THESIS

OPTIMIZATION OF A SOLAR-ASSISTED DRYING SYSTEM FOR DRYING BANANAS

ROSARIN SMITABHINDU

GRADUATE SCHOOL, KASETSART UNIVERSITY

2008

THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Doctor of Engineering (Industrial Engineering)
DEGREE

Industrial Engineering — **FIELD**

Industrial Engineering — **DEPARTMENT**

TITLE: Optimization of a Solar-Assisted Drying System for Drying Bananas.

NAME: Ms. Rosarin Smitabhindu

THIS THESIS HAS BEEN ACCEPTED BY

P. Charn — **THESIS ADVISOR**
(Associate Professor Peerayuth Charnsethikul, Ph.D.)

Vira Chankong — **COMMITTEE MEMBER**
(Associate Professor Vira Chankong, Ph.D.)

COMMITTEE MEMBER
(Associate Professor Saeree Svetasreni, Ph.D.)

COMMITTEE MEMBER
(Associate Professor Prapaisri Sudasna-Na-Ayudthya, Ph.D.)

Serm Janjai — **COMMITTEE MEMBER**
(Associate Professor Serm Janjai, Ph.D.)

Anan Mungwattana — **DEPARTMENT HEAD**
(Associate Professor Anan Mungwattana, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON 5 June 2008

Gunjana Theeragool — **DEAN**
(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

OPTIMIZATION OF A SOLAR-ASSISTED DRYING SYSTEM FOR DRYING BANANAS

ROSARIN SMITABHINDU

**A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Doctor of Engineering (Industrial Engineering)
Graduate School, Kasetsart University
2008**

Rosarin Smitabhindu 2008: Optimization of a Solar-Assisted Drying System for Drying Bananas. Doctor of Engineering (Industrial Engineering), Major Field: Industrial Engineering, Department of Industrial Engineering. Thesis Advisor: Associate Professor Peerayuth Charnsethikul, Ph.D. 212 pages.

This paper presents a mathematical model for optimal design of a solar assisted drying system and the optimization model consists of a simulation model of a solar assisted system combined with an economic model. The simulation model consists of two systems of differential equations: one of the collector and other for the dryer cabinet and these systems of the differential equation are solved using the finite difference method. Values of the parameters of the model are determined experimentally. A computer program in FORTRAN is developed to simulate the model. The model was validated by comparing the simulation results with the experimental results and agreement was good. This simulation model was used for the optimization of solar assisted drying system.

The economic model is a mathematical model of the annual drying cost and the optimization problem is defined as the optimization of the dryer geometry and drying process so as to minimize the drying cost per unit product dried. Currently used collector area and the air recycle factor were considered as the parameters for basic mode of operation of the dryer and adaptive pattern search technique was adopted to find the optimum values of the solar collector area and the recycle factor. The optimum value of the collector area is 26 m^2 and the recycle factor is 90%. The computer program developed in this study can be used to optimize similar drying systems.

Student's signature

Thesis Advisor's signature

5 / 06 / 2008

ACKNOWLEDGEMENTS

First of all, this thesis has been successfully achieved by my scholarship sponsor, H.R.H. Princess Maha Chakri Sirindhorn and supporting for all experiments from my boss, the Lord Camberlain, Keokhwan Vajarodhaya, Director of Royal Chitralada Projects.

With the kind help and suggestion of many people who have provided support and guidance for this work. I would like to thank my thesis advisor Assoc. Professor Peerayuth Chansethikul, thesis advisor, Assoc. Professor Serm Janjai, Assoc. Professor Vira Chankong, Assistant Professor Saeree Svetasreni and Associate Professor Prapaisri Sudasna-Na-Ayudthya, committee member.

I also would like to extent special thanks to Prof. Muhlbauer, Institute for Agriculture Engineering in Tropics and Subtropics, Hohenheim University, Stuttgart, Germany and Prof. B. K. Bala, Bangladesh Agricultural University, Mymensingh, Bangladesh , both visiting professors for Physics Department , Silpakorn University, in supporting and guidance during experiments in Hohenheim University, in Silpakorn University and in Royal Chitralada Projects.

Finally, my appreciation devotes to my dad, mom and my sister who always give me heartful support.

Rosarin Smitabhindu

April 2008

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF ABBREVIATIONS

a,b	=	empirical constant
a_w	=	water activity [-]
A_c	=	Collector area of solar-assisted batch dryer [m^2]
A_{max}	=	the maximum limit of collector area
$C_{airduct}$	=	the cost of air duct
C_{annual}	=	the annual cost of system [US$ /year]
C_b	=	specific heat of the absorber [J/kg-K]
C_{blower}	=	the cost of blower
C_{burner}	=	the cost of gas burner
C_c	=	specific heat of the cover [J/kg-K]
$C_{cabinet}$	=	the cost of the dryer cabinet
$C_{capital}$	=	the total capital cost
$C_{collector}$	=	the cost of collector
$C_{electric}$	=	the cost of the electricity consumption [kWh / year]
C_f	=	specific heat of the air [J/kg-K]
C_f	=	cost of the drying cabinet construction and installation
C_{gas}	=	the cost for the consumption of the gas
C_{labor}	=	the costs of construction and installation
$C_{labor,op}$	=	labor cost for operating the system per year
C_{maint}	=	the maintenance cost
$C_{maint,i}$	=	the maintenance cost in the i^{th} year
C_{op}	=	the operating cost of the dryer [USD]
$C_{op,i}$	=	the operating cost in the i^{th} year [USD]
C_p	=	specific heat of the product [J/kg-K]
C_{pa}	=	specific heat of the drying air [J/kg-K]
C_T	=	the total capital cost for solar drying-assisted system [USD]
$C_{unit.col}$	=	the unit cost of collector [47 USD/m^2]
$C_{unit,elect}$	=	the unit price of the electricity [USD/kWh]
$C_{unit,gas}$	=	cost of gas per unit [USD/kg]
C_v	=	specific heat of the vapor [J/kg-K]

LIST OF ABBREVIATIONS (Continued)

C_w	=	specific heat of water [J/kg-K]
D_d	=	width of the gap between the trays [m]
D_c	=	width of the air channel of the collector [m]
D_h	=	hydraulic diameter [m]
D_t	=	width of gap between trays [m]
GC	=	gas consumption
G_c	=	specific mass flow rate of the air in the collector [kg/s-m^2]
G_d	=	specific mass flow rate of the air in the dryer [kg/s-m^2]
$h_{c,\ b\text{-}f}$	=	convective heat transfer coefficient between the absorber and the air in the collector [W/m^2-K]
$h_{c,\ c\text{-}f}$	=	convective heat transfer coefficient between the cover and the air in the collector [W/m^2-K]
$h_{c,\ f\text{-}c}$	=	convective heat transfer coefficient between the air in the collector and the cover [W/m^2-K]
$h_{c,\ f\text{-}c}$	=	convective heat transfer coefficient between the air in the collector and the cover [W/m^2-K]
$h_{c,\ p\text{-}f}$	=	convective heat transfer coefficient between the product and the drying air [W/m^2-K]
h_{fg}	=	latent heat of vaporization [J/kg]
$h_{r,\ c\text{-}s}$	=	radiative heat transfer coefficient between the cover and the sky [W/m^2-K]
$h_{r,\ b\text{-}c}$	=	radiative heat transfer coefficient between the cover and the absorber [W/m^2-K]
$h_{r,\ c\text{-}s}$	=	radiative heat transfer coefficient between the cover and the sky
$h_{w,\ c\text{-}a}$	=	convective heat loss coefficient of the cover caused by wind [W/m^2-K]
H	=	humidity ratio of the drying air [-]
H_{gas}	=	heating value of gas [J/kg]
$H_{i,inlet}$	=	humidity ratio of the air at the inlet of the drying chamber [-]
H_{in}	=	humidity ratio of the air from the collector [-]
$H_{i,outlet}$	=	humidity ratio of the air at the outlet of the drying chamber [-]

LIST OF ABBREVIATIONS (Continued)

H_{mix} = humidity ratio of the mixed air obtained from the recycled air and air from collector [-]
i_f = inflation rate (%)
i_{in} = the interest rate (%)
I_t = solar radiation incident on the collector [W/m^2]
k = heat conductivity [W/m-k]
k_b = heat conductivity of the black insulator [W/m-k]
L = Collector length [m.]
L_b = thickness of the insulator [m.]
m_1 = mass flow rate from the collector
m_2 = mass flow rate of the recycle air
m_3 = mass flow rate of the air to the drying chamber
M = moisture of the product at time t [kg_{water}/kg_{solid}]
M_{batch} = the weight of dried product per batch [kg]
MC = Means moisture contents of banana in the dryer [db]
$M_{dryproduct}$ = dried product form system per year [kg]
M_e = equilibrium moisture content [kg_{water}/kg_{solid}]
M_{elect} = the amount of the electricity per year [kWh / year.]
M_{gas} = amount of gas consumption per year [kg]
M_{final} = final moisture content of product [kg_{water}/kg_{solid}]
M_0 = initial moisture content of product [kg_{water}/kg_{solid}]
$M_{unit.elect}$= unit cost of electricity [USD/kWh]
N = life span of the dryer [years]
N_{sect} = number of collector section for the finite difference calculation
N_{batch} = the number of batch to be dried per year
Nu = Nusselt number (dimensionless) [-]
Q_{gas} = heat energy required by the gas burner [J]
rh = relative humidity of the air [%]
R_c = recycle factor [-]
Re = Reynolds number (dimensionless) [-]

LIST OF ABBREVIATIONS (Continued)

Rh	=	relative humidity of the air [%]
R_{max}	=	maximum limit of the recycle factor [-]
t	=	drying time [s]
T	=	drying air temperature in the mass balance equation of the product [K]
T_a	=	temperature of ambient air [K]
T_b	=	temperature of the absorber [K]
T_c	=	temperature of the cover [K]
T_{fd}	=	temperature of the drying air in the dryer [K]
T_{f0}	=	outlet air temperature from the collector [K]
T_p	=	temperature of the product [K]
T_s	=	sky temperature [K]
T_{set}	=	set point temperature of the dryer [60 °C]
U_b	=	heat loss coefficient [w/m^2-k]
V	=	speed of the air in the collector [m/s]
V_a	=	wind speed [m/s]
W_c	=	width of the collector [m]
W_T	=	Total weight of banana [kg]
x	=	distance [m]
Z	=	objective function or drying cost [USD/kg]
ε_c	=	emittance of the cover [-]
ε_b	=	emittance of the absorber [-]
ρ	=	density of the air [kg/m^3]
ρ_b	=	density of the absorber [kg/m^3]
ρ_c	=	density of the cover [kg/m^3]
$\rho_{s,p}$	=	density of dried product [kg/m^3]
δ_b	=	thickness of the absorber [m]
δ_c	=	thickness of the cover [m]
$\tau\alpha$	=	transmittance – absorptance product of the system consisting of the cover and the absorber [-]

LIST OF ABBREVIATIONS (Continued)

ν	=	kinetic viscosity of air [m^2/s]
ω	=	parameter for the annualization of the cost calculation [-]
α_c	=	absorbance of the cover [-]
σ	=	Stefan-Boltzmann constant [5.6697×10^{-8} [W/m^2K^4]

OPTIMIZATION OF A SOLAR-ASSISTED DRYING SYSTEM FOR DRYING BANANAS

INTRODUCTION

1. Background and statement of the problem

Thailand is one of the countries that produce a lot of fruits and vegetables available for internal consumption and export. More than 30 millions of populations are still in the agriculture sector. Some agricultural products such as fruits and vegetables are consumed not only as fresh products but also as processed agro-industrial products. One of the common processes for prolonging shelf life and increasing value addition is drying. In general, natural sun drying is still used by Thai farmers because it is very cheap method. However, in this method dried products are usually destroyed by rodents, birds, and insects. To overcome this problem, Royal Chitralada Projects in collaboration with the University of Hohenheim and Silpakorn University has developed a solar-assisted fruit dryer. This dryer uses solar energy and heat energy from LPG gas burner as energy sources. It has a drying area of 8 m^2. The dryer has been used for producing dried banana since 1997. These dried products are sold in supermarkets and cooperatives stores in Bangkok.

Originally, the dryer was designed for drying general fruits. As drying characteristics and sale price of fruits depend on their varieties. Therefore, to calculate estimation of optimum drying parameters was not possible without specifying the varieties. (Bala, 1998; Ratti and Mujumdar, 1997; Sodha et al., 1985) Since this dryer is now used only for banana drying, it is necessary to investigate these optimum parameters in order to improve its technical-economic performance. Moreover, this dryer is also used for demonstration purposes of the Royal Chitralada Projects, it should have an optimum design. Therefore, it is a great necessity to optimize the design of this dryer.

Banana is a potential cash crop in Thailand and dried banana is a popular snack food. Huge amount of bananas is consumed in every part of Thailand. The international market can also be promoted with dried banana if the quality of the dried banana can be maintained at international standard. There is a bright future for dried banana for both internal consumption and export. Since the problem of dried banana is aggravated by the lack of viable drying technology, a systematic research on solar drying of banana will contribute to the knowledge on the production of quality dried banana for local consumption and export. This research will also generate much scientific information for researchers, processors and users.

This work will cover all aspects is solar drying of banana. Hence such a piece of research has a great economic potential for Thailand as well as for the other developing countries in line tropics and subtropics.

OBJECTIVES

The major objectives of this work are to optimize design parameters of the solar-assisted fruit drying system at the Royal Chitralada Projects based on minimizing the drying cost with the following minor objectives;

- To formulate a model of solar–assisted fruit dryer for industrial operations.
- To simulate the performance of the dryer for industrial operations.
- To determine the optimal operation parameters of the dryer.

LITERATURE REVIEW

1. Solar drying of bananas

Bananas (*Musa* x *paradisiace* L.) are one of the major tropical fruits in Thailand. In general three main varieties of bananas, namely Kai (AA group), Hom (AAA group) and Namwa (ABB group) are grown in this country. The Namwa variety grown mainly in Thailand is generally consumed as both fresh and dried fruit. This variety is dried not only for preservation purpose, but also for modification of the taste, flavor and texture of the product to meet consumers' requirements and consequently increase market value of the product. The annual production of fresh bananas in Thailand is 1.8 million tons and the area covered by banana cultivation is 139,000 ha (FAOSTAT, 2004). In Thailand bananas are traditionally dried. In traditional sun dried method, bananas are spread on mats and directly exposed to solar radiation. Drying by this method takes 5 - 7 days, depending on the weather conditions. In the sun drying method the bananas become contaminated with dust, dirt, animals, insect infection, and microorganisms. More over drying processes can not be controlled, and a relatively low quality of dried product is obtained. The product quality is far below the international market. Under these conditions losses can be as high as 10 to 15%.

Banana drying is an important post harvest operation and it is dried from 65% to 30% (w.b.). Banana should be properly dried and the quality is affected by the method of drying. Banana drying is profitable since it is mainly consumed as a dried product.

Sun drying offers a cheap method of drying but often results in a inferior quality of dried banana due to its dependence on weather conditions and vulnerability to the attack of rains, insects, pests and microorganism and dust and dirt. These drying problems are more serious particularly during the rainy reason, when bananas are rewetted by rain, which induces tine growth of moulds. To overcome these problems, more efficient drying equipment is needed to replace the traditional sun drying method.

As an alternative to sun drying, mechanical and solar dryers are available. The mechanical dryer used in industrialized countries are not applicable to small farms in developing countries due to high investment and operating cost (Bala, 1998). Solar drying is environmentally friendly and economically viable in developing countries (Esper, 1998). Solar drying has been the oldest and most widely used method of drying crops in the world especially in the developing countries. For better utilization of abundant of solar energy dryers have been designed based on the specific needs.

Thailand, located in the tropical region of South East Asia, receives annual daily solar radiation of 18.2 $MJ/m^2/day$, which is relatively high for solar energy applications. Therefore, the utilization of solar dryers has been considered to be one of the most promising options for solar drying of banana in Thailand.

Natural convection solar dryers appear to have potential of adoption and application in the tropics and subtropics. This type of dryer is low cost, can be locally constructed and does not require any power from electrical grid or fossil fuels. It is suitable for household level for drying of 10 to 15 kg of bananas. But the natural convection solar dryer suffers from the limitations due to extremely low buoyancy induced airflow inside the dryers (Bala and Woods, 1994a; Bala and Woods, 1995). Comparatively high investment, limited capacity and the risk of crop spoilage during adverse conditions have up to now prevented the wide acceptance of natural convection solar dryers (Esper, 1998; Schirmer et al., 1996). Recently researchers and users are opting for forced convection solar assisted dryers for drying perishable agricultural products. The drying rate and drying capacity of a forced convection/solar assisted forced convection drier is much higher in comparison to natural convection solar dryer. It is used for commercial drying of fruits, vegetables, cereals, grains, legumes, oil seeds and spices. Even fish and meat can be dried in the forced convection solar dryer. The solar assisted forced convection dryer, therefore, may be considered for adoption and application for industrial production of dried banana.

To maintain the quality of dried banana to the international standard, a systematic research on drying of banana is essential. A lot of works have been conducted on physical properties and thermal properties, thin layer drying and drying characteristic of grains but limited work has been reported on solar drying of banana.

Recently, researchers and users are showing an upsurge of interest in the forced convection solar drying and solar assisted forced convection drying of agricultural products. But systematic researches on the solar drying of banana have not been carried out. No study on computer modeling of solar assisted dryer has been reported. Although, the solar assisted dryer is promising for drying of banana, its component performance and the overall process have not been optimized mathematically.

2. Optimization of solar dryer

In the past twenty years, many types of solar dryers have been developed, both in developing countries and industrialized countries. As solar dryers use solar energy that has diurnal and seasonal variations, the design of the dryers is more complicated than the mechanical dryers, those use gas and electricity with constant energy supply. In addition, the design of dryers depends on varieties of products to be dried, which complicates optimum design of the dryers. Many attempts have been made to optimize solar dryers of various products. Radajewski et al. (1987) carried out a study on optimization of solar grain drying in a continuous flow dryer. The schematic diagram of their dryer is depicted in Figure 1

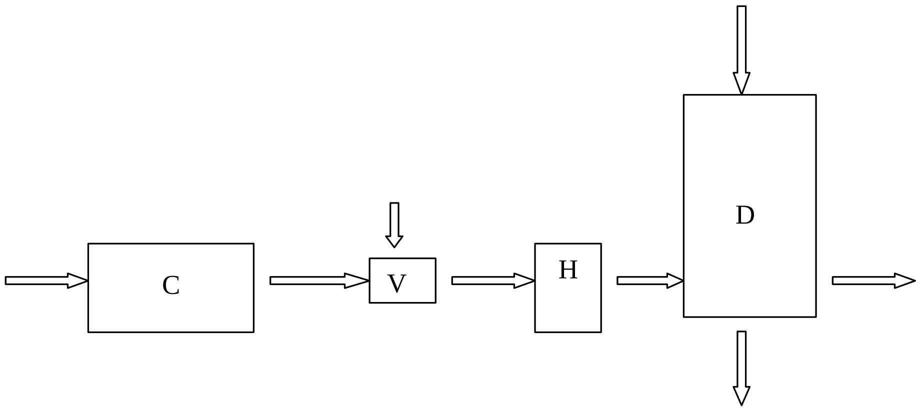


Figure 1 Schematic diagram of continuous flow grain dryer with solar supplemented air heating system (Radajewski, et. al, 1987). C, solar collector; V, mixing value; H, auxiliary heater; D, continuous flow dryer; ⟹ , air flow.

In their study, a simulation model was developed to minimize the cost of drying to obtain the optimum values of the geometry of the collector and the specific rate of airflow through the collector. In addition, the ratio for the cost of drying in a solar-supplemented system to the cost of drying in a conventional system was minimized. This ratio represents the maximum possible saving for the given drying conditions.

Janjai et al. (1994) developed a procedure for determining the optimum collector area for a solar paddy drying system. The system subjected to their study is schematically shown in Figure 2.

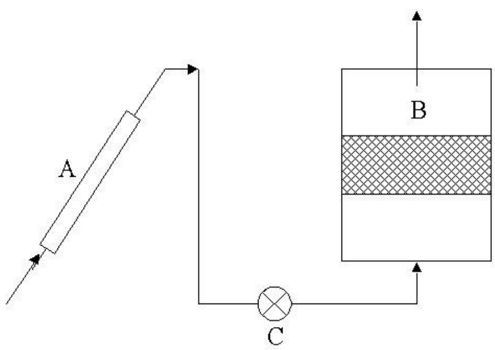


Figure 2 Schematic diagram of the solar paddy drying system of Janjai et al. (1994). A, Solar collector; B, Drying bin; C, Blower.

They formulated a simulation model of this drying system and used this model for analyzing the system performance. Two correlation parameters P and Q, which are function of system parameter and weather data, were formulated. They simulated the system performance to obtain a correlation relating drying time to P and Q. Finally, the solar collector area obtained from this procedure was compared to that calculated from detail simulations and good agreement was found.

Bala and Wood (1995) undertook a study on the optimization of natural convection solar drying systems. Their system configuration is show in Figure 3. The air flow is driven by buoyancy effect alone. The air passes through the single cover collector between the cover and absorber.

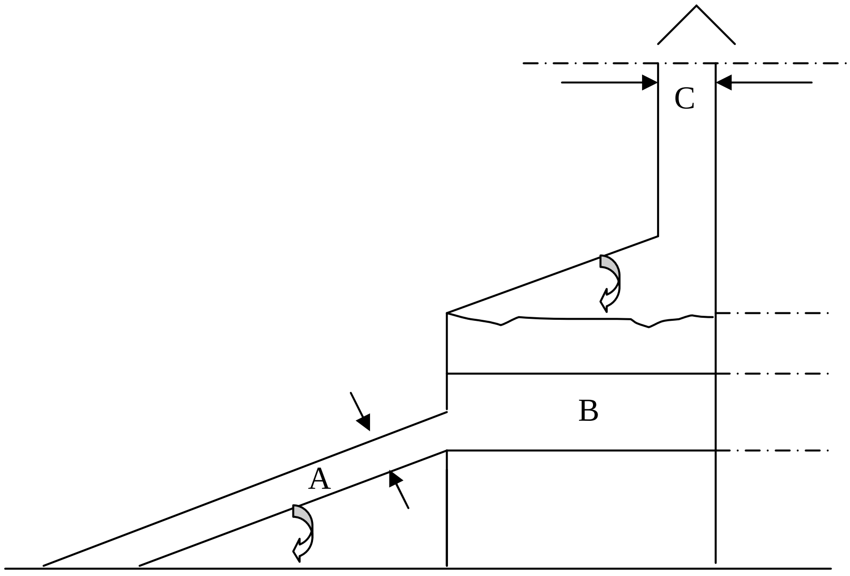


Figure 3 Solar dryer under the investigation of Bala and Wood (1995)
A, Solar Collector; B, Drying Chamber; C, Chimney

In their study, the physical simulation is combined with a cost prediction and a search technique, which finds the constrained minimum of total cost per unit moisture removal. From sensitivity analyses, they found that design geometry is not very sensitive to material or fixed cost but grain capacity has some effect.

MATERIALS AND METHOD

In this section, the description of the solar-assisted fruit drying system subjected to this study will be given. Then, the description the method used for the optimization of the system including intermediate results will be presented.

1. Description of the solar - assisted fruit drying system

The solar assisted forced convection drying system has been installed at the Royal Chitralada Projects in Chitralada Villa, Dusit Palace in Bangkok to demonstrate the potentiality of solar assisted convection dryer for production of quality dried fruits. This dryer has a capacity of drying 250 kg of bananas and it is being used for drying other fruits since the middle of 2000.

This drying system consists of 2 main parts namely (1) the solar collector and (2) the drying cabinet. For the solar collector, it is placed on the rooftop of the drying processing center where the drying cabinet is installed. It composes of polyurethane black insulator and cover glass. There is an air gap between the cover glass and the black insulator in which ambient air is sucked from both ends of the collector. This air is sucked at the midpoint of the collector and supplied into the cabinet with additional heat using LPG if needed. Each part of the collector was designed with the modular concept. The parts of the collector such as insulator and cover glass are in module form so that these can be easily transported and connected to each other. The schematic diagram of this collector is shown in Figure 4.

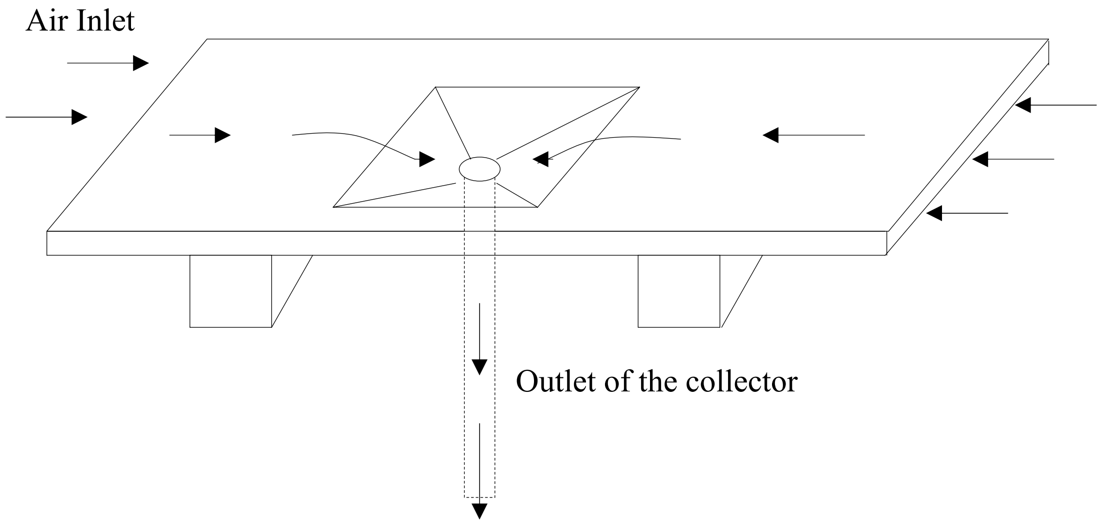


Figure 4 Schematic diagram of the solar collector

The drying cabinet is tray type with a dimension of 1 × 2 × 1.5 m^3 for the drying area. There are 15 trays with a total drying area of 8 m^2. This drying cabinet was specially designed in such a way that hot air is guided to flow parallel through the product placed in the trays. This design has the advantage of allowing uniform distribution of air temperature in the cabinet from hot air for drying the products. Ambient air preheated by solar collector is sucked by an electrical fan and additional heat if needed is supplied by LPG gas burner. Then heated air is supplied to the cabinet. The schematic views of the drying cabinet are shown in Figure 5.

The ambient air from both ends of the collector is sucked by the fan of the LPG burner to move through the air gap of the collector. The collector receives the energy from solar radiation and heats air inside the collector. The polyurethane black insulator is used to reduce heat loss. The hot air from the solar collector is moved through the air duct to the LPG gas burner placed on the top of the cabinet. This air is heated up again by the gas burner if needed and forced to the drying cabinet for drying the product. Part of this drying air is recycled and circulated in the cabinet to save energy and the other part of this air is released to ambient surrounding. The pictorial views of the solar collector and drying cabinet are shown in Figure 6 and Figure 7, respectively.

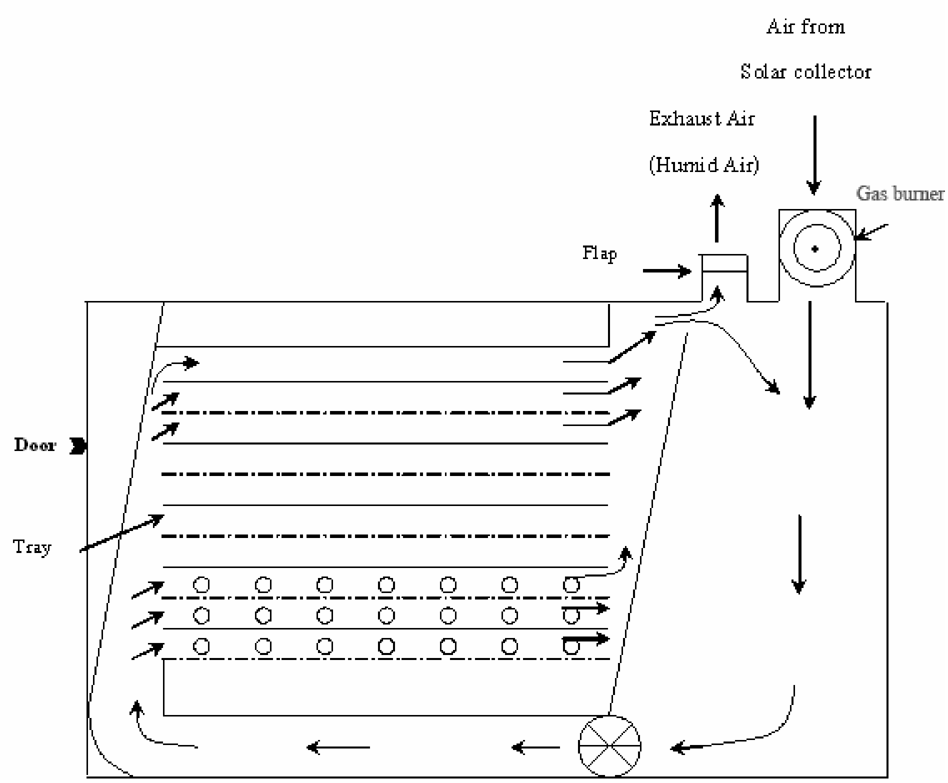


Figure 5 Schematic diagram of the drying cabinet.

Figure 6 The pictorial views of the solar collector.

Figure 7 The pictorial views of the drying cabinet.

2. Operation of solar-assisted fruit drying system for drying banana

Bananas of Namwa variety, grown in the central region of Thailand, were used in these experiments. The size of each banana fruit was approximately 4 cm diameter and 10 cm length. As the banana obtained for the experiments were not ripe enough, they were first ripened in a chamber. In general, the ripe bananas had an initial moisture content of 69% w.b. and sugar content of 27 ^{o}Brix, depending on the state of ripeness.

The solar assisted dryer under investigation was installed at the Royal Chitralada Projects in Bangkok, Thailand, where other pilot-scale agro-industrial research, development and dissemination of new technologies are conducted. The dryer was installed in the ground floor in the drying processing center and its collector was placed on the rooftop of that building. Ninety six drying tests were carried out during March 2002 to December 2004.

2.1 Instrumentation

Equipment for measuring parameters affecting the drying processes were installed for investigating the performance of the dryer. K-type thermocouples were used to measure the air temperature inside the collector along the flow direction. A pyranometer (Kipp and Zonen CM11) was employed to measure the global solar radiation at the middle of the length of the collector. The electrolytic sensor of hygrometer (Novasina TR24) was installed at the air inlet of the collector in order to monitor the relative humidity of the ambient air. All the signals from these sensors were connected to a data acquisition unit (Cambell, XL) controlled by a microcomputer to collect the data every 10 minutes. To monitor the weight loss of the product during drying, samples of the bananas were taken from 6 positions in the trays of the dryer and weighed every 2 – 3 hours. The sun dried control samples were weighed as well. These samples were used to determine the moisture content by the oven method.

2.2 Procedure

For all drying tests, ripe bananas were first peeled carefully and then spread in a single layer on the tray inside the dryer. The bananas were dried as whole fruits without any chemical pretreatment. Drying was started when loading was completed, normally at 8:00 a.m. and stopped at 5:00 p.m.. Afterwards, the bananas in the dryer were collected and placed in plastic boxes in order to induce fermentation and for diffusion of moisture within the banana fruits. In case of wet samples the product was dried to some extent using LPG gas and some extent to save the product from spoilage during rainy day. These were again spread in the dryer the next morning and the process was repeated until final moisture content of about 30% w.b. was reached. The average sugar content of the dried banana was increased to 55 ^{o}Brix. Finally, dried bananas were collected and flattened. The thickness of the flattened bananas was approximately 1 cm. In general, up to 250 kg of ripe bananas can be dried in the dryer, but for the first three tests, only 50 – 100 kg of bananas were dried in order to investigate the influence of the stage of ripeness on the quality of the final product. The moisture content of the banana sample was measured at the starting and end of each experiment by standard air oven method.

3. Optimization of solar assisted fruit drying system

In general the optimization of a product process can be done by collecting data of process parameters from different cases. Then analyze these data in order to find the optimum value of the parameters which minimize or maximize the objective function. But for our case, the solar drying system is only the prototype and there no other similar systems. In addition, we cannot vary the parameters of the system because the variation will disturb the production process of the Royal Chitralada Projects, the owner of the system. Therefore, we do not have the performance data for various values of the system parameters for the optimization procedure.

To overcome this problem, we propose to use the simulation method to obtain the data on the system performance with various values of system parameter. To accomplish this, we have to start with the formulation mathematical model of the system. As there are some parameters in the model are not available in literature such as equilibrium moisture content and thing layer drying characteristics, we to determine these parameters by mean of experiment. To ensure the accuracy of the model, we will compare the performance of the system obtained from the simulation to that obtained from the measurement under the real operation. Once the model has been validated, we will use the model to generate the data on the performance of the system with different values of system parameters. Finally, we will use these generated data for our optimization process. All the above-mentioned steps can be summarized as the following. The schematic diagram of these steps is shown in Figure 8.

1) Formulation of mathematical model of the system
2) Determination of model parameters

3) Experimental validation of the models
4) Formulation of the cost model of the system
5) Identification of system parameters to be optimized by using simulation model
6) Formulation of the objective function for the optimization process and define constraints.
7) Selection of the method for the optimization process.

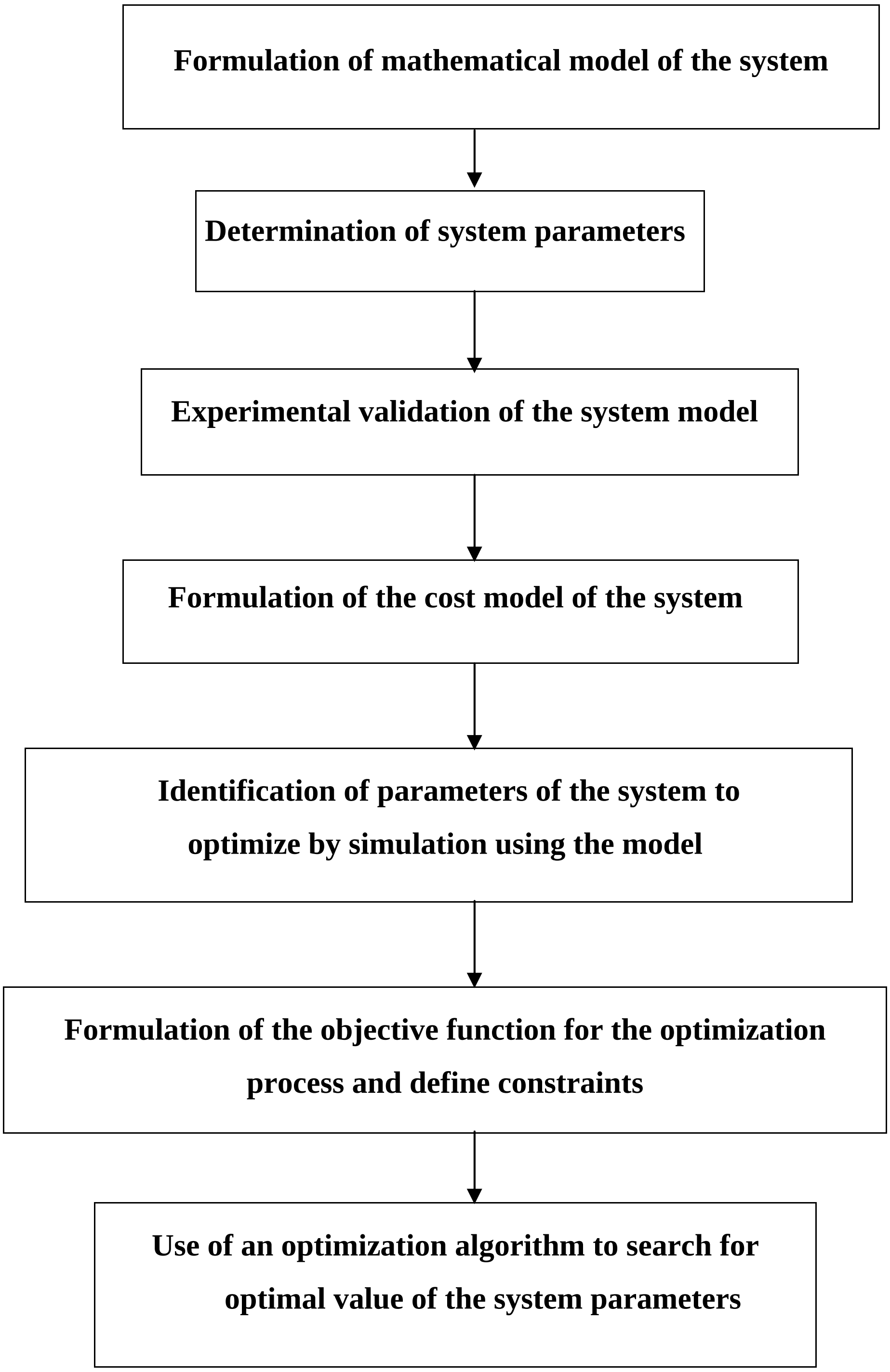


Figure 8 Schematic diagram showing the method used in this work.

3.1 Formulation of the model for the thermal performance of the solar-assisted fruit drying system

The computer simulation model of the solar assisted fruit dryer consists of two components. These are:

(a) The collector model, which predicts the temperatures at different positions of the collector at different drying time.

(b) The model of the drying unit that predicts the drying behaviour of banana at different time in the dryer unit.

3.1.1 Formulation of the model for collector

Flat plate collector is designed for applications requiring moderate temperature not exceeding 100°C. It is relatively cheap and can be easily constructed. It uses both beam and diffuses solar radiation and is well suited for the drying of agricultural crops. In the solar assisted dryer, the airflow is driven by a fan operated by electricity and air passes through the single cover collector between the cover and the absorber. The heated air is sucked at the mid - point of the collector and heated if necessary using LPG gas and is then passed over the products spread in a thin layer in trays inside the drying unit. The moisture evaporated from the products is carried away and a fraction of the exhaust air from the dryer is recirculated.

Considering an element, Δx of collector at a distance x from the inlet (Figure 9), the energy balances on the collector components give the following equations.

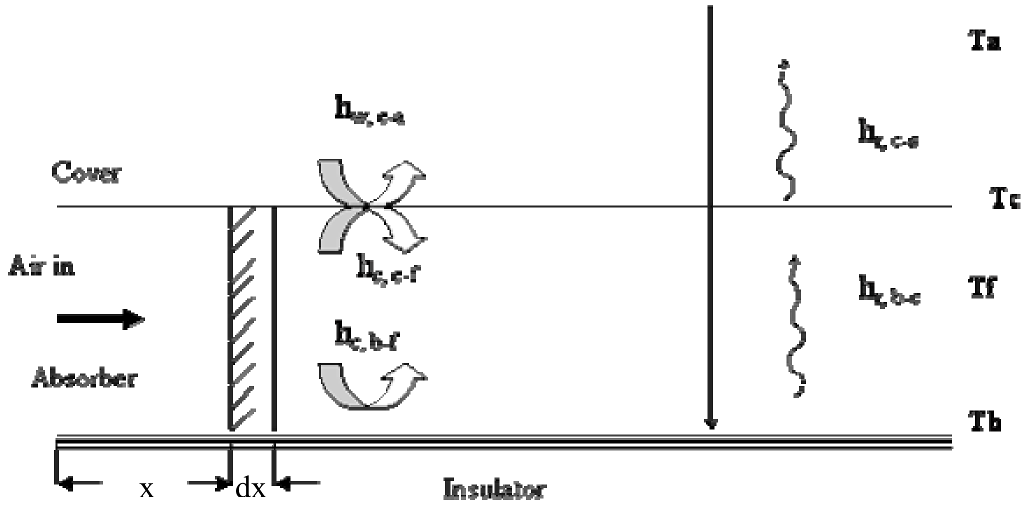


Figure 9 Schematic diagram showing heat transfers in the flat plate solar collector.

- The cover of the collector

The heat balance in the element (x, x+Δx) of the cover and of width W in time Δt yields the following equation;

$$\rho_c\delta_c\Delta xWC_c(T_c+\frac{\partial T_c}{\partial t}\Delta t)-\rho_c\delta_c\Delta xWC_cT_c = h_{c,f-c}\Delta xW(T_f-T_c)\Delta t$$
$$+h_{r,b-c}\Delta xW(T_b-T_c)\Delta t+h_{w,c-a}\Delta xW(T_a-T_c)\Delta t+h_{r,c-s}\Delta xW(T_s-T_c)\Delta t+\Delta xW\alpha_cI_t\Delta t \quad (3.1)$$

This equation can be expressed as

$$\rho_c\delta_cC_c\frac{\partial T_c}{\partial t} = h_{r,b-c}(T_b-T_c)+h_{c,f-c}(T_f-T_c)+h_{w,c-a}(T_a-T_c)+h_{r,c-s}(T_s-T_c)+\alpha_cI_t \quad (3.2)$$

Where

T_c = temperature of the cover [K]

T_f = temperature of the air in the collector [K]

T_b = temperature of the absorber [K]

T_a = temperature of ambient air [K]

$h_{r,\,b\text{-}c}$ = radiative heat transfer coefficient between the cover and the absorber [W/m^2-K]

$h_{c,\,f\text{-}c}$ = convective heat transfer coefficient between the air in the collector and the cover [W/m^2-K]

$h_{w,\,c\text{-}a}$ = convective heat loss coefficient of the cover caused by wind [W/m^2-K]

$h_{r,\,c\text{-}s}$ = radiative heat transfer coefficient between the cover and the sky [W/m^2-K]

ρ_c = density of the cover [kg/m^3]

δ_c = thickness of the cover [m]

C_c = specific heat of the cover [J/kg-K]

α_c = absorbance of the cover [-]

I_t = solar radiation incident on the collector [W/m^2]

- Air in the collector

The heat balance in the air inside collector in the element (x, x+ Δx) yields

$$D_c G_c W C_f (T_f + \frac{\partial T_f}{\partial x}\Delta x)\Delta t - DGW(C_f + C_v H)T_f \Delta t = \Delta x W h_{c,f-c}(T_c - T_f)\Delta t + \Delta x W h_{c,b-f}(T_b - T_f)\Delta t \tag{3.3}$$

This equation can be rewritten as

$$D_c G_c C_f \frac{\partial T_f}{\partial x} = h_{c,c-f}(T_c - T_f) + h_{c,b-f}(T_b - T_f) \tag{3.4}$$

where

T_f = temperature of the air in the collector [K]

T_c = temperature of the cover [K]

T_b = temperature of the absorber [K]

$h_{c,\,c\text{-}f}$ = convective heat transfer coefficient between the cover and the air in the collector [W/m^2-K]

$h_{c,\,b\text{-}f}$ = convective heat transfer coefficient between the absorber and the air in the collector [W/m^2-K]

D_c = width of the air channel of the collector [m]

G_c = specific mass flow rate of the air in the collector [kg/s-m^2]

C_f = specific heat of the air [J/kg-K]

- Absorber of the Collector

The heat balance in the element (x, x+ Δx) of the absorber and of width W in time Δt also yields the equation;

$$\rho_b \Delta x W \delta_b C_b (T_b + \frac{\partial T_b}{\partial t}\Delta t) - \rho_b \Delta x W \delta_b C_b T_b \tag{3.5}$$
$$= h_{c,f-b} \Delta x W (T_f - T_b)\Delta t + h_{r,b-c} \Delta x W (T_c - T_b)\Delta t + u_b \Delta x W (T_a - T_b)\Delta t + \Delta x W (\tau\alpha) I_t \Delta t$$

This equation can be rearranged as

$$\rho_b \delta_b c_b \frac{\partial T_b}{\partial t} = h_{c,b-f}(T_f - T_b) + h_{r,b-c}(T_c - T_b) + U_b (T_a - T_b) + (\tau\alpha) I_t \tag{3.6}$$

where

T_b = temperature of the absorber

T_f = temperature of the air in the collector

T_a = temperature of ambient air

U_b = heat loss coefficient through the back side of the collector to ambient air

$h_{c, b-f}$ = convective heat transfer coefficient between the absorber and the air in the collector

$h_{r, b-c}$ = radiative heat transfer coefficient between the absorber and the cover

$(\tau\alpha)$ = transmittance – absorptance product of the system consisting of the cover and the absorber

I_t = solar radiation incident on the collector

- Model parameters

Radiation heat transfer coefficient, $h_{r,c-s}$ between the cover and the sky is given as (Duffie and Beckman, 1991):

$$h_{r,c-s} = \varepsilon_c \sigma (T_c^2 + T_s^2)(T_c + T_s) \quad (3.7)$$

and $T_s = 0.552 T_a^{1.5}$

where

T_c = temperature of the cover [K]

T_s = sky temperature [K]

ε_c = emittance of the cover [-]

σ = Stefan-Boltzmann constant [5.6697×10^{-8} W/m^{2}K^4]

The radiative heat transfer coefficient, $h_{r,b-c}$ between the cover and the absorber is (Duffie and Beckman, 1991)

$$h_{r,b-c} = \frac{\sigma (T_b^2 + T_c^2)(T_b + T_c)}{\frac{1}{\varepsilon_b} + \frac{1}{\varepsilon_c} - 1} \quad (3.8)$$

where

T_b = temperature of the absorber [K]

T_c = temperature of the cover [K]

ε_c = emittance of the cover [-]

ε_b = emittance of the absorber [-]

Convective heat transfer coefficient, $h_{w,\ c-a}$ at the upper surface of the cover from ambient air is computed from the following equation (McAdams, 1954).

$$h_{w,c-a} = 5.7 + 3.8V \quad (3.9)$$

where

V = wind speed [m/s]

The convective heat transfer in the inner surface of the cover from the air inside the collector is computed from the following correlation suggested by Kays and Crawford (1980).

$$Nu = 0.0158\, Re^{0.8} \quad (3.10)$$

where

Nu = Nusselt number [-]

Re = Raynolds number [-]

The Reynolds number is given by

$$Re = \frac{D_h V \rho}{\nu} \quad (3.11)$$

where

D_h = hydraulic diameter [m]

V = speed of the air in the collector [m/s]

ρ = density of the air [kg/m^3]

ν = kinetic viscosity [m^2/s]

The hydraulic diameter, D_h is computed from the relationship

$$D_h = \frac{4W_c D_c}{2(W_c + D_c)} \quad (3.12)$$

where

W_c = width of the collector [m]

D_c = width of the air channel of the collector [m]

The convective heat transfer coefficient, $h_{c,b-f}$ is computed from Nusselt number Nu computed from equation (3.10), hydraulic diameter from equation (3.12) and the thermal conductivity, k from the following equation;

$$h_{c,b-f} = \frac{Nu\, k}{D_h} \quad (3.13)$$

where k = heat conductivity [W/m-K]

The value of $h_{c,c\text{-}f}$ was assumed to be equal to $h_{c,b\text{-}f}$ because of the similarity of the materials and geometry of media: air-cover and absorber-air

Conduction heat loss coefficient through the absorber is given by

$$U_b = \frac{k_b}{L_b} \tag{3.14}$$

where

U_b = heat loss coefficient [W/m^2-k]

k_b = heat conductivity of the black insulator [W/m-k]

L_b = thickness of the insulator [m]

- Method of solution

The collector model expressed in equation (3.1)–(3.13) is a system of differential equation which we cannot solve using analytical method. To solve these equations, we proposed to use the implicit finite difference method. These systems of equation were written in the finite difference form as follows:

- Equation for the cover

$$\rho_c\delta_c C_c \frac{T_{c,t+\Delta t} - T_{c,t}}{\Delta t} = h_{r,b-c}(T_{b,t+\Delta t} - T_{c,t+\Delta t}) + h_{c,f-c}(T_{f,t+\Delta t} - T_{c,t+\Delta t}) + h_{w,c-a}(T_{a,t+\Delta t} - T_{c,t+\Delta t}) + h_{r,c-s}(T_{s,t+\Delta t} - T_{c,t+\Delta t}) + \alpha_c I_{t,t+\Delta t} \tag{3.15}$$

- Equation for the air

$$D_c G_c C_f\, 0.5[\frac{T_{f,x+\Delta x,t+\Delta t} - T_{f,x-\Delta x,t+\Delta t}}{2\Delta x} + \frac{T_{f,x+\Delta x,t} - T_{f,x-\Delta x,t}}{2\Delta x}] = h_{c,b-f}(T_{b,x,t+\Delta t} - T_{f,x,t+\Delta t}) + h_{c,c-f}(T_{c,x,t+\Delta t} - T_{f,x,t+\Delta t}) \tag{3.16}$$

- Equation for the absorber

$$\rho_b\delta_b C_b \frac{T_{b,t+\Delta t} - T_{b,t}}{\Delta t} = h_{c,b-f}(T_{f,t+\Delta t} - T_{b,t+\Delta t}) + h_{r,b-c}(T_{c,t+\Delta t} - T_{b,t+\Delta t}) + U_b(T_{a,t+\Delta t} - T_{b,t+\Delta t}) + (\tau\alpha)I_{t,t+\Delta t} \tag{3.17}$$

The length of the collector is divided into a number of layers so that the properties of the material are constant or nearly so within each layer. The time interval should be small enough for the air conditions to be constant at inlet to and exit from the layer. But for the economy of computing time, a compromise between the acceptability of the results and intervals must be used. Figure 10 shows that Schematic diagram of the collector for the numerical solution.

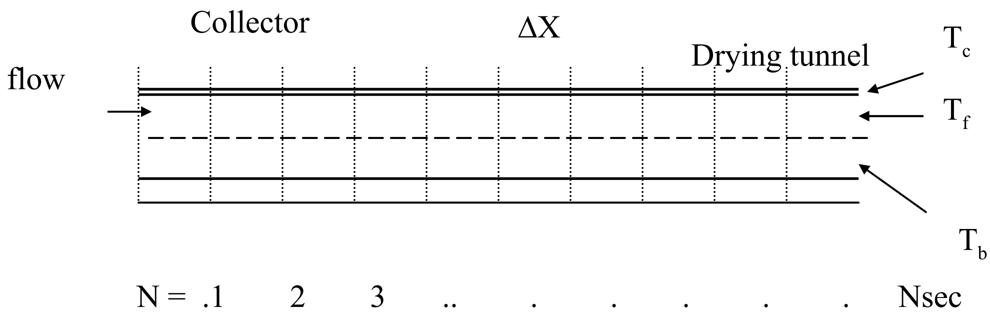


Figure 10 Schematic diagram of the collector for the numerical solution using finite difference method

The system of equations consisting of equation (3.15), equation (3.16) and equation (3.17) is expressed in the following form for the interval Δt for the entire length of collector unit.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{vmatrix} T_c \\ T_f \\ T_b \end{vmatrix} = \begin{vmatrix} b_1 \\ b_2 \\ b_3 \end{vmatrix} \qquad (3.18)$$

This system of equations is a set of implicit equations for the time interval Δt for the entire length of the drying unit and is solved using the Gauss – Jordan elimination method.

- Results

A program computer in FORTRAN was written to solve the equation. We obtained the values of temperature of the air in the collector for every time step Δt using the input data such as solar radiation and ambient air temperature from the measurement. Figure 11 shows the predicted variation of air temperature in the collector.

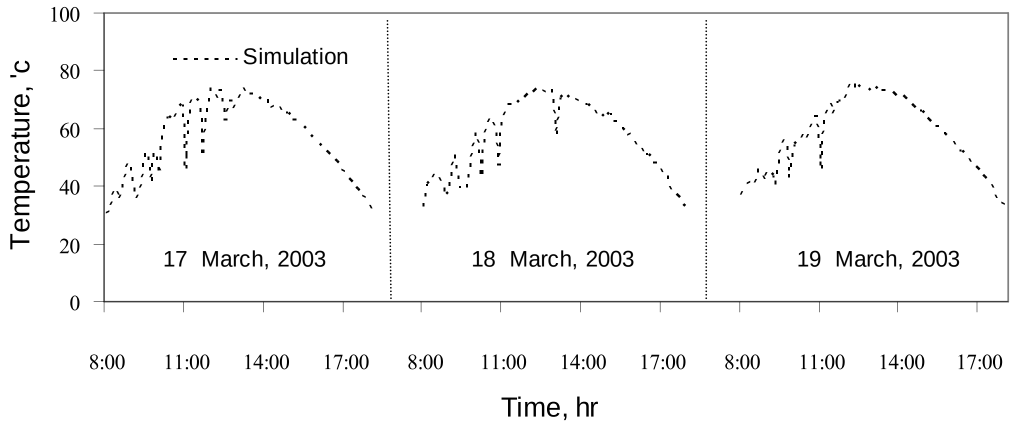


Figure 11 Predicted variation in the collector temperature.

3.1.2 Formulation of the model for drying unit

The bananas are placed in a single layer in a series of trays, which are parallel (Figure 12) so that the bananas in the trays almost equally at a uniform temperature and only thin layer equation is sufficient to describe the drying behavior inside the dryer.

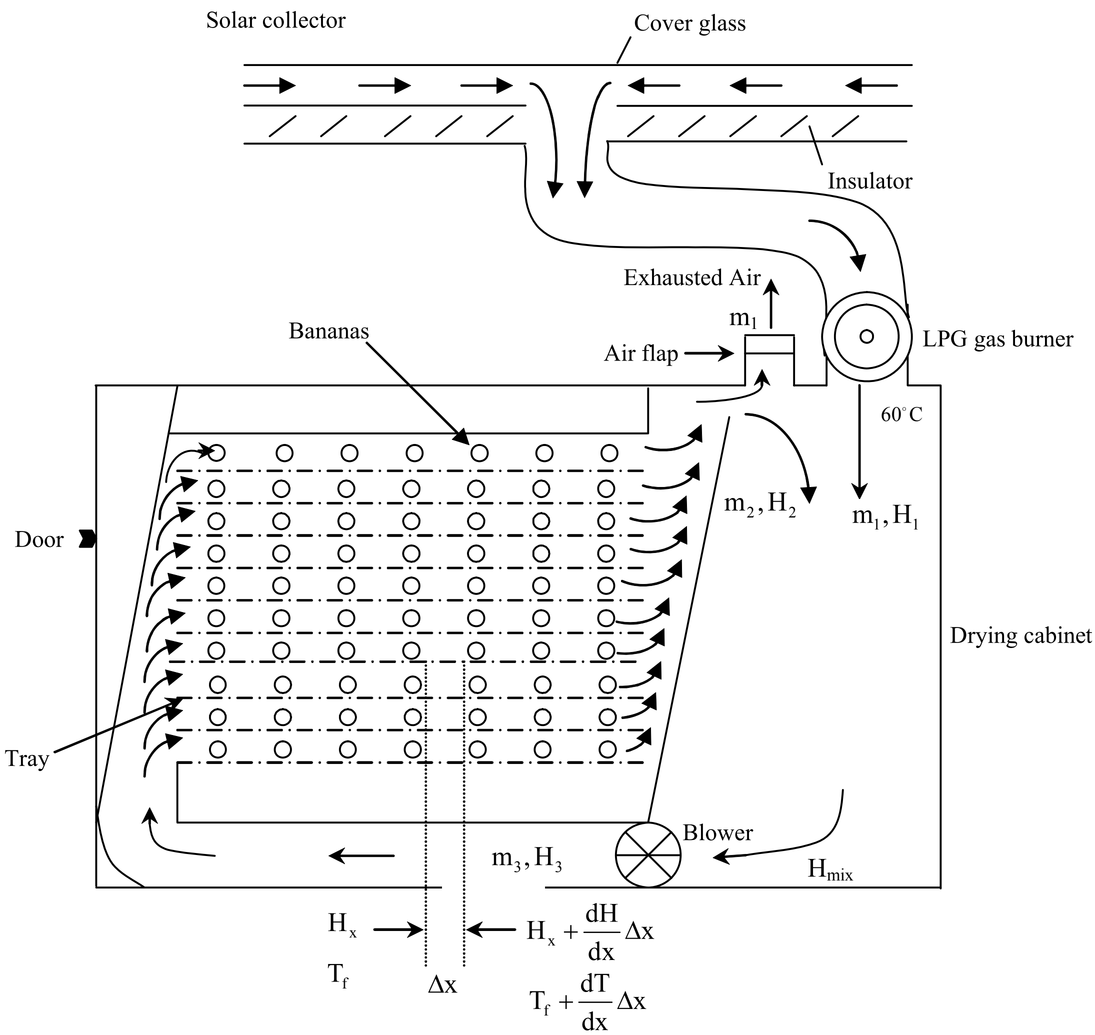


Figure 12 Schematic diagram of heat and moisture transfer in the drying unit (m = mass flow rate, H = humidity ratio, T_f = drying air temperature)

- Energy balance of the drying air

The change in enthalpy of the drying air is equal to the convective heat transfer to the product, the cover and the absorber.

$$D_d G_d W_d (C_f + C_v H)(T_{fd} + \frac{\partial T_{fd}}{\partial x}\Delta x)\Delta t - D_d G_d W_d (C_f + C_v H) T_{fd} \Delta t = \Delta x W_d h_{c,p-f}(T_p - T_{fd})\Delta t \qquad (3.19)$$

This equation can be rewritten as

$$D_d G_d (C_f + C_v H) \frac{\partial T_{fd}}{\partial x} = h_{c,p-f} (T_p - T_{fd}) \tag{3.20}$$

where

T_s = temperature of the drying air [K]

T_p = temperature of the product [K]

H = humidity ratio of the drying air [-]

$h_{c,\,p\text{-}f}$ = convective heat transfer coefficient between the product and the drying air [W/m^2-K]

D_d = width of the gap between the trays [m]

W_d = width of tray [m]

- Energy balance of the product inside the dryer

The change in enthalpy of the product is equal to the convective heat transfer to the product minus heat supplied to evaporate the moisture to the air. This energy balance yields;

$$\begin{aligned} &\Delta x W_d \rho_{s,p} (C_p + C_w M)(T_p + \frac{\partial T_p}{\partial t} \Delta t) + \Delta x W_d \rho_{s,p} (C_p + C_w M) T_p \\ &= h_{fg} D_d G_d \frac{\partial H}{\partial x} \Delta x \Delta t + C_v (T_{fd} - T_p) D_d G_d \frac{\partial H}{\partial x} \Delta x \Delta t \\ &\quad + \Delta x W_d h_{c,p-f} (T_{fd} - T_p) \Delta t \end{aligned} \tag{3.21}$$

This equation can be rearranged as

$$\begin{aligned} \rho_{s,p} (C_p + C_w M) \frac{\partial T_p}{\partial t} &= [h_{fg} + C_w (T_p - T_{fd})] D_d G_d \frac{\partial H}{\partial x} \\ &\quad + h_{c,p-f} (T_{fd} - T_p) \end{aligned} \tag{3.22}$$

where

T_p = temperature of the product [K]

T_{fd} = temperature of the drying air [K]

H = humidity ratio of the drying air [-]

m = moisture content of the product [kg_{water}/kg_{solid}]

$h_{c, p\text{-}f}$ = convective heat transfer coefficient between the product and the drying air [W/m^2-K]

C_p = specific heat of the product [J/kg-K]

C_w = specific heat of water [J/kg-K]

$\rho_{s,p}$ = density of product [kg/m^3]

W_d = width of tray [m]

D_d = width of the gap between the trays [$kg/s\text{-}m^2$]

G_d = specific mass flow rate of the drying air [$kg/s\text{-}m^2$]

- Moisture Balance of the air

Moisture balance states that the moisture gain by the air is equal to the moisture loss by the product.

$$D_d G_d W_d \Delta t(H + \frac{\partial H}{\partial x}\Delta x) - D_d G_d W_d \Delta t H \quad = \quad \rho_{s,p}\Delta x W_d M - \rho_{s,p}\Delta x W_d (M + \frac{\partial M}{\partial t}\Delta t) \qquad (3.23)$$

Rearrangement of this equation yields;

$$D_d G_d \frac{\partial H}{\partial x} \quad = \quad -\rho_{s,p}\frac{\partial M}{\partial t} \qquad (3.24)$$

where

H = humidity ratio of the drying air [-]

m = moisture content of the product to be dried [kg_{water}/kg_{solid}]

D_d = width of the gap between the trays [m]

W_d = width of tray [m]

G_d = specific mass flow rate of the air [$kg/s\text{-}m^3$]

$\rho_{s,p}$ = density of product [kg/m^3]

x = distance [m]

t = time [s]

- Moisture balance of the product to be dried

The moisture content of the product is expressed by an appropriate thin layer equation as follows.

$$\frac{M - M_e}{M_0 - M_e} = f(rh, T, t) \tag{3.25}$$

where

M	= moisture of the product at time t [kg_{water}/kg_{solid}]
M_0	= initial moisture content [kg_{water}/kg_{solid}]
M_e	= equilibrium moisture content [kg_{water}/kg_{solid}]
rh	= relative humidity of the air [%]
T	= temperature of the drying air [K]
t	= drying time [s]

The function $f(rh, T, t)$ will be determined by experiment. The detail of the experiment will be explained in the next section.

- Recirculation

A part of the exhaust air from the dryer was recycled and this amount was fixed by design. The original design of recirculation ratio is 0.95. The temperature and humidity of the mixed air were computed as follows:

The humidity and the temperature of the mixed air were computed from the following equations.

Mass balance equations

$$m_2 = m_1 + m_3 \tag{3.26}$$

where

m_1 = mass flow rate from the collector

m_2 = mass flow rate at the outlet of the drying chamber

m_3 = mass flow rate of the air entering the drying chamber

3.2 Equilibrium moisture content

The equilibrium moisture content of an agricultural product is defined as the moisture content of the material after it has been exposed to a particular temperature and relative humidity for an infinite long period of time. The equilibrium moisture content is dependent upon the relative humidity and temperature of the environment as well as on the species, variety and maturity of the product (Bala et al., 1997). There are two types of equilibrium moisture content – static equilibrium moisture content and dynamic equilibrium moisture content. Static equilibrium moisture content is obtained after a long time of exposure of a product to a constant atmosphere where as dynamic equilibrium moisture content is obtained by the best fitting of thin layer drying equation to experimental data.

The concept of dynamic equilibrium moister content was introduced by McEwen et al., (1954). They suggested that dynamic and static equilibrium moisture content should be used for drying and storage design respectively. Allen (1960) also mentioned that the dynamic equilibrium moisture content is a logical choice for the grain drying process, but static equilibrium moisture content is more relevant for storage problems.

Moisture sorption isotherm of food materials is essential in drying, packaging and storage. Moisture isotherm data are extremely valuable in the prediction of food stability. Food quality is usually related to equilibrium moisture content. Most foods have critical moisture content below which the rate of quality loss is negligible (Gal, 1983; Rahman and Lubuza, 1999). The moisture content isotherm can be used to calculate the moisture changes which may occur during storage and also predicting shelf life stability with respect to physical, biochemical, and microbial stability of foods, which in turn determines their quality criteria. In general data on moisture isotherms are indispensable in food product and process development and quality control. The data on moisture isotherms are also essential in simulation and optimization of drying processes and hence the drying systems.

The objective of this subsection study was to conduct experimental measurement of isotherms for banana at various temperatures and relative humilities and to fit the moisture isotherms data to sorption isotherm models.

- Mathematical models of sorption isotherms

Many investigators have developed mathematical equations - theoretical, semi theoretical and empirical to describe the sorption isotherms of agricultural materials. Chirife and Iglesias (1978) reviewed 23 isotherm equations, both theoretical and empirical, and their use for fitting sorption isotherms of foods and food products. None of those equations described adequately the sorption isotherms over the whole range of relative humidity and for all types of food materials tested. Lomauro et al., (1985) evaluated two two-parameter equations and one three parameter equation for 163 food materials including fruits, vegetables, spices and starchy foods. They found that the three parameter Guggenheim, Anderson and van den Berg (GAB) equation (Van den Berg, 1984) described the sorption isotherms for most food better than two parameters equations.

Chen and Morey (1989) proposed four models to described sorption isotherms of various agricultural products and also faced that there is no universal equation for sorption isotherms of agricultural products. The modified Henderson equation and Chung - Pfost equation were the best models for starchy grains and fibrous materials. The modified Halsey equation was the best for high oil and protein products. The modified Oswin model was good for popcorn, peanut pods and other variety of corn and wheat.

More than 200 EMC/ERH purely empirical equations have been developed for cereal grains and food materials (Sun and Woods, 1994). But no single equation accurately describes the EMC/ERH relationships for various crops over a broad range of the relative humidity and temperature. Sun (1998) reported that the modified Chung - Pfost equation, modified Oswin equation, Strohman - Yoerger equation, and modified Halsey equation are the most appropriate equations for describing the EMC/ERH sorption isotherms of wheat, shelled corn, rice and rape-seed respectively.

Recently the GAB model has been proposed by food engineers as the universal model to fit the sorption data for all foods. Lomauro et al. (1985) reported that moisture sorption of foods can be described by more than one sorption model and the GAB gives the best fit for more than 50% of the fruits, meats and vegetables analysed. But Chen and Jayas (1998) reported that GAB model could not be used to describe the sorption isotherms of starchy grains. Furthermore, it is difficult to extend the model by adding a temperature term. Hussain et al. (2001) developed six two - parameter and one - three parameter models to fit the observed data of pineapple and the modified BET model was found to be the best model for pineapple.

Soysal and Oztekin (1999) used the most commonly used equilibrium moisture content and equilibrium relative humidity equations to compare their ability to best fit the published sorption data of selected medicinal and aromatic plants. Both the modified Oswin equation and Halsey equation have the ability to accurately describe the EMC/ERH relationships for selected medicinal and aromatic plants over a broad range of relative humidity and temperature and the Henderson equation is the least successful equation. The modified Oswin equation produces much more reliable results for the temperature range from 5 to 45^{o}C, compared to other two equations. The Halsey equation gives the most reliable results for the 60^{o}C temperature level. They suggested the use of the modified Oswin equation for the 5 to 45^{o}C temperature ranges and the use of Halsey equation for the 60^{o}C level to obtain maximum accuracy.

- Determination of equilibrium moisture content of banana

As the product of this study is banana, we determined experimentally the equilibrium moisture content of banana. Equilibrium moisture content of banana was determined experimentally in the Department of Physics, Silpakorn University, Nakhon Pathom, Thailand using dynamic method. To determine the equilibrium moisture contents three sets of equipments were constructed and each set consists of a hot air chamber and six sample boxes (Figure 13). The hot air chamber was equipped with a 3 kW electrical heater and an electronic temperature controller. The sample box was essentially an airtight plastic box containing saturated salt solution to maintain constant relative humidity inside the box. The box is divided by plastic wall into sections for holding two samples separately and simultaneously. Two small electric fans were used to circulate the air inside the box to accelerate moisture transfer between the samples and air inside the box. The boxes were placed inside the chamber. Thus, the temperature and relative humidity were controlled by temperature controller and salt solution respectively. In conducting the experiments the samples of the products were placed inside the boxes, which were placed inside the chamber. The samples were weighed regularly until they reached equilibrium. The final moisture contents of the product were determined by standard air oven method.

Figure 13 Equipment used for determination of the equilibrium moisture content of banana

- Model Selection

The models selected to fit the sorption isotherms of banana are shown in Table 1. These models were selected on the basis of their effectiveness for describing isotherms of several food and plant materials and simplicity of computation. The parameters of the models were determined by regression analysis and the value of the root mean square error (RMSE) was computed as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(M_p - M_o)^2}{N}} \qquad (3.27)$$

Table 1 Selected isotherm models for experimental data fitting.

Models	Mathematical expression
a. Two variables models:	
1. Bradley (1936)	$\ln \frac{1}{a_w} = b_1 b_2^{M_e}$
2. Halsey (1948)	$a_w = \exp - \left[\frac{b_2}{M_e^{b_1}} \right]$
3. Henderson (1952)	$1 - a_w = \exp - \left[\frac{b_2}{M_e^{b_1}} \right]$
4. Iglesia and Chirife (1982)	$M_e = b_1 \left(\frac{a_w}{1 - a_w} \right) + b_2$
5. Kuhn (1967)	$M_e = \frac{b_1}{\ln a_w} + b_2$
6. Oswin (1946)	$M_e = b_2 \left[\frac{1}{1 - a_w} \right]^{b_1}$
7. Smith (1947)	$M_e = b_2 - b_1 \ln (1 - a_w)$
8. BET (1938)	$\frac{a_w}{(1 - a_w) M_e} = \frac{1}{b_1 b_2} + \frac{a_w (b_2 - 1)}{b_1 b_2}$
9. Empirical	$M_e = b_0 + b_1 a_w + b_2 a_w^2 + b_1 a_w^3$

Table 1 (Continued)

Models	Mathematical expression
b. Three variables models:	
1. Chung- Pfost (1967)	$\ln a_w = -\frac{b_1}{RT} \exp(-b_2 M_e)$
2. Day and Nelson (1965)	$1 - a_w = \exp(-b_1 T M_e^{b_2})$

Sorption isotherm of banana is shown in Figure 14 at three temperatures 30°C, 40°C and 50°C in the range of 15% - 98%. The isotherm curves have sigmoid shape and similar patterns. Eleven isotherm models were fitted to the experimental data of banana. Computed parameters and their coefficients of determination for the best fitted sorption isotherms of banana are shown in Table 1. The observed (points) and the best predicted (solid lines) sorption isotherms are shown in Figure 15. The agreement between the observed and the best fitted results are excellent for banana. The EMC values decreases with increase temperature at all levels of relative humidity. The kinetic energy associated with water molecules present in banana increase with increasing temperature. This in turn, resulted in decreasing alternative forces and consequently escape of water molecules. This led to a decrease in EMC values with increasing temperature at a given relative humidity. Several researches (Iglesias and Chirife, 1982; McLaughlin and Magee, 1998; Hussain et al., 2001 and Shivhare et al., 2004) have reported similar trends for food materials.

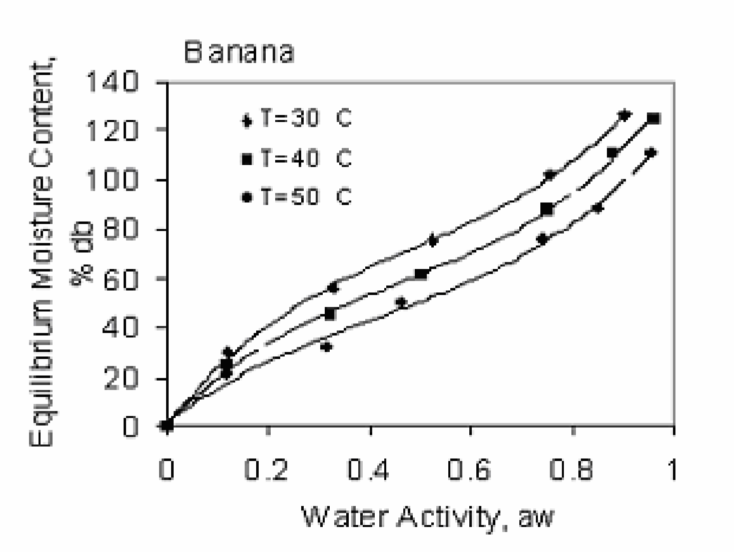


Figure 14 Measured isotherms for 30°C, 40°C and 50°C of temperatures.

Table 2 The coefficients of the best models, root mean square error (RMSE) and the coefficient of determinant (R^2) for banana.

Crop	Temp. (°C)	Model	Coefficients				RMSE	R^2
			b_0	b_1	b_2	b_3		
Banana	30	Henderson		2.2187	0.00005		1.0763	0.9991
	40	Empirical	1.732	215.27	-299.02	218.4	1.9922	0.9978
	50	Henderson		1.8754	0.00044		2.9587	0.9864

Both the Henderson equation and the empirical equation have the ability to accurately describe the EMC/ERH relationship for banana. On the basis of the standard error of estimate (RMSE) and the coefficient of determination (R^2) the Henderson equation and the empirical equation were found to describe adequately the equilibrium moisture content of banana. The Henderson equation produces much more reliable results for the temperature of 30°C and 50°C. The empirical equation gives the much reliable results for 40°C temperature level.

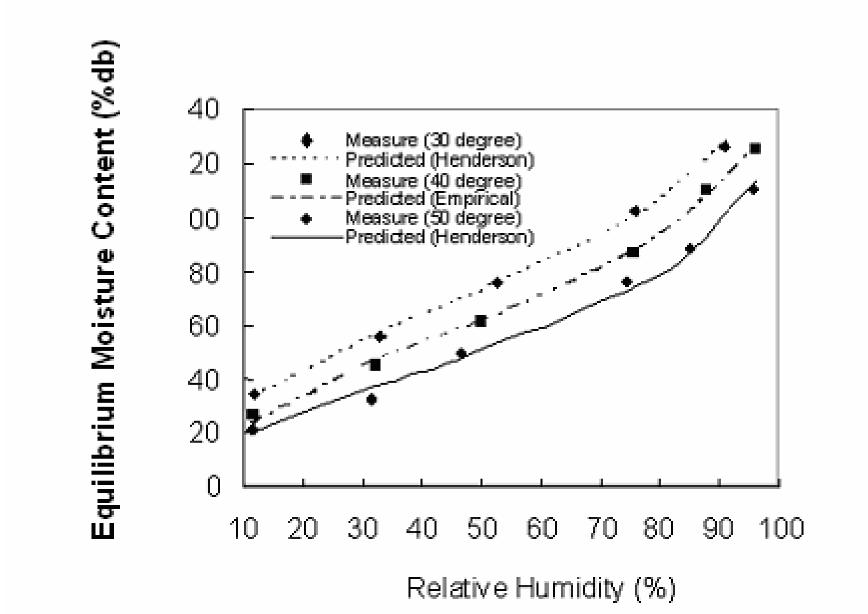


Figure 15 The experimental and the best predicted isotherms of banana for 30°C, 40°C and 50°C of temperatures.

3.3 Thin layer drying of banana

Banana is known as a crop of tropical climates and it is grown in Thailand for both fresh and dried consumption. Dried banana is a popular in Thailand and it has a wide domestic market. Sun drying is largely used for preserving it. Small-scale food industries are also now becoming popular in Thailand and these industries are now intensely supported by Thai government.

Drying is an energy intensive operation. Understanding the drying behavior of banana is very important for controlling drying process and for production of quality dried banana. Furthermore thin layer drying equation of banana is needed for simulation of the performance of mechanical / solar dryer and its component optimization (Bala, 1998).

Solar drying system must be properly designed in order to meet particular drying requirement of specific crops and to give satisfactory performance with respect to energy requirement (Steinfeld and Segal, 1986). Simulation models are needed in the solar dryer designs and in the operation of drying systems (Diamante and Munro, 1993). The prediction of drying rate of the specific crops under various conditions is important for the design of solar drying system (Zaman and Bala, 1989).

Thin layer Drying models are required in simulating the drying behavior and drying time under any drying condition. The designing of drying systems and optimization of dryer performance can be made using simulation models and it provides an opportunity for the assessment of the energy conservation and saving alternative fuels. This process is advantageous since full-scale experimentation for different configuration of drying system for different product is very time consuming, and therefore not always feasible (Bala et al., 1997). Thin layer drying models are essential for simulation studies of the dryers. The efficiency of the drying system can be improved by the analysis of the drying process. Analysis of the drying system can be greatly expedited by using computer simulation. A thin layer drying equation is, therefore, needed for banana and this equation must be suitable for use at any temperature and relative humidity of the drying air for banana drying.

A large amount of work on thin layer drying of agricultural crops has been conducted and numerous mathematical models have been proposed. But limited study has been conducted on thin layer drying of banana (Ploungchandang and Woods, 2000). This subsection presents laboratory experiments on thin layer drying of banana under controlled conditions of the drying air and development of a thin layer drying equation.

Thin layer drying of banana has been reported by some researchers. Queroz and Nebra (2001) reported a theoretical and experimental study on the drying kinetics of banana under different air drying conditions. The diffusion model with constant diffusion coefficient, equilibrium boundary conditions and without shrinkage assumption did not adequately represent the banana drying process. The convective boundary condition and shrinkage phenomenon when both are included in the diffusion models improved significantly the fitting of theoretical results to the experimental data. However, the best model was obtained when only the convective boundary condition was included in the diffusion model. Dandamrongrak et al. (2002) studied thin layer drying behavior of banana under pre-treatments and examined three models: simple exponential model, two-term exponential model and page model. The rate of drying was higher for pretreatment involving freezing but the sample, which was blanched only, did not show any improvement in drying rate. While all the models closely fitted to the drying data, simple model showed greater deviation from the experimental data. The two-term exponential model found to be the best model for describing the drying behavior of banana. Demirel and Turhan (2003) reported air-drying behavior of untreated and sodium bisulphite and ascorbic/citric acid treated Dwarf Cavendish and Gros Michel banana slices between 40°C and 70°C. Pretreatment and increasing temperature decreased browning and the color change in the untreated samples was acceptable. Pretreatment and temperature did not affect the shrinkage. The drying of banana was evaluated using diffusion equation for an infinite slab and in all the cases the equation fitted the experimental data well.

- Thin Layer Drying Equation

The general approaches to the study of thin layer drying are:

(i) The development of empirical equation
(ii) The development of theoretical equation
(iii) The development of semi theoretical equation.

The theoretical approach concerns with either the diffusion equation or the simultaneous heat and mass transfer equation. The semi theoretical concerns approximated theoretical equation. The main justification for the empirical equation is a satisfactory fit to all experimental data and subsequent benefit in the simulated drying. The reasons for using theoretical equations are to give some physical explanation and understanding for the transfer process. The approximated equations are simpler and take less computing time in comparison to the theoretical model and provide some understanding of the transfer process (Bala, 1998)

Theoretical modeling predicts the moisture transfer within the product involving capillary flow, liquid diffusion, vapor diffusion, surface diffusion, thermal diffusion and thermodynamic flow (Parry, 1985). Luikov (1966) developed the following equation for describing the drying of capillary porous product based on physical mechanisms:

$$\frac{\partial M}{\partial t} = \nabla^2 K_{11} M + \nabla^2 K_{12} T_p + \nabla^2 K_{13} P \tag{3.28}$$

$$\frac{\partial T_p}{\partial t} = \nabla^2 K_{21} M + \nabla^2 K_{22} T_p + \nabla^2 K_{23} P \tag{3.29}$$

$$\frac{\partial P}{\partial t} = \nabla^2 K_{31} M + \nabla^2 K_{32} T_p + \nabla^2 K_{33} P \tag{3.30}$$

The moisture flow due to a total pressure gradient is not significant in the temperature ranges employed in the drying of agricultural product. Thus, the system of differential equation (3.28) to (3.29) reduces to

$$\frac{\partial M}{\partial t} = \nabla^2 K_{11} M + \nabla^2 K_{12} T_p \tag{3.31}$$

$$\frac{\partial T_p}{\partial t} = \nabla^2 K_{21} M + \nabla^2 K_{22} T_p \tag{3.32}$$

Thermal diffusion is negligible for agricultural materials and may be assumed to be zero. Again the contribution of $\nabla^2 K_{21} M$ to $\partial T_p/\partial t$ is also negligible. Hence the system of equations (3.31) to (3.32) reduces to

$$\frac{\partial M}{\partial t} = \nabla^2 K_{11} M \tag{3.33}$$

$$\frac{\partial T_p}{\partial t} = \nabla^2 K_{22} T_p \tag{3.34}$$

In agricultural materials the thermal diffusivity is large compared to moisture diffusivity. Neglecting the moisture diffusivity leads to the following equation:

$$\frac{\partial M}{\partial t} = \nabla^2 K_{11} M \tag{3.35}$$

Equation (3.35) is the well known fundamental differential equation of diffusion and is referred to as Fick's second law of diffusion. The equation (3.35) can be expressed in term of vector analysis as

$$\frac{\partial M}{\partial t} = \text{div}(D_v \,\text{grad}\, M) \tag{3.36}$$

The solution of the diffusion equation (3.36) in cartesian, spherical and cylindrical coordinates have been discussed by Crank (1979). Thus, the solution for sphere is

$$\frac{M - M_e}{M_o - M_e} = \frac{6}{\pi^2}\sum_{n=1}^{\infty}\frac{1}{n^2}\exp(-n^2\frac{D_v \pi^2}{r^2}t) \tag{3.37}$$

and solution for a plan sheet of half thickness z is

$$\frac{M - M_e}{M_o - M_e} = \frac{8}{\pi^2}\sum_{n=1}^{\infty}\frac{1}{(2n-1)^2}\exp(-(2n-1)^2\frac{D_v \pi^2}{r^2}t) \tag{3.38}$$

This equation converges rapidly because of the omission of alternate terms. A Simplification of equation (3.37) to (3.38) has been frequently used to describe the thin layer drying of agricultural products. If only the first term is considered, these equations can be approximated to the form:

$$\frac{M - M_e}{M_o - M_e} = a\ \exp(\text{-}Kt) \tag{3.39}$$

when a = 1, the equation reduces to the single exponential or Newton equation.

A relationship analogous to Newton's law of cooling (Lewis, 1921) is often used in drying analysis and the rate of moisture loss is assumed proportional to the moisture remaining to be lost and it is as follows:

$$\frac{dM}{dt} = -K(M - M_e) \tag{3.40}$$

The integration of the equation (3.40) yields the Newton or single exponential equation as follows:

$$\frac{M - M_e}{M_o - M_e} = \exp(-Kt) \tag{3.41}$$

Several investigators (Boyce, 1966; Kachru et al., 1991; Watson and Bhargava, 1974; Bala and Woods, 1992; Shei and Chen, 1998; Buser et al., 1999; Hussain and Bala, 2002 and Dandamrongrak et al., 2002) have fitted the above equation for barley, shelled corn, malt, rough rice, marigold flower, green chili and banana.

Page (1949) proposed the flowing empirical equations for shelled corn

$$\frac{M - M_e}{M_o - M_e} = \exp(-Kt^n) \tag{3.42}$$

Several investigators (Bala and Woods, 1992; Guarte, 1996; Bashir et al., 1998; Afzal and Abe 1999; Yaldyz and Ertekyn, 2001) have reported that page equation adequately predicts the thin layer drying of malt, copra, sliced onion, potato and green bean. Dandamrongrak et al., (2002) fitted the page equation to the thin layer data of banana and found good agreement between the simulated and experimented data.

Nellist and O' Callaghan (1971) proposed the following double exponential equation for thin layer drying of ryegrass seed:

$$M = M_e = A \exp(-K_1 t) + B \exp(-K_2 t) \tag{3.43}$$

Several investigators (Sharaf-Eldeen et al., 1979; Yaldyz and Ertekyn, 2001; Bala, 1983 and Dandamrongrak et al., 2002) reported that double exponential equation describes adequately the thin layer drying behavior of corn, red chili, malt and banana.

It is quite reasonable to assume that thermal diffusivity is large as compared to moisture diffusivity for banana and the Newton equation is accepted as a standard equation. Considering these factors it is proposed to fit the experimental data in the Newton equation and to develop thin layer drying equation in the following form:

$$\frac{M - M_e}{M_o - M_e} = \exp(-Kt) \qquad (3.44)$$

- Thin Layer Drying Apparatus

The drying experiments were carried out using the laboratory dryer in the Institute for Agricultural Engineering, the University of Hohenheim, Stuttgart, Germany (Figure 16 and 17). The thin layer drying apparatus is described in details by Guarte (1996).

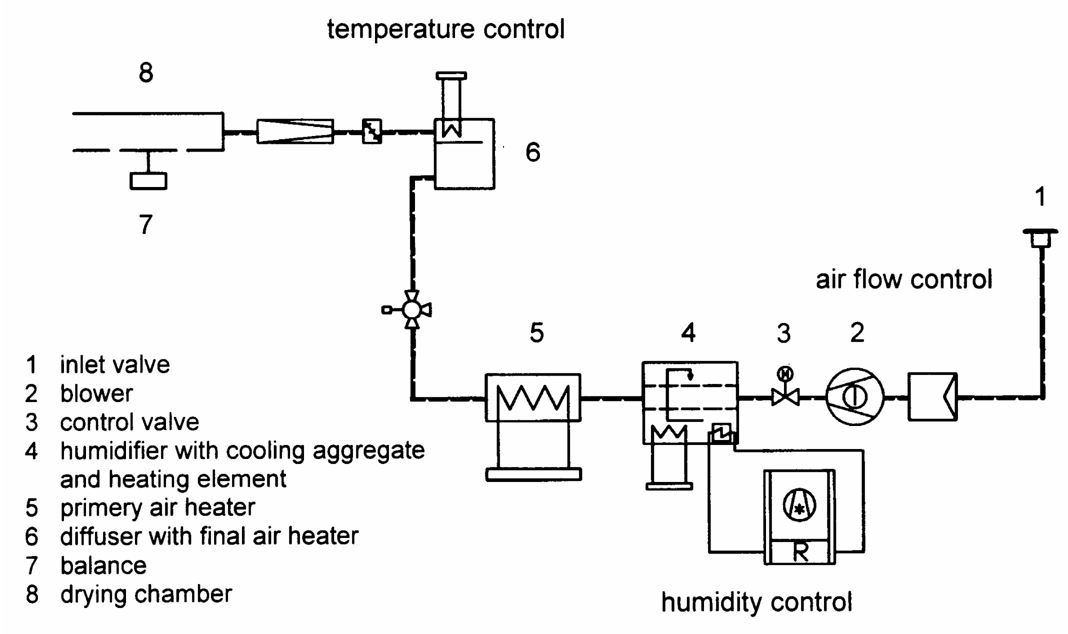


Figure 16 Schematic diagram of the drying apparatus (Hohenheim Type, the University of Hohenheim, Stuttgart, Germany).

Figure 17 Pictorial view of the drying apparatus (Hohenheim Type, the University of Hohenheim, Stuttgart, Germany).

The laboratory dryer consisted of an airflow control assembly (1-3), humidity control assembly (4), temperature control assembly (5-6), balance (7) and the drying chamber (8) as shown in Figure 16. Each assembly was electronically controlled by a proportional plus-reset (integral) (PI) controller. The airflow control assembly regulated the amount of fresh air entering the drying chamber at an accuracy of plus or minus 0.05 m/s. The desired initial dew point temperature of the drying air was attained by the humidity control assembly at an accuracy of plus or minus 0.5 K. The air passed through humidity control assembly and then heated to the desired temperature and channeled to the drying chamber. At the drying chamber, the hot air was further exposed to secondary heating elements (6) to obtain uniform temperature distribution and accurate temperature control of plus or minus 0.5 K. Weighing of samples inside the drying chamber was carried out automatically through a load cell at 10-min intervals, although any desired interval could be achieved. During weighing, the drying air was temporarily diverted through a by-pass to get accurate values. All data were recorded automatically in the computer.

- Thin Layer Drying Experiments

Thin layer drying experiments under controlled conditions were conducted using laboratory dryer at the University of Hohenheim, Germany. Different combinations of air temperature and relative humidity were considered and a total of 9 experimental runs were conducted. The air conditions of thin layer drying of banana are given in Table 3.

Table 3 Condition for thin layer drying of banana.

Experiment number	Relative humidity (%)	Temperature (^{o}C)
1.	10	50
2.	10	60
3.	10	70
4.	20	50
5.	20	60
6.	20	70
7.	25	50
8.	25	60
9.	25	70

- Experiment Results

The moisture content was plotted against time for different conditions of the drying air as shown in Figure 18. The experimental results show that the drying air temperature and relative humidity have strong influence on the drying rate of banana. The drying rate increases with the increase of temperature. The reason might be that the equilibrium moisture content of banana decreases with an increase in drying air temperature. It is observed from Figure 18 that there is no constant rate period of drying; drying takes place at falling rate period. Drying rate of banana is higher at lower relative humidity and decreases with the increase of relative humidity.

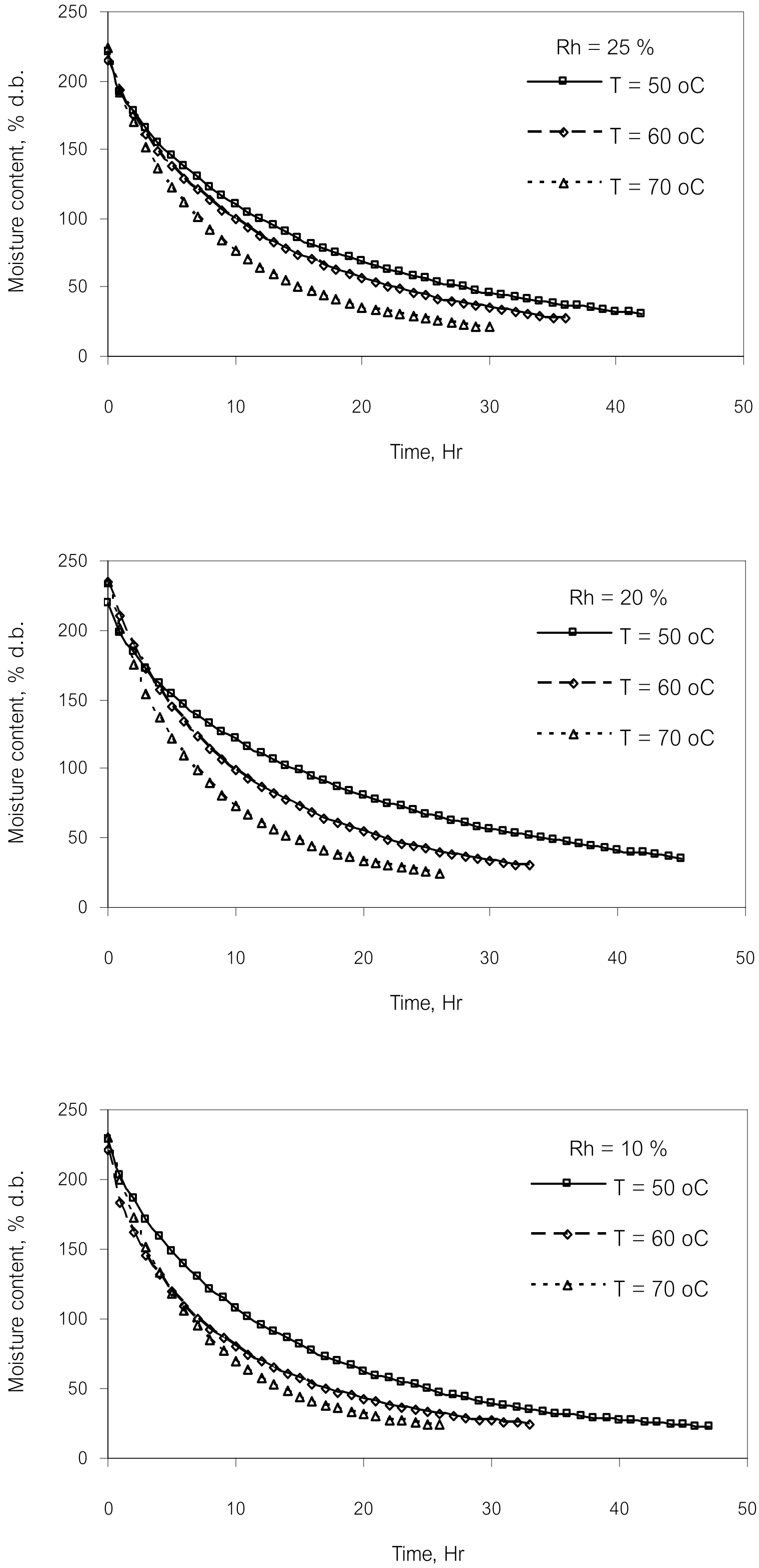


Figure 18 Thin layer drying curves of banana at different drying conditions.

The following empirical equation was fitted to experimental data (Bala, 1998).

$$\frac{M - M_e}{M_o - M_e} = Ae^{-Bt} \tag{3.45}$$

where the constant A and B were expressed as a function of relative humidity and temperature time (hr) in the following form:

$$A = a_o + a_1\,rh + a_2\,T + a_3\,rh^2 + a_4T^2 \tag{3.46}$$

$$B = b_o + b_1\,rh + b_2\,T + b_3\,rh^2 + b_4T^2 \tag{3.47}$$

Where rh = relative humidity of the drying air

T = temperature the drying air

t = time

a_i, and b_i (i = 0, 1, 2, 3, 4) are empirical constants.

The equation (3.45) was fitted to nine sets of experimental data by using regression routine in Excel. The predicted and experimental moisture contents are shown in Figure 19. The agreement between the predicted and observed moisture contents is good.

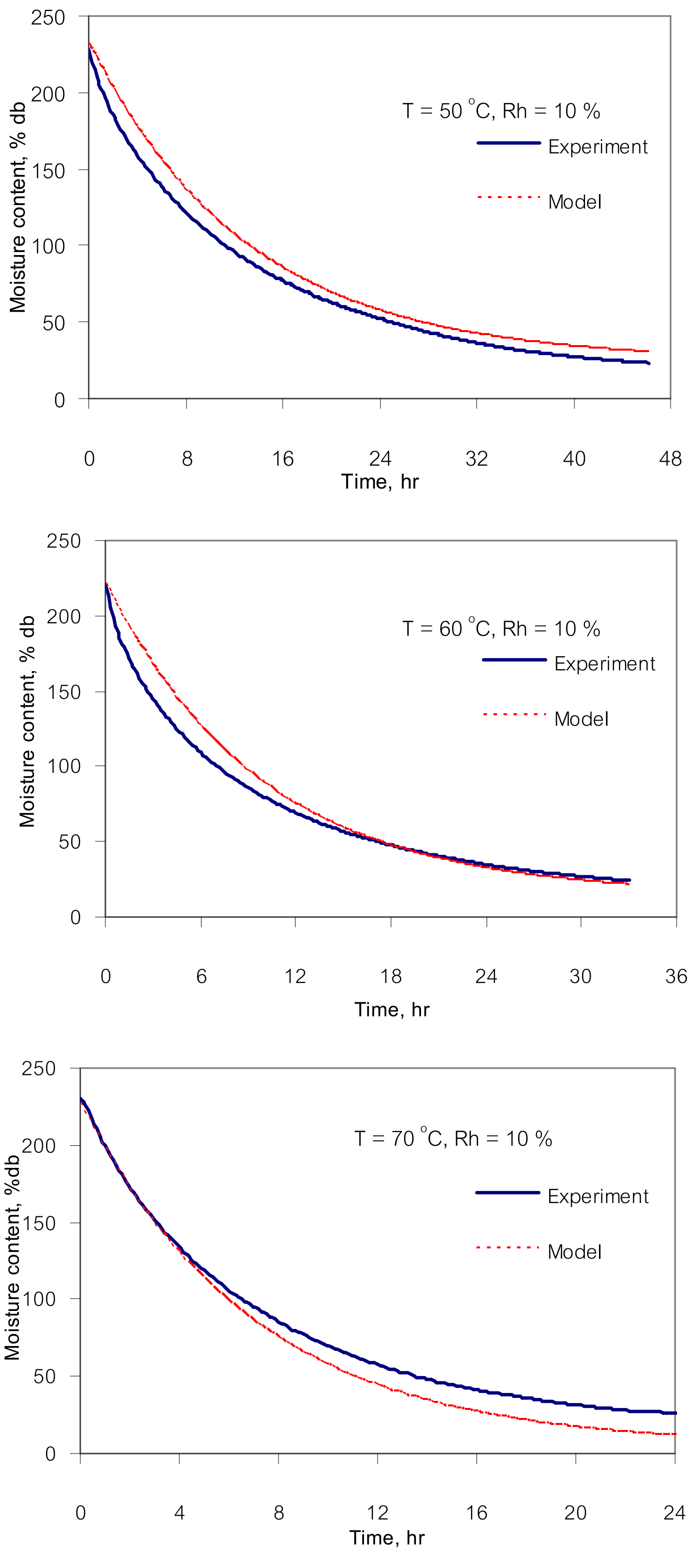


Figure 19 The predicted and experimental moisture contents during drying banana at various drying conditions.

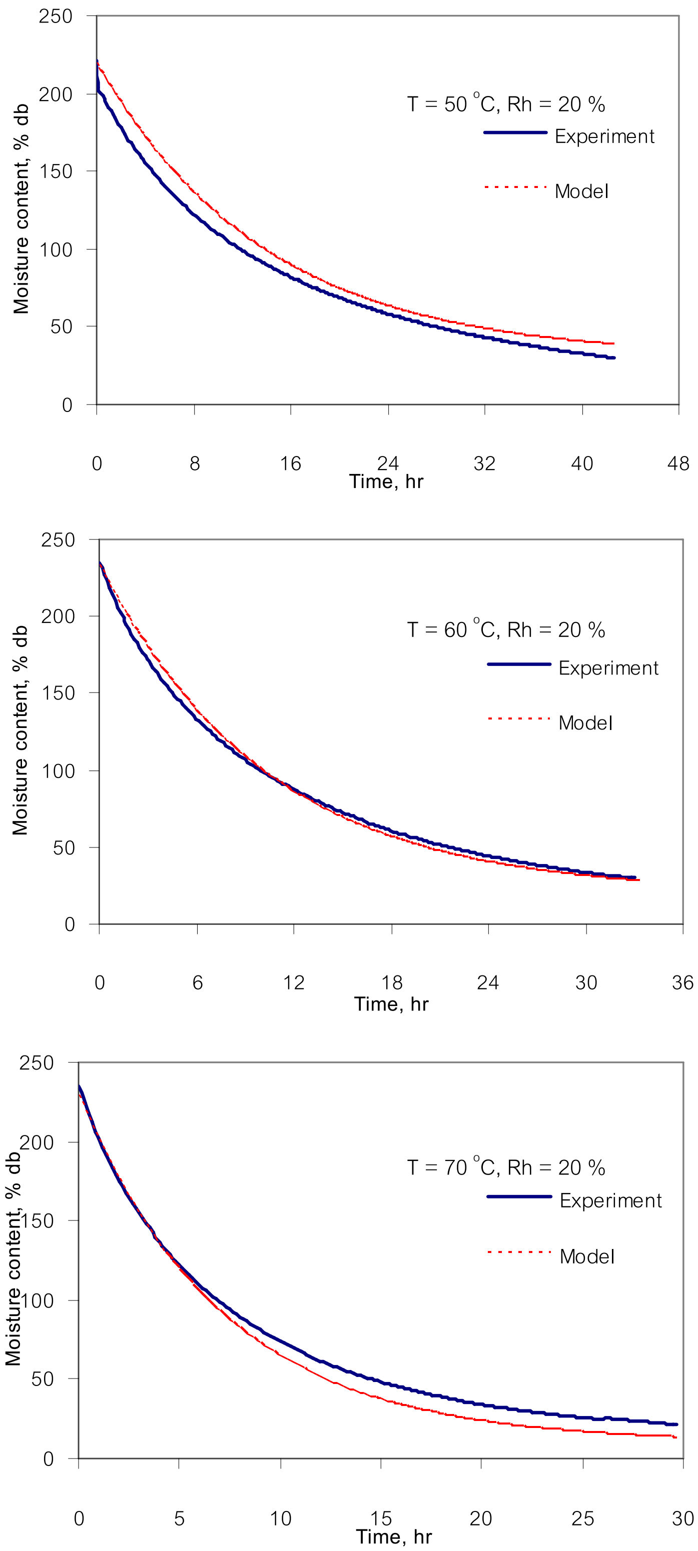
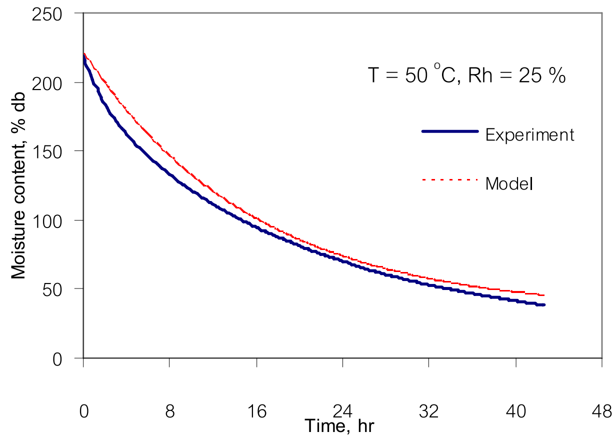
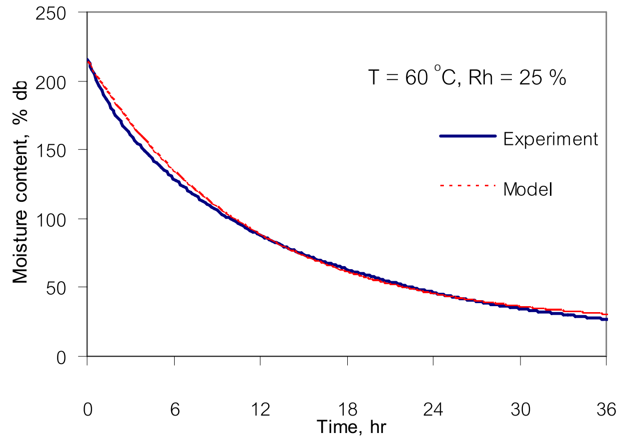


Figure 19 The predicted and experimental moisture contents during drying banana at various drying conditions. (cont.)





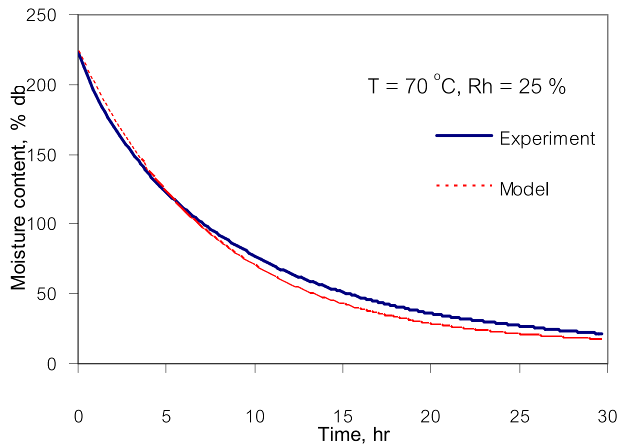


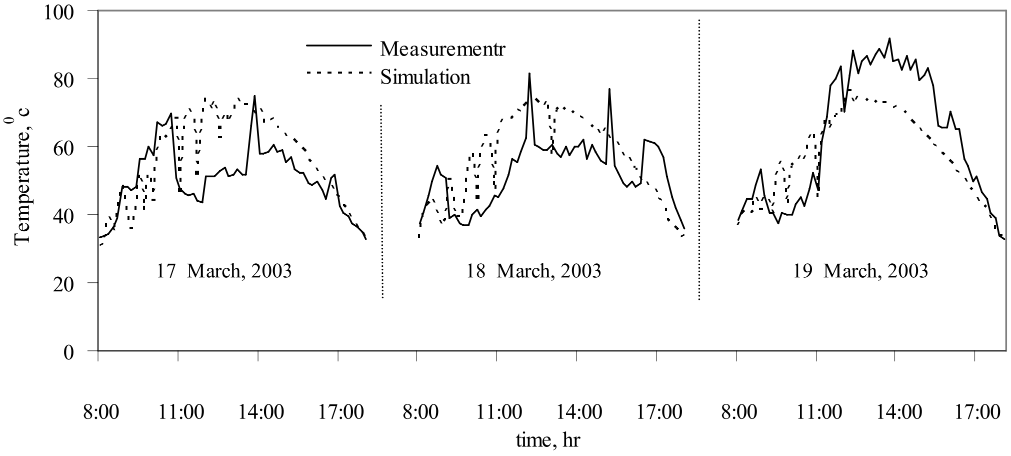
Figure 19 The predicted and experimental moisture contents during drying banana at various drying conditions. (cont.)

3.4 Experimental validation of the model

To build up confidence in the model, drying experiments were carried out. The results obtained from the experiments were compared with those calculated from the model. The details of the experiments are explained as follows.

For the solar collector, it outlet air temperature obtained from the measurement was compared with that calculated from the collector model. The results are shown in Figure 20. For most cases the variation pattern of the simulated temperature agrees with that of the measurement.

For the dryer, we compared the moisture content of bananas obtained from the measurement with those calculated from the dryer model, as the results shown in Figure 21. It is observed that the simulated results agree well with the calculated results. These simulation models will be used for the optimization process.



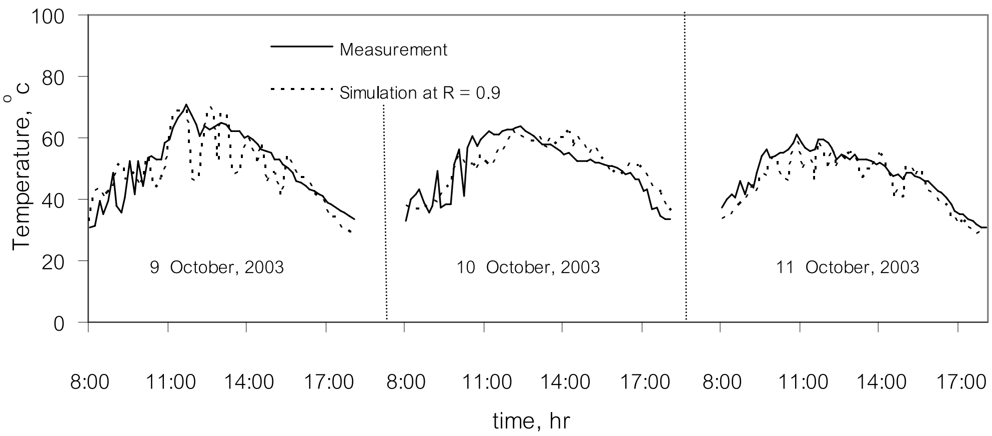
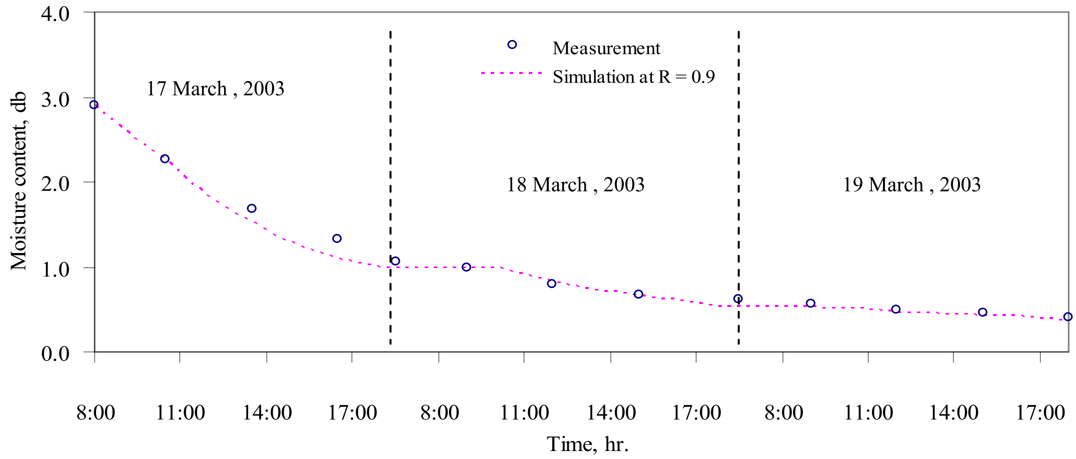


Figure 20 Predicted and observed variation in the collector temperature.



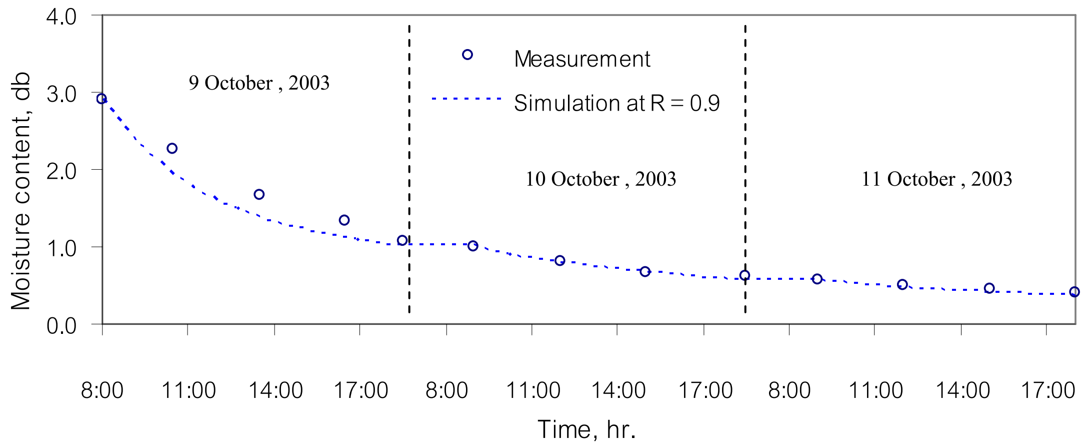


Figure 21 Simulated and observed moisture content of bananas for the drying test

3.5 Formulation of the economic model of the solar-assisted fruit drying system

The various cost of drying system was estimated based on the real cost of the construction and operation of this drying system because the system has been used since the year 2000. The capital cost of the system consists of the cost of the collector, drying cabinet and labor cost for the construction and installation. The capital cost of this drying system is listed as follows:

- Solar collector 2,000 USD
- Drying cabinet 2,833 USD
- Labor (construction and installation) 500 USD

(1 USD = 40 Bahts)

The collector material costs are those for glass and polyurethane back insulator. The drying cabinet material costs are those for the trays, LPG burner, air duct, blower, system control, insulation materials and structural materials.

The cost of the collector ($C_{collector}$) can be expressed as

$$C_{collector} = C_{unit,col} \times A_c \qquad (3.48)$$

where $C_{unit,col}$ is the unit cost of the collector and A_c is the collector area.

The total capital cost for the solar-assisted drying system (C_T) is given by next equation

$$C_T = C_{collector} + C_f \qquad (3.49)$$

where C_f is the material cost of the drying cabinet and the labor cost for construction and installation.

The annual cost calculation method proposed by Audsley and Wheeler (1978) yields:

$$C_{annual} = \left[C_T + \sum_{i=1}^{N}\left(C_{main,i} + C_{op,i}\right)\omega^i\right]\left[\frac{\omega - 1}{\omega\left(\omega^N - 1\right)}\right] \quad (3.50)$$

where C_{annual} is the annual cost of the system. $C_{maint,i}$ and $C_{op,i}$ are the maintenance cost and the operating cost at the year i, respectively. ω is expressed as

$$\omega = (100 + i_{in}) / (100 + i_f) \quad (3.51)$$

where i_{in} and i_f are the interest rate and the inflation rate in percent, respectively.

The maintenance cost of the first year was assumed to be 1% of the capital cost. The operating cost consists of the gas consumption cost (C_{gas}), electricity consumption cost ($C_{electric}$) and the labor cost for operating the system ($C_{labor,op}$). This cost can be written as follows:

$$C_{op} = C_{gas} + C_{electric} + C_{labor,op} \quad (3.52)$$

The cost of the gas consumption is calculated as follows:

$$C_{gas} = C_{unit,gas} \times M_{gas} \quad (3.53)$$

where $C_{unit,gas}$ is the unit cost of the gas and M_{gas} is the mass of the gas used per year.

The cost of the electricity consumption can be calculated as

$$C_{electric} = C_{unit,elect} \times M_{elect} \quad (3.54)$$

where $C_{electric}$ is the unit cost of the electricity and M_{elect} is the amount of the electricity used per year.

The annual cost per unit of dried product is called the drying cost (Z). It can be written as

$$Z = \frac{C_{annual}}{M_{dry\ product}} \tag{3.55}$$

where $M_{dry\ product}$ is the dried product obtained from this drying system per year.

3.6 Identification of system parameters for the optimization

In drying technology, drying cost is usually used as an objective function for the optimization. For a given set of economic parameters (Bala, 1998), this drying cost is influenced by the technical parameters of the system. These parameters can be divided into 4 groups as follows:

1. Geometrical parameters such as length, width, collector area of the dryer. These parameters can be varied with the range dictated by the technical constrain.
2. Operating parameters. Examples of these parameters are air flow rate and recycle factor of the drying air parameters. These can be varied to meet a technical requirement.
3. Material properties. These are the thermal properties of materials use in drying unit and the collector such as the absorptance and emittance of the absorber. The parameters are usually fixed from the selection of materials for the construction of the dryer.
4. Environmental parameter. In a solar drying system, the environmental parameter is solar radiation, ambient relative humidity, ambient air temperature and wind speed. These parameters can not be controlled.

As the material properties are usually fixed and the environmental parameters cannot be controlled, the parameters which we can optimize are the geometrical parameters and the operating parameters. To identify the parameters, which are necessary to optimize, we simulate the performance of the model based on the technical and economic model developed in the last sections. The simulation was done for a whole year using the typical weather data of Nakhon Pathom province and these simulated results are shown in Figure 22 and 23. From these figures it is evident that the parameters to be optimized are collector area and recycle ratio. Another important factor is drying air temperature which is related to quality of the dried product.

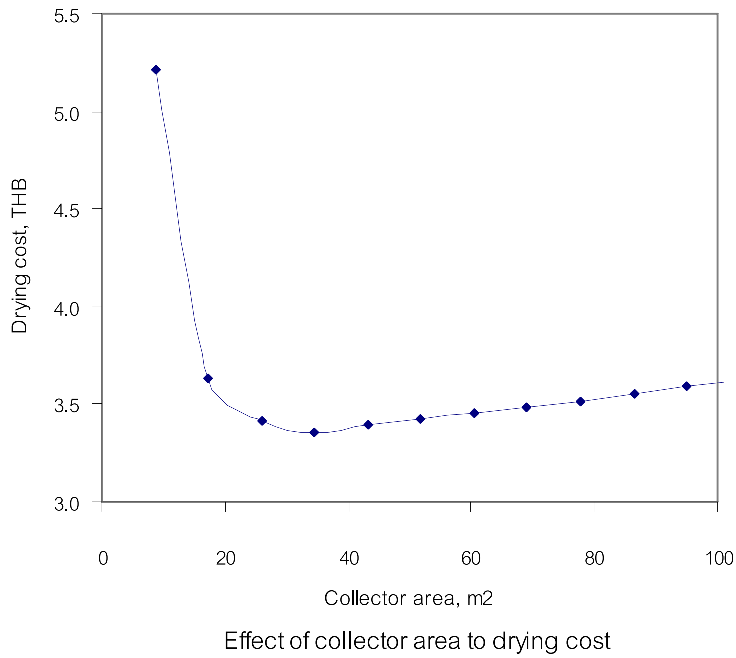


Figure 22 Influence of collector area (A_c) on the drying cost.

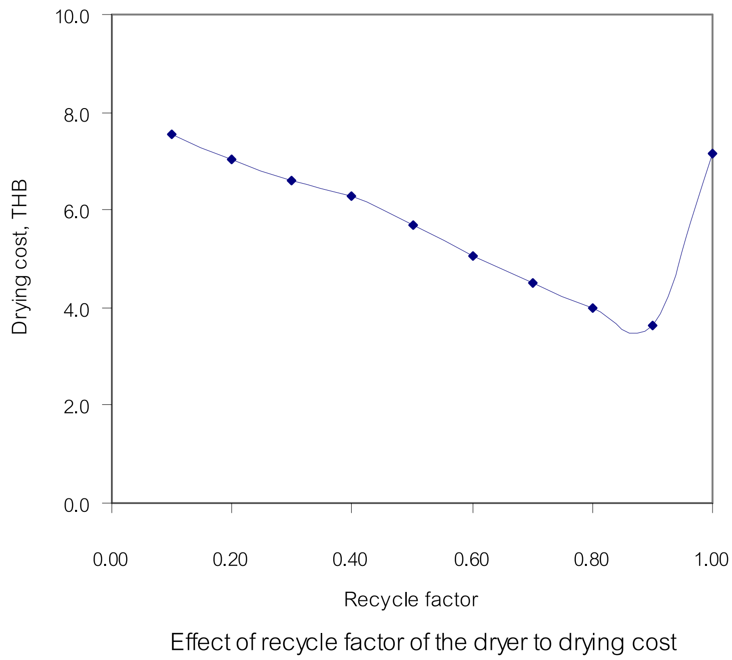


Figure 23 Influence of the recycle factor (R_c) on the drying cost.

3.7 Optimization method

3.7.1 Formulation of the objective function

Solar drying systems must be properly designed in order to meet particular drying requirements of specific products and to give satisfactory performance with respect to requirements. Designers should investigate the basic parameters such as collector area, recirculation ratio and product quality. However, full scale experiments for different products, drying seasons and system configurations are sometime costly and may not be feasible. The simulation model is a valuable tool for predicting the dryer performance and it is also least costly and less time consuming. Again, simulation model is essential to optimize the dryer component and drying process, and optimization techniques can be used for optimal design of solar drying systems.

Scientists usually search for optimum process, method and technique. The aim is to search for the optimum solution, often a better solution and sometimes at least a least bad solution (Habchi, 2000). The purpose of the optimization is either maximization or minimization of the objective function subjected to certain restrictions. In case of solar assisted dryer, a too large collector enables a large amount of drying but may not economic and further this may cause some space problem in installation on the roof top. On the other hand a small collector is good for small quantity but may not be cost effective. Again, recycling part of the exhaust air having drying potential may improve the dryer performance and the economy of drying. It is, therefore essential to optimize the collector area and recycle ratio of the solar assisted dryer to minimize the cost per kg of dried product. Thus the objective function of this problem is to minimize the cost per kg of dried product (Z) and it is expressed as:

$$Z = \frac{\left[C_T + \sum_{i=1}^{N}(C_{maint,i} + C_{op,i})\omega^i\right]\left[\frac{\omega - 1}{\omega(\omega^N - 1)}\right]}{M_{dry\ product}} \tag{3.56}$$

The construction of the solar assisted dryer at the Royal Chitralada Project was based on the approximately design. Thus, it is needed to be optimized the dryer geometry and drying process such as the area of the collector (A_c) and the percentage of air recycle (R_c).

Several studies have been reported on the optimization of forced convection solar drying systems (Radajewski et al 1987, 1989 and 1990). In a forced convection system the mass flow rate of air is controlled explicitly, whereas in a natural convection system mass flow is an implicit function of the temperature profiles. As a result the simulation and optimization of forced convection systems is relatively simple (Bala and Woods, 1995).

Bala and Woods (1995) developed an optimization technique for the dryer geometry of a natural convection solar dryer. They combined the physical simulation with a cost model and search technique. The main feathers for optimum design were a relatively long collector, a thin grain bed and a negligible chimney height. The cost of the optimum design was 60% of that of a typically design. Simate (2003) optimized the length, width, and height of a mixed mode and an indirect natural convection solar dryer using the optimization technique proposed by Bala and Woods (1995). The optimization gave a shorter collector length for the mixed mode dryer (1.8 m) than for the indirect mode dryer (3.34 m) of the same grain capacity (90 kg). The optimization showed that the drying chamber length is not sensitive to the capacity ratio.

Hossain et al. (2005) optimized the geometry of solar tunnel dryer for drying of chilli without color loss. The simulation model was combined with the economic model of the solar tunnel dryer and adaptive pattern search was used to find optimum dimensions of the collector and the drying unit. The capacity of the optimum designed dryer is higher than the typically used dryer and achieves a cost saving of about 16%. Sensitivity analysis shows that the design geometry is sensitive to the costs of major construction materials of the collector and air temperature inside the dryer

The specific objectives of the subsection are:

(i) to present a general optimization technique for solar assisted drying systems as a case study of optimal design;

(ii) to examine the sensitivity of the collector/dryer system to the price of construction materials, fixed costs and the capacity of the system;

3.7.2 Formulation of optimization of the collector area and recycle ratio

The optimization of the dryer component and recycle ratio may be stated mathematically as

$$\text{Minimize } Z = \frac{\left[C_T + \sum_{i=1}^{N} (C_{\text{maint},i} + C_{\text{op},i})\omega^i\right]\left[\frac{\omega - 1}{\omega(\omega^N - 1)}\right]}{M_{\text{dry product}}} \tag{3.57}$$

Subject to the constraints

$$0 \le A_c \le A_{max}$$

$$0 \le R_c \le R_{max}$$

where A_{max} is the maximum limit of collector area, R_c is the recycle ratio and R_{max} is the maximum limit of the recycle ratio.

3.7.3 Method for the search for the optimum parameters

There are at least a dozen of published computerised search codes and probably even a greater number of unpublished codes. Taubert (1968) selected three promising search codes with published FORTRAN programs as possible candidates for inclusion in the search decision rules (SDR) and these codes are

(1) Conjugate gradient search
(2) Variable metric search and
(3) Pattern search

In every case pattern search exhibited the fastest convergence and conjugate gradient search the slowest. As a result Taubert (1968) selected the pattern search technique for incorporation in the search decision rule. In this study the adaptive pattern search which is essentially a pattern search incorporated in search decision rule (SDR) is selected.

The problem of optimization in equation (3.57) is as a problem of minimization under constrained conditions. The problem itself is a nonlinear optimization under constrained conditions. The components of the dryer and drying of banana to be optimized are collector area and recycle ratio under constrained conditions.

The optimum collector area and the recycle ratio are determined by adaptive pattern search technique. The technique essentially consists of exploratory search and pattern search under constrained conditions. The first kind of search is included to explore the local behavior of the objective function and the second of move is included to take advantage of pattern direction within the range of constraints.

Pattern search is a procedure for systematically varying the independent variables ($x_1,x_2,x_3.....x_n$) of an objective function in an attempt to locate the minimum value of the function. The search is based on the heuristic that if move from one point to another in n-dimension space is successful, the next move should also be made is this direction. This move is termed a pattern move by Hooke and Jeeves (1961) and utilizes information concerning the local characteristics of the response surface supplied by an exploratory search conducted around each base point.

A descriptive flow chart of the exploratory search logic is given in Figure 18. In this routine vector X contains the n coordinate values of the base point to be studied and vector D contains the corresponding step sizes computed during the last exploratory search. The behavior of the function is inferred entirely from the success or failure of the objective function evaluations as the value of each coordinate is modified. The gradient vector is neither calculated nor is any significance placed on the absolute magnitude of each success or failure. Only the success/failure information is used to adjust the step sizes for future exploratory searches and to form a pattern indicating the probable direction or another successful move.

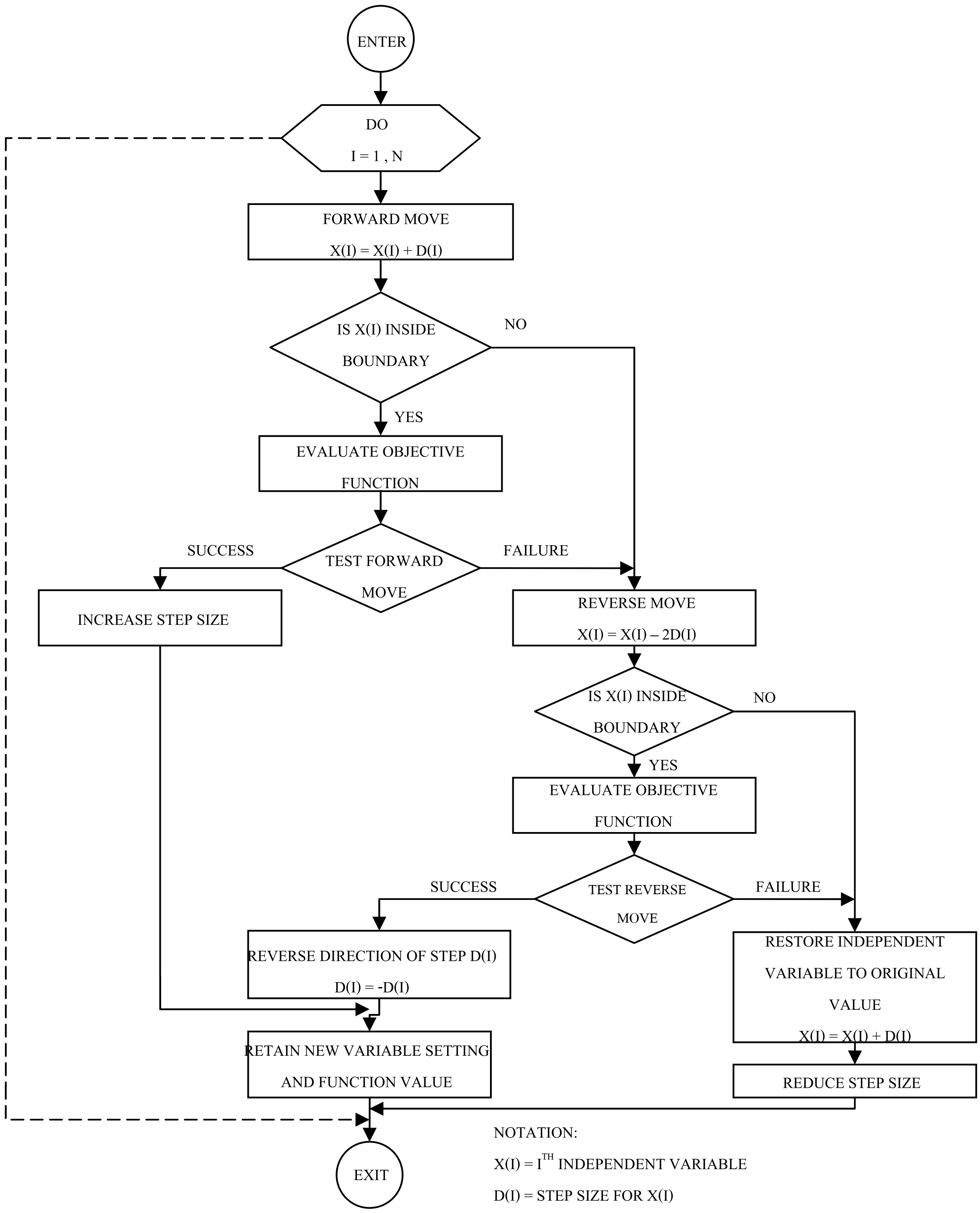


Figure 24 Descriptive flow chart of the pattern search

The pattern move logic shown in Figure 24 essentially moves the search from the current base point to a new temporary base point by simulating the combined effect of moves form the previous base point. In other words, the coordinate values are changed by an amount equal to the difference between the current value of base point and the search proves base point. The new temporary base point is accepted as the new current base point if a local exploratory search proves to be successful. Thus the pattern search exploration vector moves from one successful base point to another while continuously modifying its direction and distance or movement based on information supplied by the local exploratory searches. This feature enables the routine to locate and rapidly move along sharp ridges in n-dimensional space. There by reducing computes time requirement.

Failure of a pattern move destroys the pattern based the heuristic that it would probably be better to look in a new direction. Consequently the search returns to the last successful base point and begins an exploratory search in an attempt to establish a new pattern. If this fails the step sizes are reduced and another exploratory search is conducted. This process continues until either a new pattern is established and the search moves on, or the step sizes fall below a preset minimum, and the search terminates. The detailed operation of the computer program is shown in Figure 25.

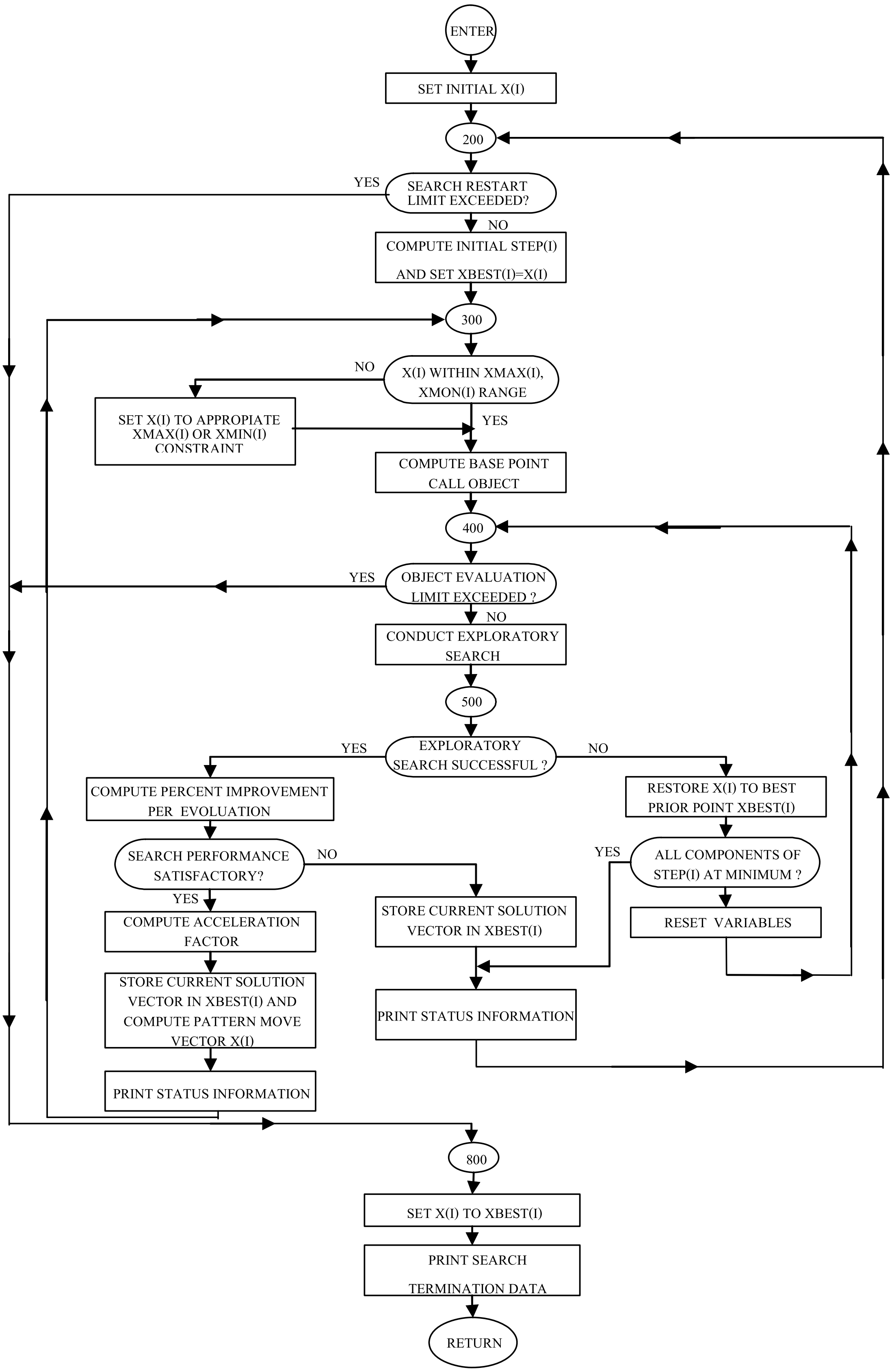


Figure 25 Detail flow chart of the computer program for the pattern search

In the optimization procedure for collector area and recycle ratio within the restrictions, the annual cost is estimated in the economic model which also calls the drying model for the amount of dried bananas produced to find the annual cost per unit of dried product. In essence, the exploratory and pattern search set step size of collector area and recycle ratio within the constrained values and search direction for optimum values and search for optimum collector area and recycle ratio for minimum cost per unit of dried product. This process is continued till the optimum values are obtained.

3.7.4 Summarized Computational Method

The solar-assisted dryer at the Royal Chitralada Projects has been continuously used since 2000. Due to the constrain of the hardware of the dryer, only two parameters namely the collector area (A_c) and the recycle factor of the drying air R_c can be adjusted to improve the thermal and economic performance of the dryer. The other parameters of the dryer are not practically adjustable because they have been fixed by the developer of the dryer.

To obtain the optimum values of the collector area (A_c) and recycle factor (R_c), the procedure can be summarized as shown in the following diagram.

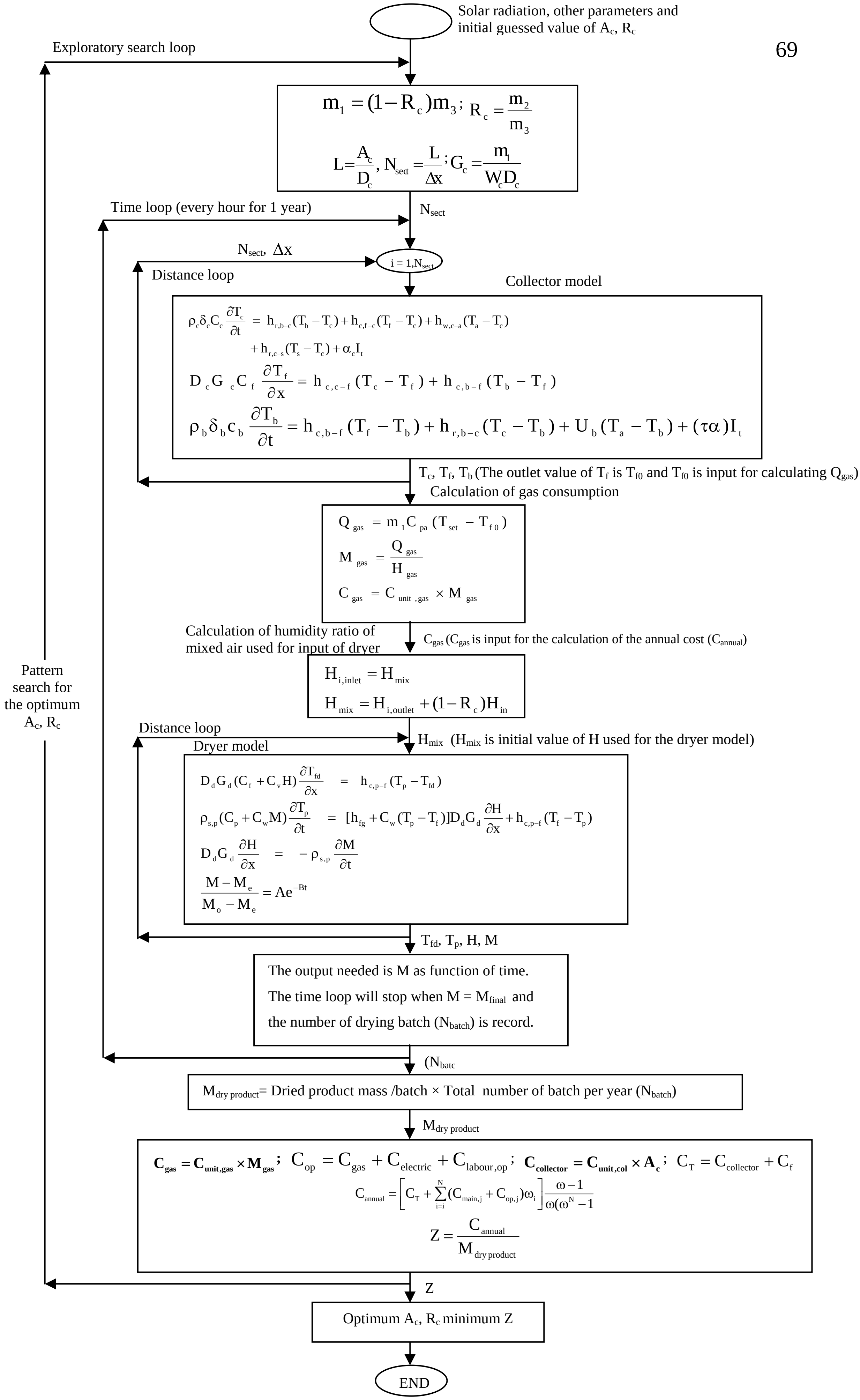


Figure 26 Diagram for the optimization process

The existing solar collector of this solar-assisted dryer has a fixed width (D_c) of 1.8 m. To increase the collector area (A_c), only the length (L) of the collector can be extended. For a given value of the collector area (A_c), the length (L) of the collector is obtained from the following equation:

$$L = \frac{A_c}{D_c} \tag{3.58}$$

where L = length of the collector

A_c = area of the collector

D_c = width of the collector

The length L is required for the numerical solution of the collector model. This is due to the fact that collector is divided into N_{sect} sections, each of which has the length of Δx .

$$N_{sect} = \frac{L}{\Delta x} \tag{3.59}$$

where N_{sect} = number of section along the length of the collector for the finite difference calculation method.

Δx = length of each section

The collector model is written as:

$$\rho_c \delta_c C_c \frac{\partial T_c}{\partial t} = h_{r,b-c}(T_b - T_c) + h_{c,f-c}(T_f - T_c) + h_{w,c-a}(T_a - T_c) + h_{r,c-s}(T_s - T_c) + \alpha_c I_t \tag{3.60}$$

$$D_c G_c C_f \frac{\partial T_f}{\partial x} = h_{c,c-f}(T_c - T_f) + h_{c,b-f}(T_b - T_f) \tag{3.61}$$

$$\rho_b \delta_b c_b \frac{\partial T_b}{\partial t} = h_{c,b-f}(T_f - T_b) + h_{r,b-c}(T_c - T_b) + U_b(T_a - T_b) + (\tau\alpha) I_t \tag{3.62}$$

Where the main parameters to be solved are as follows:

T_c = temperature of the cover

T_f = temperature of the air in the collector

T_b = temperature of the absorber

By applying the finite difference method to equation (3.60)-(3.62), the values of T_c, T_f and T_b for every section along the length of the collector are obtained. Δx and N_{sect} involves in finite difference process of the calculation. If the value Δx and N_{sect} are high, the collector length will be high, causing high outlet air temperature (T_{f0}). T_{f0} is the value of T_f at the outlet of the collector. Therefore, A_c which is affect by Δx and N_{sect} will have influence on T_{f0}.

The value of T_{f0}, in turn is used for the calculation of the energy needed by the gas burner (Q_{gas}) in order to heat up the air from the collector to the set up temperature (T_{set} = 60 °C). This energy requirement can be calculated from:

$$Q_{gas} = m_1 C_p (T_{set} - T_{f0}) \tag{3.63}$$

where m_1 = air flow rate from the collector

C_p = specific heat of air

T_{set} = set up temperature required by the drying process (T_{set} = 60 °C)

T_f = air temperature from the collector

Q_{gas} is converted into the quantity of gas (M_{gas}) by

$$M_{gas} = \frac{Q_{gas}}{H_{gas}} \tag{3.64}$$

where M_{gas} = quantity of gas

H_{gas} = heating value of gas

Q_{gas} = energy requirement from the gas burner

The value of Q_{gas} is needed for the calculation of the operating cost of the dryer ($C_{op,i}$) in the annual cost model. The annual cost model is expressed as:

$$C_{annual} = \left[C_T + \sum_{i=i}^{N} (C_{main,j} + C_{op,j})\omega_i \right] \frac{\omega - 1}{\omega(\omega^N - 1} \quad (3.65)$$

$$C_T = C_{collector} + C_f \quad (3.66)$$

$$C_{collector} = C_{unit,col} \times A_c \quad (3.67)$$

$$C_{op} = C_{gas} + C_{electric} + C_{labour,op} \quad (3.68)$$

$$C_{gas} = C_{unit,gas} \times M_{gas} \quad (3.69)$$

where C_T = total initial capital cost of the solar dryer system

$C_{main,j}$ = maintenance cost of year j

$C_{op,i}$ = operating cost of the year j

$C_{collector}$ = cost of collector

$C_{unit,col}$ = unit cost of collector

C_f = cost of the dryer

The factor $\frac{\omega - 1}{\omega(\omega^N - 1}$ is used for the annualization of all costs. The value of A_c is also involved in the calculation of the initial capital cost (C_T), as stated in equation (3.66). Therefore, A_c has influence on the annual cost (C_{annual}) of the system, which is the main parameter affecting the objective function (Z).

For the effect of recycle factor (R_c) on the objective function, it can be clarified as follows.

The balance of the air flow rate in the system can be written as:

$$m_3 = m_1 + m_2 \tag{3.70}$$

where

m_1 = air flow rate from the collector (m_1 varies with the recycle factor, R_c)

m_2 = recycle air flow rate (m_2 is adjustable)

m_3 = air flow rate in the dryer. This flow rate (m_3) is fixed to be 0.2389 m^3/s by the producer of the dryer because the power of blower for driving air in the dryer is constant)

The recycle factor (R_c) is the ratio of the recycle air flow rate (m_2) to the air flow rate in the dryer, which can be expressed as:

$$R_c = \frac{m_2}{m_3} \tag{3.71}$$

From equation (3.70) and (3.71), we can write:

$$m_3 = m_1 + R_c m_3 \tag{3.72}$$

$$m_1 = (1 - R_c) m_3 \tag{3.73}$$

As the flow rate in the dryer (m_3) is fixed from the power of the blow ($m_3 = 0.2389\, m^3$/s), the flow rate from the collector (m_1) varies with the recycle factor (R_c), as stated in equation (3.73).

From equation (3.61) of the collector model, the specific air flow rate (G_c) is related to the flow from the collector (m_1) with the following equation:

$$G_c = \frac{m_1}{D_c W_c} \tag{3.74}$$

where G_c = specific air flow rate or air flow rate per unit area perpendicular to the flow

D_c = distance between the cover and the absorber

W_c = width of the collector

Equations (3.73)-(3.74) indicate that recycle factor (R_c) has influence on the specific flow rate (G_c) of the collector. This specific flow rate (G_c) will affect the outlet air temperature of the collector (T_{f0}) through the numerical solution of equation (3.60)-(3.62). Therefore, R_c affects the annual cost which is the main parameter of the objective function (Z) described late on.

In addition to A_c, the recycle factor (R_c) will also affect the performance of the dryer as follows.

The model of the dryer is expressed as:

$$D_d G_d (C_f + C_v H)\frac{\partial T_{fd}}{\partial x} = h_{c,p-f}(T_p - T_{fd}) \tag{3.75}$$

$$\rho_{s,p}(C_p + C_w M)\frac{\partial T_p}{\partial t} = [h_{fg} + C_w (T_p - T_{fd})]D_d G_d \frac{\partial H}{\partial x} + h_{c,p-f}(T_{fd} - T_p) \tag{3.76}$$

$$D_d G_d \frac{\partial H}{\partial x} = -\rho_{sp}\frac{\partial M}{\partial t} \tag{3.77}$$

$$\frac{M - M_e}{M - ne} = Ae^{-Bt} \tag{3.78}$$

where the main parameters to be solved numerically are:

T_{fd} = temperature of air in the dryer or drying air

T_p = temperature of the product in the dryer

H = humidity ratio of drying air

M = moisture of the product

The equation (3.75)-(3.78) are solved with the finite difference method. The input of this numerical procedure is the humidity ratio of the drying air at the entrance the drying trays (H_{mix}). H_{mix} is the humidity ratio of the air which is mixed between the recycled air and the air from the collector. H_{mix} can be calculated from:

$$H_{mix} = H_m R_c + H_{in}(1 - R_c) \tag{3.79}$$

where H_m = humidity ratio at the air of the outlet of the dryer. (Part of this air will be recycled.)

H_{in} = humidity of the air flowing from the collector

The solution of equations (3.75)-(3.78) are T_{fd}, T_p, H and M as functions of x and t.

As H_{mix} is the input of equation (3.76) in the numerical process and H_{mix} depends on R_c, consequently the value of R_c has influence on the solution of equations (3.75)-(3.78) including the value of the moisture content of the product (M). If the recycle factor (R_c) is high, more moist air from the outlet of the dryer will be recycled and used again for the drying process, causing the product dried slowly. For each drying batch, the calculation of the moisture (M) will be stopped when the value of M reaches the final moisture content (M = 38%). The number of drying batch is summed up until the end of the year to obtain the total number of batch per year. This total number of batch per year will be used for the calculation of the annual production ($M_{dry\,product}$). The value of $M_{dry\,product}$, in turn will be employed for the calculation of the objective function Z as follows:

$$Z = \frac{C_{annual}}{M_{dry\,product}} \tag{3.80}$$

where C_{annual} = annual cost of dryer system

$M_{dry\,product}$ = annual production

In conclusion, both A_c and R_c have influence on both C_{annual} and $M_{dry\,product}$ through the above-mentioned models and numerical procedure.

Our task is to find the optimum values of A_c and R_c which make the drying cost (Z) minimum.

3.7.5 Organization of the computer program for optimization procedure

The computer programs written in FORTRAN were developed and the flow chart of the computer program shown in Figure 27. The steps for further elaboration are as follows:

1. The input of the programs are the initial value of the collector area ($A_{c,int}$) an air recycle factor ($R_{c,int}$) and subjected to restrictions for realistic values of drying operations (collector area: $0 < A_c < 100$ and recycle ratio: $0 < R_c < 1.0$). This program calls subroutine named SDRMIN to search for the optimum value of the collector area ($A_{c,opt}$) an air recycle factor ($R_{c,opt}$)
2. The subroutine cost is used to calculate the total cost (capital cost, maintenance cost and operating cost of the dryer), employing the cost model in section 3.3.5. In order to calculate the drying cost, this subroutine calls the subroutine DRYING to compute the annual production of dried bananas and gas and electricity consumption.
3. Subroutine DRYING is used to simulate the performance of the drying systems consisting of the dryer and the solar collector. Therefore, it calls the subroutine BACTH and subroutine COL for this purpose.
4. Subroutine BACTH computes drying air temperature, bananas temperature, humidity ratio of the air and moisture contents of bananas, using the model in section 3.3.1.2. The output of the subroutine is the annual production of dried bananas which is

used for the calculation of the drying cost in the subroutine COST. In solving the drying model equation, the subroutine MATRIX and the subroutine CHART are called.

5. Subroutine COL subroutine is employed to calculate the air temperature at the outlet of the collector. To obtain the temperature, the subroutine MATRIX is called for solving the finite difference equation representing the performance of the calculator.
6. Subroutine SDRMIN functions as a search mechanism. It tries to find the optimum value of the solar collector area ($A_{c,opt}$) an air recycle factor ($R_{c,opt}$) which made the drying cost (C_{drying}) minimum. The adaptive pattern search technique under constrained conditions is used in this subroutine. During the search operation, this subroutine calculates the drying cost for each solar collector area (A_c) an air recycle factor (R_c). Therefore, this subroutine calls the subroutine COST for this calculation.
7. If the optimum is not achieved, adaptive pattern search under constrained conditions starts to find the new values of A_c and R_c. Otherwise the iteration is stopped and the optimum values of A_c and R_c are printed out.

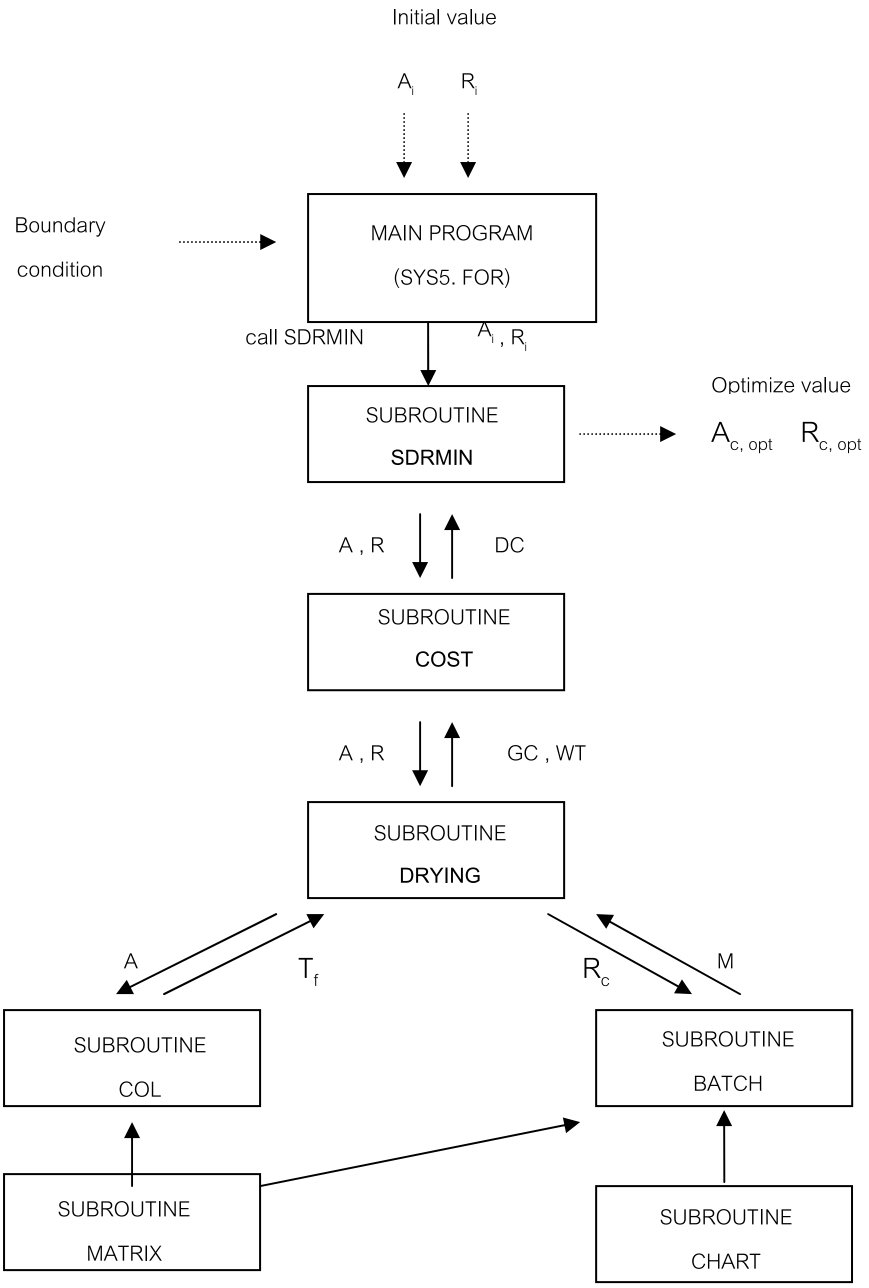


Figure 27 Descriptive flow chart of the optimization program where (A_c = Collector area of solar-assisted batch dryer, $[m^2]$, R = Recycle factor is ratio between recycle flow rate and total flow rate, [decimal], DC = Drying cost per unit of dried product, [Baht/kg], WT = Total weight of banana,[kg], TF = Temperature of drying air, [C], MC = Means moisture contents of banana in the dryer,[db], GC = gas consumption).

All the programs are in FORTRAN computer language. The source codes of this program were compiled using the COMPAQ FORTRAN compiler version 5.1 and run on PC with Microsoft Window XP. The input data required by all programs were written in text files. The output were shown on the computer screen and also recorded in the output text files.

RESULTS AND DISCUSSION

All the values of the input parameters of the simulation program corresponding to the drying system installed at the Royal Chitralada Projects were used. For the solar radiation data, we used a typical data measure at the solar monitoring if Silpakorn University, Nakhon Pathom, located at about 60 km from the Royal Chitralada Projects. This is because of the fact that these data were obtained from long-term measurement. The relative humidity and ambient air temperature measured at the same station were also used.

As initial values of the area (A_c) of the solar collector and the air recycle factor (R_c) are required by the simulation program, the values of A_c and R_c of the drying system at the Royal Chitralada Projects were first input in the program. The step of variation of R_c in the program is 1% because the control of the recycle factor (R_c) of the existing dryer can be done only the step of 1%. The variation of the solar collector area was done with the step of 0.1 m^2. This is due to the fact that the precision of the area of solar collector generally available in markets is in first decimal of a square meter. After the program received all input data, the program searched for the optimum value of the collector area (A_c) and the air recycle factor (R_c). It took about a few seconds on Pentium 4 PC computer to find the optimum value of the collector area and the air recycle factor (R_c). These optimum values used in the real drying system at the Royal Chitralada Projects are shown in Table 4.

Table 4 The optimum and the actual values of the solar collector area (A_c), the recycle factor (R_c) and the drying cost (Z).

	Optimum value	Actual value
Area of solar collector, A_c (m^2)	26.0	18
Air recycle factor, R_c (%)	90.0	95
Drying Cost, Z (USD/Kg.)	0.225	0.45

From the result in the table, it was found that the collector area of the real drying system is 18 m^2 less than the optimum value are nearly the same values.

CONCLUSION

From the design of the solar drying system in the Royal Chitralada Projects, which was not for specific product, we have optimized the solar collector area and the air recycle factor of this system for drying bananas. According to the optimization process, the thermal performance of the system components namely the solar collector and the drying unit were modified. Heat and mass balanced in the form of differential equations for each component were formulated. This system of the differential equation was solved by using the finite difference method. Values of various parameters required by the model were experimentally determined. For the sorption isotherm of bananas, it was determined by the dynamic method with the air flow through the product in relative and humidity control boxes. The thin layer drying model required by simulation model of the drying unit was also experimentally determined. As a result of this experiment, a thin layer drying model of bananas was established. Computer programs for each component of the dryer were developed and inter-connected to obtain the simulation program of the system. This simulation program was validated by comparing the simulation results with those obtained from the measurement and good agreement has been found. After the validation, this simulation program was used as a tool for the optimization process.

For the optimization process, first area of the solar collector and the air recycle factor were identified as parameter to optimize. Next, an objective function representing the drying cost was formulated. Then, the pattern search technique was adopted to search for the optimum values of the solar collector area and the recycle factor. These optimum values are resulted in the minimum drying cost. The search technique was done through the simulation model. Finally, the optimum value of the collector area of 26 m^2 and the recycle factor of 90% was obtained. As this drying system is for the demonstration of drying technology for small scale food industry, the result obtained from this work can be used as a guideline for the construction of other unit of the solar dryer. This will help to increase benefit to users. In addition the simulation program and the optimization method developed in this work can be applied to optimize a similar drying system. It is recommended that such drying system be disseminated for wide-spread used to produce dried bananas in a small industrial scale using the finding from this work as a guideline.

LITERATURE CITED

Allen, J.R. 1960. Application of grain drying theory to the drying of maize and rice. **Journal of Agricultural Engineering Research.** 5(4): 363-385.

Afzal, T.M. and T. Abe. 1999. Some fundamental attribute of the infrared radiation drying of potato. **Drying Technology.** 17(182): 137-155.

Audsley, E and J. Wheeler. 1978. The annual cost of machinery calculated actual cash flows. **Journal of Agricultural Engineering Research.** 23: 189-201.

Bala, B.K. 1983. **Deep bed drying of malt.** Ph. D. Thesis, University of Newcastle upon Tyne, UK.

Bala, B.K. 1997a. **Drying and Storage of cereal grains.** Oxford and IBH Publishing Co. PVT. LTD., New Delhi.

Bala, B.K., Esper, A. and W. Muhlbauer. 1997. Simulation of photovoltaic, forced convection, solar drying of rough rice. **Journal of Energy, Heat and Mass Transfer.** 19: 145-151.

Bala, B.K. 1998. **Solar Drying Systems: Simulation and Optimization.** Agrotech Publishing Academy, Udaipur, India.

Bala, B.K. and J.L. Woods. 1992. Thin layer drying models for malt. **Journal of Food Engineering.** 16: 239-249.

Bala, B.K. and J.L. Woods. 1994a. Simulation of the indirected natural convection solar drying system. **Solar Energy.** 53(3): 259-266.

Bala, B.K. and J.L. Woods. 1995. Optimization of the natural - convection solar drying system. **Energy.** 20(4): 285-294.

Bashir, El Fadil Adam Ahmed. 1998. Solar drying of sliced onions and quality attributes as affected by the drying process and storage condition. Dissertation, Institute for Agricultural Engineering in the Tropics und Subtropics, The University of Hohenheim, Stuttgart, Germany.

Boyce, D.S. 1966. **Heat and moisture transfer in ventilated grain.** Ph.D. Thesis, University of Newcastle Upon Tyne, UK.

Bradley, R.S. 1936. Polymolecular adsorbed films. Part 1. The adsorption of argon on salt crystals at low temperature and the determinations of surface fields. **Journal of American Chemical Society**. 58: 1467-1474.

Brunauer, S., Emmett, P. and E. Teller. 1938. Adsorption of gases in multimolecular layers. **Journal of the American Chemical Society.** 60: 309-319.

Buser, M.D., Stone, M.L., Brusewitz, G.H., Maness, N.O. and D.P., Whitlock. 1999. Thin layer drying of marigold flowers and flower components for petal removal. **Transactions of the ASAE.** 42(5): 1367-1373.

Chen, C. and D.S., Jayas. 1998. Evaluation of the GAB equation for the isotherms of agricultural products. **Transactions of the ASAE**. 41: 1755-1760.

Chen, C.C. and R.V. Morey 1989. Comparisons of four EMC/ERH equations. **Transactions of the ASAE.** 32: 983-990.

Chirfe, J. and H.A. Iglesias. 1978. Equations for fitting water sorption isotherms of foods. **A Review Journal of Food Technology.** 13: 159-174.

Chung, D. and H. Pfost. 1967. Adsorption and desorption of water vapor by cereal grains and their products. Part1: Heat and free energy changes of adsorption and desorption. **Transactions of the ASAE.** 10(3): 549.

Dandamrongrak, R., Young, G. and R. Mason. 2002. Evaluation of various pre-treatment for the dehydration of banana and selection of suitable drying models. **Journal of Food Engineering.** 55: 139-146.

Day, D. L. and G. L., Nelson. 1965. Desorption isotherms for wheat. **Transactions of the ASAE.** 8(2): 293-296.

Demirel, D and M. Turhan. 2003. Air-drying behaviour of Dwarf Cavendish and Gros Michael banana slices. **Journal of Food Engineering.** 59: 1-11.

Diamante, L.M. and P.A. Munro. 1993. Mathematical modeling of the thin layer drying of sweet potato slices. **Solar Energy.** 51(4): 271-276.

Duffie, J.A and W.A. Beckman. 1991. **Solar Engineering of Thermal Process.** John Wiley & Sons, New York.

El-Shiatry, M.A., Muller, J. and W. Muhlbauer. 1991. **Drying fruits and vegetables with solar energy in Egypt.** Agricultural Mechanization in Asia, Africa and Latin America. 22(2): 61-64.

Esper, A. 1998. **Solar energy in agriculture.** The workshop on “Solar-Technology for Everyday Use. 10-11 January, 1998 held at CIRDAP, Dhaka, Bangladesh.

Esper, A. and W. Muhlbauer. 1996. Solar tunnel dryers for fruits. **Plant Research and Development.** 44: 61-80.

FAOSTAT data, 2004. “Last updated February 2004”.

Fleming, P.D., Ekechukwu, O.V., Norton, B. and S. D. Probert. 1987. Design, installation and preliminary testing of a natural convection solar energy tropical crop drier. Solar Drying in Africa. Proceeding of the work shop held in Dakar, Sen-161.

Gal, S. 1983. The used for and practical applications of sorption data. In: Physical Properties of Foods; Eds. Jowitt, Ronald et al., Applied Science Publishers, London, pp. 425.

Guarte, R. 1996. Modeling the drying behavior copra and development of a natural convection dryer for production of high quality copra in the Philippines. Dissertation, Institute for Agricultural Engineering in the Tropics und Subtropics, The University of Hohenheim, Stuttgart, Germany.

Habchi, G. 2000. Modeling and simulation of complex production systems. Simulation: Practice and Theory. 8(2): 281-282.

Halsey, G. 1948. Physical adsorption in non-uniform surfaces. **Journal of Chemical Physics.** 16: 931-937.

Henderson, S.M. 1952. A basic concept of equilibrium moisture. **Agricultural Engineering.** 33: 29-32

Hooke, R. and T.A. Jeeves. 1961. “Direct search” Solution of numerical and statistical problem. **J. Ass. Comput. Mach.** 8: 212-229.

Hossain, Ayub.M.D. 2003. **Forced convection solar drying of chilli.** Ph. D. Thesis, Bangladesh, Agricultural University, Mymensingh, Bangladesh.

Hussain, M.A. and B.K. Bala. 2002. Thin layer drying characteristics for green chili. **Drying Technology**. 21(2): 489-505.

Hussain, M.D., Bala, B.K., Hussain, M.A. and M.R.A. Mondol. 2001. Sorption isotherms and heat of sorption of pineapple. **Journal of Food Engineering**. 48: 103-107.

Hussain, M.A., Woods, J. L. and B.K. Bala. 2005. Optimization of solar tunnel drier for drying of chilli without color loss. **Renewable Energy.** 30: 729-742.

Iglesias, H.A. and Chirife, J 1982. **Handbook of food isotherms: Water sorption parameters for food and food components.** Academic Press, New York.

Janjai, S., Esper, A. and W. Muhlbauer. 1994. A procedure for determining the optimum collector area for solar paddy drying systems. **Renewable Energy.** 4(4): 409-416.

Kachru, R.P. Ojha, T.P. and G.T. Kurup. 1991. Drying characteristics of Indian paddy varieties. **Journal of Agricultural Engineering.** 8(1): 16-23.

Kays, W.M. and M.E. Crawford. 1980. **Convection Heat and Mass Transfer.** Mc.Graw-Hill, New York.

Kuhn, I. 1967. Generalized potential theory of adsorption. **Journal of Colloid Interface Science.** 23: 563.

Lewis, W. K. 1921. The rate of drying solid materials. **Journal of Industrial Engineering and Chemistry.** 13(5): 427-432.

Lomauro, C.J., Bakshi, A.S. and T.P. Labuza. 1985. Evaluation of food moisture sorption isotherm equation. **Part 1. Fruit vegetable and meat products. Lebenmittel-Wissenchaft und Technologie.** 18: 111-117.

Luikov, A.V. 1966. **Heat and Mass Transfer in Capillary Porous Bodies.** Pergamon Press, Oxford, UK.

McAdams, W.H. 1954. **Heat Transmission** 3rd Ed. McGraw-Hill.

McEwen, E., Simmonds, W.C.H. and G.T. WARD. 1954. Drying of wheat grain part III: Interpretation in terms of biological structure. **Transactions of the Institution of Chemical Engineering.** 32: 115-120.

McLaughlin, C.P. and T.R.A. Magee. 1998. The determination of sorption isotherm and isoteric heats of sorption for potatoes. **Journal of Food Engineering.** 35: 267-280.

Nellist, M.E. and J. R. O. O'Callaghan. 1971. The measurement of drying rates in thin layer of ryegrass seed. **Journal of Agricultural Engineering Research.** 16(3): 192-212.

Oswin, C.R. 1946. The kinetics of package life III. The isotherm. **Journal of Chemical Industry**. 65: 419-42l.

Parry, J.L. 1985. Mathematical modelling and computer simulation of heat and mass transfer in agricultural grain drying: a review. **Journal of Agricultural Engineering.**

Page, G.E. 1949. **Factors influencing the maximum rate of air drying of shelled corn in thin layers.** M.S. Thesis, Purdue University, West Lafayette, Indiana.

Ploungchandang, S. and J.L. Woods. 2000. Solar drying of banana: mathematical model, laboratory simulation, and field data compared. **Journal of Food Science.** 65(6): 990-996.

Queroz, M.R. and S.A. Nebra. 2001. Theoretical and experimental analysis of the drying kinetics of bananas. **Journal of Food Engineering.** 47(2): 127-132.

Radajewski, W., Jolly, P. and G. Y. Abawi. 1987. Optimization of solar grain drying in a continuous flow dryer. **Journal of Agricultural Engineering Research**. 38: 127-144.

Radajewski, W., Jolly, P. and A.S. Wollin. 1989. Optimization of solar drying system. Proceedings of 11th International Congress on Agricultural Engineering, Dublin, 2243-2248.

Radajewski, W., Graydon, D. and E. J. McGahan. 1990. Optimization of solar hay drying in a step-flow dryer. **Transaction of the ASAE.** 33: 1423-1431.

Rahman, M.S. and T.P. Lubuza. 1999. **Water Activity of food preservation.** In: Hand book of Food Preservation. Rahman, M.S. (Ed.), Marcel Dekka Inc., NewYork., 339-382.

Schirmer, P., Janjai, S., Esper, A., Smitabhindu, R. and W. Muhlbauer. 1996. Experimental investigation of the performance of the solar dryer for drying banana. **Renewable Energy.** 7(2): 199-129.

Sharaf- Eldeen, Y.I., Hamdy, M. Y. and J. L. Blaisdell. 1979. Mathematical simulation of drying of fully- exposed ear corn and its components. **Transactions of the ASAE.** 79: 6523.

Shei, H.J. and K.L. Chen. 1998. Intermittent drying of rough rice. **Drying Technology.** 16(3-5): 839-851.

Shivhare, W.S., Arora, S., Ahmed, J. and G. V. S. Raghavan. 2004. Moisture adsorption isotherms for mushroom. **Lebensmittel-Wissenschaft und Technology.** 37: 133-137.

Simate, I.N. 2003. Optimization of mixed –mode and indirect mode natural convection dryers. **Renewable Energy.** 28: 435-453.

Smith, S.E. 1947. Sorption of water vapor by proteins at high polymers. **Journal of the American Chemical Society.** 69: 646-651.

Sodha, M.S., Dang, A., Bansal P.K. and S.B. Sharma. 1985. An analytical and experimental study of open sun drying and a cabinet type drier. **Solar Energy.** 63(1): 263-271.

Soysal, Y and S. Oztekin. 1999. Moisture content equations for some medicine and aromatic plants. **Journal of Agricultural Engineering Research.** 74: 317-324.

Steinfeld, A. and I. Segal. 1986. Simulation model of solar thin layer drying process. **Drying Technology.** 4(4): 535-554.

Sun, Da-Wen. 1998. Selection of EMC/ERH isotherm equations for drying and storage. Proceedings of CIGR 13th International Congress on Agricultural engineering Rue Haroun Errachid, Rabat, Morocco. (vol. 6): 331-336.

Sun, Da-Wen and J.L. Woods. 1994. Low temperature characteristics of wheat in thin layers. **Transactions of the ASAE.** 57(6): 1919-1926.

Taubert, W.H. 1968. A search decision rule for the aggregate scheduling program. **Management Science.** 14B: 343-359.

Watson, E.L. and V.K. Bhargava. 1974. Thin layer drying studies on wheat. Canadian **Journal of Agricultural Engineering.** 16(1): 18-22.

Van den Berg, C. 1984. Description of water activity of food for engineering purposes by means of the GAB Model of sorption. In: Engineering and Food vol. 1, B.M. Mckenne (Ed.). Elsevier Applied Science, London, UK. 311-321.

Yaldyz, Osman and Extekyn, Can. 2001. Thin layer solar drying of some uegetasses. **Drying Technology.** 19(384): 583-597

Zaman, M.A. and B.K. Bala. 1989. Thin layer solar drying of rough rice. **Solar Energy.** 42(2): 167-171.

Appendix A

Flow chart of the programs

The programs used to calculate optimum values of A_c and R_c and the minimum value of the objective function Z are as follows :

1. Main program

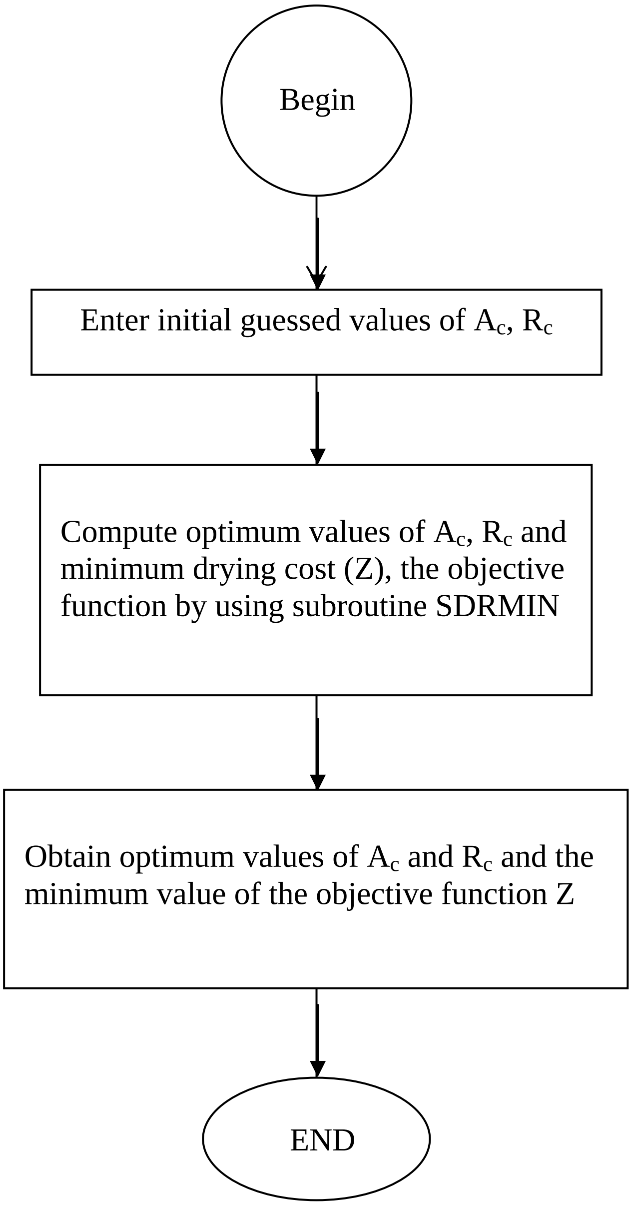


Figure A1.1 Flow chart of the main program

2. Subroutine SDRMIN is employed for the optimization process using the exploratory search method proposed by Taubert (1968)

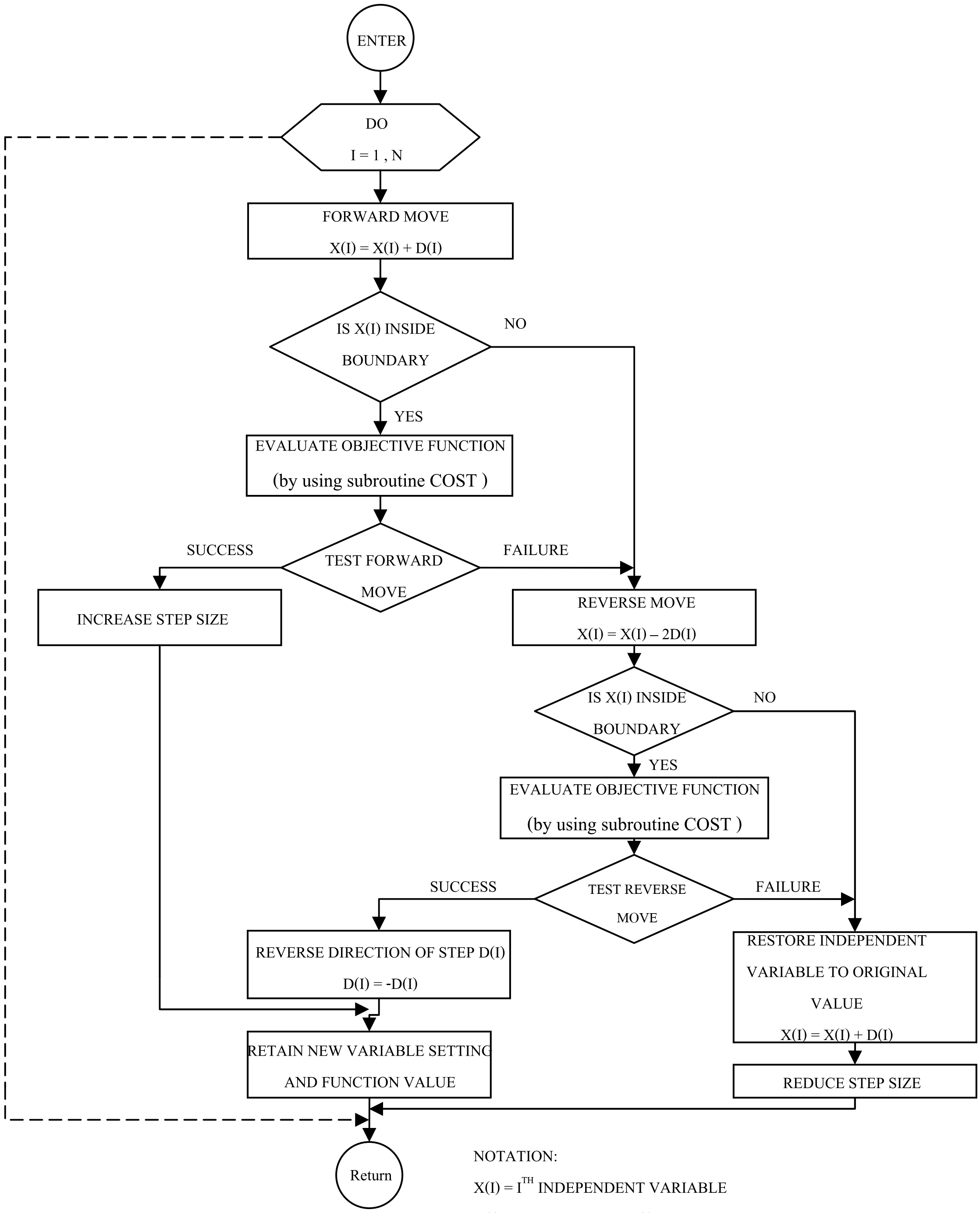


Figure A1.2 Flow chart of subroutine SDRMIN

3. Subroutine COST is used for computing the objective function, Z (drying cost)

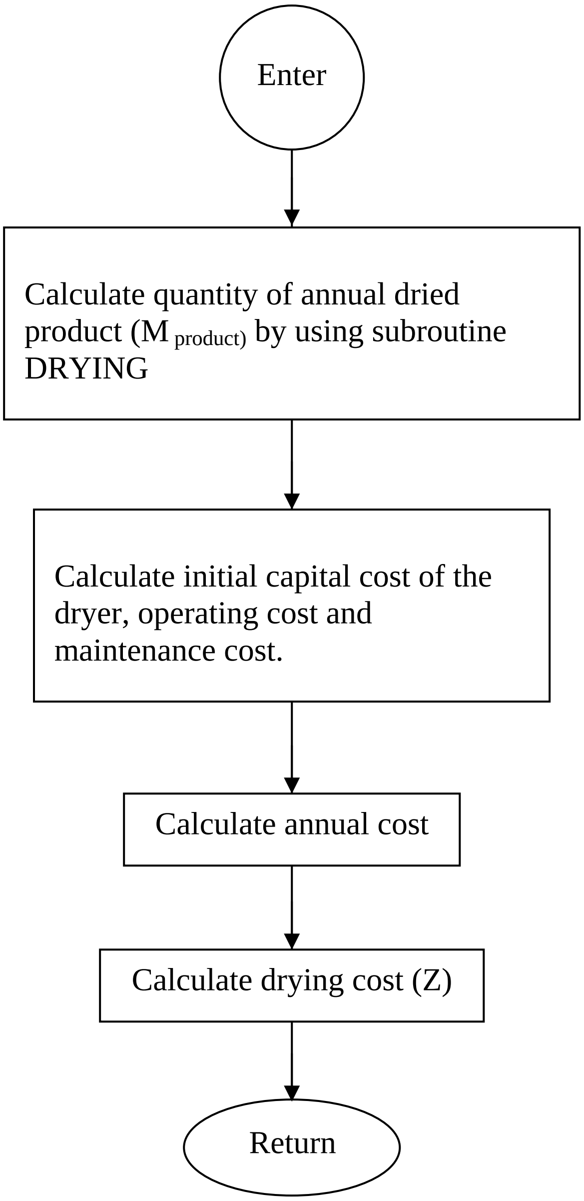


Figure A1.3 Flow chart of subroutine COST

4. Subroutine DRYING is for the calculation of the quantity of annual dried product and gas consumption of the dryer.

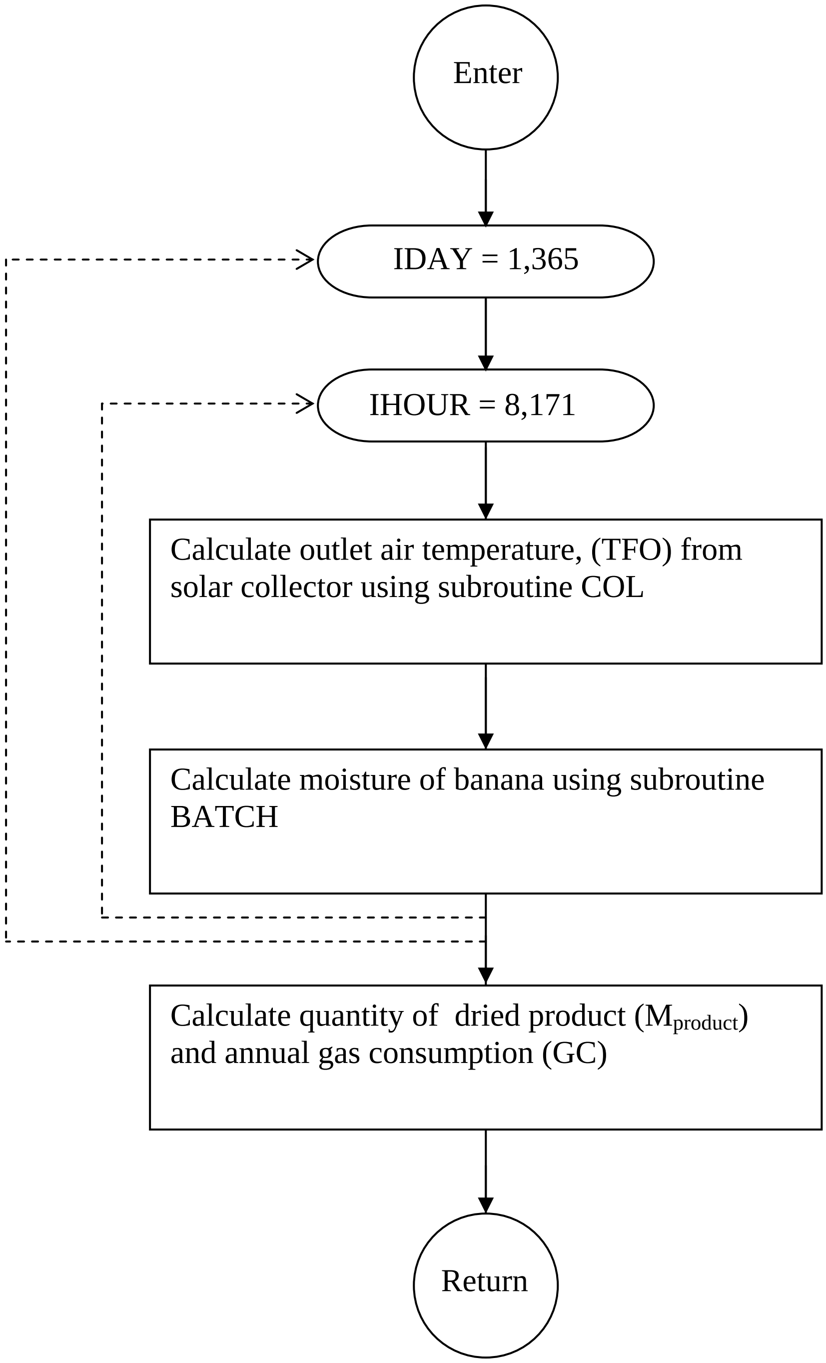


Figure A1.4 Flow chart of subroutine DRYING

5. Subroutine COL is employed to compute outlet air temperature (TFO) of the collector using the collector model developed in this work.

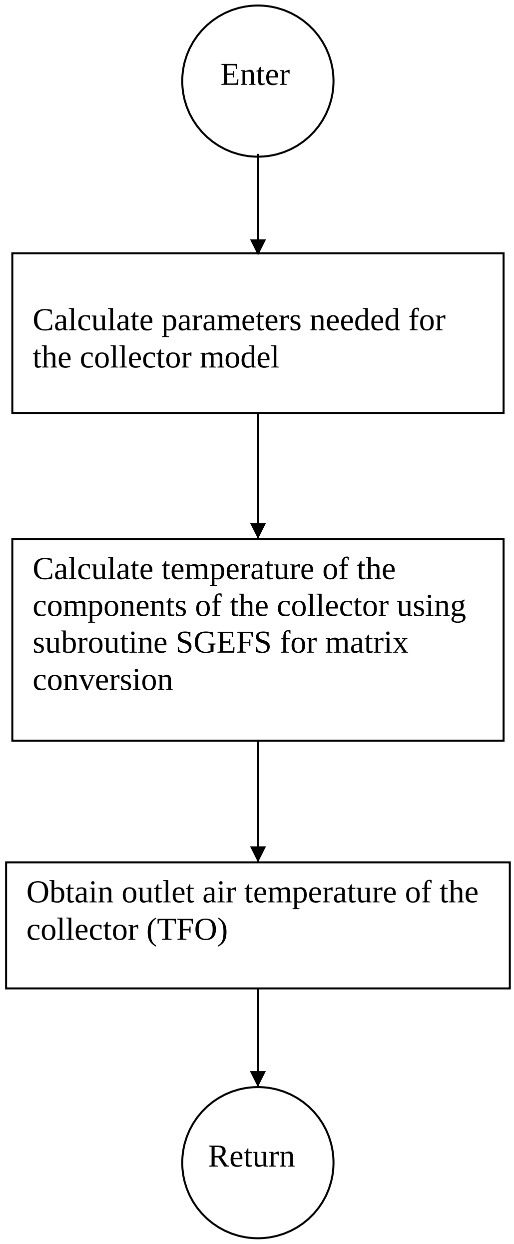


Figure A1.5 Flow chart of subroutine COL

6. Subroutine SGEFS for solving the matrix equation of the collector model based on the method presented in Kahaner et al (1988)

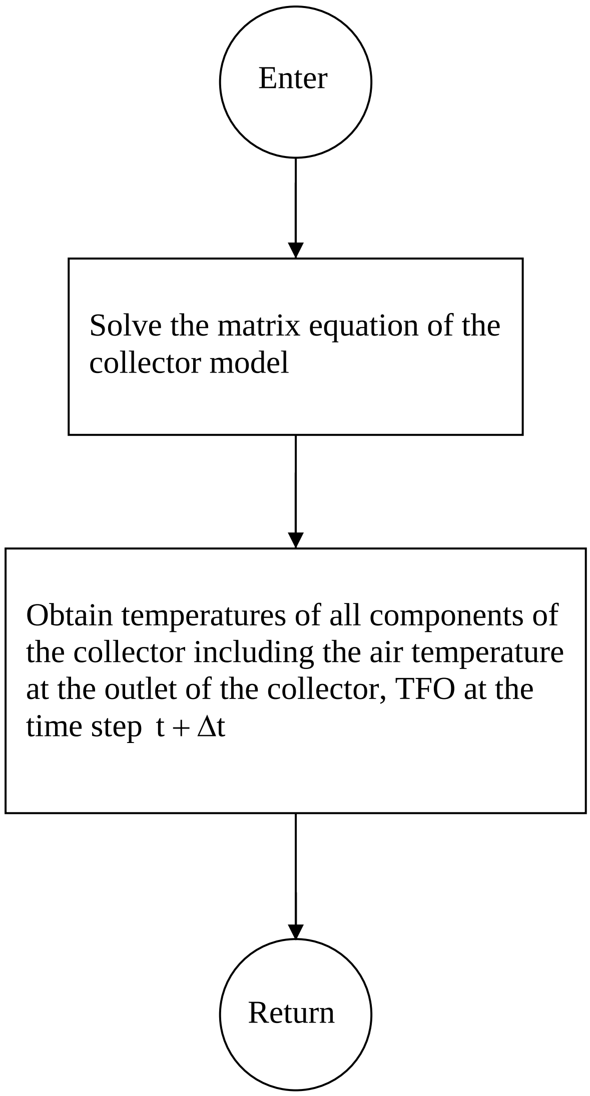


Figure A1.6 Flow chart of subroutine SGEFS

7. Subroutine BATCH is for the computation of moisture of the product and air temperature in the dryer based on the dryer model developed in this work.

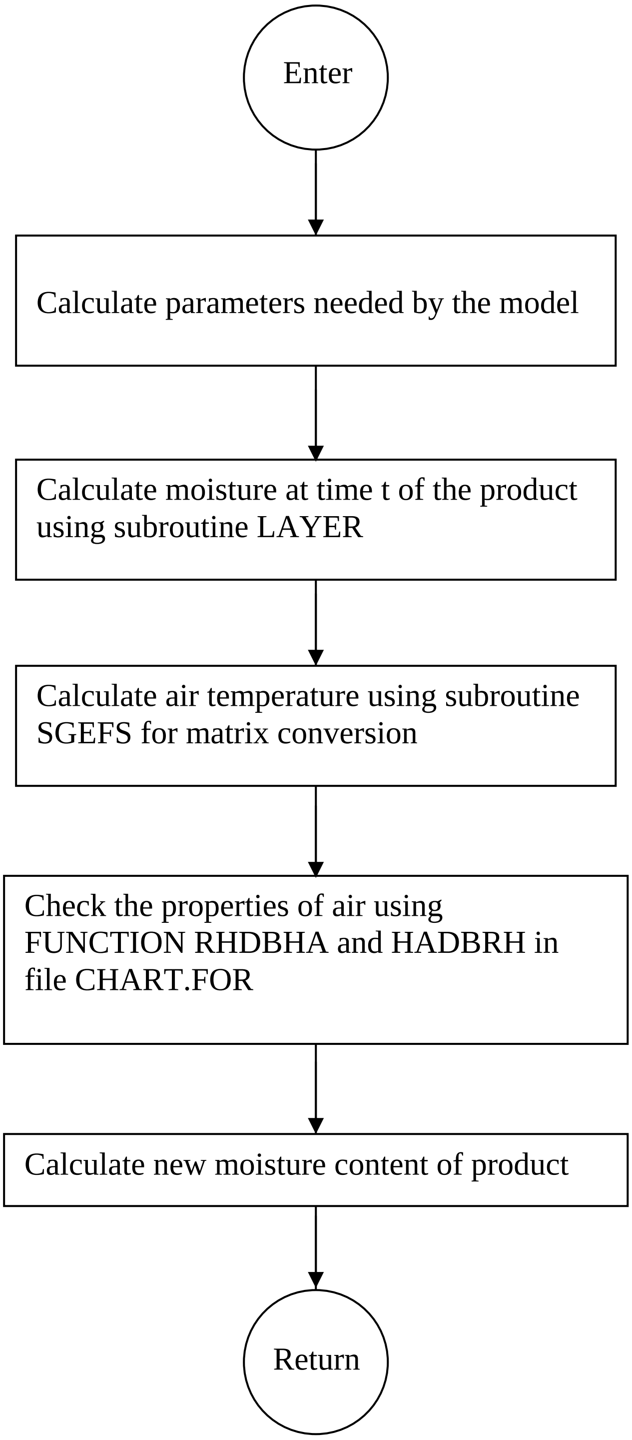


Figure A1.7 Flow chart of subroutine BATCH

8. Subroutine LAYER is used for computing thin layer drying equation of the dryer model. The thin layer drying equation was developed in this work by carrying out drying experiments.

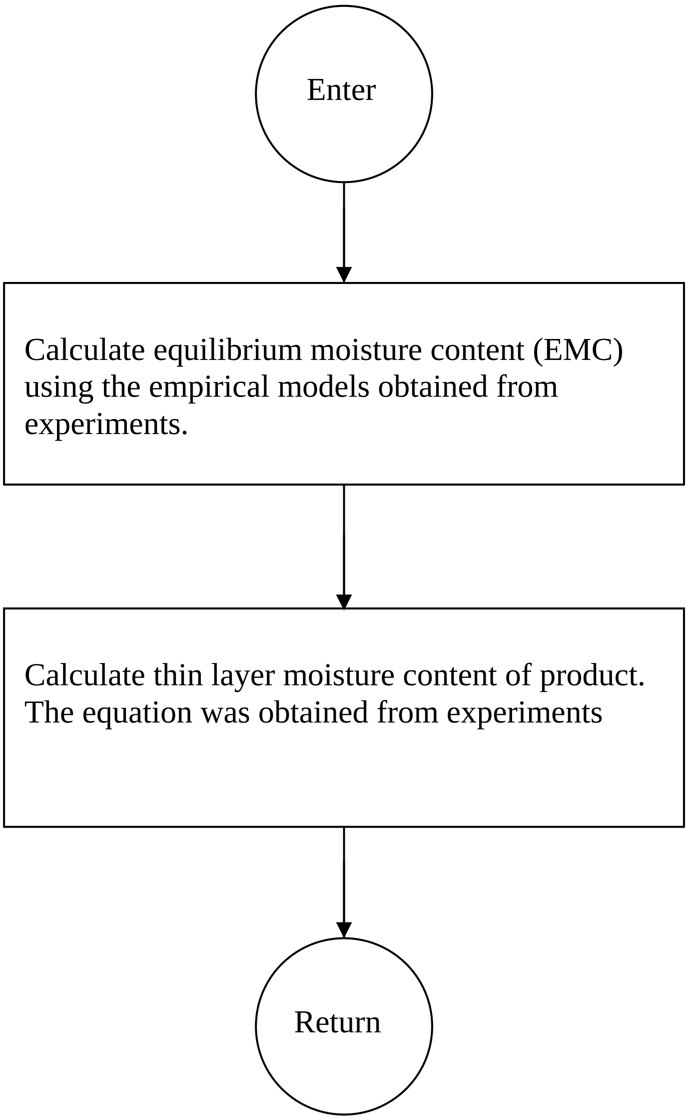


Figure A1.8 Flow chart of subroutine LAYER

9. Subfunctions in the file CHART.FOR are used to compute properties of moist air in the subroutine BATCH. They are composed of the following subfunctions :

- FUNCTION RHDBHA (DB, HA)
- FUNCTION PVHA (HA)
- FUNCTION PSDB (DB)
- FUNCTION HADBRH (DB, RH)
- FUNCTION WBDBHA (DB, HA, G_1, G_2, EPS)
- FUNCTION WBL (TWB)
- FUNCTION HLDB (DB)

Appendix B

Listing of the programs

1. Main program

```
C      File name : SYS5.FOR
C      Version :  22 April 05
C      Purpose : To optimize the solar-assisted batch dryer.
C      Note : This program (Sys5) is used to define a value of initial value, upper
C                 and lower boundary of optimizing parameter. And transfer these value to
C                 optimizing program (SDRMIN)
C      Subroutines used are written in the following files :
C                               1) COL.FOR
C                               2) BATCH.FOR
C                               3) COST.FOR
C                               4) DRYING.FOR
C                               5) OPTIMUM.FOR
C     This program uses simulation models of the solar collector and dryer.
C      Weather data used in this program is TRY data.The
C      1st column is hourly global radiation and 2nd column is ambient temperature.
C     A subroutine used for optimization is SDRMIN.This subroutine was written, based on the
C      exploratory search method proposed by Taubert (1968)
C
        PROGRAM MAIN
        PARAMETER (LDA=180)
        COMMON /TEMP/ T
        COMMON X(100),XMAX(100),XMIN(100),N,MAXTRY,AWR,ADR
        DIMENSION T(LDA),TIM(300),IT(8760*6),TA(8760*6),TF(8760*6),
        1                  HT(365,24),TDB(365,24)
```

```
      REAL IT,LENGTH,XLOSS,LOAD,XLOAD,l,DC
      INTEGER NHOUR
       DATA CPA/1002./
C
C     A........Collector area of solar-assisted batch dryer,  [m2]
C     R........Recycle factor is ratio between recycle flow rate and total flow rate, [decimal]
C     X........A value of optimizing parameter
C     XMAX.....Maximum value of optimizing parameter
C     XMIN.....Minimum value of optimizing parameter
C     N........Number of optimizing parameter
C
      WRITE(*,*)'Enter a value of collector area and recycle factor'
      READ(*,*) A,R
      X(1) = A
      X(2) = R
      XMAX(1) = 100.0
      XMIN(1) = 0.0
      XMAX(2) = 1.0
      XMIN(2) = 0.0
      N = 2
      MAXTRY = 10000

      CALL SDRMIN(X(1),X(2),XMAX(1),XMAX(2),XMIN(1),XMIN(2),MAXTRY,N,
      &     TRIALV)

      STOP
      END
```

2. Subroutine SDRMIN

```
      SUBROUTINE SDRMIN(XINI,YINI,XUPP,YUPP,XLOW,YLOW,MAXTRY,N,TRIALV)
C
C     Note : This program is used for the optimization based on the exploratory search method
          proposed by
C      The value of STPSET and EPSV is very important.
C
C     COMMON TM(50),SR (50),TB(50),TW(50)
C     COMMON X(10),XMAX(10),XMIN(10),N,MAXTRY,AWR,ADR
C     BESTV.....Final value of objective function
C     FIRSTV..  Initial value of objective function
C     NEVAL.....Number of times for evaluate
C     GOODV.....Good value of objective function
C     STEP......Size of moving step in searching
C     TRIALV....Trial value of objective function
C     X.........A value of optimizing parameter
C     XMAX......Maximum value of optimizing parameter
C     XMIN......Minimum value of optimizing parameter
C
      DIMENSION X(100),XMAX(100),XMIN(100)
      DIMENSION STEP(100),XBEST(100)
      OPEN(2,FILE='OPTIMIZE1.DAT',STATUS='OLD')
C
C     INITIAL CONTROL PARAMETERS
C
```

```
      X(1) = XINI
      XMAX(1) = XUPP
      XMIN(1) = XLOW
      X(2) = YINI
      XMAX(2) = YUPP
      XMIN(2) = YLOW
      MAXABT=100
      PCTHI=0.5
      PCTMED=0.25
      PCTMIN=1.0E-2
      STPDEC=0.5
c     STPINC=2.0
      STPINC=1.5
      STPMIN=1.0E-6
      STPSET=0.1
      EPSV=0.000001
C     INITIALIZE COUNTERS
      IPAGE=1
      LINE=100
      NDEC=0
      NEVAL=0
      NFWDF=0
      NFWDO=0
      NFWDS=0
      NMIN=0
      NPAT=0
      NPRIOR=0
      NPROG=0
      NREST=0
```

```
      NREVF=0
      NREVO=0
      NREVS=0
C     COMPUTE PATTERN MOVE PRINT FREQUENCY
      IPRINT=10/N
      IF(IPRINT.LT.1)IPRINT=1
C     TERMINATE THE SEARCH IF MAXABT(SEARCH RESART LIMIT) IS EXCEEDED
200   NABORT=NMIN+NPROG
200   NABORT=NMIN+NPROG
      IF(NABORT.GT.MAXABT) GO TO 800
C     RESET EXP SEARCH SUCCESS STATUS FLAG+PATTERN MOVE ACCELERATOR
      IFLAG=0
      ACCEL=2.2
C     COMPUTE INITIAL STEP SIZE VECTOR STEP (I)+SAVE TRIAL VECTORX(I)
      DO 250 I=1,N
      STEP (I)=STPSET*(XMAX(I)-XMIN(I))
250   XBEST(I)=X(I)
C     TEST TO INSURE ALL SOLUTION VECTOR COMPONENTS ARE WITHIN BOUNDS
C     IF OUTSIDE RESET TO XMAX(I) OR XMIN(I) BOUNDARY
300   DO 350 I=1,N
      IF(X(I).GT.XMAX(I))X(I)=XMAX(I)
      IF(X(I).LT.XMIN(I))X(I)=XMIN(I)
350   CONTINUE
C     EVALUATE OBJECTIVE FUNCTION+INCREMENT EVALUATION COUNTER
      CALL COST(X(1),X(2),TRIALV)
      NEVAL=NEVAL+1
C     RECORD INITIAL SOLUTION VALUE FOR COMPARISON DURING EXP SEARCH
      GOODV=TRIALV
C     IF INITIAL PASS OF SEARCH INITIALIZE+PRINT INITIAL CONDITIONS
```

```
      IF(NEVAL.GT.1) GO TO 400
      PCTDEC=0.0
C     INITIALIZE WITH INITIAL SOLUTION VALUE(TRIALV)
      BESTV=TRIALV
      FIRSTV=TRIALV
      PRIORV=TRIALV
C     PRINT INITIAL CONDITIONS
      WRITE(6,930) IPAGE
      WRITE(6,932)
      WRITE(6,934)N
      WRITE(6,936)MAXTRY
      WRITE(6,938)MAXABT
      WRITE(6,940)STPSET
      WRITE(6,942)STPINC
      WRITE(6,944)STPDEC
      WRITE(6,946)STPMIN
      WRITE(6,948)PCTMIN
      WRITE(6,950)PCTMED
      WRITE(6,952)PCTHI
      WRITE(6,954)ACCEL
      WRITE(6,960)(X(I),I=1,N)
      WRITE(6,962)(XMAX(I),I=1,N)
      WRITE(6,964)(XMIN(I),I=1,N)
      WRITE(6,966)(STEP(I),I=1,N)
C     PRINT NEW PAGE HEADING,INITIAL SEARCH VALUE+RETURN TO STMT 400
      JPRINT=1
      GO TO 900
C     TERMINATE SEARCH IF MAXTRY IS EXCEEDED
400   IF(NEVAL.GT.MAXTRY) GO TO 800
```

```
C       INITIALIZE MINIMUM STEP SIZE COUNTER
        NSTEP=0
C

C       START EXPLORATORY SEARCH SELECTION
C
        DO 500 I=1,N
C       IF STEP SIZE IS ZERO OMIT EXP SEARCH FOR THAT COMPONENT
        IF (STEP(I).EQ.0) GO TO 490
C       FORWARD MOVE LOGIC
        X(I)=X(I)+STEP(I)
C       IF OUTSIDE BOUNDS TRY REVERSE MOVE
        IF(X(I).GT.XMAX(I).OR.X(I).LT.XMIN(I)) GO TO 410
        CALL COST(X(1),X(2),TRIALV)
        NEVAL=NEVAL+1
C       IF NEW POINT NOT BETTER TRY REVERSE MOVE
        IF(TRIALV.GE.GOODV) GO TO 420
        GOODV=TRIALV
        NFWDS=NFWDS+1
405     IF(IFLAG.GT.0)STEP(I)=STPINC*STEP(I)
        GO TO 500
C       REVERSE MOVE LOGIC
410     NFWDO=NFWDO+1
        GO TO 430
420     NFWDF=NFWDF+1
C430    X(I)=2.0*STEP(I)
430     X(I)=X(I)-2.0*STEP(I)
C       IF OUTSIDE BOUNDS RESTORE X(I)+REDUCE STEP SIZE
        IF (X(I).GT.XMAX(I).OR.X(I).LT.XMIN(I)) GO TO 440
```

```
      CALL COST(X(1),X(2),TRIALV)
      NEVAL=NEVAL+1
C     IF NEW POINT IS NOT BETTER RESTORE X(I)+REDUCE STEP SIZE
      IF (TRIALV.GT.GOODV) GO TO 450
      GOODV=TRIALV
      STEP(I)=-STEP(I)
      NREVS=NREVS+1
      GO TO 500
C     RESTORE X(I)=REDUCE STEP LOGIC
440   NREVO=NREVO+1
      GO TO 460
450   NREVF=NREVF+1
460   X(I)=X(I)+STEP(I)
      STEP(I)=STPDEC*STEP(I)
C     TEST STEP SIZE-RESTORE TO MIN VALUE IF BELOW MINIMUM
      IF(ABS(STEP(I)).GE.STPMIN) GO TO 500
      IF(STEP(I).GE.0.0) STEP (I)=STPMIN
      IF(STEP(I).LT.0.0) STEP (I)=-STPMIN
490   NREST=NREST+1
      NSTEP=NSTEP+1
500   CONTINUE
      IF(ABS((BESTV-GOODV)/GOODV).GT.EPSV)GO TO 560
      BESTV=GOODV
      DO 550 I=1,N
550   XBEST(I)=X(I)
      GO TO 800
560   CONTINUE
C     TEST FOR SUCCESS OR FAILURE OF EXPLORATORY SEARCH
      IF (GOODV.LT.BESTV) GO TO 700
```

```
C       EXPLORATORY SEARCH IS A FAILURE
        DO 650 I=1, N
650     X(I)=XBEST(I)
        IF(NSTEP.LT.N) GO TO 660
C       TEST FOR NEW PAGE, PRINT SEARCH DATA+GOTO STMT 200 TO RESTART
        NMIN=NMIN+1
        JPRINT=2
        GO TO 900
660     GOODV=BESTV
        IFLAG=0
        NDEC=NDEC+1
        GO TO 400
C       EXPLORATORY SEARCH IS SUCCESS
700     BESTV=GOODV
        XX=NEVAL-NPRIOR
        PCTDEC=ABS(100.0*(PRIORV-GOODV)/(PRIORV*XX))
        PRIORV=GOODV
        NPRIOR=NEVAL
C       TEST FOR SATISFACTORY PROGRESS
        IF(PCTDEC.GE.PCTMIN) GO TO 720
        NPROG=NPROG+1
        DO 710 I=1,N
710     XBEST(I)=X(I)
C        TEST FOR NEW PAGE, PRINT SEARCH DATA+GO TO STMT 200TO RESTART
        JPRINT=2
        GO TO 900
720     IF(PCTDEC.LT. PCTMED) ACCEL=ACCEL+0.10
        IF(PCTDEC.LT. PCTHI) ACCEL=ACCEL+0.05
        IF (ACCEL. GT. 3.5) ACCEL=2.2
```

```
        DO 750 I=1, N
        XX=XBEST(I)
        XBEST(I)=X(I)
750     X(I)=XX+ACCEL*(X(I)-XX)
        IFLAG=1
        NPAT=NPAT+1
        JPRINT=3
        IF(MOD(NPAT, IPRINT))300,900,300
900     IF(LINE.LE.40) GOTO 901
        IF(JPRINT.NE.1)WRITE(6,990)
C       WRITE NEW PAGE HEADING
        IPAGE=IPAGE+1
        LINE=0
        WRITE(*,930)IPAGE
        WRITE(*,990)
        WRITE(*,991)
        WRITE(*,990)
        WRITE(*,992)
        WRITE(*,993)
        WRITE(*,990)
C       WRITE LINE STATUS OF INFORMATION
901     WRITE(*,995)NEVAL, BESTV, NPAT,NDEC, NMIN,NPROG,PCTDEC,NFWDO,
        NFWDS,
        1NFWDF,NREVO,NREVS,NREVF,NREST
        LINE=LINE+1
        GO TO(400,200,300),JPRINT
C       SEARCH TERMINATION -REST X(I)+PRINT TERMINATION DATA
800     DO 810 I=1,N
810     X(I)=XBEST(I)
```

```
      WRITE(*,995)NEVAL,BESTV,NPAT,NDEC,NMIN,NPROG,PCTDEC,NFWDO,
      NFWDS,
      1NFWDF,NREVO,NREVS,NREVF,NREST
      WRITE(*,990)
      IPAGE=IPAGE+1
      WRITE(*,930)IPAGE
      WRITE(*,970)
      WRITE(*,972)NEVAL
      WRITE(*,974)FIRSTV
      WRITE(*,976)BESTV
      XX=NEVAL
      PCTDEC=ABS(100.0*(FIRSTV-BESTV)/FIRSTV)
      WRITE(*,977)PCTDEC
      PCTDEC=PCTDEC/XX
      WRITE(*,978)PCTDEC
      WRITE(*,980) (XBEST(I),I=1,N)
      WRITE(*,982) (STEP(I),I=1,N)
930   FORMAT(1H1,34X,'* * * A D A P T I V  P A T T E R N  S E A R
      1C * * * ',27X,'PAGE',I3 //    )
932   FORMAT('I N I T I A L  C O N D I T I O N S')
934   FORMAT(5X,'NO OF INDEPENDENT VARIABLES',47X,I10)
936   FORMAT(5X,'MAX NO OF OBJECTIVE FUNTION EVALUATIONS',34X,I10)
938   FORMAT(5X,'MAX NO OF RESARTS AFTER STEPSIZE OR UNSAT PROGRESS AB
      1ORT',16X,I10)
940   FORMAT(5X,'STEP SIZE MULTIPLIER-INITIAL START',38X,E10.3)
942   FORMAT(5X,'STEP SIZE MULTIPLIER-FWD MOVE SUCCESS FOLLOWING EXP
      1SEARCH SUCCESS',6X,E10.3)
944   FORMAT(5X,'STEP SIZE MULTIPLIER-FORWARD+REVERSE MOVE FAILUR',21X,
      1E10.3)
```

```
946   FORMAT(5X,'MIN STEP SIZE FOR ALL INDEPENDENT VARIABLES',31X,E10.3)
948   FORMAT(5X,'UNSAT PROGRESS ABORT(MIN PERCENT IMPROVEMENT PER
      EVALL
      1ATION)',13X,E10.3)
950   FORMAT(5X,'PATTERN MOVE ACCELERATOR THRESHOLD-LOW/MED PCT PER
      EV
      1ALUATION',11X,E10.3)
952   FORMAT(5X,'PATTERN MOVE ACCELERATOR THRESHOLD-MED/HI PCT PER EV
      1ALUATION',11X,E10.3)
954   FORMAT(5X,'INITIAL PATTERN MOVE ACCELWRATOR',42X,E10.3)
960   FORMAT(   ///   ,'INITIAL SOLUTION VECTOR - X(I)',/,(10F12.4))
962   FORMAT(   ///   ,'UPPER RANGE CONSTRAINT VECTOR - XMAX(I)',/,
      1F12.4)
964   FORMAT(   ///   ,'LOWER RANGE CONSTRAINT VECTOR - XMIN(I)',/,
      1F12.4)
966   FORMAT(   ///   ,'INITIAL STEP SIZE VECTOR - STEP(I)',/,(10F12.4))
970   FORMAT('S E A R C H  T E R M I N A T I O N  R E S U L T S ')
972   FORMAT(5X,'NO OF OBJECTIVE FUNCTION EVALUATIONS',8X,I12)
974   FORMAT(5X,'INITIAL VALUE OF OBJECTIVE FUNCTION',9X,E12.5)
976   FORMAT(5X,'FINAL VALUE OF OBJECTIVE FUNCTION',9X,E12.5)
977   FORMAT(5X,'OVERALL PERCENT DECREASE',20X,E12.5)
978   FORMAT(5X,'OVERALL PERCENT DECREASE PER EVALUATION',5X,E12.5)
980   FORMAT( ///   ,'FINAL SOLUTION VECTOR-XBEST(I) AND X(I)',
      1/,(10F12.4))
982   FORMAT( /// ,'FINAL STEP SIZE VECTOR-STEP(I)',/,(10F12.4))
990   FORMAT(1X,129(1H-))
991   FORMAT(':S D R S U M M A R Y :E X P L O R A T O R Y  S E A R C
      1H D A T A : F O R W A R D M O V E : R E V E R S E M O V E :RE
      2STORE:')
```

```
992    FORMAT(': OBJ FCT OBJ FCT :PATTERN DEC STEP MIN STEP ON PROG
       1PCT    DEC     :OUTSIDE TRIAL TRIAL :OUTSIDE TRIAL TRIAL :ST
       2EP AT:')
993    FORMAT(':EVAL EVAL :MOVE SEARCH ABORT ABORT
       1/EVAL :RANGE SUCCESS FAILURE: RANGE SUCCESS FAILURE:MI
       2NIMUM')
995    FORMAT(' : ',I5,E15.5,2H :,I5,3I9,E14.5,2H :,I6,2I8,3H :,I6,2I8,
       1 3H  :,I5,' :')
       CLOSE(2)
       RETURN
       END
```

3. subroutine COST

```
C	234567  Solar Air Collector, file name : COST.FOR
C	This program was written for optimization of a solar-assisted batch dryer
C	located at Royal Chitlada palace. The programme is developed based on the
C	economic model as Prof.Bala 's
C	Version : 20 December 2004
C	 Importants parameters used in the main programme :
C	CANNUAL..Annual cost for investment of the dryer per year, [THB]
C	CAU......Cost of solar collector per unit area, [THB]
C	CCOLLEC..Cost of solar collector, [THB]
C	CCAPITAL.Capital cost of solar assisted batch dryer, [THB]
C	CDRYER...Cost of all of batch dryer part, [THB]
C	CELEC....Cost of electricity consumption, [THB]
C	CGAS.....Cost of gas consumption, [THB]
C	CLABOR...Cost of construction and installation, [THB]
C	CMAN.....Cost of man hour to operated the dryer, [THB]
C	CMAIN....Coat for maintenance of the dryer, [THB]
C	CMATER...Cost of material for construction the dryer, [THB]
C	COPER....Cost of present value of the dryer, [THB]
C	CTOTAL...Total cost for investment of the dryer, [THB]
C	DC...... Drying cost per unit of dried product, [THB/kg]
C	EC.......Electricity consumption, [unit]
C	GC.......Gas consumption, [kg]
C	INFLA....Inflation rate, [decimal]
C	INTER....Interest rate, [decimal]
```

```
C       N........Number of operation year of dryer, [year]
C       WT.......Total weight of dried banana per year, [kg] C       AC......Annual cost for invesment
        of the dryer per year, [baht]

        SUBROUTINE COST(A,R,DC)
        REAL W,INFLA,INTER
        CALL DRYING(GC,WT,A,R)
        N = 10
        CD = 66636.00
        CAU = 3468.52
        CDRYER = CD
        CCOLLE = CAU*A
        CMATER = CDRYER+CCOLLE
        CLABOR = 0.1*CMATER
        CTOTAL = CMATER+CLABOR
        CMAINTAIN = 0.01*CTOTAL
        CMAN = 2.*240.*120.
        CGAS = 16.0*GC
        EC = 100.
        CELEC = 3.0*EC
        COPERATE = CMAN + CGAS + CELEC
C
C       Initialization
C
        COPER = 0.0
        CMAIN = 0.0
C
C       Begin to calculate present value of solar tunnel dryer
C
```

```
      INFLA = 0.05
      INTER = 0.07
      W = (INFLA+1.)/(INTER+1.)
      DO I = 1,10
      COPER = COPER + COPERATE*W**I
      CMAIN = CMAIN + CMAINTAIN*W**I
      CPRESENT=(CTOTAL+COPER+CMAIN)
      CTOTAL = 0.0
      END DO
C
C     Calculate annual cost
C
      CANNUAL = CPRESENT*(W-1.)/(W*((W**N)-1.))
C
C     Finally, calculate the drying cost per unit,DC, of dried product,WT
C
      DC = CANNUAL/WT
      WRITE(2,*) A,R,DC
      END SUBROUTINE
```

4. Subroutine DRYING

```
      SUBROUTINE  DRYING(GC,WT,A,R)
C
C     Version :  22 April 05
C     Purpose : To calculate annual product of dried bananas (WT) form the solar-assisted batch
C               dryer and gas consumption of the dryer (GC)
C      Note : In this system, the heater is controlled by temperature in the dryer
C                         If TDO > TUPP then gas burner is off
C                         Else if TDO > TLOW then gas burner is off
C                         Else gas burner is on
C      Apparatus :
C
C    <--------------------------Solar collector------------------------>
C    ------------------------------------------------------------------
C    |              flow/2 ->          O          <- flow/2          |
C    ------------------------------------------------------------------
C                             c   flow   c
C                               c         c
C                                 c         c
C                                   c        c
C                                   c  Heater c
C                             outletC                  c
C                   cccccccccccccccccC  cC    |         c
C                   c  ============           V         c
C                   c  ============= c   TUPP=60   c
C                   c  ============= c  Mixing box   c
C                   c  ============= c   TLOW=55   c
C                   c  ccccccccccccccc                     c
```

```
C                    cccccccccccccccccccccccccccccccccc
C                    <--Batch dryer--->
C

      PARAMETER (LDA=180)
      COMMON /TEMP/ T
      COMMON X(10),XMAX(10),XMIN(10),N,MAXTRY,AWR,ADR
       DIMENSION T(LDA),TIM(300),IT(8760*6),TA(8760*6),TF(8760*6),
      1                 HT(365,24),TDB(365,24)
      REAL IT,LENGTH,XLOSS,LOAD,XLOAD,l,DC
      INTEGER NHOUR
      DATA CPA/1002./
C
C     Define the constant parameters of the collector and dryer.
C
      ISTART = 1
      ISTOP = 8760*6
      WEIGHT = 200.0
      XMC = 3.0
      WIDTH   = 1.8
      LENGTH  = A/WIDTH
      DELT    = 600.0
      THICKI  = 0.06
      D       = (0.18+0.14)/2.0
      EC      = 0.6
      EP      = 0.9
      TRANC = 0.82
      ALPHAP = 0.9
      RHOC = 0.08
```

```
      FLOW = 0.2628
C
C     Read hourly weather data
C
      OPEN(1,FILE='TMYNP.DAT',STATUS='OLD')
      DO 11 IDAY = 1,365
      DO 12 IHOUR = 1,24
      READ(1,10) HT(IDAY,IHOUR),TDB(IDAY,IHOUR)
 10   FORMAT(F6.1,X,F3.0)
C
C     Interpolate value for simulation
C
      DO 13 I = 1,6
C
C     Transform hourly value(kJ/hour) to secondly value(W)
C
      NHOUR = 24*(IDAY-1)+IHOUR
      ITIME = I+(NHOUR-1)*6
      IT(ITIME) = HT(IDAY,IHOUR)/3600.*1000.
      TA(ITIME) = TDB(IDAY,IHOUR)
 13   CONTINUE
 12   CONTINUE
 11   CONTINUE
      CLOSE(1)
C
      OPEN(4,FILE='OUT.DAT',STATUS='OLD')
C
C     Initialize the temperature of all element of the collector
C
```

```
      DO 70 J=1,NT
      T(J)=TA(ISTART)+273.0
   70 CONTINUE
C
C       Initialize the initial value for annual calculating
C
      QSOLT=0.0
      QCOLT=0.0
      QAUXT=0.0
       QBINT=0.0
      QEJECT=0.0
      LOAD = 0.0
      WT = 0.0
      RT = 0.0
C
C      initialize the necessary parameter.
C
      RHI = 40.0
      RHIN = RHI
      TDI = TA(ISTART)
      HM = HADBRH(TA(ISTART)+273.,RHI/100.)
      WC = WEIGHT
C
C   Increase time to the next time step, dt, after calculating temperature of all nodes.
C
      WRITE(4,*) 'DAY  TIME   TFI    TFO    IT     MC      W'
      DO 80 IDAY=1,365
      DO 90 IHOUR=8,17
      DO 85 I = 1,6
```

```
C
C       Transform hourly value(kJ/hour) to secondly value(W)
C
        NHOUR = 24*(IDAY-1)+IHOUR
        ITIME = I+(NHOUR-1)*6
        IT(ITIME) = HT(IDAY,IHOUR)/3600.*1000.
        TA(ITIME) = TDB(IDAY,IHOUR)
C
C      Calculate the temperature of the collector for each time step
C
        NSEC = 60
        NT=3*NSEC
        DELX=LENGTH/NSEC
        TFI = TA(ITIME)
        HIN = HADBRH(TFI+273.,RHI/100.)
C       G = (FLOW)/(D*WIDTH)
        G = (FLOW)/(D*WIDTH)*(1-R)
        HFLOW = G*D*WIDTH
        HFF = HCF(G/2.)
C
C       Note:Flow in the collector is half of this in dryer.
C
        CALL COL(TFO,NSEC,IT(ITIME),TFI+273.,TA(ITIME)+273.,
        1       HFLOW/2.,DELX,DELT,WIDTH,THICKI,D,EC,EP,
        2       TRANC,ALPHAP,RHOC,HFF)

        IF (IT(ITIME).EQ.0.0) THEN
        TF(ITIME) = TFI
        ELSE
```

```
      TF(ITIME) = TFO
      END IF
C

C     Compute energy obtained from solar collector, QCOL, J/hr
C
      QSOL=A*IT(ITIME)*600.
      QCOL=HFLOW*CPA*(TFO-TFI)*600.
C
C     Simulate drying process in batch dryer
C
      WS = WC/(1+XMC)
      W = 1.60
      H = 1.20
      L = 1.22
      NBAT = 9
      XMO = 3.0
      XMEND = 0.35
      TUPP=60.0
      TLOW=55.0
      HMIX = HM*R+HIN*(1-R)
      DFLOW = R*FLOW
      CALL BATCH(TDO,TDI,TA,DFLOW,HMIX,HM,XMC,XMT,W/2,H,L,NBAT,WC)
C
C     Compute energy supplied by the auxiliary heat source, QAUX, J/10 mins
C
      TMIX = TDO*R+TFO*(1-R)
C     RHMIX = RHDBHA(TMIX+273.15,HMIX)
      RHMIX = RHDBHA(TMIX+273.15,HM)
```

```
      IF ((TMIX-TUPP).GT.0.0) THEN
      QAUX=0.0
      QCOl=XLOAD
      ELSE
      IF ((TMIX-TLOW).GT.0.0) THEN
      QAUX=0.0
      QCOl=XLOAD
      ELSE
      QCOL=HFLOW*CPA*(TFO-TFI)*600.
      QAUX=HFLOW*CPA*(TUPP-TMIX)*600.
      XLOAD=HFLOW*CPA*(TUPP-TFI)*600.
      TMIX = 60.
      END IF
      END IF
      QSOLT=QSOLT+QSOL
      QCOLT=QCOLT+QCOL
      QAUXT=QAUXT+QAUX
      LOAD=LOAD+XLOAD
C
C     Calculate weight of dried bananas and removal water in each batch
C
      WC = (XMT-XMC)*WS+WC
C
C     Send new value as last value in next time step
C
      XMC = XMT
      TDI = TMIX
   85 CONTINUE
C
```

```
C         If moisture content of dried bananas is less than 0.4 db then
C          let start to operate for new batch in next morning.
C

          IF (XMC.LT.0.40) EXIT
   90     CONTINUE
          IF (XMC.LT.0.40) THEN
          WT = WT+WC
          WC = WEIGHT
          XMC = XMO
          ELSE
          END IF
   80     CONTINUE
C
C         Heating value of natural gas (Methane and Butane) is 50,000 kJ/kg
C
          HEATVALUE = 5E7
          GC = QAUXT/HEATVALUE
          END SUBROUTINE
```

5. Subroutine COL.FOR

```
C       234567  Solar Air Collector, file name : COL.FOR
C
C       This program was written for simulation the thermal performance of the
C       solar collector developed at Silpakorn University. The programme is developed based on the
C       mathematical model for which the
C       the energy balance for the covers and fluid air was written seperately.
C       The finite difference method with implicit approach was used to solve the
C       differrential equation of the model.
C
C       Version : 25  December 2003 (Crank-Nicolson)
C                 Use middle decretisation and Crank-Nicolson approach for
C                 Segment 1 and 1<N<NSEC   and use backward descretisation
C                 Crank-Nicolson for the last segment
C                 dT/dx= 0.5(Tx+dx,t+dt - Tx-dx,t+dt)/2dx +
C                        0.5(Tx+dx,t - Tx-dx,t)/2dx
C                  for the last secment dT/dx= 0.5(Tx,t+dt-Tx-dx,t+dt)/dt
C                                               +0.5(Tx,t-Tx-dx,t)/dt
C       Subroutine used :  This programme CALL SUBROUTINE SGEFS to solve the
C                     matrix equation [A].T=B based on the Gaussian elimination
C                     method.The subroutine was built upon  routines from LINPACK
C                     library at the Argonne National Laboratory.
C
C       Importants parameters used in the main programme :
C
```

C A...... Matrix element (left hand side), [W/m2-K]

C ALPHAC.. Absorption coefficient of the cover (for incoming solar radiation ,[decimal]

C ALPHAP...Absorption coefficient of the absorber (for incoming solar radiation ,[decimal]

C B...... Matrix element (right hand side), [W/m2]

C CONDI... Conductivity of back insulator, [W/m]

C CPA..... Specific heat of fluid air, [J/kg-K]

C CPC..... Specific heat of cover glass, [J/kg-K]

C CPP..... Specific heat of absorber plate, [J/kg-K]

C D...... Average high of air gab in the collector, [m]

C DENA.....Density of fluid air, [kg/m3]

C DENC.....Density of cover glass, [kg/m3]

C DENP.....Density of absorber plate, [kg/m3]

C DELX.....Length of each discrete section in finite difference method, [m]

C DELT.....Time step in finite difference with implicit approach, [s]

C EC...... Emissivity of the cover, [decimal]

C EP...... Emissivity of the cover, [decimal]

C FLOW.....Total mass flow rate of the collector, [kg/s]

C G........Mass flow rate per unit area perpendicular to the flow, [kg/s-m2]

C HCBF, HCPF, HFF..Force convective heat transfer coefficient of the fluid [W/m2-K]

C HRCS.....Radiative heat transfer coefficient of the cover to sky, [W/m2-K]

C HRCS.....Radiative heat transfer coefficient of the cover to sky, [W/m2-K]

C HW.......Convective heat transfer coefficient of the cover to air because wind,[W/m2-K]

C IND,IWORK,ITASK..Variable required for solving in subroutine SGEFS

C IT.......Incident solar radiation on the collector, [W/m2]C

C LENGTH...Length of the collector, [m]

C N........Section number, [-]

C NSEC.....Total number of sections of the collector for the finite difference approach, [m]

C NT.......Total number of equations in the matrix system

C QSC......Solar radiation absorbed by the cover2 [W/m2]

```
C       QSP......Solar radiation absorbed by the absorber [W/m2]
C       RCOND....Variable required for solving in subroutine SGEFS
C       RHOC.....Reflectance of cover glass,[decimal]
C       SIXMA....Stefan-Boltzmann constant
C       T........Temperature of elements, [K]
C       TA.......Ambient air temperature, [K]
C       TAUC.....Transmition coefficient of the cover (for thermal radiation) [decimal]
C       TFI......Inlet air temperature of the collector, [K]
C       TFO......Outlet air temperature of the collector, [K]
C       THICKI...Thickness of back insulator, [m]
C       TS.......Sky temperature, [K]
C       UB.......Conductive heat transfer coefficient of the back insulator [W/m2-K]
C       WORK.....Variable required for solving in subroutine SGEFS
C       WIDTH....Width of the collector [m]
C       WIND.....Wind speed caused convective heat loss from the cover2, [m/s]
C       THICKI...Thickness of the back insulator, [m]
C
        SUBROUTINE COL(TFO,NSEC,IT,TFI,TA,FLOW,DELX,DELT,
        1         WIDTH,THICKI,D,EC,EP,
        2         TRANC,ALPHAP,RHOC,HFF)
        PARAMETER (LDA=180)
        COMMON /TEMP/ T
        DIMENSION A(LDA,LDA),B(LDA),WORK(LDA),T(LDA)
        INTEGER IWORK(LDA),I,J,N,NT,ITASK,IND
        REAL IT,LENGTH,RCOND
        DATA SIGMA/5.6697E-8/,CONDI/0.09/,DENA/1.000/,
        1    CPA/1002./,DENC/200./,DENP/6000./,CPC/500./,CPP/400./
C
C       Optical properties of the collector
```

```
C
      ALPHAC=1.-TRANC-RHOC
      ITASK=1
      NT=3*NSEC
C
C     Calculation of parameters which are independent of time.
C
      LENGTH=DELX*NSEC
      AREA=LENGTH*WIDTH
       G=FLOW/(D*WIDTH)
      TS=0.0552*TA**1.5
C
      WIND = 0.0
      HW=5.7+3.8*WIND
      UB=CONDI/THICKI
      CONC=5.0*CPC/DELT
      CONP=0.002*CPP*DENP/DELT
      CONX=0.25*D*G*CPA/DELX
      CONT=DENA*D*CPA/DELT
C
      HCCF = 5*HFF
      HCPF = 5*HFF
C
C     Calculate the solar radiation absorbed by the components of the collector
C
      QSC=ALPHAC*IT
      QSP=ALPHAP*TRANC*IT
C
C    Increase time to the next time step, dt, after calculating temperature of all nodes.
```

```
C
C     Initialisation of the matrix elements
C
      DO 20 I=1,NT
      B(I)=0.0
      DO 20 J=1,NT
      A(I,J)=0.0
 20   CONTINUE
C
C     Begin the calculation of the temperature of all nodes at t+dt,
C     starting with the computation of the matrix element
C
      DO 90 N=1,NSEC
C
C     Calculate radiative heat transfer coefficient (depend on temperature)
C
      HRCS=EC*SIGMA*(TS*TS+T(1+3*(N-1))*T(1+3*(N-1)))*(TS+T(1+3*(N-1)))
      HRPC=SIGMA*(T(1+3*(N-1))*T(1+3*(N-1))+T(3+3*(N-1))*T(3+3*(N-1)))
     1     *(T(1+3*(N-1))+T(3+3*(N-1)))/(1./EC+1./EP-1)
C
C     Calculate the matrix elements
C
C     The matrix elements of Tc
C
      A(1+3*(N-1),1+3*(N-1))= CONC+HCCF+HW+HRCS+HRPC
      A(2+3*(N-1),1+3*(N-1))=-HCCF
```

```
      A(3+3*(N-1),1+3*(N-1))=-HRPC
C
C     The matrix element of Tf (normal elements)
C
      A(1+3*(N-1),2+3*(N-1))=-HCCF
C     A(2+3*(N-1),2+3*(N-1))= HCPF+HCCF
      A(3+3*(N-1),2+3*(N-1))=-HCPF
C
C     The matrix element of Tp
C
      A(1+3*(N-1),3+3*(N-1))=-HRPC
      A(2+3*(N-1),3+3*(N-1))=-HCPF
      A(3+3*(N-1),3+3*(N-1))= CONP+HRPC+HCPF+UB
C
C     Right hand side vector B, except for the term from Tf eq.
C
      B(1+3*(N-1))=CONC*T(1+3*(N-1))+HW*TA+HRCS*TS+QSC
      B(3+3*(N-1))=CONP*T(3+3*(N-1))+UB*TA+QSP
C
C     Matrix elements of Tfx-dx,Tf,Tfx+dx and B
C
      IF(N.EQ.1) THEN
      A(2+3*(N-1),5+3*(N-1))=CONX
      A(2+3*(N-1),2+3*(N-1))=HCPF+HCCF+CONT
      B(2+3*(N-1))=CONX*TFI -CONX*T(5+3*(N-1))+CONX*TFI
     1                        +CONT*T(2+3*(N-1))
```

```
      ELSE
      IF(N.GT.1.AND.N.LT.NSEC) THEN
      A(2+3*(N-1),3*(N-1)-1)=-CONX
      A(2+3*(N-1),2+3*(N-1))= HCPF+HCCF+CONT
      A(2+3*(N-1),5+3*(N-1))= CONX
      B(2+3*(N-1))=CONX*T(3*(N-1)-1)-CONX*T(5+3*(N-1))
      1        +CONT*T(2+3*(N-1))

C
      ELSE
      IF(N.EQ.NSEC) THEN
      A(2+3*(N-1),2+3*(N-1))= 2.*CONX+HCPF+HCPF+CONT
  2
       A(2+3*(N-1),3*(N-1)-1)=-2.*CONX
       B(2+3*(N-1))=-2.*CONX*T(2+3*(N-1))+
  1              2.*CONX*T(3*(N-1)-1)+CONT*T(2+3*(N-1))
       ELSE
       WRITE(*,30)
 30    FORMAT(' Some calculation of matrix elements are missing')
      ENDIF
       ENDIF
      ENDIF
C
 90   CONTINUE
C
C     End of the computation of the matrix element
C
C     Use a subroutine to solve the matrix equation [A].T=B
C     NT in the argument, original is N
```

```
C
      CALL SGEFS(A,LDA,NT,B,ITASK,IND,WORK,IWORK,RCOND)
C

C     Write the solution of the equation [A].T=B
C
C     Transfer the solution of matrix equation to the temperature
C     values for use to calculate the heat transfer coeff. for the next time step.
C
      DO 110 I=1,NT
      T(I)=B(I)
 110  CONTINUE
      TFO=T(2+3*(NSEC-1))-273.15
C     write(*,*) T(1+3*(NSEC-1))-273.,TFO,T(3+3*(NSEC-1))-273.
      RETURN
      END SUBROUTINE

      FUNCTION HCF(V)
C
C     This program is used to calculate force convective heat transfer coefficient from
C     model of Heaton et.al.(1964) in Dufie & Beckman
C
C     D.......Average high of air gab in the collector, [m]
C     DE......Hydrolic diameter, [m]
C     DENS....Density of fluid,[kg/m3]
C     HCF.....Heat transfer coefficient for forced convection in the collector,[W/K-m2]
C     KA......Thermal conductivity of fluid ,[W/m-K]
C     L.......Length of the collector, [m]
C     NU......Nusselt number
```

```
C      RE......Renold number
C      V.......Air velocity above plate,[m/s]
C      VISC....Viscosity of fluid air,[m2/s]
C      W.......Width of the collector [m]
C      Calculate characteristic diameter
C
       W = 1.80
       D = 0.16
       DE = 4*W*D/(2*(W+D))
C
C      Calculate Reynold number
C
       DENS = 1.05
       VISC = 1.88E-05
       RE = DE*V*DENS/VISC
C
C      Calculate Nusselt number
C
       L=8.4
       NU=0.0158*(RE**0.8)*(1+(DE/L)**0.7)
C
C      Calculate the force convective heat transfer coefficient
C
       KA = 0.029
       HCF = NU*0.029/DE
       RETURN
       END FUNCTION
```

6. Subroutine SGEFS

```
      SUBROUTINE SGEFS(A,LDA,N,V,ITASK,IND,WORK,IWORK,RCOND)
C***BEGIN PROLOGUE  SGEFS
C***DATE WRITTEN   800317   (YYMMDD)
C***REVISION DATE  870916   (YYMMDD)
C***CATEGORY NO.  D2A1
C***KEYWORDS  GENERAL SYSTEM OF LINEAR EQUATIONS,LINEAR EQUATIONS
C***AUTHOR  VOORHEES, E., (LOS ALAMOS NATIONAL LABORATORY)
C***PURPOSE  SGEFS solves a GENERAL single precision real
C            NXN system of linear equations.
C***DESCRIPTION
C
C      From the book "Numerical Methods and Software"
C      by  D. Kahaner, C. Moler, S. Nash
C      Prentice Hall 1988
C
C      Subroutine SGEFS solves a general NxN system of single
C      precision linear equations using LINPACK subroutines SGECO
C      and SGESL.  That is, if A is an NxN real matrix and if X
C      and B are real N-vectors, then SGEFS solves the equation
C
C                A*X=B.
C
C      The matrix A is first factored into upper and lower tri-
C      angular matrices U and L using partial pivoting.  These
```

```
C          factors and the pivoting information are used to find the
C          solution vector X.  An approximate condition number is
C          calculated to provide a rough estimate of the number of
C          digits of accuracy in the computed solution.
C
C          If the equation A*X=B is to be solved for more than one vector
C          B, the factoring of A does not need to be performed again and
C          the option to only solve (ITASK .EQ. 2) will be faster for
C          the succeeding solutions.  In this case, the contents of A,
C          LDA, N and IWORK must not have been altered by the user follow-
C          ing factorization (ITASK=1).  IND will not be changed by SGEFS
C          in this case.  Other settings of ITASK are used to solve linear
C          systems involving the transpose of A.
C
C          Argument Description ***
C
C   A      REAL(LDA,N)
C           on entry, the doubly subscripted array with dimension
C             (LDA,N) which contains the coefficient matrix.
C           on return, an upper triangular matrix U and the
C             multipliers necessary to construct a matrix L
C             so that A=L*U.
C   LDA    INTEGER
C           the leading dimension of the array A.  LDA must be great-
C           er than or equal to N.  (terminal error message IND=-1)
C   N      INTEGER
C           the order of the matrix A.  The first N elements of
C           the array A are the elements of the first column of
C           the  matrix A.  N must be greater than or equal to 1.
```

```
C           (terminal error message IND=-2)
C   V       REAL(N)
C           on entry, the singly subscripted array(vector) of di-
C             mension N which contains the right hand side B of a
C             system of simultaneous linear equations A*X=B.
C           on return, V contains the solution vector, X .
C   ITASK  INTEGER
C           If ITASK=1, the matrix A is factored and then the
C             linear equation is solved.
C           If ITASK=2, the equation is solved using the existing
C             factored matrix A and IWORK.
C           If ITASK=3, the matrix is factored and A'x=b is solved
C           If ITASK=4, the transposed equation is solved using the
C             existing factored matrix A and IWORK.
C           If ITASK .LT. 1 or ITASK .GT. 4, then the terminal error
C             message IND=-3 is printed.
C   IND    INTEGER
C           GT. 0  IND is a rough estimate of the number of digits
C                  of accuracy in the solution, X.
C           LT. 0  see error message corresponding to IND below.
C   WORK   REAL(N)
C           a singly subscripted array of dimension at least N.
C   IWORK  INTEGER(N)
C           a singly subscripted array of dimension at least N.
C   RCOND  REAL
C           estimate of 1.0/cond(A)
C
C  Error Messages Printed ***
C
```

```
C   IND=-1  fatal   N is greater than LDA.
C   IND=-2  fatal   N is less than 1.
C   IND=-3  fatal   ITASK is less than 1 or greater than 4.
C   IND=-4  fatal   The matrix A is computationally singular.
C                   A solution has not been computed.
C   IND=-10 warning    The solution has no apparent significance.
C                   The solution may be inaccurate or the matrix
C                   A may be poorly scaled.
C
C***REFERENCES  SUBROUTINE SGEFS WAS DEVELOPED BY GROUP C-3, LOS
                ALAMOS SCIENTIFIC LABORATORY, LOS ALAMOS, NM 87545.
C               THE LINPACK SUBROUTINES USED BY SGEFS ARE DESCRIBED IN
C               DETAIL IN THE *LINPACK USERS GUIDE* PUBLISHED BY
C               THE SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS
C               (SIAM) DATED 1979.
C***ROUTINES CALLED  R1MACH,SGECO,SGESL,XERROR
C***END PROLOGUE  SGEFS
C
      INTEGER LDA,N,ITASK,IND,IWORK(*)
      REAL A(LDA,*),V(*),WORK(*),R1MACH
      REAL RCOND
      CHARACTER MSG*54
C***FIRST EXECUTABLE STATEMENT  SGEFS
      IF (LDA.LT.N)  GO TO 101
      IF (N.LE.0)  GO TO 102
      IF (ITASK.LT.1) GO TO 103
      IF (ITASK.GT.4) GO TO 103
       IF (ITASK.EQ.2 .OR. ITASK.GT.3) GO TO 20
C
```

```
C         FACTOR MATRIX A INTO LU
          CALL SGECO(A,LDA,N,IWORK,RCOND,WORK)
C
C         CHECK FOR COMPUTATIONALLY SINGULAR MATRIX
          IF (RCOND.EQ.0.0)  GO TO 104
C         COMPUTE IND (ESTIMATE OF NO. OF SIGNIFICANT DIGITS)
          IND=-INT(ALOG10(R1MACH(4)/RCOND))
C
C         CHECK FOR IND GREATER THAN ZERO
          IF (IND.GT.0)  GO TO 20
          IND=-10
          CALL XERROR( 'SGEFS ERROR (IND=-10) -- SOLUTION MAY HAVE NO SIGNIF
     1I   CANCE',58,-10,0)
C
C         SOLVE AFTER FACTORING
   20     JOB=0
          IF (ITASK.GT.2) JOB=1
          CALL SGESL(A,LDA,N,IWORK,V,JOB)
          RETURN
C
C         IF LDA.LT.N, IND=-1, FATAL XERROR MESSAGE
  101     IND=-1
           WRITE(MSG, '(
          * "SGEFS ERROR (IND=-1) -- LDA=", I5, " IS LESS THAN N=",
          *    I5      )'   ) LDA, N
          CALL XERROR(MSG(1:54), 54, -1, 0)
          RETURN
C
C         IF N.LT.1, IND=-2, FATAL XERROR MESSAGE
```

```
 102    IND=-2
         WRITE(MSG, '(
        * "SGEFS ERROR (IND=-2) -- N=", I5, " IS LESS THAN 1.") ')N
        CALL XERROR(MSG(1:47), 47, -2, 0)
        RETURN
C
C       IF ITASK.LT.1, IND=-3, FATAL XERROR MESSAGE
 103    IND=-3
        WRITE(MSG, '(
        * "SGEFS ERROR (IND=-3) -- ITASK=", I5, " IS LT 1 OR GT 4.")
        *              ') ITASK
        CALL XERROR(MSG(1:52), 52, -3, 0)
        RETURN
C
C       IF SINGULAR MATRIX, IND=-4, FATAL XERROR MESSAGE
 104    IND=-4
        CALL XERROR( 'SGEFS ERROR (IND=-4) -- SINGULAR MATRIX A - NO SOLUT
        1ION',55,-4,0)
        RETURN
C
        END

        SUBROUTINE SGECO(A,LDA,N,IPVT,RCOND,Z)
C***BEGIN PROLOGUE  SGECO
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C        by  D. Kahaner, C. Moler, S. Nash
C            Prentice Hall 1988
```

```
C***ROUTINES CALLED  SASUM,SAXPY,SDOT,SGEFA,SSCAL
C***END PROLOGUE  SGECO
      INTEGER LDA,N,IPVT(*)
      REAL A(LDA,*),Z(*)
      REAL RCOND
C
      REAL SDOT,EK,T,WK,WKM
      REAL ANORM,S,SASUM,SM,YNORM
      INTEGER INFO,J,K,KB,KP1,L
C
C     COMPUTE 1-NORM OF A
C
C***FIRST EXECUTABLE STATEMENT  SGECO
      ANORM = 0.0E0
      DO 10 J = 1, N
      ANORM = AMAX1(ANORM,SASUM(N,A(1,J),1))
   10 CONTINUE
C
C     FACTOR
C
      CALL SGEFA(A,LDA,N,IPVT,INFO)
C
C   RCOND = 1/(NORM(A)*(ESTIMATE OF NORM(INVERSE(A)))) .
C   ESTIMATE = NORM(Z)/NORM(Y) WHERE  A*Z = Y  AND  TRANS(A)*Y = E .
C   TRANS(A)  IS THE TRANSPOSE OF A .  THE COMPONENTS OF  E  ARE
C   CHOSEN TO CAUSE MAXIMUM LOCAL GROWTH IN THE ELEMENTS OF W  WHERE
C   TRANS(U)*W = E .  THE VECTORS ARE FREQUENTLY RESCALED TO AVOID
C   OVERFLOW.
C
```

```
C     SOLVE TRANS(U)*W = E
C
            EK = 1.0E0
            DO 20 J = 1, N
             Z(J) = 0.0E0
   20       CONTINUE
            DO 100 K = 1, N
            IF (Z(K) .NE. 0.0E0) EK = SIGN(EK,-Z(K))
            IF (ABS(EK-Z(K)) .LE. ABS(A(K,K))) GO TO 30
            S = ABS(A(K,K))/ABS(EK-Z(K))
            CALL SSCAL(N,S,Z,1)
            EK = S*EK
   30       CONTINUE
            WK = EK - Z(K)
            WKM = -EK - Z(K)
            S = ABS(WK)
            SM = ABS(WKM)
            IF (A(K,K) .EQ. 0.0E0) GO TO 40
            WK = WK/A(K,K)
            WKM = WKM/A(K,K)
            GO TO 50
   40       CONTINUE
            WK = 1.0E0
            WKM = 1.0E0
   50       CONTINUE
            KP1 = K + 1
            IF (KP1 .GT. N) GO TO 90
            DO 60 J = KP1, N
             SM = SM + ABS(Z(J)+WKM*A(K,J))
```

```
       Z(J) = Z(J) + WK*A(K,J)
       S = S + ABS(Z(J))
 60    CONTINUE
       IF (S .GE. SM) GO TO 80
       T = WKM - WK
       WK = WKM
       DO 70 J = KP1, N
       Z(J) = Z(J) + T*A(K,J)
 70    CONTINUE
 80    CONTINUE
 90    CONTINUE
       Z(K) = WK
100    CONTINUE
       S = 1.0E0/SASUM(N,Z,1)
       CALL SSCAL(N,S,Z,1)
C
C      SOLVE TRANS(L)*Y = W
C
       DO 120 KB = 1, N
       K = N + 1 - KB
       IF (K .LT. N) Z(K) = Z(K) + SDOT(N-K,A(K+1,K),1,Z(K+1),1)
       IF (ABS(Z(K)) .LE. 1.0E0) GO TO 110
       S = 1.0E0/ABS(Z(K))
       CALL SSCAL(N,S,Z,1)
110    CONTINUE
       L = IPVT(K)
       T = Z(L)
       Z(L) = Z(K)
       Z(K) = T
```

```
 120    CONTINUE
        S = 1.0E0/SASUM(N,Z,1)
        CALL SSCAL(N,S,Z,1)
C
        YNORM = 1.0E0
C
C       SOLVE L*V = Y
C
        DO 140 K = 1, N
        L = IPVT(K)
        T = Z(L)
        Z(L) = Z(K)
        Z(K) = T
         IF (K .LT. N) CALL SAXPY(N-K,T,A(K+1,K),1,Z(K+1),1)
        IF (ABS(Z(K)) .LE. 1.0E0) GO TO 130
        S = 1.0E0/ABS(Z(K))
        CALL SSCAL(N,S,Z,1)
        YNORM = S*YNORM
 130    CONTINUE
 140    CONTINUE
        S = 1.0E0/SASUM(N,Z,1)
        CALL SSCAL(N,S,Z,1)
        YNORM = S*YNORM
C
C       SOLVE  U*Z = V
C
        DO 160 KB = 1, N
        K = N + 1 - KB
        IF (ABS(Z(K)) .LE. ABS(A(K,K))) GO TO 150
```

```
      S = ABS(A(K,K))/ABS(Z(K))
      CALL SSCAL(N,S,Z,1)
      YNORM = S*YNORM
  150 CONTINUE
      IF (A(K,K) .NE. 0.0E0) Z(K) = Z(K)/A(K,K)
      IF (A(K,K) .EQ. 0.0E0) Z(K) = 1.0E0
      T = -Z(K)
      CALL SAXPY(K-1,T,A(1,K),1,Z(1),1)
  160 CONTINUE
C     MAKE ZNORM = 1.0
      S = 1.0E0/SASUM(N,Z,1)
      CALL SSCAL(N,S,Z,1)
      YNORM = S*YNORM
C
      IF (ANORM .NE. 0.0E0) RCOND = YNORM/ANORM
      IF (ANORM .EQ. 0.0E0) RCOND = 0.0E0
      RETURN
      END
      SUBROUTINE SGEFA(A,LDA,N,IPVT,INFO)
C***BEGIN PROLOGUE  SGEFA
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C        by  D. Kahaner, C. Moler, S. Nash
C            Prentice Hall 1988
C***END PROLOGUE  SGEFA
      INTEGER LDA,N,IPVT(*),INFO
      REAL A(LDA,*)
C
```

```
      REAL T
      INTEGER ISAMAX,J,K,KP1,L,NM1
C
C   GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
C***FIRST EXECUTABLE STATEMENT  SGEFA
      INFO = 0
      NM1 = N - 1
      IF (NM1 .LT. 1) GO TO 70
       DO 60 K = 1, NM1
      KP1 = K + 1
C
C      FIND L = PIVOT INDEX
C
      L = ISAMAX(N-K+1,A(K,K),1) + K - 1
      IPVT(K) = L
C
C      ZERO PIVOT IMPLIES THIS COLUMN ALREADY TRIANGULARIZED
C
      IF (A(L,K) .EQ. 0.0E0) GO TO 40
C
C        INTERCHANGE IF NECESSARY
C
      IF (L .EQ. K) GO TO 10
       T = A(L,K)
       A(L,K) = A(K,K)
       A(K,K) = T
   10  CONTINUE
C
```

```
C         COMPUTE MULTIPLIERS
C
          T = -1.0E0/A(K,K)
          CALL SSCAL(N-K,T,A(K+1,K),1)
C
C         ROW ELIMINATION WITH COLUMN INDEXING
C
          DO 30 J = KP1, N
           T = A(L,J)
           IF (L .EQ. K) GO TO 20
           A(L,J) = A(K,J)
           A(K,J) = T
   20      CONTINUE
           CALL SAXPY(N-K,T,A(K+1,K),1,A(K+1,J),1)
   30     CONTINUE
          GO TO 50
   40     CONTINUE
          INFO = K
   50     CONTINUE
   60     CONTINUE
   70     CONTINUE
          IPVT(N) = N
          IF (A(N,N) .EQ. 0.0E0) INFO = N
          RETURN
          END

          SUBROUTINE SGESL(A,LDA,N,IPVT,B,JOB)
C***BEGIN PROLOGUE  SGESL
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
```

```
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C         by  D. Kahaner, C. Moler, S. Nash
C              Prentice Hall 1988
C***END PROLOGUE  SGESL
      INTEGER LDA,N,IPVT(*),JOB
      REAL A(LDA,*),B(*)
C
      REAL SDOT,T
      INTEGER K,KB,L,NM1
C***FIRST EXECUTABLE STATEMENT  SGESL
      NM1 = N - 1
      IF (JOB .NE. 0) GO TO 50
C
C       JOB = 0 , SOLVE  A * X = B
C       FIRST SOLVE  L*Y = B
C
      IF (NM1 .LT. 1) GO TO 30
      DO 20 K = 1, NM1
      L = IPVT(K)
      T = B(L)
      IF (L .EQ. K) GO TO 10
       B(L) = B(K)
       B(K) = T
 10   CONTINUE
      CALL SAXPY(N-K,T,A(K+1,K),1,B(K+1),1)
 20   CONTINUE
 30   CONTINUE
C
```

```
C        NOW SOLVE  U*X = Y
C
         DO 40 KB = 1, N
         K = N + 1 - KB
         B(K) = B(K)/A(K,K)
         T = -B(K)
         CALL SAXPY(K-1,T,A(1,K),1,B(1),1)
  40     CONTINUE
         GO TO 100
  50     CONTINUE
C
C      JOB = NONZERO, SOLVE  TRANS(A) * X = B
C      FIRST SOLVE  TRANS(U)*Y = B
C
         DO 60 K = 1, N
         T = SDOT(K-1,A(1,K),1,B(1),1)
         B(K) = (B(K) - T)/A(K,K)
  60     CONTINUE
C
C        NOW SOLVE TRANS(L)*X = Y
C
         IF (NM1 .LT. 1) GO TO 90
         DO 80 KB = 1, NM1
         K = N - KB
         B(K) = B(K) + SDOT(N-K,A(K+1,K),1,B(K+1),1)
         L = IPVT(K)
         IF (L .EQ. K) GO TO 70
          T = B(L)
          B(L) = B(K)
```

```
      B(K) = T
 70   CONTINUE
 80   CONTINUE
 90   CONTINUE
100   CONTINUE
      RETURN
      END

      INTEGER FUNCTION ISAMAX(N,SX,INCX)
C***BEGIN PROLOGUE  ISAMAX
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C       by  D. Kahaner, C. Moler, S. Nash
C           Prentice Hall 1988
C***END PROLOGUE  ISAMAX
C
      REAL SX(*),SMAX,XMAG
C***FIRST EXECUTABLE STATEMENT  ISAMAX
      ISAMAX = 0
      IF(N.LE.0) RETURN
      ISAMAX = 1
      IF(N.LE.1)RETURN
      IF(INCX.EQ.1)GOTO 20
C
C      CODE FOR INCREMENTS NOT EQUAL TO 1.
C
      SMAX = ABS(SX(1))
      NS = N*INCX
```

```
      II = 1
      DO 10 I=1,NS,INCX
      XMAG = ABS(SX(I))
      IF(XMAG.LE.SMAX) GO TO 5
      ISAMAX = II
      SMAX = XMAG
  5   II = II + 1
 10   CONTINUE
      RETURN
C
C     CODE FOR INCREMENTS EQUAL TO 1.
C
      20 SMAX = ABS(SX(1))
      DO 30 I = 2,N
      XMAG = ABS(SX(I))
      IF(XMAG.LE.SMAX) GO TO 30
      ISAMAX = I
      SMAX = XMAG
 30   CONTINUE
      RETURN
      END

      REAL FUNCTION SASUM(N,SX,INCX)
C***BEGIN PROLOGUE  SASUM
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C       by  D. Kahaner, C. Moler, S. Nash
C           Prentice Hall 1988
```

```
C***END PROLOGUE  SASUM
C
      REAL SX(*)
C***FIRST EXECUTABLE STATEMENT  SASUM
      SASUM = 0.0E0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1)GOTO 20
C
C     CODE FOR INCREMENTS NOT EQUAL TO 1.
C
      NS = N*INCX
      DO 10 I=1,NS,INCX
       SASUM = SASUM + ABS(SX(I))
   10 CONTINUE
      RETURN
C
C     CODE FOR INCREMENTS EQUAL TO 1.
C
C
C     CLEAN-UP LOOP SO REMAINING VECTOR LENGTH IS A MULTIPLE OF 6.
C
   20 M = MOD(N,6)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
      SASUM = SASUM + ABS(SX(I))
   30 CONTINUE
       IF( N .LT. 6 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,6
```

```
      SASUM = SASUM + ABS(SX(I)) + ABS(SX(I + 1)) + ABS(SX(I + 2))
      1  + ABS(SX(I + 3)) + ABS(SX(I + 4)) + ABS(SX(I + 5))
   50 CONTINUE
      RETURN
      END
      SUBROUTINE SAXPY(N,SA,SX,INCX,SY,INCY)
C***BEGIN PROLOGUE  SAXPY
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C         by  D. Kahaner, C. Moler, S. Nash
C             Prentice Hall 1988
C***END PROLOGUE  SAXPY
C
      REAL SX(*),SY(*),SA
C***FIRST EXECUTABLE STATEMENT  SAXPY
      IF(N.LE.0.OR.SA.EQ.0.E0) RETURN
      IF(INCX.EQ.INCY) IF(INCX-1) 5,20,60
    5 CONTINUE
C
C       CODE FOR NONEQUAL OR NONPOSITIVE INCREMENTS.
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
      SY(IY) = SY(IY) + SA*SX(IX)
      IX = IX + INCX
```

```
      IY = IY + INCY
 10   CONTINUE
      RETURN
C
C     CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C     CLEAN-UP LOOP SO REMAINING VECTOR LENGTH IS A MULTIPLE OF 4.
C
 20   M = MOD(N,4)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
      SY(I) = SY(I) + SA*SX(I)
 30   CONTINUE
      IF( N .LT. 4 ) RETURN
 40   MP1 = M + 1
      DO 50 I = MP1,N,4
      SY(I) = SY(I) + SA*SX(I)
      SY(I + 1) = SY(I + 1) + SA*SX(I + 1)
      SY(I + 2) = SY(I + 2) + SA*SX(I + 2)
      SY(I + 3) = SY(I + 3) + SA*SX(I + 3)
 50   CONTINUE
      RETURN
C
C     CODE FOR EQUAL, POSITIVE, NONUNIT INCREMENTS.
C
 60   CONTINUE
      NS = N*INCX
      DO 70 I=1,NS,INCX
      SY(I) = SA*SX(I) + SY(I)
```

```
   70   CONTINUE
        RETURN
        END

        SUBROUTINE SCOPY(N,SX,INCX,SY,INCY)
C***BEGIN PROLOGUE  SCOPY
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C         by  D. Kahaner, C. Moler, S. Nash
C             Prentice Hall 1988
C***END PROLOGUE  SCOPY
C
        REAL SX(*),SY(*)
C***FIRST EXECUTABLE STATEMENT  SCOPY
        IF(N.LE.0)RETURN
        IF(INCX.EQ.INCY) IF(INCX-1) 5,20,60
    5   CONTINUE
C
C       CODE FOR UNEQUAL OR NONPOSITIVE INCREMENTS.
C
        IX = 1
        IY = 1
        IF(INCX.LT.0)IX = (-N+1)*INCX + 1
        IF(INCY.LT.0)IY = (-N+1)*INCY + 1
        DO 10 I = 1,N
        SY(IY) = SX(IX)
        IX = IX + INCX
        IY = IY + INCY
```

```
   10    CONTINUE
         RETURN
C

C      CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C      CLEAN-UP LOOP SO REMAINING VECTOR LENGTH IS A MULTIPLE OF 7.
C
   20    M = MOD(N,7)
         IF( M .EQ. 0 ) GO TO 40
         DO 30 I = 1,M
         SY(I) = SX(I)
   30    CONTINUE
         IF( N .LT. 7 ) RETURN
   40    MP1 = M + 1
         DO 50 I = MP1,N,7
         SY(I) = SX(I)
         SY(I + 1) = SX(I + 1)
         SY(I + 2) = SX(I + 2)
         SY(I + 3) = SX(I + 3)
         SY(I + 4) = SX(I + 4)
         SY(I + 5) = SX(I + 5)
         SY(I + 6) = SX(I + 6)
   50    CONTINUE
         RETURN
C
C      CODE FOR EQUAL, POSITIVE, NONUNIT INCREMENTS.
C
   60    CONTINUE
```

```
      NS = N*INCX
      DO 70 I=1,NS,INCX
      SY(I) = SX(I)
   70 CONTINUE
      RETURN
      END

      REAL FUNCTION SDOT(N,SX,INCX,SY,INCY)
C***BEGIN PROLOGUE  SDOT
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C         by  D. Kahaner, C. Moler, S. Nash
C              Prentice Hall 1988
C***END PROLOGUE  SDOT
C
      REAL SX(*),SY(*)
C***FIRST EXECUTABLE STATEMENT  SDOT
      SDOT = 0.0E0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.INCY) IF(INCX-1)5,20,60
    5 CONTINUE
C
C       CODE FOR UNEQUAL INCREMENTS OR NONPOSITIVE INCREMENTS.
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
```

```
      DO 10 I = 1,N
      SDOT = SDOT + SX(IX)*SY(IY)
      IX = IX + INCX
      IY = IY + INCY
   10 CONTINUE
      RETURN
C
C     CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C     CLEAN-UP LOOP SO REMAINING VECTOR LENGTH IS A MULTIPLE OF 5.
C
   20 M = MOD(N,5)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
      SDOT = SDOT + SX(I)*SY(I)
   30 CONTINUE
      IF( N .LT. 5 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,5
      SDOT = SDOT + SX(I)*SY(I) + SX(I + 1)*SY(I + 1) +
      1   SX(I + 2)*SY(I + 2) + SX(I + 3)*SY(I + 3) + SX(I + 4)*SY(I + 4)
   50 CONTINUE
      RETURN
C
C     CODE FOR POSITIVE EQUAL INCREMENTS .NE.1.
C
   60 CONTINUE
      NS=N*INCX
      DO 70 I=1,NS,INCX
```

```
      SDOT = SDOT + SX(I)*SY(I)
   70 CONTINUE
      RETURN
      END
      REAL FUNCTION SNRM2(N,SX,INCX)
C***BEGIN PROLOGUE  SNRM2
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C       by  D. Kahaner, C. Moler, S. Nash
C           Prentice Hall 1988
C***END PROLOGUE  SNRM2
      INTEGER         NEXT
      REAL   SX(*),  CUTLO, CUTHI, HITEST, SUM, XMAX, ZERO, ONE
      DATA   ZERO, ONE /0.0E0, 1.0E0/
C
      DATA CUTLO, CUTHI / 4.441E-16,  1.304E19 /
C***FIRST EXECUTABLE STATEMENT  SNRM2
      IF(N .GT. 0) GO TO 10
      SNRM2  = ZERO
      GO TO 300
C
   10 ASSIGN 30 TO NEXT
      SUM = ZERO
      NN = N * INCX
C     BEGIN MAIN LOOP
      I = 1
   20 GO TO NEXT,(30, 50, 70, 110)
   30 IF( ABS(SX(I)) .GT. CUTLO) GO TO 85
```

```
      ASSIGN 50 TO NEXT
      XMAX = ZERO
C
C     PHASE 1.  SUM IS ZERO
C
 50   IF( SX(I) .EQ. ZERO) GO TO 200
      IF( ABS(SX(I)) .GT. CUTLO) GO TO 85
C
C     PREPARE FOR PHASE 2.
      ASSIGN 70 TO NEXT
      GO TO 105
C
C     PREPARE FOR PHASE 4.
C
 100  I = J
      ASSIGN 110 TO NEXT
      SUM = (SUM / SX(I)) / SX(I)
 105  XMAX = ABS(SX(I))
      GO TO 115
C
C     PHASE 2.  SUM IS SMALL.
C     SCALE TO AVOID DESTRUCTIVE UNDERFLOW.
C
 70   IF( ABS(SX(I)) .GT. CUTLO ) GO TO 75
C
C     COMMON CODE FOR PHASES 2 AND 4.
C      IN PHASE 4 SUM IS LARGE.  SCALE TO AVOID OVERFLOW.
C
 110  IF( ABS(SX(I)) .LE. XMAX ) GO TO 115
```

```
      SUM = ONE + SUM * (XMAX / SX(I))**2
      XMAX = ABS(SX(I))
      GO TO 200
C
  115 SUM = SUM + (SX(I)/XMAX)**2
      GO TO 200
C
C
C     PREPARE FOR PHASE 3.
C
   75 SUM = (SUM * XMAX) * XMAX
C
C
C   FOR REAL OR D.P. SET HITEST = CUTHI/N
C   FOR COMPLEX      SET HITEST = CUTHI/(2*N)
C
   85 HITEST = CUTHI/FLOAT( N )
C
C      PHASE 3.  SUM IS MID-RANGE.  NO SCALING.
C
      DO 95 J =I,NN,INCX
      IF(ABS(SX(J)) .GE. HITEST) GO TO 100
   95 SUM = SUM + SX(J)**2
      SNRM2 = SQRT( SUM )
      GO TO 300
C
  200 CONTINUE
      I = I + INCX
      IF ( I .LE. NN ) GO TO 20
```

```
C
C             END OF MAIN LOOP.
C
C             COMPUTE SQUARE ROOT AND ADJUST FOR SCALING.
C
      SNRM2 = XMAX * SQRT(SUM)
  300 CONTINUE
      RETURN
      END

      SUBROUTINE SSCAL(N,SA,SX,INCX)
C***BEGIN PROLOGUE  SSCAL
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
C    From the book "Numerical Methods and Software"
C         by  D. Kahaner, C. Moler, S. Nash
C              Prentice Hall 1988
C***END PROLOGUE  SSCAL
C
      REAL SA,SX(*)
C***FIRST EXECUTABLE STATEMENT  SSCAL
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1)GOTO 20
C
C        CODE FOR INCREMENTS NOT EQUAL TO 1.
C
      NS = N*INCX
      DO 10 I = 1,NS,INCX
      SX(I) = SA*SX(I)
```

```
   10    CONTINUE
         RETURN
C

C     CODE FOR INCREMENTS EQUAL TO 1.
C
C     CLEAN-UP LOOP SO REMAINING VECTOR LENGTH IS A MULTIPLE OF 5.
C
   20    M = MOD(N,5)
         IF( M .EQ. 0 ) GO TO 40
         DO 30 I = 1,M
         SX(I) = SA*SX(I)
   30    CONTINUE
         IF( N .LT. 5 ) RETURN
   40    MP1 = M + 1
         DO 50 I = MP1,N,5
         SX(I) = SA*SX(I)
         SX(I + 1) = SA*SX(I + 1)
         SX(I + 2) = SA*SX(I + 2)
         SX(I + 3) = SA*SX(I + 3)
         SX(I + 4) = SA*SX(I + 4)
   50    CONTINUE
         RETURN
         END

         SUBROUTINE SSWAP(N,SX,INCX,SY,INCY)
C***BEGIN PROLOGUE  SSWAP
C    THIS PROLOGUE HAS BEEN REMOVED FOR REASONS OF SPACE
C    FOR A COMPLETE COPY OF THIS ROUTINE CONTACT THE AUTHORS
```

```
C     From the book "Numerical Methods and Software"
C          by  D. Kahaner, C. Moler, S. Nash
C               Prentice Hall 1988
C***END PROLOGUE  SSWAP
C
      REAL SX(*),SY(*),STEMP1,STEMP2,STEMP3
C***FIRST EXECUTABLE STATEMENT  SSWAP
      IF(N.LE.0)RETURN
      IF(INCX.EQ.INCY) IF(INCX-1) 5,20,60
    5 CONTINUE
C
C       CODE FOR UNEQUAL OR NONPOSITIVE INCREMENTS.
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
      STEMP1 = SX(IX)
      SX(IX) = SY(IY)
      SY(IY) = STEMP1
      IX = IX + INCX
      IY = IY + INCY
   10 CONTINUE
      RETURN
C
C       CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
```

```
C     CLEAN-UP LOOP SO REMAINING VECTOR LENGTH IS A MULTIPLE OF 3.
C
   20    M = MOD(N,3)
         IF( M .EQ. 0 ) GO TO 40
         DO 30 I = 1,M
         STEMP1 = SX(I)
         SX(I) = SY(I)
         SY(I) = STEMP1
   30    CONTINUE
         IF( N .LT. 3 ) RETURN
   40    MP1 = M + 1
         DO 50 I = MP1,N,3
         STEMP1 = SX(I)
         STEMP2 = SX(I+1)
         STEMP3 = SX(I+2)
         SX(I) = SY(I)
         SX(I+1) = SY(I+1)
         SX(I+2) = SY(I+2)
         SY(I) = STEMP1
         SY(I+1) = STEMP2
         SY(I+2) = STEMP3
   50    CONTINUE
         RETURN
   60    CONTINUE
C
C    CODE FOR EQUAL, POSITIVE, NONUNIT INCREMENTS.
C
         NS = N*INCX
         DO 70 I=1,NS,INCX
```

```
      STEMP1 = SX(I)
      SX(I) = SY(I)
      SY(I) = STEMP1
   70 CONTINUE
      RETURN
      END

      REAL FUNCTION R1MACH(I)
C***BEGIN PROLOGUE  R1MACH
C***DATE WRITTEN   790101   (YYMMDD)
C***REVISION DATE  831014   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C          HALL, A. D., (BELL LABS)
C          SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns single precision machine dependent constants
C***DESCRIPTION
C    From the book, "Numerical Methods and Software" by
C              D. Kahaner, C. Moler, S. Nash
C              Prentice Hall, 1988
C
C
C    R1MACH can be used to obtain machine-dependent parameters
C    for the local machine environment.  It is a function
C    subroutine with one (input) argument, and can be called
C    as follows, for example
C
C        A = R1MACH(I)
```

```
C
C    where I=1,...,5.  The (output) value of A above is
C    determined by the (input) value of I.  The results for
C    various values of I are discussed below.
C
C  Single-Precision Machine Constants
C  R1MACH(1) = B**(EMIN-1), the smallest positive magnitude.
C  R1MACH(2) = B**EMAX*(1 - B**(-T)), the largest magnitude.
C  R1MACH(3) = B**(-T), the smallest relative spacing.
C  R1MACH(4) = B**(1-T), the largest relative spacing.
C  R1MACH(5) = LOG10(B)
C***REFERENCES  FOX, P.A., HALL, A.D., SCHRYER, N.L, *FRAMEWORK FOR
C                 A PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHE-
C                 MATICAL SOFTWARE, VOL. 4, NO. 2, JUNE 1978,
C                 PP. 177-188.
C***ROUTINES CALLED  XERROR
C***END PROLOGUE  R1MACH
C
      INTEGER SMALL(2)
      INTEGER LARGE(2)
      INTEGER RIGHT(2)
      INTEGER DIVER(2)
      INTEGER LOG10(2)
C
      REAL RMACH(5)
C
    EQUIVALENCE (RMACH(1),SMALL(1))
    EQUIVALENCE (RMACH(2),LARGE(1))
    EQUIVALENCE (RMACH(3),RIGHT(1))
```

```
      EQUIVALENCE (RMACH(4),DIVER(1))
      EQUIVALENCE (RMACH(5),LOG10(1))
C

C     MACHINE CONSTANTS FOR THE CDC CYBER 170 SERIES (FTN5).
C
C      DATA RMACH(1) / O"00014000000000000000" /
C      DATA RMACH(2) / O"37767777777777777777" /
C      DATA RMACH(3) / O"16404000000000000000" /
C      DATA RMACH(4) / O"16414000000000000000" /
C      DATA RMACH(5) / O"17164642023241175720" /
C
C     MACHINE CONSTANTS FOR THE CDC CYBER 200 SERIES
C
C     DATA RMACH(1) / X'9000400000000000' /
C     DATA RMACH(2) / X'6FFF7FFFFFFFFFFF' /
C     DATA RMACH(3) / X'FFA3400000000000' /
C     DATA RMACH(4) / X'FFA4400000000000' /
C     DATA RMACH(5) / X'FFD04D104D427DE8' /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C     DATA RMACH(1) / 00564000000000000000B /
C     DATA RMACH(2) / 37767777777777777776B /
C     DATA RMACH(3) / 16414000000000000000B /
C     DATA RMACH(4) / 16424000000000000000B /
C     DATA RMACH(5) / 17164642023241175720B /
C
C     MACHINE CONSTANTS FOR THE CRAY 1
```

```
C
C     DATA RMACH(1) / 20003400000000000000B /
C     DATA RMACH(2) / 57776777777777777776B /
C     DATA RMACH(3) / 37722400000000000000B /
C     DATA RMACH(4) / 37723400000000000000B /
C     DATA RMACH(5) / 37777464202324117572OB /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86  AND
C     THE PERKIN ELMER (INTERDATA) 7/32.
C
C     DATA RMACH(1) / Z00100000 /
C     DATA RMACH(2) / Z7FFFFFFF /
C     DATA RMACH(3) / Z3B100000 /
C     DATA RMACH(4) / Z3C100000 /
C     DATA RMACH(5) / Z41134413 /
C
C     MACHINE CONSTANTS FOR THE IBM PC FAMILY (D. KAHANER NBS)
C
      DATA RMACH/1.18E-38,3.40E+38,0.595E-07,1.19E-07,0.30102999566/
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA OR KI PROCESSOR).
C
C     DATA RMACH(1) / "000400000000 /
C     DATA RMACH(2) / "377777777777 /
C     DATA RMACH(3) / "146400000000 /
C     DATA RMACH(4) / "147400000000 /
C     DATA RMACH(5) / "177464202324 /
C
```

```
C
C    MACHINE CONSTANTS FOR THE SUN-3 (INCLUDES THOSE WITH 68881 CHIP,
C      OR WITH FPA BOARD. ALSO INCLUDES SUN-2 WITH SKY BOARD. MAY ALSO
C      WORK WITH SOFTWARE FLOATING POINT ON EITHER SYSTEM.)
C    DATA SMALL(1) / X'00800000' /
C    DATA LARGE(1) / X'7F7FFFFF' /
C    DATA RIGHT(1) / X'33800000' /
C    DATA DIVER(1) / X'34000000' /
C    DATA LOG10(1) / X'3E9A209B' /
C
C
C    MACHINE CONSTANTS FOR THE VAX 11/780
C   (EXPRESSED IN INTEGER AND HEXADECIMAL)
C  *** THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS***
C
C    DATA SMALL(1) /       128 /
C    DATA LARGE(1) /    -32769 /
C    DATA RIGHT(1) /     13440 /
C    DATA DIVER(1) /     13568 /
C    DATA LOG10(1) / 547045274 /
C
C  ***THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSTEMS***
C    DATA SMALL(1) / Z00000080 /
C    DATA LARGE(1) / ZFFFF7FFF /
C    DATA RIGHT(1) / Z00003480 /
C    DATA DIVER(1) / Z00003500 /
C    DATA LOG10(1) / Z209B3F9A /
C
C
```

```
C***FIRST EXECUTABLE STATEMENT  R1MACH
      IF (I .LT. 1  .OR.  I .GT. 5)
     1   CALL XERROR ( 'R1MACH -- I OUT OF BOUNDS',25,1,2)
C
      R1MACH = RMACH(I)
      RETURN
C
      END

      DOUBLE PRECISION FUNCTION D1MACH(I)
C***BEGIN PROLOGUE  D1MACH
C***DATE WRITTEN   750101   (YYMMDD)
C***REVISION DATE  831014   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C           HALL, A. D., (BELL LABS)
C           SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns double precision machine dependent constants
C***DESCRIPTION
C    From the book, "Numerical Methods and Software" by
C              D. Kahaner, C. Moler, S. Nash
C              Prentice Hall, 1988
C
C
C    D1MACH can be used to obtain machine-dependent parameters
C    for the local machine environment.  It is a function
C    subprogram with one (input) argument, and can be called
C    as follows, for example
```

```
C
C         D = D1MACH(I)
C
C    where I=1,...,5.  The (output) value of D above is
C    determined by the (input) value of I.  The results for
C    various values of I are discussed below.
C
C  Double-precision machine constants
C  D1MACH( 1) = B**(EMIN-1), the smallest positive magnitude.
C  D1MACH( 2) = B**EMAX*(1 - B**(-T)), the largest magnitude.
C  D1MACH( 3) = B**(-T), the smallest relative spacing.
C  D1MACH( 4) = B**(1-T), the largest relative spacing.
C  D1MACH( 5) = LOG10(B)
C***REFERENCES  FOX P.A., HALL A.D., SCHRYER N.L.,*FRAMEWORK FOR A
C               PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHEMATICAL
C               SOFTWARE, VOL. 4, NO. 2, JUNE 1978, PP. 177-188.
C***ROUTINES CALLED  XERROR
C***END PROLOGUE  D1MACH
C
      INTEGER SMALL(4)
       INTEGER LARGE(4)
      INTEGER RIGHT(4)
      INTEGER DIVER(4)
      INTEGER LOG10(4)
C
   DOUBLE PRECISION DMACH(5)
C
      EQUIVALENCE (DMACH(1),SMALL(1))
      EQUIVALENCE (DMACH(2),LARGE(1))
```

```
      EQUIVALENCE (DMACH(3),RIGHT(1))
      EQUIVALENCE (DMACH(4),DIVER(1))
      EQUIVALENCE (DMACH(5),LOG10(1))
C
C
C     MACHINE CONSTANTS FOR THE CDC CYBER 170 SERIES (FTN5).
C
C      DATA SMALL(1) / O"00604000000000000000" /
C      DATA SMALL(2) / O"00000000000000000000" /
C
C      DATA LARGE(1) / O"37767777777777777777" /
C      DATA LARGE(2) / O"37167777777777777777" /
C
C      DATA RIGHT(1) / O"15604000000000000000" /
C      DATA RIGHT(2) / O"15000000000000000000" /
C
C      DATA DIVER(1) / O"15614000000000000000" /
C      DATA DIVER(2) / O"15010000000000000000" /
C
C      DATA LOG10(1) / O"17164642023241175717" /
C      DATA LOG10(2) / O"16367571421742254654" /
C
C     MACHINE CONSTANTS FOR THE CDC CYBER 200 SERIES
C
C     DATA SMALL(1) / X'9000400000000000' /
C     DATA SMALL(2) / X'8FD1000000000000' /
C
C     DATA LARGE(1) / X'6FFF7FFFFFFFFFFF' /
C     DATA LARGE(2) / X'6FD07FFFFFFFFFFF' /
```

```
C
C    DATA RIGHT(1) / X'FF74400000000000' /
C    DATA RIGHT(2) / X'FF45000000000000' /
C
C    DATA DIVER(1) / X'FF75400000000000' /
C    DATA DIVER(2) / X'FF46000000000000' /
C
C    DATA LOG10(1) / X'FFD04D104D427DE7' /
C    DATA LOG10(2) / X'FFA17DE623E2566A' /
C
C
C    MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C    DATA SMALL(1) / 00564000000000000000B /
C    DATA SMALL(2) / 00000000000000000000B /
C
C    DATA LARGE(1) / 37757777777777777777B /
C    DATA LARGE(2) / 37157777777777777777B /
C
C    DATA RIGHT(1) / 15624000000000000000B /
C    DATA RIGHT(2) / 00000000000000000000B /
C
C    DATA DIVER(1) / 15634000000000000000B /
C    DATA DIVER(2) / 00000000000000000000B /
C
C    DATA LOG10(1) / 17164642023241175717B /
C    DATA LOG10(2) / 16367571421742254654B /
C
C    MACHINE CONSTANTS FOR THE CRAY 1
```

```
C
C     DATA SMALL(1) / 20135400000000000000B /
C     DATA SMALL(2) / 00000000000000000000B /
C
C     DATA LARGE(1) / 57776777777777777777B /
C     DATA LARGE(2) / 00000777777777777774B /
C
C     DATA RIGHT(1) / 37643400000000000000B /
C     DATA RIGHT(2) / 00000000000000000000B /
C
C     DATA DIVER(1) / 37644400000000000000B /
C     DATA DIVER(2) / 00000000000000000000B /
C
C     DATA LOG10(1) / 377774642023241175717B /
C     DATA LOG10(2) / 000007571421742254654B /
C
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86, AND
C     THE PERKIN ELMER (INTERDATA) 7/32.
C
C     DATA SMALL(1),SMALL(2) / Z00100000, Z00000000 /
C     DATA LARGE(1),LARGE(2) / Z7FFFFFFF, ZFFFFFFFF /
C     DATA RIGHT(1),RIGHT(2) / Z33100000, Z00000000 /
C     DATA DIVER(1),DIVER(2) / Z34100000, Z00000000 /
C     DATA LOG10(1),LOG10(2) / Z41134413, Z509F79FF /
C
C     MACHINE CONSTATNS FOR THE IBM PC FAMILY (D. KAHANER NBS)
C
```

```
      DATA DMACH/2.23D-308,1.79D+308,1.11D-16,2.22D-16,
     *  0.301029995663981195D0/
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR).
C
C     DATA SMALL(1),SMALL(2) / "033400000000, "000000000000 /
C     DATA LARGE(1),LARGE(2) / "377777777777, "344777777777 /
C     DATA RIGHT(1),RIGHT(2) / "113400000000, "000000000000 /
C     DATA DIVER(1),DIVER(2) / "114400000000, "000000000000 /
C     DATA LOG10(1),LOG10(2) / "177464202324, "144117571776 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR).
C
C     DATA SMALL(1),SMALL(2) / "000400000000, "000000000000 /
C     DATA LARGE(1),LARGE(2) / "377777777777, "377777777777 /
C     DATA RIGHT(1),RIGHT(2) / "103400000000, "000000000000 /
C     DATA DIVER(1),DIVER(2) / "104400000000, "000000000000 /
C     DATA LOG10(1),LOG10(2) / "177464202324, "476747767461 /
C
C
C     MACHINE CONSTANTS FOR THE SUN-3 (INCLUDES THOSE WITH 68881 CHIP,
C       OR WITH FPA BOARD. ALSO INCLUDES SUN-2 WITH SKY BOARD. MAY ALSO
C       WORK WITH SOFTWARE FLOATING POINT ON EITHER SYSTEM.)
C
C      DATA SMALL(1),SMALL(2) / X'00100000', X'00000000' /
C      DATA LARGE(1),LARGE(2) / X'7FEFFFFF', X'FFFFFFFF' /
C      DATA RIGHT(1),RIGHT(2) / X'3CA00000', X'00000000' /
C      DATA DIVER(1),DIVER(2) / X'3CB00000', X'00000000' /
C      DATA LOG10(1),LOG10(2) / X'3FD34413', X'509F79FF' /
```

```
C
C
C    MACHINE CONSTANTS FOR VAX 11/780
C    (EXPRESSED IN INTEGER AND HEXADECIMAL)
C   *** THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS***
C
C    DATA SMALL(1), SMALL(2) /       128,          0 /
C    DATA LARGE(1), LARGE(2) /    -32769,         -1 /
C    DATA RIGHT(1), RIGHT(2) /      9344,          0 /
C    DATA DIVER(1), DIVER(2) /      9472,          0 /
C    DATA LOG10(1), LOG10(2) /  546979738,  -805796613 /
C
C   ***THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSYEMS***
C    DATA SMALL(1), SMALL(2) / Z00000080, Z00000000 /
C    DATA LARGE(1), LARGE(2) / ZFFFF7FFF, ZFFFFFFFF /
C    DATA RIGHT(1), RIGHT(2) / Z00002480, Z00000000 /
C    DATA DIVER(1), DIVER(2) / Z00002500, Z00000000 /
C    DATA LOG10(1), LOG10(2) / Z209A3F9A, ZCFF884FB /
C
C  MACHINE CONSTANTS FOR VAX 11/780 (G-FLOATING)
C    (EXPRESSED IN INTEGER AND HEXADECIMAL)
C   *** THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS***
C
C    DATA SMALL(1), SMALL(2) /        16,          0 /
C    DATA LARGE(1), LARGE(2) /    -32769,         -1 /
C    DATA RIGHT(1), RIGHT(2) /     15552,          0 /
C    DATA DIVER(1), DIVER(2) /     15568,          0 /
C    DATA LOG10(1), LOG10(2) /  1142112243, 2046775455 /
C
```

```
C    ***THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSYEMS***
C     DATA SMALL(1), SMALL(2) / Z00000010, Z00000000 /
C     DATA LARGE(1), LARGE(2) / ZFFFF7FFF, ZFFFFFFFF /
C     DATA RIGHT(1), RIGHT(2) / Z00003CC0, Z00000000 /
C     DATA DIVER(1), DIVER(2) / Z00003CD0, Z00000000 /
C     DATA LOG10(1), LOG10(2) / Z44133FF3, Z79FF509F /
C
C
C***FIRST EXECUTABLE STATEMENT  D1MACH
      IF (I .LT. 1  .OR.  I .GT. 5)
      1   CALL XERROR( 'D1MACH -- I OUT OF BOUNDS',25,1,2)
C
      D1MACH = DMACH(I)
      RETURN
C
      END

       INTEGER FUNCTION I1MACH(I)
C***BEGIN PROLOGUE  I1MACH
C***DATE WRITTEN   750101   (YYMMDD)
C***REVISION DATE  840405   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C          HALL, A. D., (BELL LABS)
C          SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns integer machine dependent constants
C***DESCRIPTION
C
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C   These machine constant routines must be activated for
C   a particular environment.
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     I1MACH can be used to obtain machine-dependent parameters
C     for the local machine environment.  It is a function
C     subroutine with one (input) argument, and can be called
C     as follows, for example
C
C          K = I1MACH(I)
C
C     where I=1,...,16.  The (output) value of K above is
C     determined by the (input) value of I.  The results for
C     various values of I are discussed below.
C
C  I/O unit numbers.
C    I1MACH( 1) = the standard input unit.
C    I1MACH( 2) = the standard output unit.
C    I1MACH( 3) = the standard punch unit.
C    I1MACH( 4) = the standard error message unit.
C
C  Words.
C    I1MACH( 5) = the number of bits per integer storage unit.
C    I1MACH( 6) = the number of characters per integer storage unit.
C
C  Integers.
C    assume integers are represented in the S-digit, base-A form
C
C               sign ( X(S-1)*A**(S-1) + ... + X(1)*A + X(0) )
```

```
C
C            where 0 .LE. X(I) .LT. A for I=0,...,S-1.
C   I1MACH( 7) = A, the base.
C   I1MACH( 8) = S, the number of base-A digits.
C   I1MACH( 9) = A**S - 1, the largest magnitude.
C
C  Floating-Point Numbers.
C   Assume floating-point numbers are represented in the T-digit,
C   base-B form
C            sign (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C            where 0 .LE. X(I) .LT. B for I=1,...,T,
C            0 .LT. X(1), and EMIN .LE. E .LE. EMAX.
C   I1MACH(10) = B, the base.
C
C  Single-Precision
C   I1MACH(11) = T, the number of base-B digits.
C   I1MACH(12) = EMIN, the smallest exponent E.
C   I1MACH(13) = EMAX, the largest exponent E.
C
C  Double-Precision
C   I1MACH(14) = T, the number of base-B digits.
C   I1MACH(15) = EMIN, the smallest exponent E.
C   I1MACH(16) = EMAX, the largest exponent E.
C
C  To alter this function for a particular environment,
C  the desired set of DATA statements should be activated by
C  removing the C from column 1.  Also, the values of
C  I1MACH(1) - I1MACH(4) should be checked for consistency
```

```
C  with the local operating system.
C***REFERENCES  FOX P.A., HALL A.D., SCHRYER N.L.,*FRAMEWORK FOR A
C                 PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHEMATICAL
C                 SOFTWARE, VOL. 4, NO. 2, JUNE 1978, PP. 177-188.
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  I1MACH
C
       INTEGER IMACH(16),OUTPUT
      EQUIVALENCE (IMACH(4),OUTPUT)
C
C
C    MACHINE CONSTANTS FOR THE CDC CYBER 170 SERIES (FTN5).
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   60 /
C     DATA IMACH( 6) /   10 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   48 /
C     DATA IMACH( 9) / O"00007777777777777777" /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   48 /
C     DATA IMACH(12) / -974 /
C     DATA IMACH(13) / 1070 /
C     DATA IMACH(14) /   96 /
C     DATA IMACH(15) / -927 /
C     DATA IMACH(16) / 1070 /
```

```
C
C    MACHINE CONSTANTS FOR THE CDC CYBER 200 SERIES
C
C    DATA IMACH( 1) /      5 /
C    DATA IMACH( 2) /      6 /
C    DATA IMACH( 3) /      7 /
C    DATA IMACH( 4) /      6 /
C    DATA IMACH( 5) /     64 /
C    DATA IMACH( 6) /      8 /
C    DATA IMACH( 7) /      2 /
C    DATA IMACH( 8) /     47 /
C    DATA IMACH( 9) / X'00007FFFFFFFFFFF' /
C    DATA IMACH(10) /      2 /
C    DATA IMACH(11) /     47 /
C    DATA IMACH(12) / -28625 /
C    DATA IMACH(13) /  28718 /
C    DATA IMACH(14) /     94 /
C    DATA IMACH(15) / -28625 /
C    DATA IMACH(16) /  28718 /
C
C
C    MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C    DATA IMACH( 1) /    5 /
C    DATA IMACH( 2) /    6 /
C    DATA IMACH( 3) /    7 /
C    DATA IMACH( 4) /6LOUTPUT/
C    DATA IMACH( 5) /   60 /
C    DATA IMACH( 6) /   10 /
```

```
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   48 /
C     DATA IMACH( 9) / 00007777777777777777B /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   47 /
C     DATA IMACH(12) / -929 /
C     DATA IMACH(13) / 1070 /
C     DATA IMACH(14) /   94 /
C     DATA IMACH(15) / -929 /
C     DATA IMACH(16) / 1069 /
C
C     MACHINE CONSTANTS FOR THE CRAY 1
C
C     DATA IMACH( 1) /   100 /
C     DATA IMACH( 2) /   101 /
C     DATA IMACH( 3) /   102 /
C     DATA IMACH( 4) /   101 /
C     DATA IMACH( 5) /    64 /
C     DATA IMACH( 6) /     8 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    63 /
C     DATA IMACH( 9) /  777777777777777777777B /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    47 /
C     DATA IMACH(12) / -8189 /
C     DATA IMACH(13) /  8190 /
C     DATA IMACH(14) /    94 /
C     DATA IMACH(15) / -8099 /
C     DATA IMACH(16) /  8190 /
```

```
C
C
C    MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C    THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86, AND
C    THE PERKIN ELMER (INTERDATA) 7/32.
C
C    DATA IMACH( 1) /   5 /
C    DATA IMACH( 2) /   6 /
C    DATA IMACH( 3) /   7 /
C    DATA IMACH( 4) /   6 /
C    DATA IMACH( 5) /  32 /
C    DATA IMACH( 6) /   4 /
C    DATA IMACH( 7) /  16 /
C    DATA IMACH( 8) /  31 /
C    DATA IMACH( 9) / Z7FFFFFFF /
C    DATA IMACH(10) /  16 /
C    DATA IMACH(11) /   6 /
C    DATA IMACH(12) / -64 /
C    DATA IMACH(13) /  63 /
C    DATA IMACH(14) /  14 /
C    DATA IMACH(15) / -64 /
C    DATA IMACH(16) /  63 /
C
C    MACHINE CONSTANTS FOR THE IBM PC FAMILY (D. KAHANER NBS)
C
      DATA IMACH/5,6,0,6,32,4,2,31,2147483647,2,24,
     * -125,127,53,-1021,1023/
C            NOTE! I1MACH(3) IS NOT WELL DEFINED AND IS SET TO ZERO.
C
```

```
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR).
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / "377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   54 /
C     DATA IMACH(15) / -101 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR).
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
```

```
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / "377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   62 /
C     DATA IMACH(15) / -128 /
C     DATA IMACH(16) /  127 /
C
C
C     MACHINE CONSTANTS FOR THE SUN-3 (INCLUDES THOSE WITH 68881 CHIP,
C       OR WITH FPA BOARD. ALSO INCLUDES SUN-2 WITH SKY BOARD. MAY ALSO
C       WORK WITH SOFTWARE FLOATING POINT ON EITHER SYSTEM.)
C
C      DATA IMACH( 1) /    5 /
C      DATA IMACH( 2) /    6 /
C      DATA IMACH( 3) /    6 /
C      DATA IMACH( 4) /    0 /
C      DATA IMACH( 5) /   32 /
C      DATA IMACH( 6) /    4 /
C      DATA IMACH( 7) /    2 /
C      DATA IMACH( 8) /   31 /
C      DATA IMACH( 9) / 2147483647 /
C      DATA IMACH(10) /    2 /
C      DATA IMACH(11) /   24 /
C      DATA IMACH(12) / -125 /
C      DATA IMACH(13) /  128 /
C      DATA IMACH(14) /   53 /
```

```
C     DATA IMACH(15) / -1021 /
C     DATA IMACH(16) /  1024 /
C

C    MACHINE CONSTANTS FOR THE VAX 11/780
C
C    DATA IMACH(1) /    5 /
C    DATA IMACH(2) /    6 /
C    DATA IMACH(3) /    5 /
C    DATA IMACH(4) /    6 /
C    DATA IMACH(5) /   32 /
C    DATA IMACH(6) /    4 /
C    DATA IMACH(7) /    2 /
C    DATA IMACH(8) /   31 /
C    DATA IMACH(9) /2147483647 /
C    DATA IMACH(10)/    2 /
C    DATA IMACH(11)/   24 /
C    DATA IMACH(12)/ -127 /
C    DATA IMACH(13)/  127 /
C    DATA IMACH(14)/   56 /
C    DATA IMACH(15)/ -127 /
C    DATA IMACH(16)/  127 /
C
C***FIRST EXECUTABLE STATEMENT  I1MACH
      IF (I .LT. 1  .OR.  I .GT. 16)
      1   CALL XERROR ( 'I1MACH -- I OUT OF BOUNDS',25,1,2)
C
      I1MACH=IMACH(I)
      RETURN
```

```
C
      END


      SUBROUTINE XERROR(MESSG,NMESSG,NERR,LEVEL)
C***BEGIN PROLOGUE  XERROR
C***DATE WRITTEN   790801   (YYMMDD)
C***REVISION DATE  870930   (YYMMDD)
C***CATEGORY NO.  R3C
C***KEYWORDS  ERROR,XERROR PACKAGE
C***AUTHOR  JONES, R. E., (SNLA)
C***PURPOSE  Processes an error (diagnostic) message.
C***DESCRIPTION
C   From the book "Numerical Methods and Software"
C      by  D. Kahaner, C. Moler, S. Nash
C          Prentice Hall 1988
C    Abstract
C       XERROR processes a diagnostic message. It is a stub routine
C       written for the book above. Actually, XERROR is a sophisticated
C       error handling package with many options, and is described
C       in the reference below. Our version has the same calling sequence
C       but only prints an error message and either returns (if the
C       input value of ABS(LEVEL) is less than 2) or stops (if the
C       input value of ABS(LEVEL) equals 2).
C
C    Description of Parameters
C     --Input--
C       MESSG - the Hollerith message to be processed.
C       NMESSG- the actual number of characters in MESSG.
```

```
C           (this is ignored in this stub routine)
C      NERR  - the error number associated with this message.
C           NERR must not be zero.
C           (this is ignored in this stub routine)
C      LEVEL - error category.
C           =2 means this is an unconditionally fatal error.
C           =1 means this is a recoverable error.  (I.e., it is
C             non-fatal if XSETF has been appropriately called.)
C           =0 means this is a warning message only.
C           =-1 means this is a warning message which is to be
C             printed at most once, regardless of how many
C             times this call is executed.
C            (in this stub routine
C                LEVEL=2 causes a message to be printed and then a
C                            stop.
C                LEVEL<2 causes a message to be printed and then a
C                            return.
C
C    Examples
C      CALL XERROR('SMOOTH -- NUM WAS ZERO.',23,1,2)
C      CALL XERROR('INTEG  -- LESS THAN FULL ACCURACY ACHIEVED.',
C             43,2,1)
C      CALL XERROR('ROOTER -- ACTUAL ZERO OF F FOUND BEFORE INTERVAL F
C    1ULLY COLLAPSED.',65,3,0)
C      CALL XERROR('EXP    -- UNDERFLOWS BEING SET TO ZERO.',39,1,-1)
C
C***REFERENCES  JONES R.E., KAHANER D.K., "XERROR, THE SLATEC ERROR-
C            HANDLING PACKAGE", SAND82-0800, SANDIA LABORATORIES,
C            1982.
```

```
C***ROUTINES CALLED  XERRWV
C***END PROLOGUE  XERROR
      CHARACTER*(*) MESSG
C***FIRST EXECUTABLE STATEMENT  XERROR
         CALL XERRWV(MESSG,NMESSG,NERR,LEVEL,0,0,0,0,0.,0.)
         RETURN
         END

         SUBROUTINE XERRWV(MESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2)
C***BEGIN PROLOGUE  XERRWV
C***DATE WRITTEN   800319   (YYMMDD)
C***REVISION DATE  870930   (YYMMDD)
C***CATEGORY NO.  R3C
C***KEYWORDS  ERROR,XERROR PACKAGE
C***AUTHOR  JONES, R. E., (SNLA)
C***PURPOSE  Processes error message allowing 2 integer and two real
C            values to be included in the message.
C***DESCRIPTION
C   From the book "Numerical Methods and Software"
C      by  D. Kahaner, C. Moler, S. Nash
C          Prentice Hall 1988
C    Abstract
C       XERRWV prints a diagnostic error message.
C       In addition, up to two integer values and two real
C       values may be printed along with the message.
C       A stub routine for the book above. The actual XERRWV is described
C       in the reference below and contains many other options.
C
C    Description of Parameters
```

```
C    --Input--
C      MESSG - the Hollerith message to be processed.
C      NMESSG- the actual number of characters in MESSG.
C              (ignored in this stub)
C      NERR  - the error number associated with this message.
C              NERR must not be zero.
C              (ignored in this stub)
C      LEVEL - error category.
C              =2 means this is an unconditionally fatal error.
C              =1 means this is a recoverable error.  (I.e., it is
C                non-fatal if XSETF has been appropriately called.)
C              =0 means this is a warning message only.
C              =-1 means this is a warning message which is to be
C                printed at most once, regardless of how many
C                times this call is executed.
C               (in this stub LEVEL=2 causes an error message to be
C                             printed followed by a stop,
C                        LEVEL<2 causes an error message to be
C                             printed followed by a return.)
C      NI    - number of integer values to be printed. (0 to 2)
C      I1    - first integer value.
C      I2    - second integer value.
C      NR    - number of real values to be printed. (0 to 2)
C      R1    - first real value.
C      R2    - second real value.
C
C    Examples
C      CALL XERRWV('SMOOTH -- NUM (=I1) WAS ZERO.',29,1,2,
C   1   1,NUM,0,0,0.,0.)
```

```
C       CALL XERRWV('QUADXY -- REQUESTED ERROR (R1) LESS THAN MINIMUM (
C    1R2).,54,77,1,0,0,0,2,ERRREQ,ERRMIN)
C
C***REFERENCES  JONES R.E., KAHANER D.K., "XERROR, THE SLATEC ERROR-
C                 HANDLING PACKAGE", SAND82-0800, SANDIA LABORATORIES,
C                 1982.
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  XERRWV
      CHARACTER*(*) MESSG
C***FIRST EXECUTABLE STATEMENT  XERRWV
      WRITE(*,*) MESSG
      IF(NI.EQ.2)THEN
      WRITE(*,*) I1,I2
      ELSEIF(NI.EQ.1) THEN
      WRITE(*,*) I1
      ENDIF
      IF(NR.EQ.2) THEN
      WRITE(*,*) R1,R2
      ELSEIF(NR.EQ.1) THEN
      WRITE(*,*) R1
      ENDIF
      IF(ABS(LEVEL).LT.2)RETURN
      STOP
      END
```

7. Subroutine BATCH.FOR

```
C        234567  Drying Batch, file name : BATCH.FOR
C
C        This program was written for simulation the drying performance of the
C        drying batch developed at Royal Chitlada project.
C        The programme is developed based on the mathematical model for which the
C        the moisture balance between drying fluid air and product.
C
C        Version : 18 Mar 2004
C
C        Importants parameters used in the main programme :
C
C        D........Average thickness of air gab between tray, [m]
C        DELT.....Time step in finit difference with implicit approach, [s]
C        FLOW.....Total mass flow rate of the collector, [kg/s]
C        G........Specific mass flow rate per unit area perpendicular to the flow, [kg/s-m2]
C        HEIGHT...Height of dryer,[m]
C        HM.......Humidity ratio of drying air, [kg/kg]
C        HIN......Inlet humidity ratio of fluid air to the dryer, [kg/kg]
C        HMO......Outlet humidity ratio of fluid air from the dryer , [kg/kg]
C        HMIX.....Humidity ratio of air in mixing box being inlet humidity in next time step, [kg/kg]
C        LENGTH...Length of dryer per one box, [m]
C        MC.......Means moisture contents of banana in the dryer,[db]
C        NBAT.....Number of tray per one box
C        NSEC.....Number of divided section along flow direction
```

```
C       XMO......Moisture content of bananas at t=0 s,[db]
C       XMC......Moisture content of bananas at any time and any distant,[db]
C       XMT......Moisture content of bananas at next time step ,[db]
C       WS.......Solid weight of banana per one box, [kg]
C       R........Recycle factor is ratio between recycle flow rate and total flow rate, [decimal]
C       RH.......Relative humidity of drying air, [decimal]
C       RHI......Inlet relative humidity of fluid air, [decimal]
C       RHOSP....Area density of dried banana, [kg/m**2]
C       RHP......Relative humidity of drying air, [%]
C       TDO......Outlet temperature of fluid air from the dryer, [C]
C       TF.......Temperature of drying air, [C]
C       TFI......Inlet temperature of fluid air, [C]
C       TFO......Outlet temperature of fluid air setted at 60 degree of Celcius, [C]
C       WF.......Final weight of banana per one box,[kg]
C       WI.......Initial weight of banana per one box,[kg]
C       WIDTH....Width of dryer per one box,[m]
C       WS.......Weight of solid or dried bananas,[kg]
C       WT.......Total weight of banana,[kg]
C
        SUBROUTINE BATCH(TDO,TDI,TA,FLOW,HIN,HMO,XMC,MC,
        1       WIDTH,HEIGHT,LENGTH,NBAT,WT)
        PARAMETER (LDA=240)
        COMMON /TEMP/ T
        DIMENSION A(LDA,LDA),B(LDA),WORK(LDA),T(LDA)
        REAL    LENGTH,FLOW,XMT,XMC,TFC,RHP,HIN,R,TF,RH,HM,MC,RHIN,
                CONH,CONEX
        INTEGER ITIME,NSEC
        DATA DENA/1.000/,CA/1002./,CW/4186./,CV/1890./,HFG/2600./
C
```

```
C       Assumption of parameters which are independent of time.
C
        NSEC = 5
        DELX = LENGTH/NSEC
        D = HEIGHT/NBAT
        WI = WT/2.
        XMO = 3.5
        WS = WI/(1+XMO)
        RHOSP = WI/(1+XMO)/(LENGTH*WIDTH)/NBAT
        DELT= 600.0
        G=FLOW/(HEIGHT*WIDTH)/NBAT
C
C        Define boundary conditions at x = 0 of distant of tray.
C
        XM = 0.0
        RHI = RHDBHA(TDI+273.15,HIN)
        HM = HIN
        RH = RHI
        TF = TDI
        ITASK=1
        NT=2*NSEC
C
C       Calculation of parameters which are independent of time.
C
        LENGTH=DELX*NSEC
        AREA=LENGTH*WIDTH
        G=FLOW/(D*WIDTH)
        CONH=RHOSP*DELX/(G*D)/DELT
        HCPF = 3.0
```

```
C
C        Increase time to the next time step, dt, after calculating temperature of all nodes.
C
         DO 15 I=1,NT
         T(I)=TDI
   15    CONTINUE
         XMS = 0.0
C     Initialisation of the matrix elements
C
         DO 20 I=1,NT
         B(I)=0.0
         DO 20 J=1,NT
         A(I,J)=0.0
   20    CONTINUE
C
C       Begin the calculation of the temperature of all nodes at t+dt,
C       starting with the computation of the matrix element
C
         DO 90 N=1,NSEC
C
C     Transfer new temperature and humidity as those at next element.
C
                 TFC=TF
                 RHP = 100.*RH
C
C         Calculate moisture content and humidity at any distant with subroutine layer
C
         CALL LAYER (XMT,XMC,TFC,RH,DELT,XMO)
         HM = HM+CONH*(XMC-XMT)
```

```
      MC = XMT
      XMS = XMS+XMT
      XM  = XMS/N
C     write(*,*) XMT,XMC,XM
C
C      Calculate variable term for calculating matrix element
C
      CPP  = 2890.-30.3*(T(2+2*(N-1)))
      CONX = 0.25*D*G*(CA+CV*HM)/DELX
      CONP = RHOSP*(CPP+CW*XMC)/DELT
      DELM = RHOSP*(XMT-XMC)/DELT
c     write(*,*) t(n),cpp,ca,cv,cw,hm,rhosp
C
C     Calculate the matrix elements
C
C     The matrix element of Tf (normal elements)
C
C     A(1+2*(N-1),1+2*(N-1))= (HCPF+HCCF+HCBF)
      A(2+2*(N-1),1+2*(N-1))= (CV*DELM-HCPF)
C
C     The matrix element of Tp
C
      A(1+2*(N-1),2+2*(N-1))= (CV*DELM-HCPF)
      A(2+2*(N-1),2+2*(N-1))= (CONP+(CV-CW)*DELM+HCPF)
C
C     Right hand side vector B, except for the term from Tf eq.
C
      B(2+2*(N-1))=CONP*T(2+2*(N-1))-HFG*DELM
C    Matrix elements of Tfx-dx,Tf,Tfx+dx and B
```

```
C
      IF(N.EQ.1) THEN
      A(1+2*(N-1),3+2*(N-1))=CONX
      A(1+2*(N-1),1+2*(N-1))=HCPF-DELM*CV
C     B(2+4*(N-1))=CONX*(TDI-T(6+4*(N-1))+TFI)
       B(1+2*(N-1))=CONX*(TDI-T(3+2*(N-1))+TDI)
      1
C
      ELSE
      IF(N.GT.1.AND.N.LT.NSEC) THEN
      A(1+2*(N-1),2*(N-1)-1)=-CONX
C     A(2+4*(N-1),2+4*(N-1))= HCPF+HCCF+HCBF+CONT
      A(1+2*(N-1),3+2*(N-1))= CONX
      B(1+2*(N-1))=CONX*(T(2*(N-1)-1)-T(3+2*(N-1)))
C      write(*,*) CONX,T(2*(N-1)-1),T(3+2*(N-1)),B(1+2*(N-1))
              A(1+2*(N-1),1+2*(N-1))= HCPF-(DELM*CV)
      ELSE
      IF(N.EQ.NSEC) THEN
      A(1+2*(N-1),1+2*(N-1))= 2.*CONX+HCPF+HCPF-DELM*CV
      A(1+2*(N-1),2*(N-1)-1)=-2.*CONX
      B(1+2*(N-1))=-2.*CONX*(T(1+2*(N-1))-T(2*(N-1)-1))
      1
       ELSE
       WRITE(*,30)
  30   FORMAT(' Some calculation of matrix elements are missing')
      ENDIF
      ENDIF
      ENDIF
C
```

```
C      Protect relative humidity of air is greater than 1.0 due to over increasing of humidity
C
       RH = RHDBHA(TF+273.15,HM)
       IF (RH.EQ.0.99) THEN
       HM = HADBRH(TFC+273.,RH)
       ELSE
       HM = HM-RHOSP*DELM/D/G/DELT*DELX
       END IF

       RH = RHDBHA(TFC+273.15,HM)
       RHP = 100.*RH
C      write(*,*) RHP,XM
  90   CONTINUE
C
C      End of the computation of the matrix element
C
C      Use a subroutine to solve the matrix equation [A].T=B
C      NT in the argument, original is N
C
       CALL SGEFS(A,LDA,NT,B,ITASK,IND,WORK,IWORK,RCOND)
 110   CONTINUE
       DO 118 N=1,NSEC
       TDO = T(1+2*(NSEC-1))
 118   CONTINUE
       HMO = HM
C      write(*,*) RHP,TDO,TA,MC
       END SUBROUTINE
```

8. Subroutine LAYER

```
C    FILE NAME : TLYM.FOR
C    VERSION : 18 SEP 03
C    PURPOSE : COMPUTE MOISTURE CONTENT OF  THIN LAYER BANANAS USED THE
C                      SEMI EMPIRICAL MODEL FOR NORMAL DRYING
C    FUNCTION USED : 1. EMC, for computing equilibrium moisture content
C                               2. PSDB, for computing saturated vapor pressure
C                                  evaluated at the paddy temperature
C    PARAMETERS USED :
C     TP.....Temperature of bananas ,K
C     XMC....Actual moisture content of bananas , kg/kg,db
C     XMO....Initial moisture content of bananas, kg/kg,db
C     XME....Equilibrium moisture content of bananas, kg/kg,db
C     TI.....Equivalent time,time at which the thin layer equation
C              gives the value of bananas moisture to be equal to that
C              calculate using actual value of RH,T and XMR , min
C     DELT...Time step of for calculation of the moisture content, s
C     DELM...Difference between the initial and equilibrium moisture content, kg/kg, db
C     RH.....Relative humidity value transferred from main program ,decimal
C
        SUBROUTINE LAYER(XMT,XMC,TFC,RHP,DELT,XMO)
C       Compute equilibrium moisture content (RH and TH come from both t
C       t+delt
C       write(*,12)
C  12   format(' Beginning of the LAYER ')
```

```
C
      AW  = RHP/100.
      EMC = 74.66023-(1.144253*TFC)+(37.07224*AW)
     1        +(0.001166*TFC**2.)+(51.55674*AW**2.)
      XME = EMC/100.0
C     write(*,*) EMC
C
C     Set initial moisture content for thin layer calculation.
C
      IF ((XMO-XMC).LT.0.) THEN
      DELM = XMC-XME
      ELSE
      DELM = XMO-XME
      ENDIF
C     WRITE(*,*) XMC
C
C     Compute moist ratio at time t.
C
      IF(DELM.NE.0.0) THEN
      XMR = (XMC-XME)/DELM
      ELSE
      write(*,16)
 16   format(' XMR becomes infinity')
      ENDIF
C     write(*,17) TFC,RHP,EMC
 17   format(' TF= ',F8.2,' RH= ',F8.2,' EMC= ',f8.3)
C
C       Check desorption or absorption
C
```

```
      IF (XMC.GT.XME) THEN
C       Use semiempirical model for continued drying
C

C     A = 1.503574-0.013267*TFC-0.505455*AW
C     1   +0.000094*TFC**2-2.141736*AW**2
C     B = 0.1814-0.006347*TFC+0.193*AW
C     1   +0.000081*TFC**2-0.797778*AW**2
      A = 1.503574-0.013267*TFC-0.505455*AW
      1   +0.000094*TFC**2
      B = 0.1814-0.006347*TFC+0.193*AW
       1   +0.000081*TFC**2
C     A = 1.503574-0.013267*TFC-0.505455*AW
C     1   +0.000094*TFC**2-2.141736*AW**2
      B = 0.1814-0.006347*TFC+0.193*AW
      1   +0.000081*TFC**2-0.797778*AW**2
      IF(XMR.GT.0.0.AND.B.GT.0.0) THEN
      TI=(-ALOG(XMR/A)/B)+DELT/3600.
      ELSE
      ENDIF
C     WRITE(*,*) A,B
      XMT=DELM*A*EXP(-B*TI)+XME
C
C     Assume water absorption of product isn't occurred
C      although Me is great than M
C
      ELSE
      XMT=XMC
      ENDIF
```

```
C         write(*,20) XMC,XME,XMT,XMO
C   20    format('XMC =',F10.4,' XME =',F10.4,' XMT =',F10.4,' XMO =',F10.4)
          END
```

9. Subfunctions

```
C    FILE NAME : CHART.FOR
C    VERSION : 4 MAR 92
C    FOR UNIX work station
C    MODIFIED FORM ORIGINAL PAKAGE USED BY BAKKER-ARKEMA ET AL. MSU
C    PURPOSE :TO EVALUATE  MOIST AIR PROPERTIES
C                  INPUT AND OUTPUT PARAMETERS ARE IN SI UNIT
C-------------------------------------------------------------------
C     PURPOSE:  To compute relative humidity from dry bulk
C                    temperature and humidity ratio, decimal
C     FUNCTION USED.. 1) PSDB(DB)
C                          2) PVHA(HA)
C     DB.....Dry bulk temperature ,K
C     HA.....Humidity ratio, kg vapor/kg dry air
C     PS.....Saturated vapor pressure, Pa
C     PV.....Vapor pressure, Pa
C     PATM...Atmospheric pressure, Pa
          FUNCTION RHDBHA(DB,HA)
          PATM = 101.325E03
          PS = PSDB(DB)
          PV = PVHA(HA)
          IF(PV.LT.PS) THEN
          RHDBHA = PV/PS
          ELSE
          RHDBHA=.99
```

```
C         write(*,10)
 10       format(' Air is saturated')
          ENDIF
          RETURN
          END
C      PURPOSE : To compute vapor pressure from humidity ratio of moist air
C      PVHA...Vapor pressure, Pa
C      HA.....Humidity ratio, kg vapor/kg dry air
C      PATM...Atmospheric pressure, Pa
C
          FUNCTION PVHA(HA)
          PATM = 101.325E03
          PVHA = HA*PATM/(0.6219+HA)
          RETURN
          END
C
C      PURPOSE : To compute saturated vapor pressure from dry bulk temperature
C      PSDB...Saturate vapor pressure, Pa
C      DB.....Dry bulk temperature, K
C
          FUNCTION PSDB(DB)
          R =  0.3206182232E04
          A = -0.274055258361426E05
          B =  0.541896076328951E02
          C = -0.451370384112655E-1
          D =  0.215321191636354E-4
          E = -0.462026656819982E-8
          F =  0.2416127209874E01
          G =  0.121546516706055E-2
```

```
      Q =  6.894757E03
      IF ((DB-273.16).LT.0.) THEN
      IF(DB.GT.0.0) THEN
      PSDB = Q*(EXP(23.3924-11286.6489/(1.8*DB)-0.46057*ALOG(1.8*DB)))
      ELSE
      write(*,10)
 10   format(' Argument of LOG ,DB, in PSDB is zero or negative' )
      ENDIF
      ELSE
      PSDB = Q*(R*EXP((A+1.8*DB*(B+1.8*DB*(C+1.8*DB*
      &     (D+1.8*DB*E))))/(1.8*DB*(F-G*1.8*DB))))
      ENDIF
      RETURN
      END
C
C ------------------------------------------------------
C    PURPOSE : To compute specific volume of moist air
C    VSDBHA...Specific volume of moist air, m3 moist air/kg dry air
C    DB.......Dry bulk temperature, K
C    HA.......Humidity ratio, kg vapor/kg dry air
C    PATM.....Atmospheric pressure, Pa
C
      FUNCTION VSDBHA (DB,HA)
      PATM = 101.325E03
      VSDBHA =.06243386*(53.35*1.8*DB*(0.6219+HA)/144.0/0.6219/
      &      (PATM/6.894575E03))
      RETURN
      END
C
```

```
C   -----------------------------------------------------
C

C      PURPOSE : To compute humidity ratio from dry bulk temperature
C                      and relative humidity
C      FUNCTIONS USED : 1) PSDB
C                                   2) HAPV
C      HADBRH..Humidity ratio, kg vapor/kg dry air
C      PSDB..Saturated vapor pressure, Pa
C      RH...Relative humidity, decimal
C      DB...Dry bulk temperature, K
C      PV...Vapor pressure, Pa
C
        FUNCTION HADBRH(DB,RH)
        PV = RH*PSDB(DB)
        HADBRH = HAPV(PV)
        RETURN
        END
C
C
C      PURPOSE : To compute humidity ratio from vapor pressure
C      HAPV...Humidity ratio, kg vapor/kg dry air
C      PV.....Vapor pressure, Pa
C      ATM....Atmospheric pressure, Pa
C
        FUNCTION HAPV(PV)
        PATM = 101.325E03
        HAPV = 0.6219*PV/(PATM-PV)
        RETURN
```

```
      END

C   PSDB(DB) is aldeady given previously.
C
C   ------------------------------------------------------
C     PROPOSE: To compute wet bulk temperature of moist air from
C                    dry bulk temperature and humidity ratio.
C     NOTE :   The value of the wet bulk temperature need to be
C              computed by using iterative method. The guess value
C              of wet bulk temperature G1 and G2 are choosed to
C              be G1=DB and G2=DB-35.
C     FUNCTION USED : 1) PSDB(DB)
C                     2) PVHA(HA)
C                     3) HLDB(DB)
C                     4) WBL(TWB), objective function which the zero value is searched.
C                     5) ZEROIN , subroutine used for searching the value of independent
C                        variable which make the objective function become zero.
C     PARAMETERS :
C      G1..Beginning value of the guess interval in which wet bulk temperature
C          is probably exists, K
C      G2..End value of the guess interval in which wet bulk temperature is
C          probably exists, K  (G1=DB, G2=DB-40)
C      EPS Aceptable error range in iterative process
C      DB..Dry bulk temperatur, K
C      HA..Humidity ratio, kg vapor/kg dry air
C      PV..Vapor pressure, Pa
```

```
C       A,B,TB .. Intermediate parameters
C       WBL..Objective function which zero value is searched. WBL is
C           is written as FUNCTION  its value is transferred to the
C           subroutine ZEROIN by  the subroutine argument.
C       EXTERNAL statement must de used for declaring this process.
          FUNCTION WBDBHA(DB,HA,G1,G2,EPS)
          EXTERNAL WBL
          COMMON /SPEC/PV,TB,XTRA
C
C         write(*,10) DB
  10      format('DB in WBDBHA=',F10.2)
C
          A = G1
          B = G2
          TB = DB
          PV = PVHA(HA)
          CALL ZEROIN(A,B,EPS,WBL)
          WBDBHA = (A+B)/2.0
          RETURN
          END
C
C      PURPOSE : compute the value of the objective function, WBL.
C      PARAMETERS :
C       WBL..Objective function, K
C       TWB..Wet bulk temperature, K
C       PV...Vapor pressure. Pa
C       DB...Dry bulk temperature, K
C       PATM.. Atmospheric pressure,Pa
C       PWB...Saturated vapor pressure, Pa
```

```
C
      FUNCTION WBL(TWB)
      COMMON /SPEC/PV,DB,XTRA
      PATM = 101.325E03
      PWB = PSDB(TWB)
      WBL = TWB-DB-((PWB-PV)/(0.2405*(PWB-PATM)*
     &     (1.0+0.15577*PV/PATM))*
     &     0.62194*HLDB(TWB)/2325.8377)/1.8
      RETURN
      END
C
C    PURPOSE : To compute latent heat of evaporization, J/kg
C    PARAMETERS:
C      HLDB.. Latent heat of evaporization, J/kg
C      DB.....Dry bulk temperature, K
C      R......Conversion factor for unit change
C
      FUNCTION HLDB(DB)
C
      R=2325.8377
      IF ((DB-273.16).LT.0.) THEN
      HLDB =R*(1220.884-0.05077*(1.8*DB-459.69))
      ELSE
      IF ((DB-338.72).LT.0.) THEN
      HLDB = R*(1075.8965-0.569835*(1.8*DB-459.69))
      ELSE
C
C   Original value is 0.9125275587  ,
C
```

```
      A=1354673.214-0.09125275587*1.8*DB*1.8*DB
      IF(A.GE.0.0) THEN
      HLDB = R*(SQRT(A))
      ELSE
      write(*,9) DB
    9 format(' DB= ',E15.7,/)
C
      write(*,10)
   10 format(' Argument of SQRT in HLDB is negative')
      ENDIF
      ENDIF
      ENDIF
      RETURN
      END
C
C    PVHA(HA) and PSBD(DB) are given previously.
C
C      PURPOSE : To search the  smallest interval (A,B) in which the
C                value of FUNC become nearly zero or zero. This
C                subroutine is used for searching the value of
C                 the wet bulk temperature of the function WBL.
C
       SUBROUTINE ZEROIN(A,B,EPS,FUNC)
      EXTERNAL FUNC
      REAL I,M
      FA = FUNC(A)
      FB = FUNC(B)
      FC = FA
      C = A
```

```
      IF (SIGN(1.0,FB).NE.SIGN(1.0,FC)) GO TO 1
   1  IF (ABS(FC)-ABS(FB)) 2,3,3
   2  C=B
      B=A
      A=C
      FC=FB
      FB=FA
      FA=FC
   3  IF (ABS(C-B)-2.0*EPS) 12,12,4
C
C    Replace LEGVAR statement, the statement used for preventing
C    zero division of CDC 6500 computer, with IF THEN statement.
C
   4  IF((FB-FA).GT.0.0) THEN
      I=(B-A)*FB/(FB-FA)
      J=0
      ELSE
      J=1
C     WRITE(*,20)
  20  FORMAT(' Value of I becomes infinity')
      ENDIF
C
      M=(C+B)/2.
      IF(J-0) 7,5,7
C
   5  I = -I+B
      CHINT = (B-I)*(M-I)
      IF (CHINT) 8,8,7
   7  I = M
```

```
8     IF (ABS(B-I)-EPS) 9,10,10
9     I = SIGN(1.0,(C-B))*EPS+B
10    A = B
      B = I
      FA = FB
      FB = FUNC(B)
      IF (SIGN(1.0,FB)-SIGN(1.0,FC)) 1,11,1
11    C = A
      FC = FA
      GO TO 1
12    A = (C+B)/2.0
      FA = FUNC(A)
      IF (SIGN(1.0,FA).EQ.SIGN(1.0,FB)) B = C
      RETURN
      END
```

CURRICULUM VITAE

NAME : Miss Rosarin Smitabhindu

BIRTH DATE : April 16, 1953

BIRTH PLACE : Udon Thani, Thailand

EDUCATION :

YEAR	INSTITUTION	DEGREE/DIPLOMA
1976	Kasetsart Univ.	B.S. (Food Science and Technology)
1982	Thammasart Univ.	Master of Business Administration
2001	Kasetsart Univ.	Master of Science (Information Technology)
2008	Kasetsart Univ.	Doctor of Engineering (Industrial Engineering)

POSITION/TITLE : Specialist in policy and planning

WORK PLACE : Royal Chitralada Projects

SCHOLARSHIP/AWARDS : FAO Scholarship in Finland 1982
Danish Government, 1990
Die Deutche Bank, Germany 1992
Princess Maha Chakri Sirindhorn 2001-2008