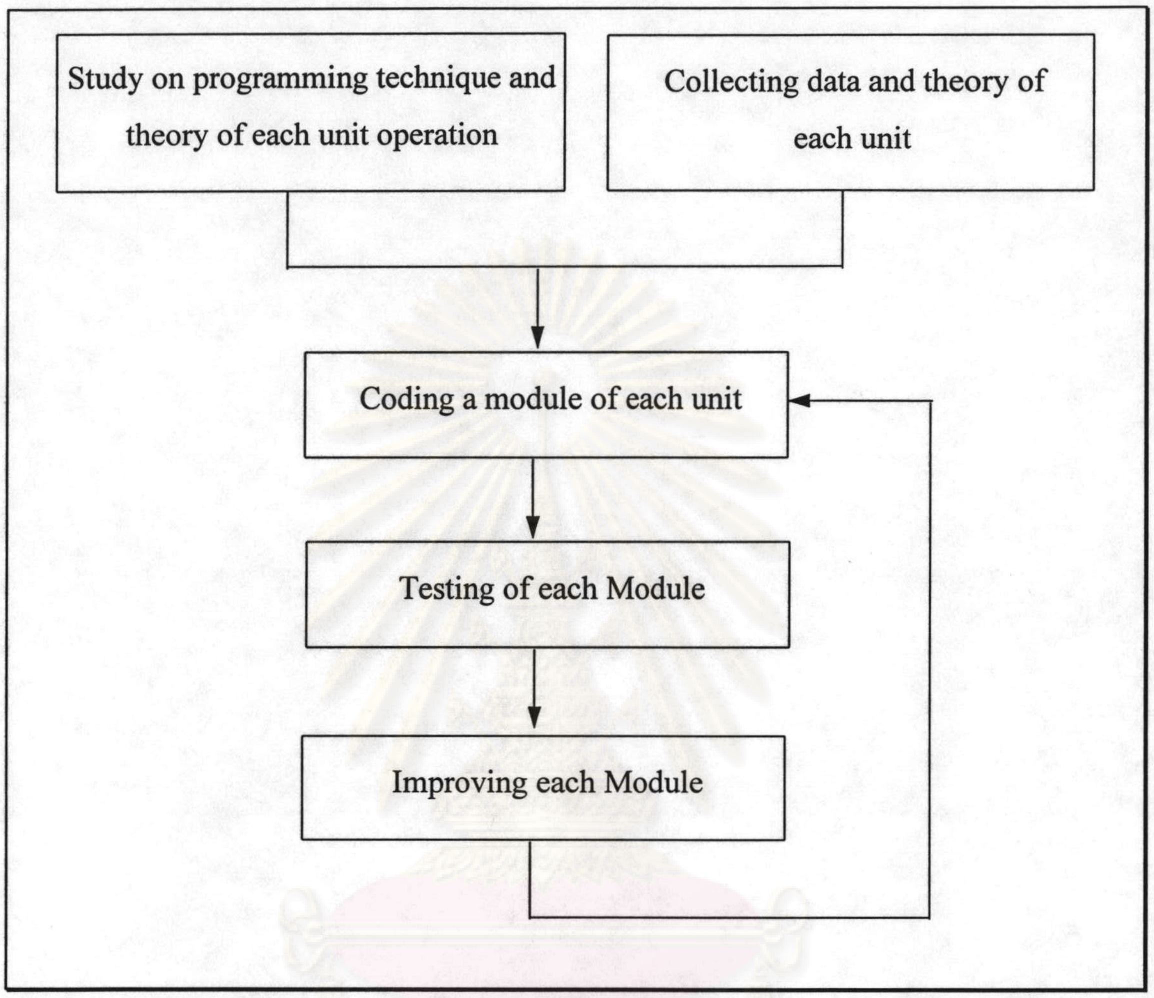

Study on programming technique and theory of each unit operation
Collecting data and theory of each unit
Coding a module of each unit
Testing of each Module
Improving each Module

Figure 3.3-2 Software development procedures diagram.

The major conceptual procedures are as follows :

Step 1. Study on Turbo C++ for Windows and Help Compiler programming technique, including study on a related technologies.

Step 2. Collecting data and theory about the selected experimental units.

Step 3. Coding a source program of each part of the system such as calculation program, GUI controlling program, and display controlling program in theory section.

Step 4. Testing this program and comparing calculation result with other calculation tools.

Step 5. Improving this CAL system by modifying the program in step 3.

As mentioned above, this work will develop each module for two sections in programming process. Each module had the same concept of programming, but differed in data and equations especially related to each unit operation. (See figure 3.3-3)

1) Calculation Section

This section was developed using Turbo C++ for Windows as a compiler to create the Windows applications. The details in developing are :

1. Creating the GUI, such as Dialog box, Menu, Icon, Bitmap, and Cursor by using Resource Workshop in .RC format.
2. Coding calculation and GUI controlling modules by using Turbo C++ for Windows Editor and save in .CPP format.
3. Creating definition file and save in .DEF format.
4. Creating project file (.PRJ) then compile and build all the files. After build all, execute file (.EXE) received and this file can use as one Windows application.

2) Theory Section

The procedure of developing this section are as follows :

1. After collect all data required for the modules, text data is saved in . RTF formatted files by using Word Process and pictures are saved in .BMP formatted file. Pictures with hot spot areas will be saved in .SHG formatted file by using Hot Spot Editor.

2. Creating help project file (.HPJ) required by Help Compiler.

3. Compiling the help project file using Help Compiler. The results is a help file (.HLP) that contains text and pictures collected in step 1., as well as keywords and hot spot areas for jumping to related topic.

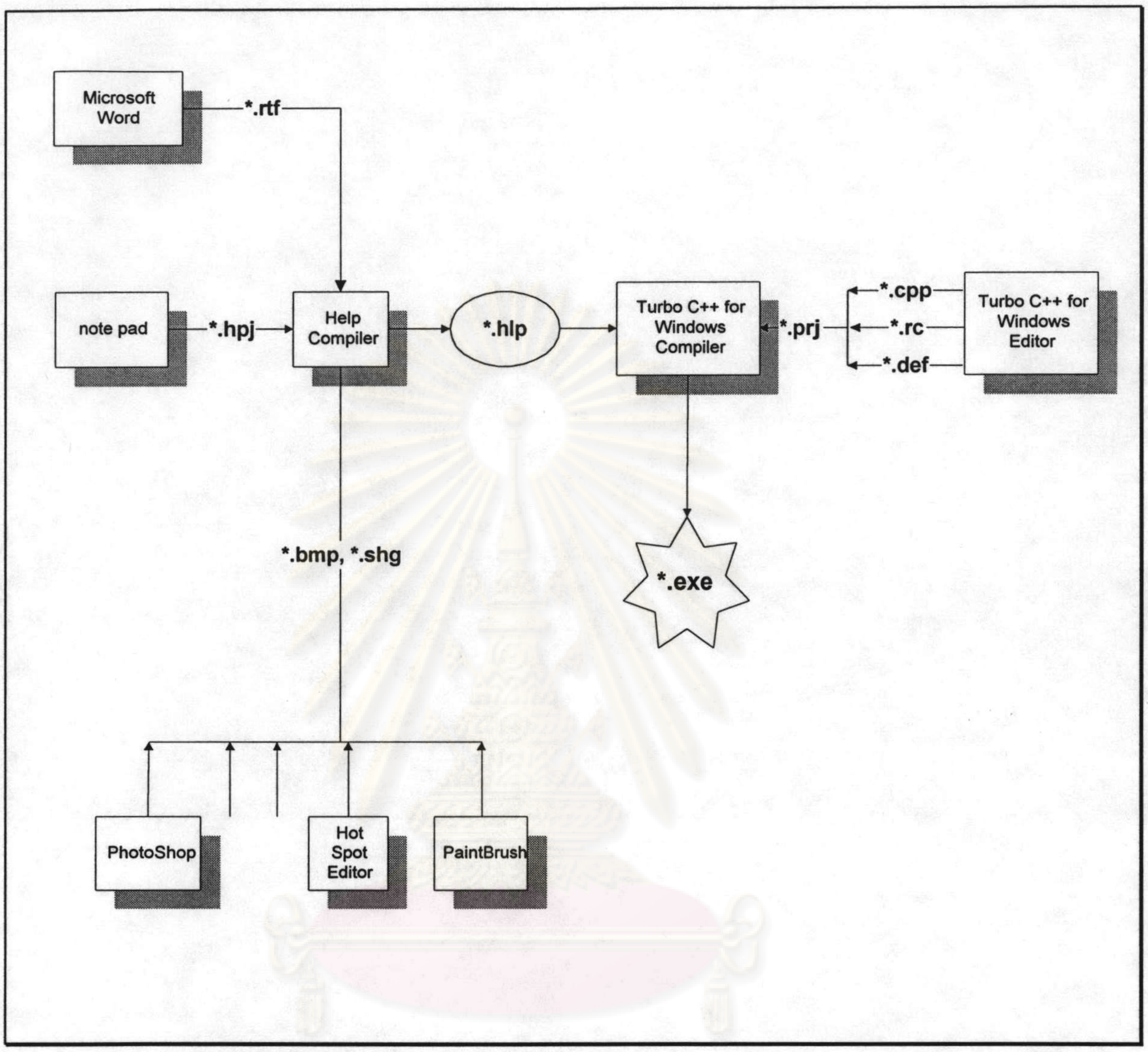


Figure 3.3-3 Programming process diagram.

3.3.2 Detail of Each Module

1) Calculation Section

Each calculation module for each experimental unit is difference in data input and procedures of calculation, including the results of calculation. The flow diagrams of each module are shown in figure 3.3-4 to 3.3-10.

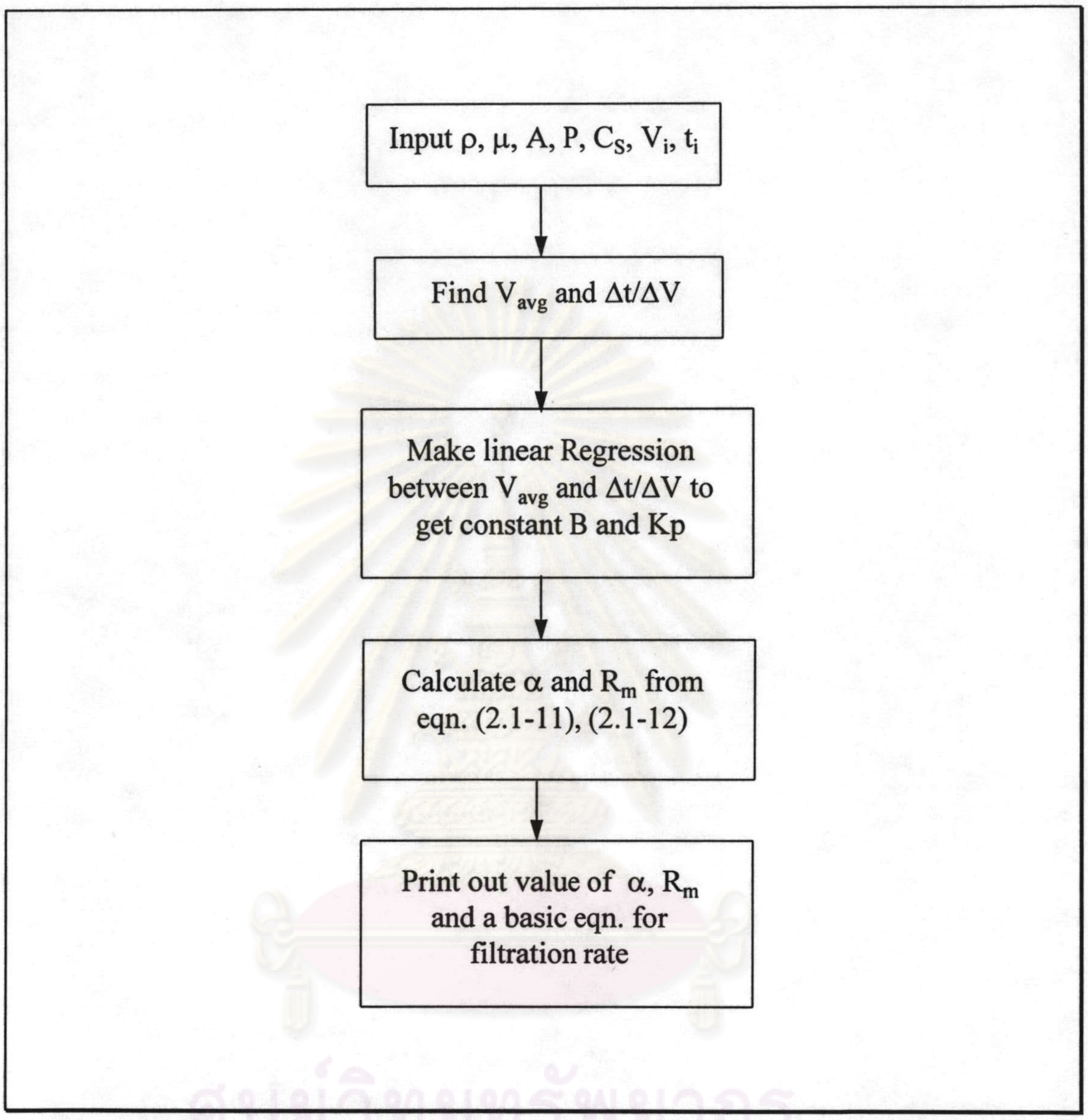

Input ρ, μ, A, P, C_S, V_i, t_i
Find V_avg and Δt/ΔV
Make linear Regression between V_avg and Δt/ΔV to get constant B and Kp
Calculate α and R_m from eqn. (2.1-11), (2.1-12)
Print out value of α, R_m and a basic eqn. for filtration rate

Figure 3.3-4 Flow diagram for Filtration unit.

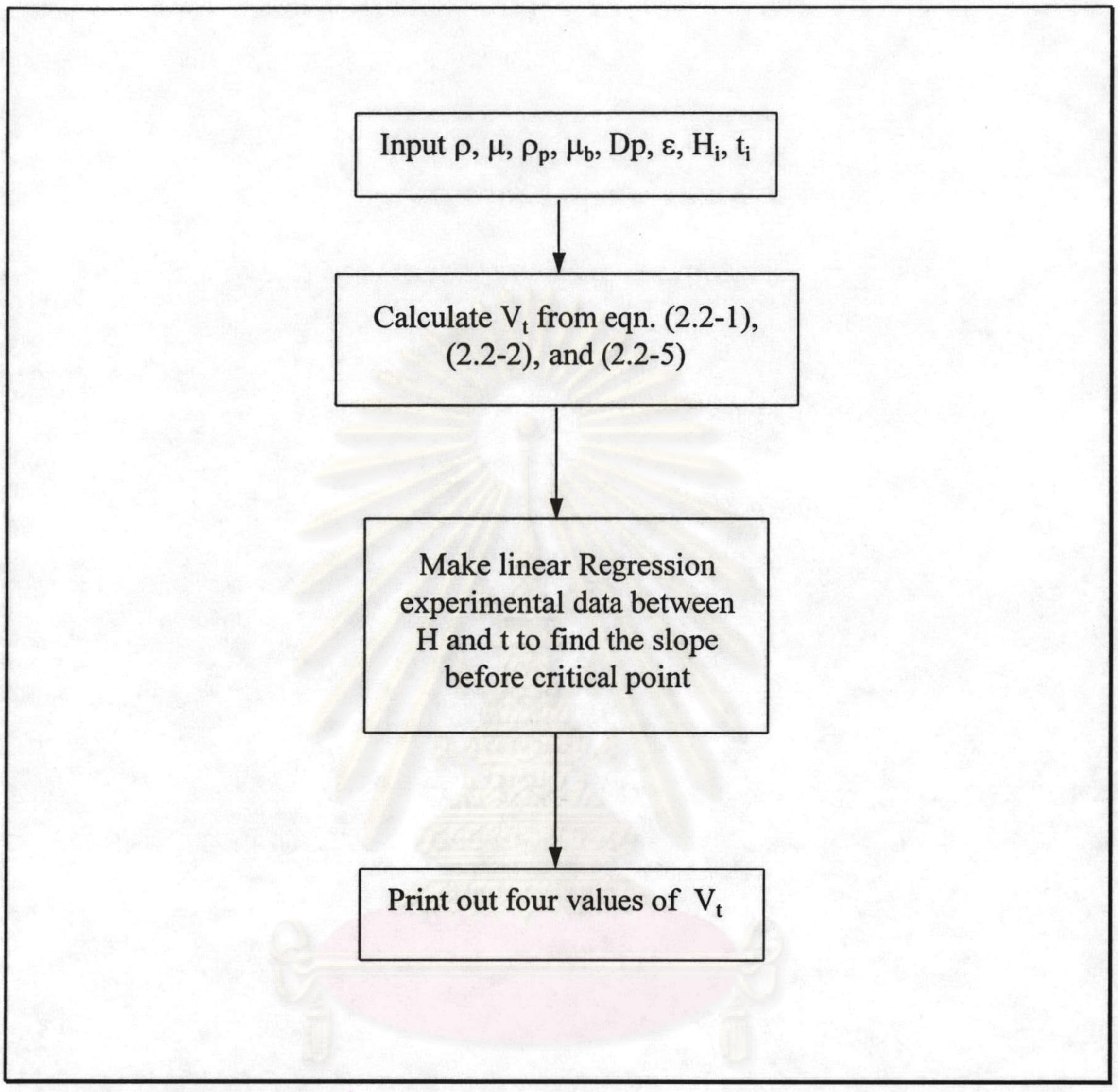
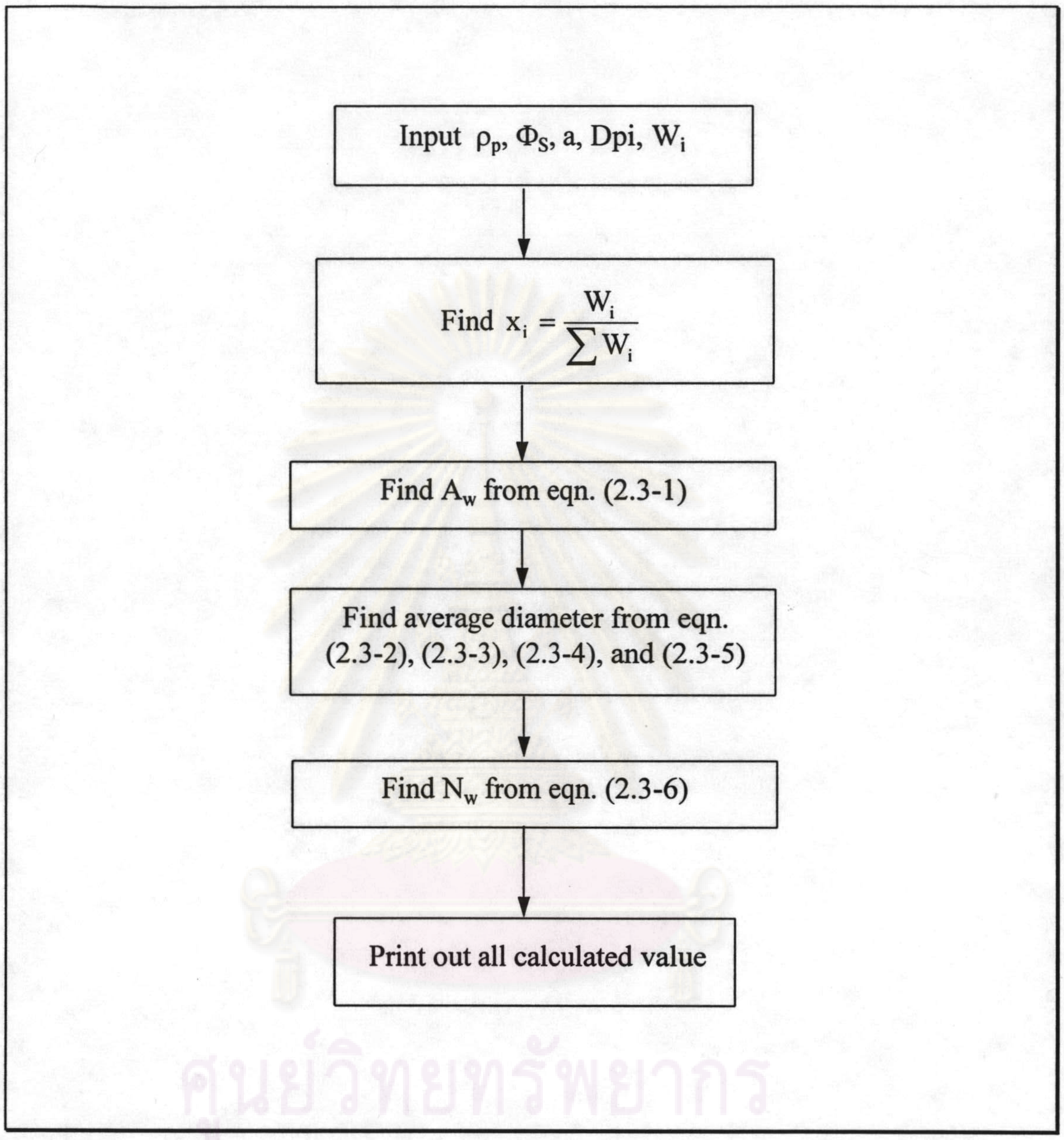

Input ρ, μ, ρ_p, μ_b, Dp, ε, H_i, t_i
Calculate V_t from eqn. (2.2-1), (2.2-2), and (2.2-5)
Make linear Regression experimental data between H and t to find the slope before critical point
Print out four values of V_t

Figure 3.3-5 Flow diagram for Sedimentation unit.



Input ρ_p, Φ_S, a, Dpi, W_i
Find $x_i = \frac{W_i}{\sum W_i}$
Find A_w from eqn. (2.3-1)
Find average diameter from eqn. (2.3-2), (2.3-3), (2.3-4), and (2.3-5)
Find N_w from eqn. (2.3-6)
Print out all calculated value

Figure 3.3-6 Flow diagram for Sieve Analysis unit.

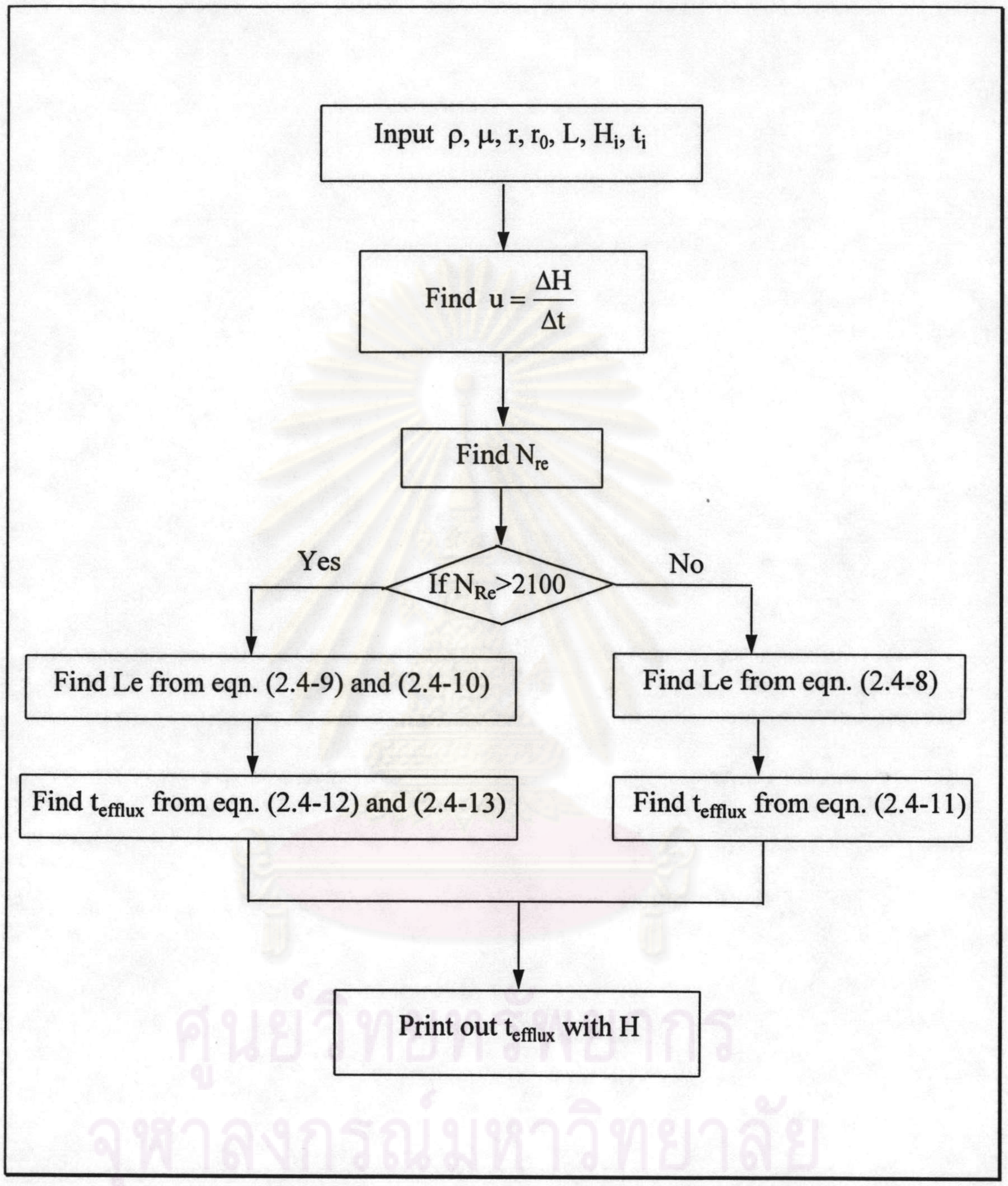

Input $\rho, \mu, r, r_0, L, H_i, t_i$
Find $u = \frac{\Delta H}{\Delta t}$
Find N_{re}
If $N_{Re}>2100$
Yes
No
Find Le from eqn. (2.4-9) and (2.4-10)
Find Le from eqn. (2.4-8)
Find t_{efflux} from eqn. (2.4-12) and (2.4-13)
Find t_{efflux} from eqn. (2.4-11)
Print out t_{efflux} with H

Figure 3.3-7 Flow diagram for Efflux Time of a Tank unit.

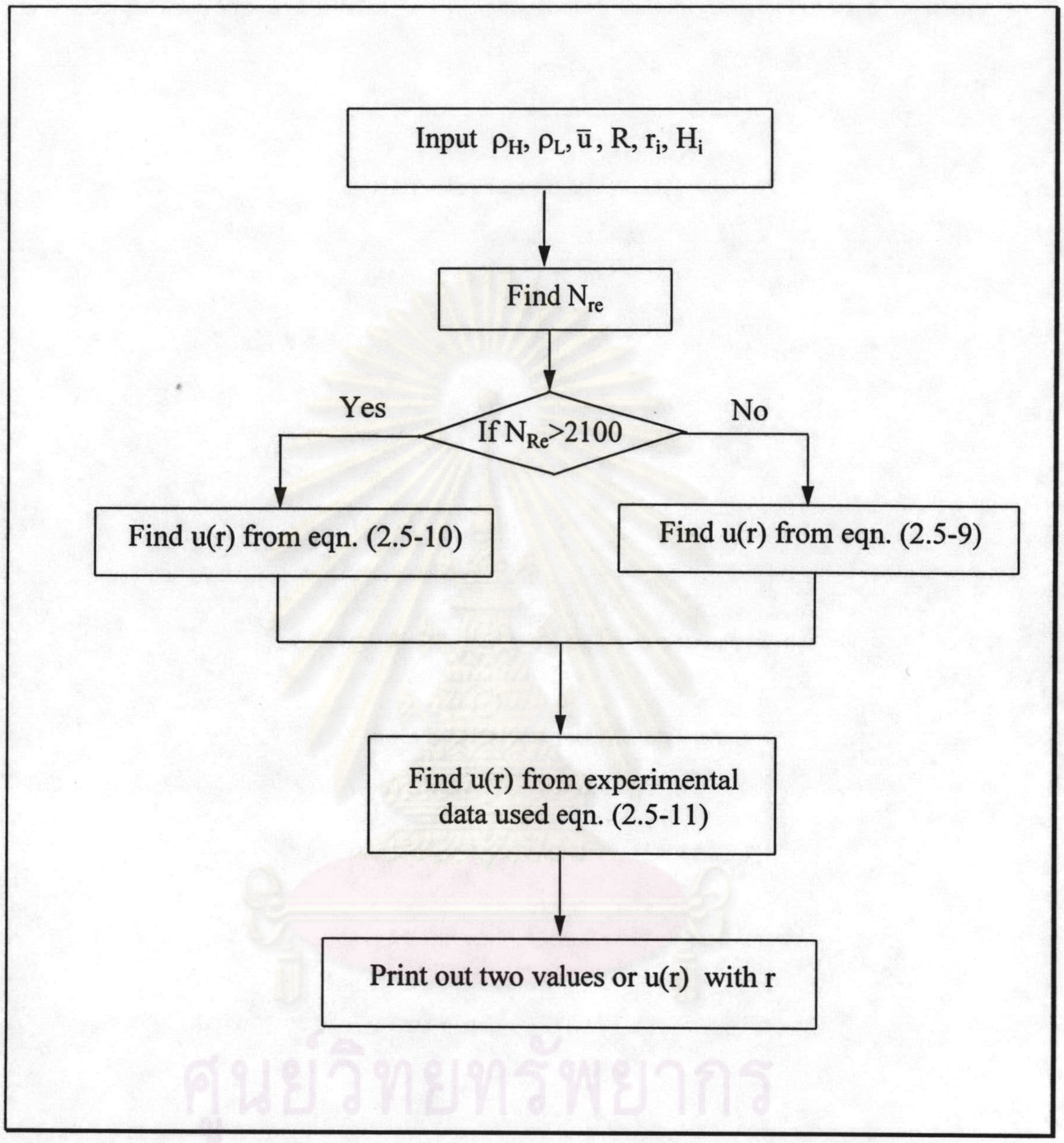
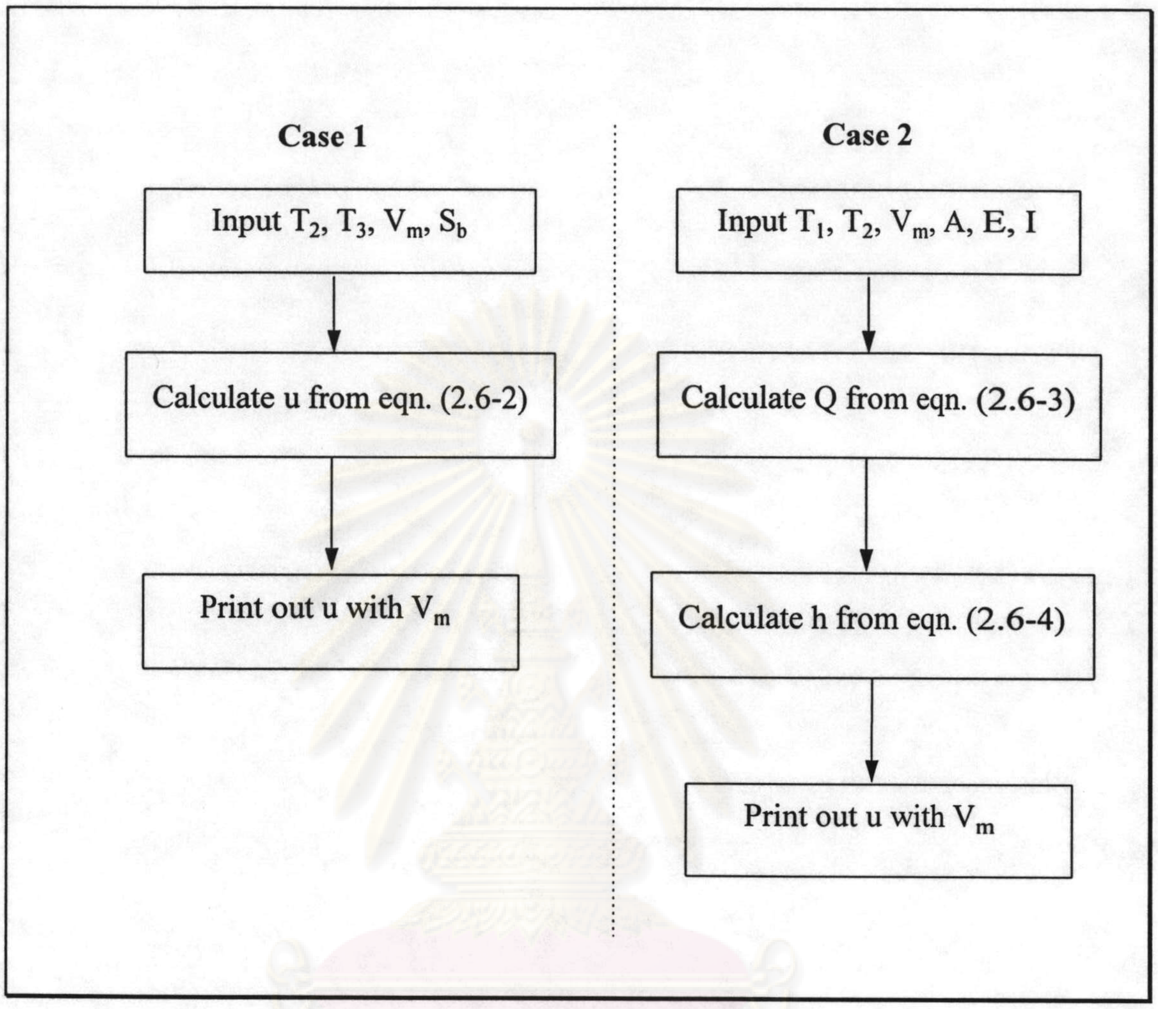
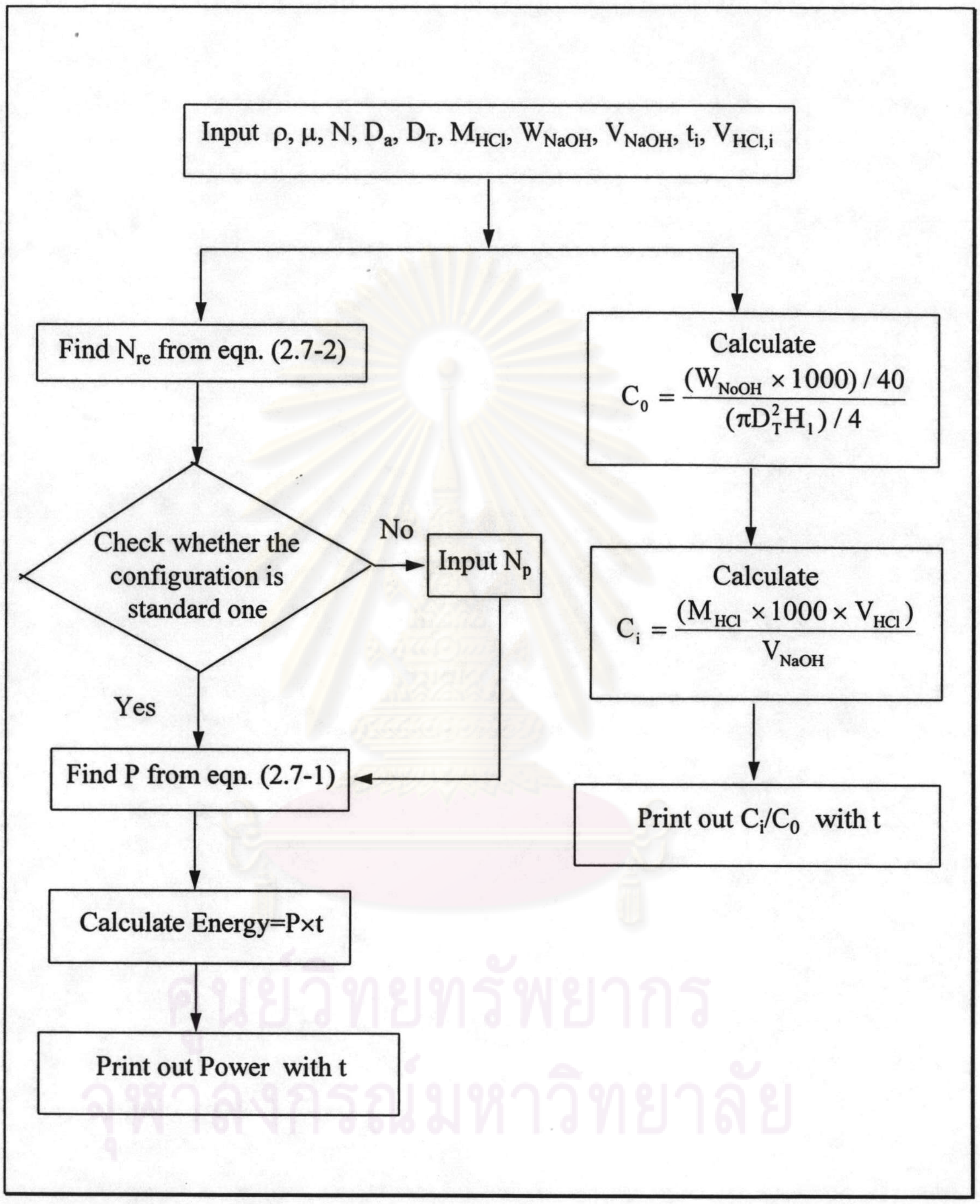

Input ρ_H, ρ_L, $\bar{u}$, R, r_i, H_i
Find N_{re}
If N_{Re}>2100
Yes
No
Find u(r) from eqn. (2.5-10)
Find u(r) from eqn. (2.5-9)
Find u(r) from experimental data used eqn. (2.5-11)
Print out two values or u(r) with r

Figure 3.3-8 Flow diagram for Flow in Pipe unit.



Case 1
Input T_2, T_3, V_m, S_b
Calculate u from eqn. (2.6-2)
Print out u with V_m
Case 2
Input T_1, T_2, V_m, A, E, I
Calculate Q from eqn. (2.6-3)
Calculate h from eqn. (2.6-4)
Print out u with V_m

Figure 3.3-9 Flow diagram for Fluidization unit.



Input ρ, μ, N, D_a, D_T, M_{HCl}, W_{NaOH}, V_{NaOH}, t_i, $V_{HCl,i}$
Find N_{re} from eqn. (2.7-2)
Check whether the configuration is standard one
No
Input N_p
Yes
Find P from eqn. (2.7-1)
Calculate Energy=P×t
Print out Power with t
Calculate $C_0 = \frac{(W_{NoOH} \times 1000)/40}{(\pi D_T^2 H_1)/4}$
Calculate $C_i = \frac{(M_{HCl} \times 1000 \times V_{HCl})}{V_{NaOH}}$
Print out C_i/C_0 with t

Figure 3.3-10 Flow diagram for Mixing unit.

Note : All input data will change to SI unit system before calculations.

2) Theory Section

Each module has the same structure of database as shown in figure 3.3-11. It has many topics in contents and some topic may have subtopics. It also has many equations shown as a bitmap with hot spots.

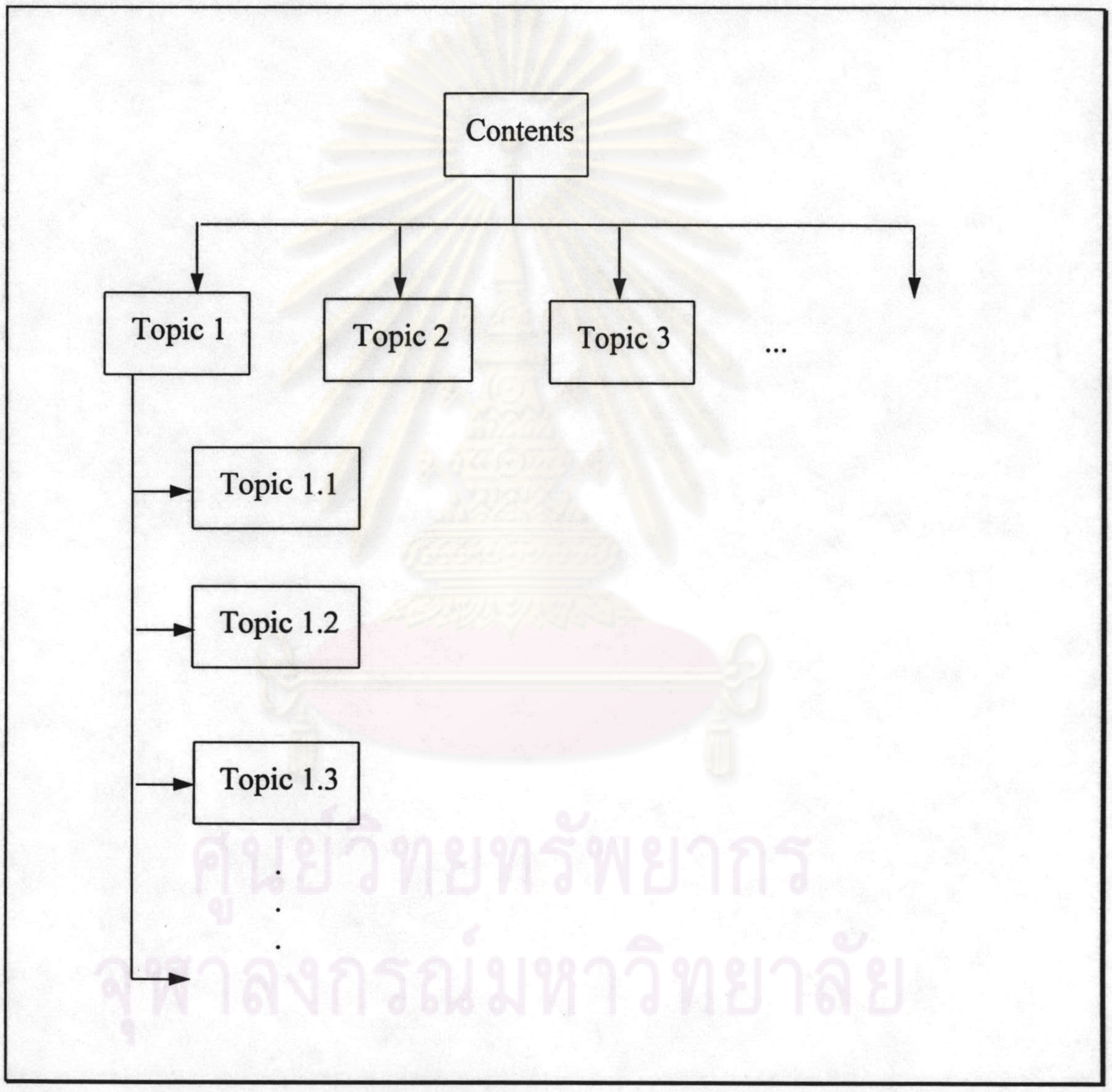


Figure 3.3-11 Theory section structure.