

**PERFORMANCE OF THERMAL ENERGY STORAGE PROTOTYPE**



**A Thesis Submitted to the Graduate School of Naresuan University  
in Partial Fulfillment of the Requirements  
for the Doctor of Philosophy Degree in Renewable Energy  
(International Program)**

**July 2015**

**Copyright 2015 by Naresuan University**

Thesis entitled "Performance of Thermal Energy Storage Prototype"

by Rungrudee Boonsu

has been approved by the Graduate School as partial fulfillment of the requirements  
for the Doctor of Science Degree in Renewable Energy of Naresuan University

**Oral Defense Committee**

*Adisak Nathakaranakule* ..... Chair  
(Associate Professor Adisak Nathakaranakule, Ph.D.)

*S. Sukchai* ..... Advisor  
(Sukruedee Sukchai, Ph.D.)

*Saranagon Hemavibool* ..... Co – Advisor  
(Assistant Professor Saranagon Hemavibool, Ph.D.)

*Sakda Somkun* ..... Co – Advisor  
(Sakda Somkun, Ph.D.)

*C. Sirisamphanwong* ..... Internal Examiner  
(Chatchai Sirisamphanwong, Ph.D.)

Approved

*Panu Putthawong*

(Panu Putthawong, Ph.D.)

Associate Dean for Administration and Planning  
for Dean the Graduate School

28 JUL 2015



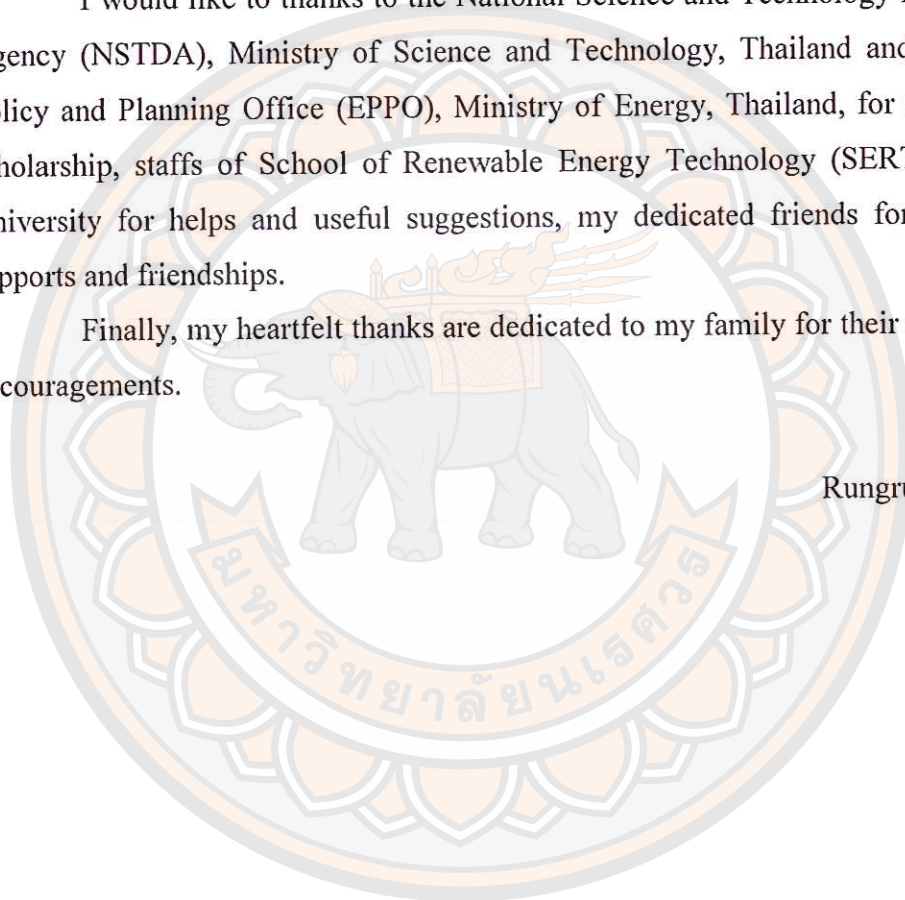
## ACKNOWLEDGEMENT

The success of this dissertation can be attributed to extensive support and assistance of the thesis advisor; Dr.Sukruedee Sukchai, the co-advisor; Assistant Professor Dr.Saranagon Hemaviboon and Dr.Sakda Somkun. I deeply thank for valuable advices and warm guidance, suggestions and encouragements throughout this study.

I would like to thanks to the National Science and Technology Development Agency (NSTDA), Ministry of Science and Technology, Thailand and the Energy Policy and Planning Office (EPPO), Ministry of Energy, Thailand, for sponsoring a scholarship, staffs of School of Renewable Energy Technology (SERT), Naresuan University for helps and useful suggestions, my dedicated friends for their warm supports and friendships.

Finally, my heartfelt thanks are dedicated to my family for their existences of encouragements.

Rungrudee Boonsu



<b>Title</b>	PERFORMANCE OF THERMAL ENERGY STORAGE PROTOTYPE
<b>Author</b>	Rungrudee Boonsu
<b>Advisor</b>	Sukruedee Sukchai, Ph.D.
<b>Co - Advisor</b>	Assistant Professor Saranagon Hemaviboon, Ph.D. Sakda Somkun, Ph.D.
<b>Academic Paper</b>	Thesis Ph.D. in Renewable Energy (International Program), Naresuan University, 2014
<b>Keywords</b>	Thermal Energy Storage, Sensible heat, Heat transfer fluid, Thermal Performance, Concrete

### ABSTRACT

The purposes of this research were to design and fabricate a thermal energy storage prototype by using local materials, and to evaluate the performance and economics of the thermal energy storage prototype. This research was carried out in four parts. Firstly, thermophysical properties test of concrete materials. Secondly, design of a thermal energy storage prototype. Thirdly, performance test of the thermal energy storage prototype. Fourthly, cost of energy analysis.

The investigations of the thermophysical properties test of concrete materials for sample 1 in which the volumetric ratios are water(1): cement(1): sand (1.5): rock(3) with a density of  $1820 \text{ kg/m}^3$ , specific heat of  $1538 \text{ J/kgK}$ , thermal conductivity of  $1.03 \text{ W/mK}$  and coefficient thermal expansion of  $8.21 \text{ micron/K}$  is suitable for use for the thermal energy storage prototype because the specific heat and volumetric heat capacity of concrete aggregates is higher than other samples.

The research was performed on thermal energy storage prototype in Thailand. Concrete was used as the solid media sensible heat material in order to fulfill local material utilization which is easy to handle and low cost. Saturated steam was used for heat transfer fluid. The thermal energy storage prototype was composed of pipes embedded in a concrete storage block. The embedded pipes were used for transporting and distributing the heat transfer medium while sustaining the pressure. The heat exchanger was composed of 16 pipes with an inner diameter of 12 mm and wall

thickness of 7 mm. They were distributed in a square arrangement of 4 by 4 pipes with a separation of 80 mm. The storage prototype had the dimensions of 0.5 x 0.5 x 4 m. The charging temperature was maintained at 180°C with the flow rates of 0.009, 0.0012 and 0.014 kg/s whereas the inlet temperature of the discharge was maintained at 110°C.

The performance evaluation of a thermal energy storage prototype was investigated in the part of charging/discharging. The experiment found that the increase or decrease in storage temperature depends on the heat transfer fluid temperature, flow rates, and initial temperature. The energy efficiency of the thermal energy storage prototype at the flow rate of 0.012 kg/s was the best because it dramatically increased and gave 41% of energy efficiency in the first 45 minutes after which it continued to rise yet only gradually. Over 180 minutes of operation time, the energy efficiency was 53% and the exergy efficiency was 38%.

From the cost of energy analysis, it is found that the capital cost structure of the thermal energy storage by using concrete material resulted that the cost of heat exchanger is more than 50% of total cost. The storage medium (concrete) in Thailand is inexpensive while the insulation and foundation cost is quite expensive. The investment cost of the thermal energy storage prototype in this research was 1,096 baht/kWh.



## LIST OF CONTENTS

Chapter	Page
<b>I INTRODUCTION</b> .....	1
Background .....	1
Statement of the study .....	3
Objectives of the study .....	4
Scopes of the study .....	4
Benefit of the study .....	5
<b>II PRINCIPLE AND RELATED LITERATURE</b> .....	6
Introduction .....	6
Solar Thermal Power.....	7
Solar thermal power plant in Thailand .....	9
Thermal Energy Storage System.....	10
Sensible Heat Storage.....	13
Sensible Heat Storage Media .....	13
Liquid media storage .....	13
Solid media storage .....	15
Reinforce storage.....	18
Heat transfer for Thermal Energy Storage System .....	19
Thermal Conduction.....	20
Thermal Convection.....	23
The Overall Heat Transfer Coefficient.....	25
Laminar and turbulent flow in tubes .....	29
Finite Element Method for Thermal Energy Storage .....	31
Two-Dimensional Heat Conduction.....	31
Economy perspective .....	37
Review of related research .....	39

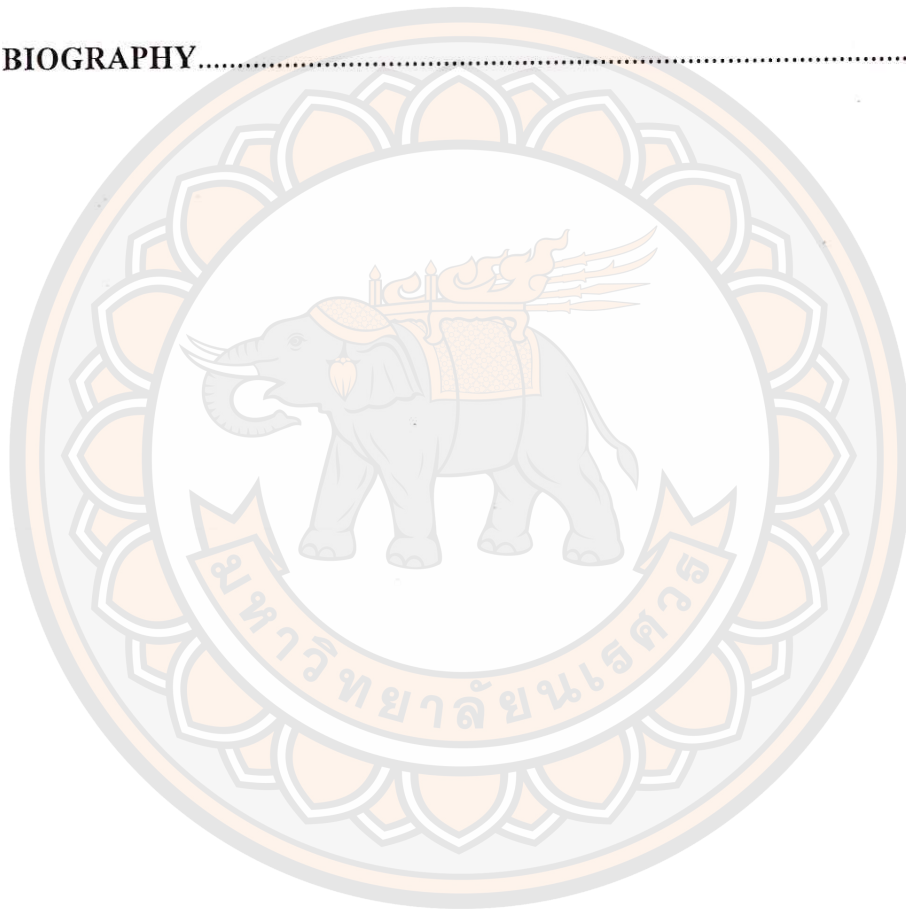
## LIST OF CONTENTS (CONT.)

Chapter	Page
<b>III RESEARCH METHODOLOGY</b> .....	51
Thermophysical properties of concrete materials test.....	51
Materials selection.....	52
Test of concrete properties.....	55
Design of thermal energy storage prototype.....	57
Experimental setup and procedures.....	61
Data collecting.....	64
Performance evaluation.....	66
Cost of Energy.....	70
<b>IV RESULTS AND DISCUSSION</b> .....	71
Properties of concrete.....	71
Design of thermal energy storage prototype.....	72
Performance test of the thermal energy storage prototype.....	75
Temperature Distribution in Radius.....	77
Temperature distribution on charging time.....	77
Temperature distribution on discharging time.....	79
Energy input.....	81
Energy recovered.....	81
Exergy input.....	82
Exergy recovered.....	83
Energy efficiency of Thermal Energy Storage prototype.....	84
Exergy efficiency of Thermal Energy Storage prototype.....	84
Cost of Energy.....	85
<b>V CONCLUSION AND RECOMMENDATION</b> .....	87
Conclusion.....	87
Recommendation.....	88



## LIST OF CONTENTS (CONT.)

Chapter	Page
REFERENCES .....	90
APPENDIX.....	97
BIOGRAPHY.....	186



## LIST OF TABLES

Table	Page
1 The four solar thermal power technologies family.....	8
2 Performance data of various concentrating solar power (CSP) technologies.....	9
3 Overview of thermal energy storage methods.....	11
4 Thermophysical Properties of Sensible Storage Material .....	15
5 Analysis of sensible heat storage.....	16
6 Thermal Characteristics of concrete aggregate material .....	18
7 Summary of transient, two-dimensional finite-difference equation ( $\Delta x = \Delta y$ ) .....	35
8 Four different compositions of concrete.....	55
9 Prediction of thermal properties of concrete .....	55
10 The thermophysical properties of the tested concrete materials in this research and the concrete material developed by others researcher .....	71
11 The amount of energy stored in thermal energy storage material .....	73
12 The capital cost structure of thermal energy storage system.....	86
13 The program list of NSimulation.....	109
14 The function list of NSimulation.....	109
15 The determinable value for Memovie program.....	111
16 The determinable value for HeatPower program .....	112
17 The determinable value for HeatTrans program.....	113
18 The determinable value for LTSolution program.....	114
19 The determinable value for MainProgram.....	115
20 The results of all calculation at flow rate 0.009 kg/s.....	172
21 The results of all calculation at flow rate 0.012 kg/s.....	174
22 The results of all calculation at flow rate 0.014 kg/s.....	177

## LIST OF FIGURES

Figure	Page
1 A classification of energy storage systems .....	6
2 Hour data on electric load during a full week, showing the possible use of a storage system. ....	7
3 Principle of concentrating solar direct radiation and the four CSP technologies .....	8
4 Temperature-enthalpy characteristics for sensible heat storage system with sensible heat transfer fluid in absorbers (left) and combined sensible/PCM storage for direct steam generation in absorbers (right). Sections in power block steam cycle: 1-2 preheating, 2-3 evaporation, 3-4 superheating. ....	12
5 A one-dimensional system in rectangular coordinates .....	20
6 Uniform flow past a heated plate.....	24
7 Heat flow through a cylinder with convection .....	25
8 One-dimensional heat flow through a hollow cylinder and electrical analog .....	27
9 One-dimensional heat flow through multiple cylindrical section and electrical analog .....	28
10 Surface nodes with convective and one-dimensional transient conduction .....	33
11 Investment costs of a 50 MW trough plant with 7-hour storage .....	37
12 Capital cost structure for two-tank molten salt storage concept according to .....	38
13 Capital cost structure for concrete storage concept.....	38
14 Concrete aggregates.....	56
15 Configuration of tested sample of concrete.....	57
16 Two-dimensional model networks of unknown temperatures within the concrete storage system .....	59
17 The modeling design flow chart .....	60
18 The heat exchanger structure of thermal energy storage prototype.....	62



## LIST OF FIGURES (CONT.)

Figure	Page
19 Foundation.....	62
20 Piping structure.....	62
21 Installation sensors and concrete materials .....	63
22 Thermal Energy Storage Prototype with insulation .....	63
23 Data collecting of thermal energy storage prototype .....	64
24 P&ID diagram .....	65
25 The three stage in a simple heat storage process: charging period (a), Storage period (b), and discharging period (c) .....	66
26 The temperature distribution all nodes in 30, 60, 90, 120,150 and 180 minutes .....	74
27 Temperature of HTF inlet and storage prototype in charging experiment at 0.009, 0.012 and 0.014 kg/s.....	76
28 Temperature of concrete storage prototype in discharging period at 0.009, 0.012 and 0.014 kg/s.....	76
29 Temperature distribution and flow rate on charging time of storage bed .....	77
30 Temperature distribution and flow rate on discharging time of storage bed. ....	79
31 Energy input during charging process at 0.009, 0.012 and 0.014 kg/s.....	81
32 Energy recovered during discharging period at 0.009, 0.012 and 0.014kg/s.....	82
33 Exergy input during charging process at 0.009, 0.012 and 0.014 kg/s.....	83
34 Energy recovered during discharging period at 0.009, 0.012 and 0.014 kg/s.....	83
35 Energy efficiency of Thermal Energy Storage at 0.009, 0.012 and 0.014 kg/s.....	84
36 Exergy efficiency of Thermal Energy Storage at 0.009, 0.012 and 0.014 kg/s.....	85

## LIST OF FUGURES (CONT.)

Figure	Page
37 Capital cost structure for Thermal Energy storage prototype.....	86
38 Instruments .....	98
39 The user interface of Memovie program .....	111
40 The user interface of HeatPower program.....	112
41 The user interface of HeatTrans program.....	113
42 The user interface of LTSolution program.....	114
43 The user interface of MainProgram.....	115
44 Front view of Thermal Energy Storge prototype .....	163
45 Concrete foundation .....	164
46 Structure components (A).....	165
47 Structure components (B).....	166
48 Rienforce concrete (A-B) .....	167
49 Rienforce concrete (C-D) .....	168
50 Insulation .....	169



## LIST OF NOMENCLATURE

Symbol	Description	Units
<b>Latin symbols</b>		
A	Cross sectional area	m <sup>2</sup>
$C_p$	Specific heat capacity	J/kg K
d	Diameter of steel pipe	m
D	Diameter of storage pipe	m
$D_m$	Pipe spacing	m
E	Energy	kJ
$F_o$	Fourier Number	-
h	Enthalpy	kJ/kg
L	Length of storage pipe element	m
$\dot{m}$	Mass flow rate	kg/s
$n_r$	Radial elements	-
$n_t$	Number of tubes	-
$N_u$	Nusselt Number	-
P	Pressure	bar
$P_r$	Prandtl Number	-
$\dot{q}$	Specific heat flux	kJ/m <sup>2</sup>
$\dot{Q}$	Heat flow rate	kJ
$Q_l$	Heat losses	kJ
$R_e$	Reynolds Number	-
s	Entropy	kJ/kg K
V	Volume	m <sup>3</sup>
$X_l$	Exergy losses	kJ

## LIST OF NOMENCLATURE (CONT.)

Symbol	Description	Units
<b>Greek symbols</b>		
$\alpha$	Heat transfer coefficient	-
$\lambda$	Thermal conductivity	-
$\pi$	3.142	-
$\Delta$	Difference	-
$\eta$	Energy efficiency	-
$\epsilon$	Exergy	-
$\psi$	Exergy efficiency	-
<b>Indices</b>		
CSP	Concentrating solar power	-
DSG	Direct steam generation	-
HTF	Heat Transfer Fluid	-
f	Final	-
i	Initial	-
<i>i</i>	Element in radial direction	-
<i>j</i>	Element in axial direction	-
L	Laminar	-
max	Maximum	-
min	Minimum	-
PCM	Phase change material	-
T	Turbulent	-
TES	Thermal energy storage	-
X	axial direction	-

# CHAPTER I

## INTRODUCTION

### Background

Currently, Carbon dioxide is deemed responsible for more than 50% of manmade climate change, making it the most critical contributor. It is released mainly by the burning of fossil fuels. Because of the time lapse between emission and their effects, the full consequences of the developing climate change have still to emerge over the coming decade, bringing increased danger to the stability of the world's economy and lifestyle. Solar thermal power emissions involve hardly any of the polluting emissions or environmental safety concerns associated with conventional, fossil or nuclear-based power generation. There is very little pollution in form of exhaust gases, dust or fumes. Most importantly, in terms of the global environment, there are no emissions of carbon dioxide in a purely solar operation of a solar thermal power plant, the main gas responsible for global climate change. Mainly, four elements or components are required in these plants: receiver, concentrator, transport/storage media system and power conversion device. At the current development level, thermal energy storage is a key component [1]. Thermal Energy storage (TES) are critical in enhancing the applicability, performance, and reliability of a wide range of energy systems as the discrepancy between energy supply and demand can be eliminated or reduced by the use of proper thermal energy storage methods [2]. The availability of storage systems is key for the increased market penetration of solar thermal power plants. The main advantages of integrated storage capacity are the extended utilization of power block, improved efficiency and extended life expectancy of components due to a reduction in thermal transients. Storage systems also facilitate the integration of solar thermal power plants into electrical grids by smoothing fluctuations caused by variations in solar radiation, thus avoiding grid instability problems and reducing the requirements for fossil peak load backup capacity [3]. Thermal energy storage in general has been one of the important topics of research for the last three decades, but most of the researchers still feel that



one of the weak points of this technology is the development of effective storage material. Concrete is sometime chosen for storage material because of its low cost, availability throughout the world, and easy processing. Inexpensive aggregates to the concrete are widely available [4]. Basically, there are three methods to store thermal energy: sensible heat storage, latent heat storage and thermo-chemical. For a sensible heat storage unit, thermal energy is stored by changing the temperature of the storage medium which can be a solid or liquid. The ability of a given material to stored sensible heat strongly depends on the value of its energy density, which is the heat capacity per unit volume. The sensible heat storage material must also be inexpensive and have good conductivity. Until now, solid materials such as cast steel, stone, rock, sand, concrete and ceramic have been selected as sensible heat storage media depending on the temperature range and application [5]. The TES system using concrete as the sensible heat storage media is usually implemented by embedding the pipes heat exchanger in concrete to transfer thermal energy to or from the heat transfer fluid, air, and synthetic oil. The advantages of concrete system include low cost of thermal storage media, high heat transfer rates into and out of concrete, ease of handling of the material, available in worldwide and uncomplicated processing. As a result, the application of concrete system is an attractive option regarding investment and maintenance costs in parabolic trough power plant. The feasibility of such systems has been already been proven in Refs. [3, 6, 7]. In a concrete system, the embedded heat exchanger accounts for most of the investment cost, and thus the tube diameter and number of metal pipes should be well designed.

Khare, et al. [8] studied the candidate materials for high temperature sensible heat storage(SHS) systems. The main difficulty in using solid media for SHS is the large size of storage bed. However, this can be minimized by using high heat capacity storage material and allowing high temperature swing. Tamme, et al. [9] suggested that castable ceramic and concrete could be used as sensible heat storage medium for high temperature heat storage applications. Laing, et al. [3] investigated ceramic and high temperature concrete for maximum storage temperature of 663 K and storage capacity of 350 kW and concluded that concrete was the more preferable storage material although ceramic is having 20 % higher storage capacity and 35 % more conductive.

Nandi et al. [10] reported concrete and castable ceramic as low cost (25-30 \$ kWh) and durable SHS systems. It has been observed that heating the concrete at elevated temperatures causes certain reactions and transformations occur due to presence of voids which influence their thermo physical properties. The compressive strength decreases by about 20 % on heating the concrete to 673 K [1]. However, such problems can be minimized by the addition of filler materials like steel needles and reinforcement to improve the mechanical and thermal strength. John et al. found that after exposure to 10 thermal cycles from ambient temperature to 723 K, concrete bed has maintained more than 50 % of their mechanical properties [11].

Therefore concrete storage for parabolic trough in power plant is potential because the cost is low. It is also world wide availability and uncomplicated processing. Although inexpensive aggregates to the concrete are widely available, the concrete storage has not been developed easily as expecting. To develop the large scale thermal concrete storage for parabolic trough power plant is very interesting. Especially, the development by using the local material materials could be beneficial for Thailand. Hence, in the present research focuses on the design and fabrication of the thermal concrete storage by using local materials and also evaluate the performance and economics.

### **Statement of the problem**

Solar energy available to Solar Thermal Power Plants is limited by diurnal, seasonal and climate changes. In order to cope with these fluctuations, the energy source to these needs to be backed up by a buffering storage system.

A buffering storage system can provide:

#### **1. Stability**

Storage smooths out transients in the solar input caused by passing clouds, which can significantly affect operations of a solar electric generating plant. The efficiency of electrical production will degrade with intermittent insulation, largely because the turbine-generator will frequently operate at partial load and in a transient mode. If regular and substantial cloudiness occurs over a short period, turbine steam conditions and/or flow can degrade enough to force turbine trips if there is no



supplementary thermal source to "ride through" the disturbance. Buffer TES systems would typically require small storage capacities.

## 2. Energy management

Storage for the purpose of delivery period displacement requires the use of a larger storage capacity. The storage shifts some or all of the energy collected during periods with sunshine to a later period with higher electricity demand or tariffs (electricity tariffs can be a function of hour of the day, day of the week, and the season). This type of TES does not necessarily increase either the solar fraction or the required collection area. The typical size ranges from 3 to 6 hours of full load operation. The size of a TES for delivery period extension will be of similar size (3 to 12 hours of full load). However, the purpose is to extend the period of power plant operation with solar energy.

## 3. Suitable storage media

Definitive selection of storage capacity is site and system dependent. Therefore, detailed statistical analysis of system electrical demand and weather patterns at a given site, along with a comprehensive economic tradeoff analysis, are desirable in any feasibility study to select the best storage capacity for a specific application.

### **Objectives of the study**

1. To design and fabricate a thermal energy storage prototype by using local materials,
2. To evaluate the performance and cost of energy analysis for the thermal energy storage prototype.

### **Scopes of the study**

1. The thermal energy storage prototype was made by concrete material.
2. The thermal property of the prototype was investigated in terms of specific heat capacity, thermal conductivity and coefficient thermal expansion.

3. The appropriate concrete mixing proportion is investigated by varying of aggregate materials.
4. The prototype is applicable for Solar Parabolic Trough Power Plant.

### **Benefit of the study**

The results from this research will provide many benefits, especially providing proven guidelines for thermal storage system design for Solar Thermal Power Plants, in the following ways:

1. More operating up-time to produce electricity. Power generation can operate during nighttime hours by using energy from the thermal energy storage.
2. It can compensate for loss of solar energy and maintain power production during periods of cloud, by discharging the stored thermal energy.
3. It can provide stability of the electricity supply to the grid. The thermal energy storage system is a backup energy source allowing continuous power supply to the grid.
4. Load management can be supported, resulting in higher revenues. Excess generation available during low-demand periods can be used to charge the TES to increase generation capacity during high-demand periods.

## CHAPTER II

### PRINCIPLE AND RELATED LITERATURE

#### Introduction

Energy in its various forms; electrical energy, magnetic energy, mechanical energy, chemical energy, Thermal Energy etc. can be stored in a variety of ways. Energy can be converted from one form into another, but cannot be created or destroyed [12]. Developing efficient and inexpensive energy storage devices is as important as developing new sources of energy. Energy storage is not a new concept, and has been used for centuries. Energy storage can reduce the time or rate of mismatching between energy supply and energy demand, and plays an important role in energy conservation. As shown in Figure 1, there are a large variety of energy storage systems under development. Energy storage improves the performance of energy systems by smoothing supply and increasing reliability, by loading leveling. Higher efficiency leads to energy conservation and improved cost effectiveness. Some types of renewable energy, such as solar, can be intermittent and fluctuating. The use of energy storage systems can result in significant benefits such as the reduction of the costs power production, and consequently reduction of costs to the power consumer. Increased flexibility of power production operations, and reduction of initial and on-going maintenance costs are also benefits.

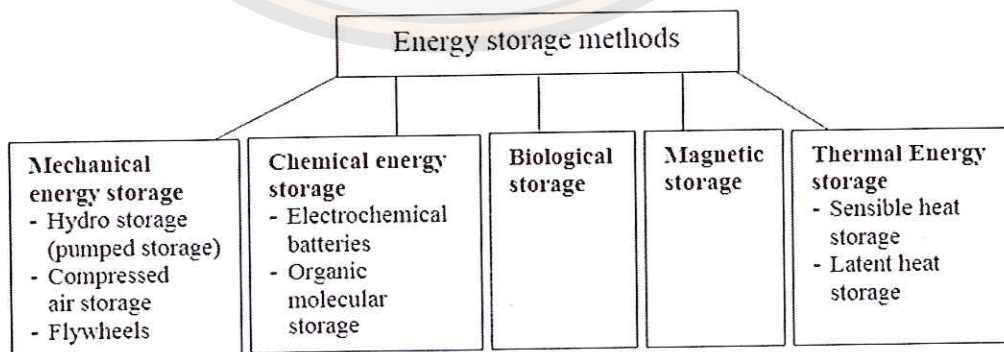
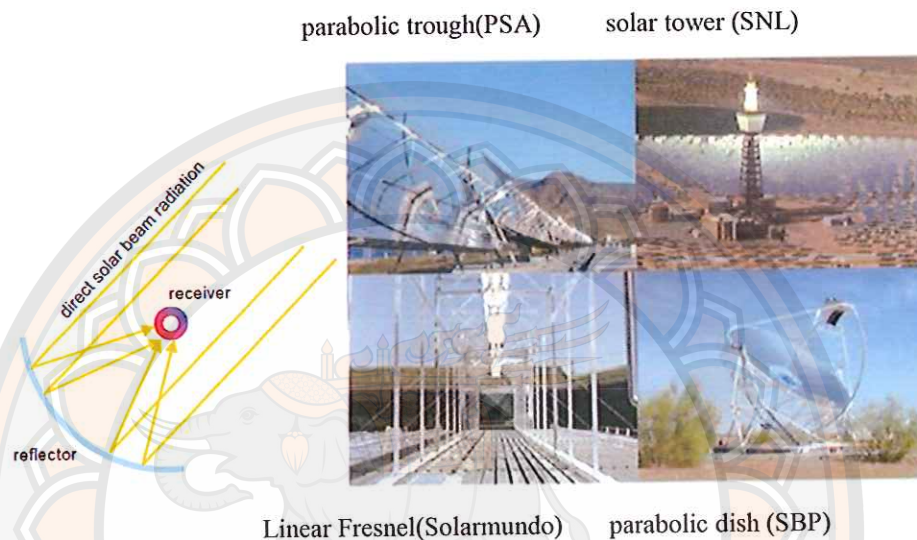


Figure 1 A classification of energy storage systems [13]





efficiencies of 30-40 %. Dish-Stirling engines are used to increase power generation in the 10 kW range. Parabolic troughs have been demonstrated to achieve annual solar to electricity efficiencies of about 10-15 % and about 17-18% projected to be achieved in the future, as shown in Table 2



**Figure 3 Principle of concentrating solar direct radiation and the four CSP technologies [14]**

**Table 1 The four solar thermal power technology families [15]**

Receiver type	Focus type	Line focus	Point focus
		Collectors track the sun along a single axis and focus irradiance on a linear receiver. This makes tracking the sun simpler.	Collectors track the sun along two axis and focus irradiance on a single point receiver. This allows for higher temperatures.
<b>Fixed</b>	Fixed receivers are stationary devices that remain in dependent of the plant's focusing device. This eases the transport of collected heat to the power block.	Linear Fresnel Reflectors	Towers(CRS)
<b>Mobile</b>	Mobile receivers move together with the focusing device. In both line focus designs, mobile receivers collect more energy	Parabolic Troughs	Parabolic Dishes



**Table 2 Performance data of various concentrating solar power (CSP) technologies [14]**

	Capacity Unit MW	Concen- tration	Peak solar efficiency	Annual solar efficiency	Thermal cycle efficiency	Capacity factor (solar)	Land use m <sup>2</sup> /MWh/y
Parabolic trough	10-200	70-80	21% (d)	10-15% (d) 17-18% (p)	30-40% ST	24%(d) 25-90%(p)	6-8
Fresnel	10-200	25-100	20% (d)	9-11% (d)	30-40% ST	25-90%(p)	4-6
Power tower	10-150	300-1000	20% (d) 35% (p)	8-10% (d) 15-25% (p)	30-40% ST 45-55% CC	25-90%(p)	8-12
Dish-stirling	0.01-0.4	1000-3000	29% (d)	16-18% (d) 18-23% (p)	30-40% Stir 20-30% GT	25%(p)	8-12

**Note:** (d) = demonstrated, (p) = projected, ST = steam turbine, GT= Gas Turbine CC = Combined Cycle, Solar efficiency = net power generation / incident direct radiation Capacity factor = solar operating hours per year/8760 hours per year

### Solar thermal power plant in Thailand

Annual direct normal irradiation in Thailand is highest in the northeast and central parts of the country, measured at 1,350-1,400 kWh/m<sup>2</sup>-yr. The yearly average direct normal irradiation, over the entire country is 1,200 kWh/m<sup>2</sup>-yr. This indicates that Thailand has moderate potential of direct solar radiation, when compared to direct radiation in other parts of the world [16].

The sole solar thermal energy power plant in Thailand is based on direct steam generation (DSG) in parabolic trough collectors. The practical feasibility of this technology has been proven in recent years. Direct steam generation (DSG) uses water inside the collectors as the heat transfer medium. Of the four plant types shown in Table 2, DSG technology demonstrates the best advantages:

1. Smaller environmental risks because oil is replaced by water.
2. The overall plant configuration is simpler.
3. Lower investment and O&M costs and also higher plant efficiency.

### Thermal Energy Storage System

Thermal energy storage (TES) deals with the storage of energy by heating, cooling, melting, solidifying, or vaporizing a material; the thermal energy becomes available when the process is reversed [13]. The technologies of thermal energy storage have been developed to the point where they can be properly acknowledged as a new energy technology. The major nontechnical use of thermal storage is to maintain a constant temperature in residences, for warmth during cold winter nights. Large stones, blocks of cast iron, and ceramics have been used to store heat from an evening fire for the whole night. With industrialization of the economy, thermal energy storage materials which were produced as a by-product of some industrial processes could be applied to energy production. A variety of new techniques of thermal energy storage became possible.

Heat storage at power plants is now typically in the form of steam or hot water, but usually for a short retention time. Recently, other materials, such as oils, which have a very high boiling point have been suggested as heat storage substances for the electricity utilities. Other materials that have a high heat of fusion at high temperatures have also been suggested for this application.

Generally, there are three basic methods for thermal energy storage:

1. Heating a liquid or a solid, without phase change: This method is called sensible heat storage. The amount of energy store depends on the temperature change of the material and can be expressed in the form

$$E = m \int_{T_1}^{T_2} C_p dT \quad (1)$$

Where  $m$  is the mass and  $C_p$  is the specific heat at constant pressure.  $T_1$  and  $T_2$  represent the lower and upper temperature levels between which the storage operates.

The difference ( $T_2 - T_1$ ) is referred to the temperature swing.



2. Heating a material which undergoes a phase change (usually melting): This is called latent heat storage. The amount of energy stored ( $E$ ) in this case depends upon the mass ( $m$ ) and latent heat of fusion ( $\lambda$ ) of the material. Thus,

$$E = m\lambda \quad (2)$$

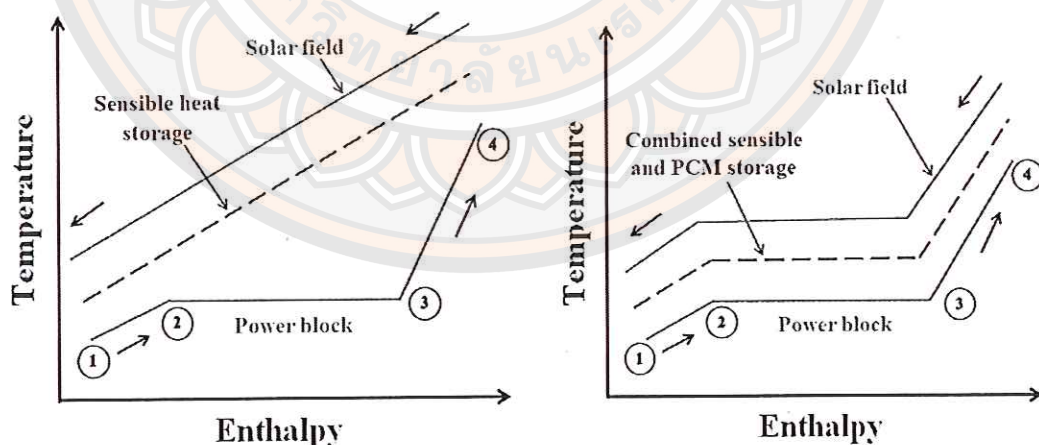
3. Using heat to produce certain physiochemical products. Absorbing and adsorbing are two examples of chemical reactions. The heat is released when the reverse reaction occurs. Also, in this case, the storage operates essentially isothermal during the reactions. However, the temperature at which heat flows from the heat supply varies, depending on the storage material.

**Table 3 Overview of thermal energy storage methods**

Type of thermal Energy Storage	Functional Principle	Phase	Examples
Sensible Heat	Temperature change of medium with highest possible heat capacity	Liquid	Hot water, organic liquids, molten salts, liquid metals
		Solid	Metals, minerals, ceramics
Latent Heat	Essentially heat of phase change	Liquid-Solid	Nitrides, chlorides, hydroxides, Carbonates, fluorides,
		Solid-Solid	eutectics Hydroxides
Chemical Energy	Large amount of chemical energy is absorbed and released due to shifting of equilibrium by changing pressure and temperature	Solid-Gas	CaO/H <sub>2</sub> O, MgO/H <sub>2</sub> O, FeCl <sub>2</sub> /NH <sub>3</sub>
		Gas-Gas	CH <sub>4</sub> /H <sub>2</sub> O
		Liquid-Gas	LiBr/H <sub>2</sub> O, NaOH/H <sub>2</sub> O, H <sub>2</sub> SO <sub>4</sub> /H <sub>2</sub> O



In terms of reducing the cost of a solar thermal power plant integrated with a storage system, direct steam generation is an attractive option. However, such solar thermal power plants experience major challenges with two-phase fluid water/steam, therefore optimized storage concepts are needed. A three-part storage system consisting of preheater) evaporator and super heater is proposed for the two phase fluid water/steam [17]. The left diagram of Figure 4, shows the temperature-enthalpy relationship for a solar thermal power plant for a single phase fluid (e.g. thermal oil). It indicates that the sensible heat storage system is employed for preheating, evaporating and superheating. The heat is stored as the temperature of the storage is increased. The diagram on the right indicates the system with direct steam generation using combined PCM and sensible heat storage system. PCM is used in an evaporator for evaporation/condensation, whereas sensible storage systems are used for preheating and superheating. The maximum amount of energy is stored in a PCM evaporator with a small temperature difference between the storage material and the working fluid. Therefore the direct steam generation concept with combined sensible and PCM storage concept is considered to be the most efficient for thermal energy storage.



**Figure 4** Temperature-enthalpy characteristics for sensible heat storage system with sensible heat transfer fluid in absorbers (left) and combined sensible/PCM storage for direct steam generation in absorbers (right). Sections in power block steam cycle: 1-2 preheating, 2-3 evaporation, 3-4 superheating [18]

### Sensible Heat Storage

In case of sensible heat storage systems, energy is stored or extracted by heating or cooling a liquid or a solid, which does not change its phase during the process. A variety of substances have been used in these systems. These include liquids such as water, heat transfer oils and certain inorganic molten salts and solids like rocks, pebbles and refractory materials. In the case of solids, the material is invariably in porous form and heat is stored or extracted by the flow of a gas or a liquid through the voids. The choice of the material used depends largely on the temperature level of the application, water being used for the temperature below 100°C and refractory bricks being used for temperatures about 1,000°C. Sensible heat storage systems are simpler in design than latent heat or chemical storage systems. However, the disadvantage of them is size; they are bigger than other configurations. An important criterion in materials selection for sensible heat storage systems is the ( $\rho C_p$ ) value. A second disadvantage associated with sensible heat systems is that they can't store or deliver energy at a constant temperature. Performance of thermal storage system is determined by storage capacity, heat input and output rates while charging and discharging, and storage efficiency. The storage capacity of sensible heat storage with a solid or liquid storage medium is given by

$$Q_s = mc_p \Delta T = \rho c_p V \Delta T \quad (3)$$

Where  $m$  is mass,  $V$  is volume,  $c$  is specific heat,  $\rho$  is density and  $\Delta T = T_{max} - T_{min}$  is the maximum temperature difference between the maximum and minimum temperatures of the medium. This expression can be used to calculate the mass and volume of storage material required to store a given quantity of energy.

#### Sensible Heat Storage Media

Sensible heat storages have 2 bases of the heat storage media as liquid media storage and solid media storage.

#### Liquid media storage

Water is a common used medium in heat storage system because of its high specific heat. Most solar water heating and space heating systems use hot water



storage tanks located either inside or outside the buildings or underground. The sizes of the tanks used vary from a hundred liters to a few thousand cubic meters. Water storage tanks are made from a variety of materials like steel, concrete and fiberglass. The tanks are suitably insulated with glass wool, mineral wool or polyurethane. The thickness of insulation used is large and ranges from 10 to 20 cm. Because of this, the cost of the insulation represents a significant part of the total cost and means to reduce this cost needs to be explored. Heat transfer oils are used in sensible heat storage systems for intermediate temperatures ranging from 100 to 300°C. The problem associated with the use of heat transfer oils is that they tend to degrade with time. The degradation is particularly serious if they are used above their recommended temperature limit. The use of oils also presents safety problems since there is a possibility of ignition above their flash point. For this reason, it is recommended that they be used in systems with an inert gas cover. A further limitation to the use of heat transfer oils is their cost. For this reason, they can be seriously considered for use only in small storage systems. A few molten inorganic salts have been considered for high temperatures (300°C and above). One is an eutectic mixture of 40 percent  $\text{NaNO}_2$ , 7 percent  $\text{NaNO}_3$  and 53 percent  $\text{KNO}_2$  (by weight). It has a low melting point of 145°C and can be used up to a temperature of 425°C. Above this temperature; decomposition and oxidation begins to take place. Another molten salt being considered for high temperature storage is sodium hydroxide, which has a melting point of 320°C and could be used for temperatures up to 800°C. However, it is highly corrosive and there is difficulty in containing it at higher temperatures. Freezing and corrosion problems can be met by using chemical additives. Water sometimes remains economically competitive at higher temperatures despite the need for pressure containment especially so when it is stored in aquifers. Organic oils, molten salts, and liquid metals circumvent the problems of vapor pressure, but have other limitations in handling, containment, cost, storage capacities, and useful temperature range, etc. In spite of the fact that these fluids have been used in commercial operations, the lifetime and cost requirements for solar thermal storage limit their use in applications such as space heating. However, oils and molten salts have been utilized in solar thermal power plants.



### Solid media storage

The difficulties of the high vapor pressure of water and the limitations of other liquids can be avoided by storing thermal energy as sensible heat in solids. For parabolic trough power plants using liquid as the heat transfer medium, the application of solid media sensible heat storage is an attractive option regarding investment and maintenance costs. Cost effective systems demand the utilization of inexpensive storage materials, which usually exhibit a low thermal conductivity. Essential for the successful development of a storage system is the sufficient heat transfer between the heat transfer fluid and the storage materials. The thermal capacity of the thermal energy storage media now in use or proposed for parabolic trough power plants present in Table 4.

**Table 4 Thermo-physical Properties of Sensible Storage Material [19]**

Storage medium	Temperature (°C)		Average Density (kg/m <sup>3</sup> )	Average Heat Conductivity (W/mK)	Average Heat Capacity (kJ/kgK)	Volume Specific Heat Capacity (kWh <sub>t</sub> /m <sup>3</sup> )	Media Cost (\$/kg)	Media Cost (\$/kWh <sub>t</sub> )
	Cold	Hot						
	<b>Solid media</b>							
Sand-Rock-mineral oil	200	300	1,700	1.0	1.3	60	0.15	4.2
Reinforced Concrete	200	400	2,200	1.5	0.85	100	0.05	1.0
Solid NaCl	200	500	2,160	7.0	0.85	150	0.15	1.5
Cast iron	200	400	7,200	37.0	0.56	160	1.00	32.0
Cast steel	200	700	7,800	40.0	0.60	450	5.00	60.0
Silica fire Bricks	200	700	1,820	1.5	1.00	150	1.00	7.0
Magnesia fire bricks	200	1,200	3,000	5.0	1.15	600	2.00	6.0

Table 4 (cont.)

Storage medium	Temperature (°C)		Average Density (kg/m <sup>3</sup> )	Average Heat Conductivity (W/mK)	Average Heat Capacity (kJ/kgK)	Volume Specific Heat Capacity (kWh/m <sup>3</sup> )	Media Cost (\$/kg)	Media Cost (\$/kWh)
	Cold	Hot						
	<b>Liquid media</b>							
Mineral oil	200	300	770	0.12	2.6	55	0.30	4.2
Synthetic oil	250	350	900	0.11	2.3	57	3.00	43.0
Silicone oil	300	400	900	0.10	2.1	52	5.00	80.0
Nitrite salts	250	450	1,850	0.57	1.5	152	1.00	12.0
Nitrite salts	265	565	1,870	0.52	1.6	250	0.70	5.2
Carbonate Salts	450	850	2,100	2.0	1.8	430	2.40	11.0
Liquid Sodium	270	530	850	71.0	1.3	80	2.00	21.0

The following table provides a summary of the main advantages and disadvantages of the various storage concepts

Table 5 Analysis of sensible heat storage [20]

Concept	Advantages	Disadvantages
Two-Tank (molten salt)	<ul style="list-style-type: none"> <li>- Provides heat at constant temperature during discharge</li> <li>- Good behavior at partial charge</li> <li>- Option for using storage medium as working fluid in the solar field as well</li> </ul>	<ul style="list-style-type: none"> <li>- risk of irreversible freezing</li> <li>- complex initial filling procedure</li> <li>- limited potential for further cost reductions resulting from technical improvements</li> <li>- total costs strongly dependent on costs of storage medium</li> <li>- tank volume approx. two times the volume of the storage material</li> </ul>

Table 5 (cont.)

Concept	Advantages	Disadvantages
Thermocline (molten salt)	<ul style="list-style-type: none"> <li>- Cost reduction due to partial substitution of molten salt by low cost filler material</li> <li>- Only single tank needed</li> </ul>	<ul style="list-style-type: none"> <li>- risk of irreversible freezing</li> <li>- complex initial filling procedure</li> <li>- filler material must be compatible with molten salt</li> <li>- about 30 % of the storage volume cannot be used due to thermal boundary layer</li> </ul>
Solid media with embedded heat exchanger	<ul style="list-style-type: none"> <li>- Cost reduction due to low cost storage material</li> <li>- No risk of freezing</li> <li>- Pressurized working fluid can be used directly without intermediate heat exchanger</li> </ul>	<ul style="list-style-type: none"> <li>- Temperature decreases during discharge</li> <li>- Repair or maintenance of embedded heat exchangers difficult</li> <li>- Thermo mechanical stress between heat exchanger and storage medium must be considered</li> </ul>
Steam accumulator	<ul style="list-style-type: none"> <li>- Low response time</li> <li>- High volume-specific power</li> <li>- Substantial operating experience in process industry</li> </ul>	<ul style="list-style-type: none"> <li>- Temperature not constant during discharge process</li> <li>- Only for small pressures cost attractive</li> <li>- Large pressurized vessels necessary</li> <li>- Storage capacity limited</li> </ul>
Packed bed with air as heat transfer fluid	<ul style="list-style-type: none"> <li>- No heat exchanger necessary</li> <li>- Suitable for high temperatures</li> <li>- Low risk potential</li> <li>- Substantial operating experience in process industry</li> </ul>	<ul style="list-style-type: none"> <li>- Pressure losses might be critical</li> <li>- Combination with solar absorbers using liquid heat transfer fluids difficult</li> <li>- Distances between solar receiver and storage must be limited</li> </ul>



### Reinforce concrete

Concrete is a nonhomogeneous manufactured stone composed of graded, granular, inert materials which are held together by the action of cementations materials and water. The inert materials usually consist of gravel or large particles of crushed stone and sand or pulverized stone. The inert materials are called aggregates. The large particles are called coarse aggregates and the small particles are called fine aggregates [21].

Concrete behaves very well when subjected to compressive forces, but ruptures suddenly when small tension forces are applied. Therefore in order to utilize this material effectively, steel reinforcement is placed in the areas subjected to tension. Reinforced concrete is a composite material which utilizes the concrete in resisting compression forces, and some other material, usually steel bars or wires, to resist the tension forces. Steel is also often used to assist the concrete in resisting compression forces. Concrete is always assumed to be incapable of resisting tension, even though it actually can resist a small amount of tension.

**Table 6 Thermal Characteristics of concrete aggregate material [22, 24]**

Material	Density (kg/m <sup>3</sup> )	Conductivity (W/m K)	Specific heat capacity (J/kg K)
Cement, Portland	1,505	0.29	840
Water (32.22 °C)	994.9	0.623	4,174
Sand	1,520	0.61	800
Stone:			
Granite	2,640	1.73-3.98	820
Limestone(100-300 °C)	2,500	1.26-1.33	900
Marble	2,500-2,700	2.07-2.94	800
Sandstone (40 °C)	2,160-2,300	1.83	710
Concrete, cinder stone,1-2-4 mix	1,900-2,300	1.37	880

Concrete is sometimes chosen because of its low cost, availability throughout the world, and easy processing. Inexpensive aggregates to the concrete are widely available. Concrete has the following characteristics as a storage medium:

1. High specific heat
2. Good mechanical properties (e.g., compressive strength)
3. A thermal expansion coefficient near that of steel (pipe material) and
4. High mechanical resistance to cyclic thermal loading.

When concrete is heated, a number of transformations and reactions take place which influence its strength and other physical properties. When concrete is heated to about 100 °C, water is expelled (up to 130 kg of water per m<sup>3</sup> of concrete). The remaining water (50 to 60 kg of water per m<sup>3</sup> concrete), either physically bound in smaller pores or held by chemisorption, is expelled as temperatures rise from 120 to 600 °C. Most dehydration occurs between 30 and 300 °C. This water loss reduces the weight of concrete by 2-4%. The specific heat decreases in the temperature range between 20 and 120 °C, and the thermal conductivity decreases between 20 and 280 °C. The mechanical properties are also slightly influenced by the loss of water; compressive strength decreases by about 20% at 400 °C compared to that at ambient temperature. Resistance to thermal cycling depends on the thermal expansion coefficients of the materials used in the concrete. To minimize such problems, a basalt concrete is sometimes used. Steel needles and reinforcement are sometimes added to the concrete to impede cracking. By doing so, the thermal conductivity is increased by about 15% at 100 °C and 10% at 250 °C. Such concrete storage can be supplied as prefabricated plates. Alternatively, the concrete maybe poured on-site into large blocks, leading to easier and more economic construction. Whether prefabricated plates or on-site pouring is advantageous depends on local conditions.

### **Heat transfer for Thermal Energy Storage System**

There are a variety of ways in which energy can be transported from one region to another, but from the continuum point of view all mechanisms can be satisfactorily categorized in terms of conduction, convection, and radiation.



The mechanisms are listed in order of increasing complexity and we will study them in that order.

### Thermal Conduction

The temperature distribution existing within a material can at most depend on three space variables and on time. If indeed the temperature is a function of time, the problem is called unsteady and the temperature distribution is referred to as being transient. If the temperature is not a function of time, the problem is referred to as steady or the temperature distribution as being in a steady state. If temperature depends only on a single space coordinate, the problem, or the temperature distribution, is referred to as one dimensional. When temperature depends on two or three space variables, the problem is referred to as a two or three dimensional problem, respectively. In a one-dimensional, unsteady problem, temperature is a function of one space variable and of time. In this thermal analysis will assume that thermal conductivity is constant even though thermal conductivity for most materials does vary with temperature, the dependence in a majority of cases is not a strong one. A material in which thermal conductivity does not vary with direction is call isotropic.

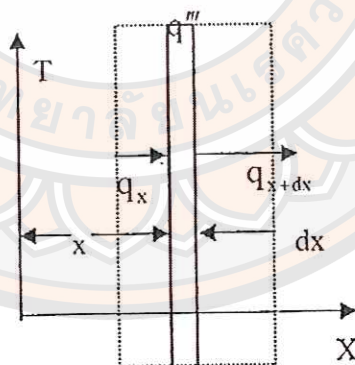


Figure 5 A one-dimensional system in rectangular coordinates [25]

Figure 5 shows a one-dimensional coordinate system. The system consists of a plane wall on which we impose between the  $T$  and  $x$  axis. A slice of the material  $dx$  thick is selected for study. The fact energy can be generated within the material is taken into account by the term  $q'''$ . Typically, internal heat generation within a solid passing through the material, or by a nuclear reaction. Some of the energy passing



through the control volume may be stored, thus increasing the internal energy of the material, which is sensed physically as an increase in the temperature of the material. This is the case for an unsteady problem.

Any geometry can be written as follows:

Rate of energy conducted into control volume	+	Rate of energy generated inside control volume	=	Rate of energy conducted out of control volume	+	Rate of energy generated inside control volume
--	---	--	---	--	---	--

From Figure 5, the equation becomes

$$q_x + q''' Adx = q_{x+dx} + \rho Adx \frac{\partial u}{\partial t} \quad (4)$$

Where  $A$  is the cross-section area,  $Adx$  is the volume of the element,  $q'''$  is the internal heat generated per unit volume,  $\rho$  is density,  $\rho Adx$  is mass, and  $\frac{\partial u}{\partial t}$  represents the rate of change in internal energy per unit mass of the control volume. The rate of energy conducted into the control volume is  $q_x$ .

The rate of energy conducted out of control volume becomes

$$q_{x+dx} = q_x + dq_x \quad (5)$$

where denote a change as  $dq_x$ . Alternatively, could write

$$q_{x+dx} = q_x + \frac{dq_x}{dx} dx \quad (6)$$

to denote more generally that the change is a function of the  $x$  coordinate. If it is anticipated that  $q_x$  will be a function of more than one variable, the following partial derivation notation is appropriate:

$$q_{x+dx} = q_x + \frac{dq_x}{dx} dx \quad (7)$$

Another way of obtaining this expression is to expand  $q_{x+dx}$  in a Taylor series and write only the first two terms. Of the various expressions written above for  $q_{x+dx}$ , Equation (7) is form to be used here. Combining with Equation (1) gives

$$q_x + q'''Adx = q_x + \frac{\partial q_x}{\partial x}dx + \rho Adx \frac{\partial u}{\partial t}$$

so that

$$-\frac{\partial q_x}{\partial x} + q'''Adx = \rho Adx \frac{\partial u}{\partial t} \quad (8)$$

This is the general energy equation for unsteady heat conduction in one dimension with internal heat generation. It is desirable to substitute something more easily measured for the various terms in this equation. Fourier's Equation is used for the heat conducted:

$$q_x = -kA \frac{\partial T}{\partial x} \quad (9)$$

where  $k$  is thermal conductivity,  $T$  is temperature, and because we anticipate that temperature will be a function of more than one variable, the partial derivative notation is used.

Conduction is the principal mode of energy transfer in solids; the above equation reduces to

$$du = dh \quad (10)$$

Substituting from the definition of specific heat, obtain

$$c_v dT = c_p dT$$

or

$$c_v = c_p = c \quad (11)$$

where  $c_v$  is the specific heat at constant volume and  $c_p$  is the specific heat at constant pressure. They are equal in a solid. So the internal energy change can be written as

$$\frac{\partial u}{\partial t} = c \frac{\partial T}{\partial t} \quad (12)$$

combining equation 9 and 10 with 11 yields

$$-\frac{\partial q_x}{\partial x} + q''' Adx = \rho Adx \frac{\partial u}{\partial t}$$

Dividing by  $Adx$  and simplifying, we obtain

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + q''' = \rho c \frac{\partial T}{\partial t} \quad (13)$$

For constant thermal conductivity the above becomes

$$\frac{\partial^2 T}{\partial x^2} + \frac{q'''}{k} = \frac{\rho c}{k} \frac{\partial T}{\partial t} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (14)$$

which is the general energy equation for unsteady heat conduction in one dimension with internal heat generation, written in terms of temperature. The term  $\alpha = \frac{k}{\rho c}$  is introduced as the thermal diffusivity of the material, with dimensions of  $L^2/T$ .

### Thermal Convection

Convection is the mode of heat transfer associated with fluid motion. If the fluid motion is due to an external motive source such as a fan or pump, the term forced convection applies. On the other hand, if fluid motion is due predominantly to the presence of a thermally induced density gradient; then the term natural convection is appropriate.



Consider a plate immersed in a uniform flow as shown in Figure 6. The plate is heated and has a uniform wall temperature of  $T_w$ . The uniform flow velocity is  $U_\infty$ , and the fluid temperature far from the plate is  $T_\infty$ . At any location we sketch the velocity distribution on the axes labeled  $V$  vs  $y$ . The velocity at the wall is zero due to the nonslip condition; that is, fluid adheres to the wall due to friction or viscous effects. The velocity increases with increasing  $y$  from zero at the wall to nearly the free-stream value at some vertical distance away, known as the hydrodynamic boundary layer. Also sketch a temperature distribution,  $T$  vs  $y$ , which is seen to decrease from  $T_w$  at the wall to  $T_\infty$  at some distance from the wall (assuming  $T_\infty < T_w$ ).

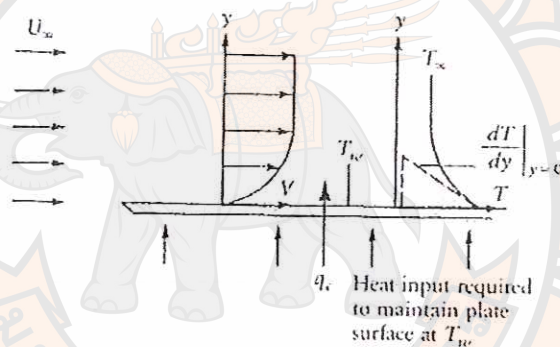


Figure 6 Uniform flow past a heated plate [25]

Heat is transferred from the wall to the fluid. Within the fluid the mechanism of heat transfer at the wall is conduction because the fluid velocity at the wall is zero. However, the rate of heat transfer depends on the slope of the  $T$  vs  $y$  curve at the wall  $dT/dy$  at  $y = 0$ . A steeper slope is indicative of a greater temperature difference and is highly dependent on the flow velocity. The flow velocity will influence the distance from the wall that we must travel before we sense that the temperature is  $T_\infty$ .

The heat transferred by convection is found to be proportional to the temperature difference. In the case of Figure 6

$$\frac{q_c}{A} \propto T_w - T_\infty$$

Introducing proportionality constant, get

$$q_c = \bar{h}_c A (T_w - T_\infty) \quad (15)$$

In which  $\bar{h}_c$  is called the average convection heat transfer coefficient or the film conductance. This coefficient accounts for the overall effects embodied in the process of convection heat transfer.

### The Overall Heat Transfer Coefficient [25]

In many problems having a polar cylindrical geometry, heat transfer by convection is quite common. Examples include steam flowing through an insulated pipe, cold Freon flowing through a copper tube, and heated crude oil flowing up from underground at an oil well. It is therefore convenient to include convective effects in the one-dimensional conduction problem for cylinders.

Figure 7 is a sketch of a pipe or tube containing a fluid at temperature  $T_{\infty 1}$ . Heat is transferred to the pipe by convection, through the pipe wall by conduction, then to the fluid outside, which is at temperature  $T_{\infty 2}$ . Also shown in the figure is the thermal circuit. From previous discussions equations can be written for each resistance as

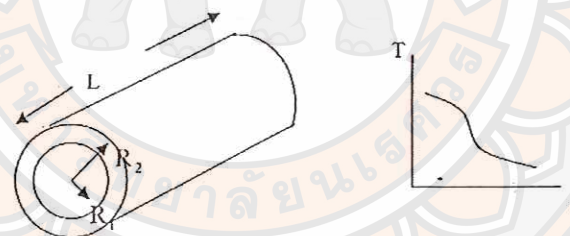


Figure 7 Heat flow through a cylinder with convection

$$R_{cl} = \frac{1}{\bar{h}_{c1} A_1} \quad (16)$$

$$R_k = \frac{\ln(R_2 / R_1)}{2\pi k L} \quad (17)$$

$$R_{cl} = \frac{1}{\bar{h}_{c2} A_2} \quad (18)$$

In these equations the convection coefficient, assumed constant, on the inside and outside surfaces are  $\bar{h}_{c1}$  and  $\bar{h}_{c2}$ , respectively. The areas for heat transfer on the inside and outside surfaces are

$$A_1 = 2\pi R_1 L \quad \text{and} \quad A_2 = 2\pi R_2 L \quad (19)$$

The heat flow from fluid to fluid, based on the overall temperature difference, is

$$q_r = \frac{T_{\infty 1} - T_{\infty 2}}{\frac{1}{\bar{h}_{c1} 2\pi R_1 L} + \frac{\ln(R_2/R_1)}{2\pi k L} + \frac{1}{\bar{h}_{c2} 2\pi R_2 L}} \quad (20)$$

or

$$q_r = \frac{2\pi L (T_{\infty 1} - T_{\infty 2})}{\frac{1}{\bar{h}_{c1} R_1} + \frac{\ln(R_2/R_1)}{k} + \frac{1}{\bar{h}_{c2} R_2}} \quad (21)$$

In some problem it is desirable to express this equation in terms of an overall heat transfer coefficient that accounts for the combined effects of convection at both surfaces and conduction:

$$q_r = UA(T_{\infty 1} - T_{\infty 2}) \quad (22)$$

Where U is an overall heat transfer coefficient and A is some area on which to base U. The question arises as to which area to select. It is customary practice to select either the outside surface area or the inside surface area of the cylinder. Based on  $A_2$ ,  $U_2$  can be determined by equations 21 and 22

$$U_2 (2\pi R_2 L) (T_{\infty 1} - T_{\infty 2}) = \frac{2\pi L (T_{\infty 1} - T_{\infty 2})}{\frac{1}{\bar{h}_{c1} R_1} + \frac{\ln(R_2/R_1)}{k} + \frac{1}{\bar{h}_{c2} R_2}} \quad (23)$$

Solving,

$$U_2 = \frac{1}{\frac{R_2}{\bar{h}_{c1} R_1} + \frac{R_2 \ln(R_2/R_1)}{k} + \frac{1}{\bar{h}_{c2}}} \quad (24)$$

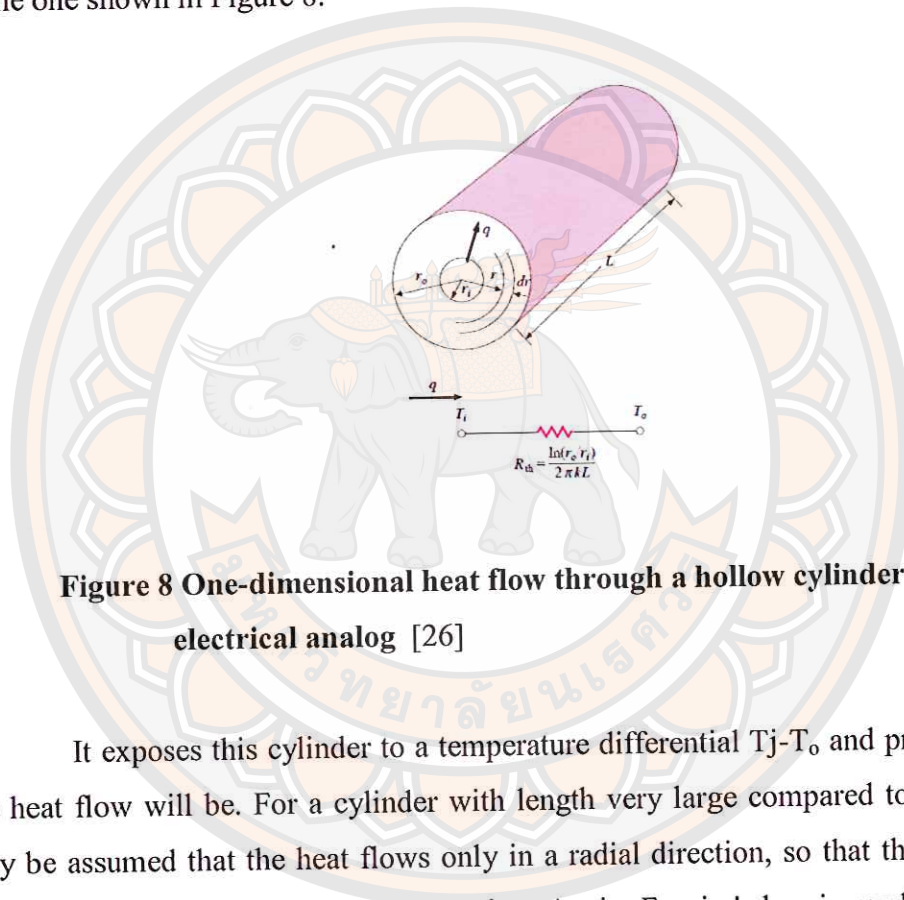


$$U_1 = \frac{1}{\frac{1}{h_{c1}} + \frac{R_1 \ln(R_2/R_1)}{k} + \frac{1}{h_{c2}R_2}} \quad (25)$$

As seen in equations 24 and 25 the overall coefficient is independent of length.

### Radial Systems of Cylinders [26]

Consider a long cylinder of inside radius  $r_i$ , outside radius  $r_o$  and length  $L$  such as the one shown in Figure 8.



**Figure 8 One-dimensional heat flow through a hollow cylinder and electrical analog [26]**

It exposes this cylinder to a temperature differential  $T_j - T_o$  and proposes what the heat flow will be. For a cylinder with length very large compared to diameter, it may be assumed that the heat flows only in a radial direction, so that the only space coordinate needed to specify the system is  $r$ . Again, Fourier's law is used by inserting the proper area relation. The area for heat flow in the cylindrical system is

$$A_r = 2\pi rL$$

So that Fourier's law is written

$$q_r = -kA_r \frac{dT}{dr} \quad (26)$$

or

$$q_r = -2\pi rL \frac{dT}{dr} \quad (27)$$

with the boundary conditions

$$T = T_i \quad \text{at } r = r_i$$

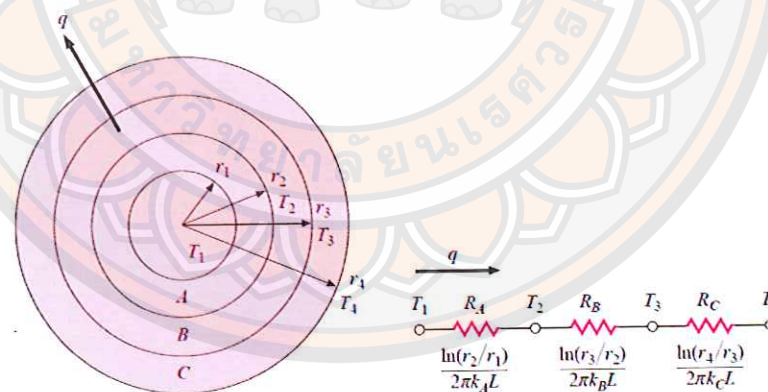
$$T = T_o \quad \text{at } r = r_o$$

The solution to Equation 27

$$q = \frac{2\pi rL(T_i - T_o)}{\ln(r_o/r_i)} \quad (28)$$

and the thermal resistance in this case is

$$R_{th} = \frac{\ln(r_o/r_i)}{2\pi kL} \quad (29)$$



**Figure 9 One-dimensional heat flow through multiple cylindrical section and electrical analog [26]**

The thermal-resistance concept may be used for multiple-layer cylindrical walls just as it was used for plane walls. For the three-layer system shown in Figure 9 the solution is

$$q = \frac{2\pi rL(T_1 - T_4)}{\ln(r_2/r_1)/k_A + \ln(r_3/r_2)/k_B + \ln(r_4/r_3)/k_C} \quad (30)$$

### Laminar and turbulent flow in tubes

The flow in the tube can be laminar or turbulent. Depending on the flow conditions. Fluid flow is streamline and thus laminar at low velocities, but turns turbulent as the velocity is increased beyond a critical value. Transition from laminar to turbulent flow does not occur suddenly; rather, it occurs over some range of velocity where the flow fluctuates between laminar and turbulent flows before it becomes fully turbulent. Most pipe flows encountered in practice are turbulent. Laminar flow is encountered when highly viscous fluids such as oils flow in small diameter tubes or narrow passages. The hydrodynamic entry length is usually taken to be the distance from the tube entrance where the friction coefficient reaches within about 2% of the fully developed value. In laminar flow, the hydrodynamic and thermal entry lengths are given approximately as the following:

$$\begin{aligned} L_H &= 0.05 \text{ Re}D \\ L_T &= 0.05 \text{ Re}Pr \end{aligned} \quad (31)$$

In the entrance region the flow in tubes is not developed, Nusselt recommended the following equation

For  $10 < L/d < 400$

$$\text{Nu} = 0.036 \text{ Re}^{0.8} \text{ Pr}^{1/3} \left(\frac{d}{L}\right)^{0.055} \quad (32)$$

Where  $L$  is the length of the tube and  $d$  is the tube diameter.

$\text{Re}$  is the Reynolds number ( $\text{Re} = \frac{4\dot{m}}{\pi D\mu}$ )

$\text{Pr}$  is the Prandtl number ( $\text{Pr} = \frac{\mu C_p}{k}$ )



The properties in Equation 31 are evaluated as the mean bulk temperature the above equations offer simplicity in computation, but errors on the order of  $\pm 25\%$  are not uncommon. Petukhov has developed a more accurate, although more complicated, expression for fully developed turbulent flow in smooth tubes.

$$\text{Nu} = \frac{(f/8)\text{RePr}}{1.07 + 12.7\sqrt{f/8}(\text{Pr}^{2/3} - 1)} \left(\frac{\mu_b}{\mu_w}\right)^n \quad (33)$$

where  $n = 0.11$  for  $T_w > T_b$ ,  $n = 0.25$  for  $T_w < T_b$  and  $n = 0$  for constant heat flux or for gases. All properties are evaluated at  $T_f = (T_w + T_b)/2$  except for  $\mu_b$  and  $\mu_w$ . The friction may be obtained from the following equation for smooth tubes.

$$f = (1.82 \log \text{Re} - 1.64)^{-2} \quad (34)$$

Equation 32 is applicable for the following ranges.

$$0.5 < \text{Pr} < 200 \text{ for } 6\% \text{ accuracy and } 0.5 < \text{Pr} < 2000 \text{ for } 10\% \text{ accuracy}$$

$$10^4 < \text{Re} < 5 \times 10^6, \text{ and } 0.08 < \mu_b / \mu_w < 40$$

To obtain agreement with data for smaller Reynolds number, Gnielinski modified the correlation and proposed an expression of the form

$$\text{Nu} = \frac{(f/8)(\text{Re} - 1000)\text{Pr}}{1 + 12.7\sqrt{f/8}(\text{Pr}^{2/3} - 1)} \quad (35)$$

The correlation is valid for  $0.5 < \text{Pr} < 200$  and  $2300 < \text{Re} < 5 \times 10^6$ . It notes that, unless specifically developed for the transition region. ( $2300 < \text{Re} < 10^4$ )

### Finite Element Method for Thermal Energy Storage

Unsteady heat conduction problems can be solved numerically by transforming the partial differential equation of heat conduction to finite-difference equations in both space and time domains. Various schemes have been developed for the finite-difference representation of the time-dependent heat conduction equation [26].

#### Two-Dimensional Heat Conduction

The finite-difference procedure is represented of the following two-dimensional, time-dependent heat conduction equation:

$$\frac{\partial T}{\partial \tau} = \alpha \left\{ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right\} \quad (36)$$

$$\alpha \text{ is thermal diffusivity, m}^2/\text{s}, \alpha = \frac{k}{\rho C_p}$$

It is supposed that this equation is to be solved over a finite region subject to a prescribed temperature at all boundary surfaces.

The regional area is divided into square sub regions by a network of mesh size  $\Delta x = \Delta y$ , and the time domain is divided into small time steps  $\Delta t$ . Then the temperature  $T(x,y,t)$  at any location  $(x,y)$  and at any time  $t$  can be denoted by

$$T(x, y, t) = T(m\Delta x, n\Delta y, i\Delta t) \equiv T_{m,n}^i \quad (37)$$

Using this notation to write the finite-difference form of the partial derivatives for  $\Delta x = \Delta y$ , as

$$\left[ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right]_{m,n,i} \cong \frac{T_{m-1,n}^i + T_{m+1,n}^i + T_{m,n-1}^i + T_{m,n+1}^i - 4T_{m,n}^i}{(\Delta x)^2} \quad (37a)$$

The partial derivative with respect to the time variable can be written as

$$\left[ \frac{\partial T}{\partial t} \right]_{m,n,i} \cong \frac{T_{m,n}^{i+1} - T_{m,n}^i}{\Delta t} \quad (37b)$$

Introducing equation 7 into equation 5 can find that the finite-difference form of the two-dimensional, time-dependent heat conduction equation is

$$T_{m,n}^{i+1} = Fo [T_{m-1,n}^i + T_{m+1,n}^i + T_{m,n-1}^i + T_{m,n+1}^i] + (1 - 4Fo)T_{m,n}^i \quad (38)$$

Where

$$Fo = \frac{\alpha \Delta t}{(\Delta x)^2}, \Delta x = \Delta y \quad (39)$$

$F_o$  is Fourier number

Equation 39 is the explicit finite-difference formulation of the heat conduction Equation 36. This equation provides a simple expression for calculating the temperature  $T_{m,n}^{i+1}$  at a node  $(m,n)$  at the time step  $i+1$  if the temperatures at the node  $(m,n)$  and at its neighboring nodes are known for the previous time step  $i$ . Therefore, by starting with the initial condition  $i = 0$ , the node temperature at the subsequent time steps are calculated.

The procedure is simple, but there is a restriction on the permissible maximum value of the parameter  $F_o$ .

Restriction on  $F_o$  by following an argument similar to that described for the one-dimensional equation, It conclude that to obtain physically meaningful results, the coefficient  $1-4 F_o$  of  $T_{m,n}^i$  in Equation 38 should not be negative. This requirement leads to the following stability criterion:

$$0 < Fo \leq \frac{1}{4} \quad (40)$$

This condition implies that for given values of  $\alpha$  and  $\Delta x$ , the time step  $\Delta t$  cannot exceed the upper limit imposed by the stability criterion given by Equations 39, 40; otherwise, the numerical calculations become unstable.



Equation 39 may also be derived by applying the energy balance method to a control volume about the interior node. Accounting for changes in thermal energy storage, a general form of the energy balance equation may be expressed as [26]

$$\dot{E}_{in} + \dot{E}_g = \dot{E}_{st} \quad (41)$$

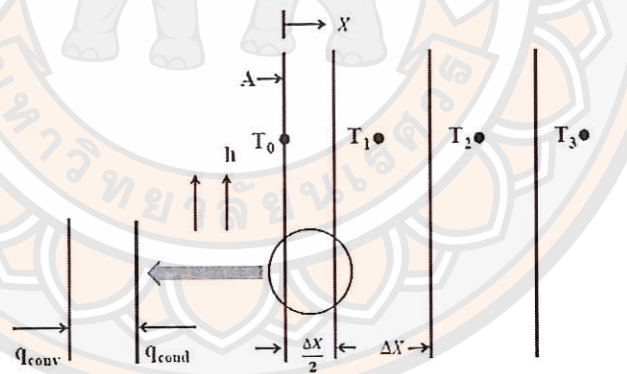
$E_{in}$  is rate of energy transfer into a control volume, W

$E_g$  is rate of energy generation, W

$E_{st}$  is rate of increase of energy stored within a control volume, W

In the interest of adopting a consistent methodology, it is again assumed that all heat flow is into the node.

To illustrate application of Equation 41, consider the surface node of the one dimensional system shown in Figure 10.



**Figure 10 Surface nodes with convective and one-dimensional transient conduction**

To more accurately determine thermal conditions near the surface, this node has been assigned a thickness that is one-half that of the interior nodes. Assuming convection transfer from an adjoining fluid and no generation, it follows from Equation 41 that

$$hA(T_\alpha - T_0^p) + \frac{kA}{\Delta x}(T_1^p - T_0^p) = \rho cA \frac{\Delta x}{2} \frac{T_0^{p+1} - T_0^p}{\Delta t} \quad (42)$$

or, solving for the surface temperature at  $t + \Delta t$ ,

$$T_0^{p+1} = \frac{2h\Delta t}{\rho c\Delta x}(T_\infty - T_0^p) + \frac{2\alpha\Delta t}{\Delta x^2}(T_1^p - T_0^p) + T_0^p \quad (43)$$

Recognizing that  $(2h\Delta t/\rho c\Delta x) = 2(h\Delta x/k)(\alpha\Delta t/\Delta x^2) = 2BiFo$  and grouping terms involving  $T_0^p$ , it follows that.

$$T_0^{p+1} = 2Fo(T_1^p + BiT_\infty) + (1 - 2Fo - 2BiFo)T_0^p \quad (44)$$

The finite-difference form of the Biot number is

$$Bi = \frac{h\Delta x}{k} \quad (45)$$

Recalling the procedure for determining the stability criterion, we require that the coefficient for  $T_0^p$  be greater than or equal to zero. Hence

$$1 - 2Fo - 2BiFo \geq 0$$

or

$$Fo(1 + Bi) \leq \frac{1}{2} \quad (46)$$

$h$  is convection heat transfer coefficient,  $W/m^2$

$k$  is thermal conductivity,  $W/m K$

$c$  is specific heat,  $J/kg K$

$\rho$  is mass density,  $kg/m^3$

$A$  is area of surface,  $m^2$

$Bi$  is Biot number

Since the complete finite-difference solution requires the use of Equation 38 for the interior nodes, as well as Equation 44 for the surface node. Equation 46 must be put in contrast with Equation 40 to determine which requirement is more stringent. Since  $Bi \geq 0$ , it is apparent that the limiting value of  $F_o$  for Equation 46 is less than that for Equation 40.

Forms of the explicit finite-difference equation for several common geometries are presented in Table 7. Each equation may be derived by applying the energy balance method to a control volume about the corresponding node. To develop confidence in your ability to apply this method, you should attempt to verify at least one of these equations.

**Table 7 Summary of transient, two-dimensional finite-difference equation**

$(\Delta x = \Delta y)$  [26]

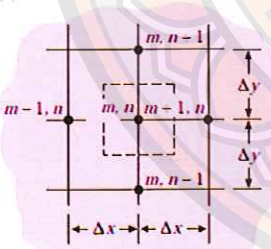
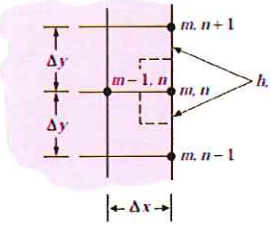
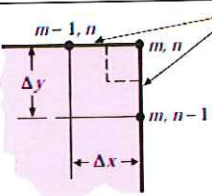
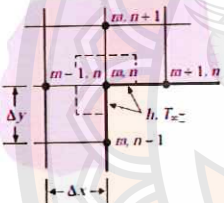
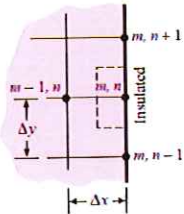
Physical situation	Nodal Equation for $(\Delta x = \Delta y)$	Stability Criterion
(A) Interior node	 $T_{m,n}^{p+1} = Fo(T_{m-1,n}^p + T_{m,n+1}^p + T_{m+1,n}^p + T_{m,n-1}^p) + [1 - 4(Fo)]T_{m,n}^p$ $T_{m,n}^{p+1} = Fo(T_{m-1,n}^p + T_{m,n+1}^p + T_{m+1,n}^p + T_{m,n-1}^p - 4T_{m,n}^p) + T_{m,n}^p$	$Fo \leq \frac{1}{4}$
(B) Convection boundary node	 $T_{m,n}^{p+1} = Fo[2T_{m-1,n}^p + T_{m,n+1}^p + T_{m,n-1}^p + 2(Bi)T_{\infty}^p] + [1 - 4(Fo) - 2(Fo)(Bi)]T_{m,n}^p$ $T_{m,n}^{p+1} = Fo[2Bi(T_{\infty}^p - T_{m,n}^p) + 2T_{m-1,n}^p + T_{m,n+1}^p + T_{m,n-1}^p - 4T_{m,n}^p] + T_{m,n}^p$	$Fo(2 + Bi) \leq \frac{1}{2}$

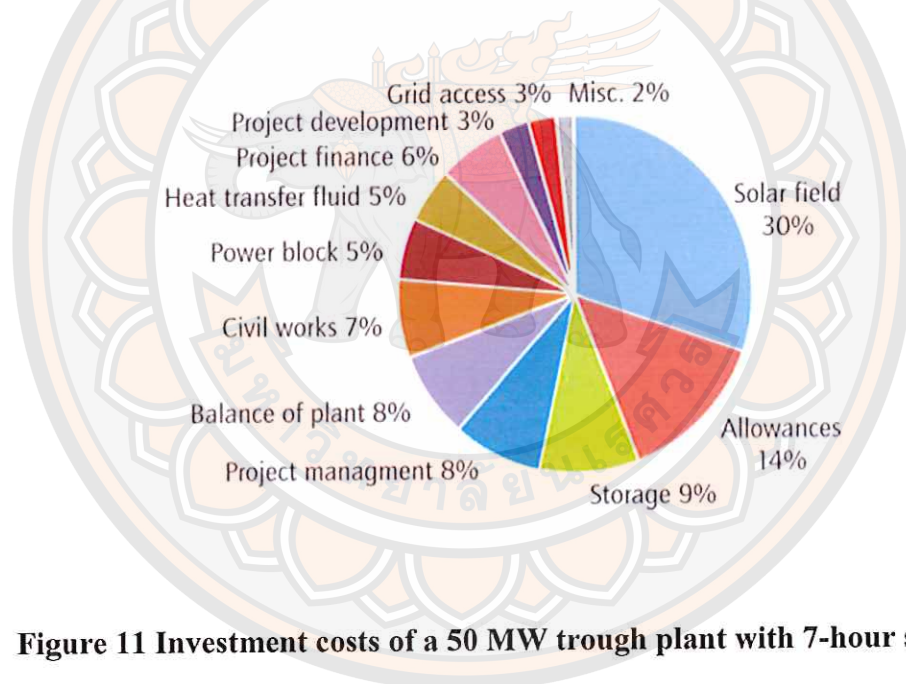


Table 7 (cont.)

Physical situation	Nodal Equation for ( $\Delta x = \Delta y$ )	Stability Criterion
	<p>(C) Exterior corner with convection boundary</p> $T_{m,n}^{p+1} = 2(Fo)[T_{m-1,n}^p + T_{m,n-1}^p + 2(Bi)T_{\infty}^p] + [1 - 4(Fo) - 4(Fo)(Bi)T_{m,n}^p]$ $T_{m,n}^{p+1} = 2Fo[T_{m-1,n}^p + T_{m,n-1}^p - 2T_{m,n}^p + 2Bi(T_{\infty}^p - T_{m,n}^p)] + T_{m,n}^p$	$Fo(1 + Bi) \leq \frac{1}{4}$
	<p>(D) Interior corner with convection boundary</p> $T_{m,n}^{p+1} = \frac{2}{3}(Fo)[2T_{m,n+1}^p + 2T_{m+1,n}^p + 2T_{m-1,n}^p + T_{m,n-1}^p + 2(Bi)T_{\infty}^p] + [1 - 4(Fo) - \frac{4}{3}(Fo)(Bi)]T_{m,n}^p$ $T_{m,n}^{p+1} = (\frac{4}{3})Fo[T_{m,n+1}^p + T_{m+1,n}^p + T_{m-1,n}^p - 3T_{m,n}^p + Bi(T_{\infty}^p - T_{m,n}^p)] + T_{m,n}^p$	$Fo(3 + Bi) \leq \frac{3}{4}$
	<p>(E) Insulated boundary</p> $T_{m,n}^{p+1} = Fo[2T_{m-1,n}^p + T_{m,n-1}^p + T_{m,n-1}^p] + [1 - 4(Fo)]T_{m,n}^p$	$Fo \leq \frac{1}{4}$

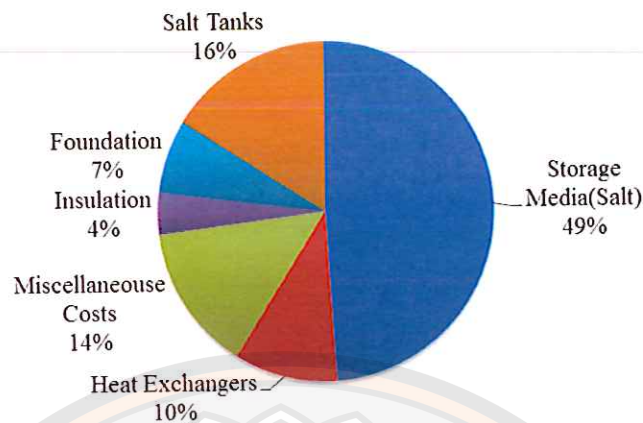
### Economy perspective

For large, state-of-the-art Parabolic trough power plants, current investment costs are 4.2-8.4 USD/W depending on labor and land costs, technologies, the amount and distribution of DNI and, above all, the amount of storage and the size of the solar field. Plants without storage that benefit from excellent DNI are on the low side of the investment cost range; plants with large storage and a higher load factor but at locations with lower DNI (about 2000 kWh/m<sup>2</sup>/year) are on the high side. Figure 1 shown break down investment costs of a trough plant with storage under Spanish skies. These investments costs are slightly higher than those of PV devices, but CSP plants have a greater energy output per MW capacity [27].



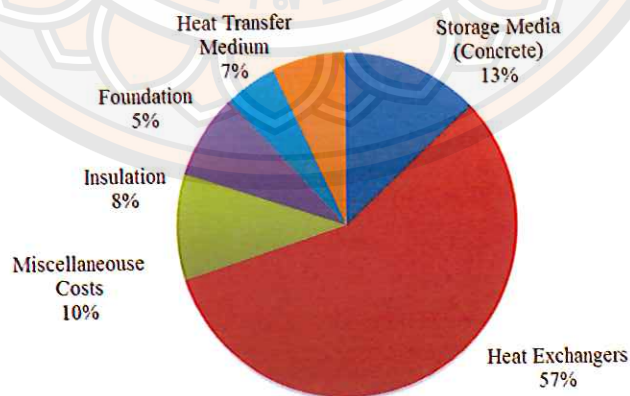
**Figure 11 Investment costs of a 50 MW trough plant with 7-hour storage**

The state of art commercial scale thermal energy storages for CSP plants use the two-tank molten salt system which is very sensitive to fluctuations of the market price of Nitrate salts, as almost 50 % of the costs are covered by the salt, as is shown in the capital cost structure for the two-tank molten salt concept in Figure 12.



**Figure 12 Capital cost structure for two-tank molten salt storage concept according to [28]**

The concrete storage technology is in a pre-commercial state with investment costs in the range of 30-35 €/kWh, which is approx. 25 % lower than for a corresponding molten salt storage. Figure 13 shows the cost structure of a concrete storage system; while the costs of the storage material can be reduced considerably, the costs of the heat exchanger become dominant at about 57 % of the total costs. Here, only little further cost reduction is expected.



**Figure 13 Capital cost structure for concrete storage concept [29]**



### Review of related research

Mortar and cement paste were used to test the effect of the sand and silica fume. The results indicated that thermal conductivity of mortar was higher than the cement paste. But specific heat of cement paste is higher than mortar. By adding sand, there is an increase of thermal conductivity and specific heat is decreased. Sand addition had the opposite effects of the silica fume addition. By adding silica fume, thermal conductivity decreased and specific heat increased [30]

This survey of thermal energy storage for parabolic trough power plants reviews that the most two-tank storage system with the heat transfer fluid (HTF) serves itself as the storage medium was successfully established in a commercial power plant. However, the power plant that uses HTF as the storage medium is very expensive. At that time, other storage systems using concrete, phase change materials and chemical storage were under progress. It was also found that an interesting option for cost-effective power plants was the use of indirect two-tank storage where liquid medium storage like molten salt was used [31].

Storage capacity is the important point for the economic point of view of solar thermal power plants. Most of today's parabolic trough power plants use the sensible heat storage system with operating temperatures between 300°C to 390°C. A simulation model is implemented in the performance of the solid sensible heat storage system of the parabolic trough power plant at PSA, Spain. The simulation results showed that many parameters that have great influence on the storage system. Storage material properties are also important for the storage system. The capacity of the system depends on the not only storage unit but also the whole system of power plant. The operational strategy of the storage unit is critical for economic optimization [9].

Two step procedures were used for developing thermal energy storage concrete (TESC). The first step is absorbing phase changing materials(PCM) using the thermal energy storage aggregates (TESAs) and in the second step, TESC was made by mixing Portland cement, and other raw materials of normal concrete and using TESAs. For the phase change material, butyl stearate (BS), a type of organic PCM is used. In porous aggregate includes one kind of shale and two kinds of clay that

provide PCM material. Mercury intrusion porosimetry (MIP) and image analysis (IA) methods were used for showing characteristics of porous structure of the materials. It was found that TESC with PCM was commercially comparable according to the result. Therefore, this two-step process of TESC is a reasonable application in the field of energy conservation [32]. Solid media sensible heat storage is popular in case of investment and maintenance costs. Two solid media sensible heat storage systems with the capacity of around 350 kWh and maximum temperature around 390°C have been investigated in the WESPE project in Spain for parabolic trough power plants. Synthetic oil is used as the heat transfer fluid. Two different storage materials, castable ceramic and high temperature concrete, were tested at the nominal temperature 390°C in order to know the characteristics of storage materials for long term behavior. The maximum thermal power is 480 kW in the operation. The castable ceramic is based on a binder containing  $\text{Al}_2\text{O}_3$ . Blast furnace cement is used as a binder for high temperature concrete. Thermo physical properties such as density, specific heat capacity, thermal conductivity, coefficient of thermal expansion and material strength and crack initiation were tested. A tubular heat exchanger with 36 steel tubes was combined with storage materials. Material properties of storage materials are, for castable ceramic, the density is  $3500 \text{ kgm}^{-3}$ , specific heat capacity is  $866 \text{ Jkg}^{-1}\text{K}^{-1}$ , thermal conductivity is  $1.35 \text{ Wm}^{-1}\text{K}^{-1}$ , coefficient of thermal expansion is  $11.8 \cdot 10^{-6}\text{K}^{-1}$ , material strength is low and no cracks and for high temperature concrete, density is  $2750 \text{ kgm}^{-3}$ , specific heat capacity is  $916 \text{ Jkg}^{-1}\text{K}^{-1}$ , thermal conductivity is  $1.00 \text{ Wm}^{-1}\text{K}^{-1}$ , coefficient of thermal expansion is  $9.3 \cdot 10^{-6}\text{K}^{-1}$ , material strength is medium and many crack at operating temperature 350°C. Although castable ceramics have a 20% higher storage capacity and 35% higher thermal conductivity, high temperature concrete seems to be the more favorable material due to lower cost, higher strength material and easier handling of the pre-mixed material. Both materials are suitable for solid media heat storage system [3].

A cost effective integrated storage system is the major factor for the commercial power plants. Moreover, the benefits of a storage system are improved output efficiency and a smooth process to the electrical grids. The storage system using synthetic oil as the heat transfer fluid is an attractive option for the parabolic



trough power plant because of the low investment and maintenance costs. This project focuses on the cost reduction of the heat exchanger and the energetic and exergetic analysis of modular storage operation including cost assessment. The analysis result of this project shows that there are many ways to improve the storage performance by technically but it is not sufficient economically. So, the tubes were embedded straight and parallel without any additional material to increase heat transfer is the best way for economic point of view. And the results of energetic and exergetic analysis of modular storage concept was also significant for economic optimization. In accordance with the techno economic view, high temperature resistant concrete became as an appropriate solid storage material for the parabolic trough power plants [33].

The improvement of thermal conductivity of stearic acid (SA) was tested as PCM using expanded graphite (EG) and carbon fiber (CF) for energy storage applications. The hot-wire method is used to measure thermal conductivity of SA/EG and SA/CF by putting different mass fraction (2%, 4%, 7% and 10%) of EG and CF. Thermal conductivity of SA improves by putting 10% of mass fraction of EG and CF. Therefore, EG and CF should be considered for the latent heat energy storage application to improve thermal conductivity of SA without reducing heat storage capacity [34].

The mechanical properties like compressive strength, flexural strength, modulus of elasticity, and weight loss were studied at high temperatures up to 900°C to know the characteristics of high temperature resistance of normal strength and autoclaved high strength mortars incorporated with polypropylene and steel fibers. Test results showed that the compressive strength of all types of samples increased with rising temperature up to 300°C. But temperature exceeds 300°C all specimens broke down to pieces except normal strength mortars, and high strength mortars with polypropylene fibers. It means that the adding polypropylene fiber has more strength than steel fiber. To prevent spoiling, minimum polypropylene dosage for autoclave and water cured high strength mortars has been determined as 0.2% and 0.1 % respectively [35].



The thermal cycling operation of the concrete storage test module was proven that concrete storage technology is a suitable option for storing sensible heat. The technology is applicable for solar trough plants, industrial waste heat and combined heat and power systems. Due to the modular design concrete storage is scalable from the kWh to GWh range. So far, the thermal storage in concrete has been tested and proven in the temperature range up to 400 °C. Currently the development is being carried on towards 500 °C. The expected storage capacity has been approved by the experimental results. The developed simulation tool allows up-scaling of the system and the design of concrete storage systems for commercial application [36].

The energy storage system is a key for both the economic and technical point of view for industrial process waste heat and solar thermal power plant. Liquid sensible heat storage using water is for the low temperature (below 100°C) storage system. For temperature above 100°C, pressurized water is needed for the storage application. And other liquid media storage such as molten salt and oil also can be used but they have some disadvantages like a high freezing point and degradation problems. So, solid sensible heat storage such as concrete is the most attractive option for high temperature storage. Concrete storage media was developed by DLR for parabolic trough power plants and concrete is also suitable for the regarding the costs. A 20 m<sup>3</sup> concrete solid sensible heat storage medium was tested with the storage capacity of 400 kWh in Stuttgart for 16 months with the operating temperature 300°C to 400°C and thermal cycle is about 270 cycles. The simulation result of this project shows that this plant with a 1,100 MWh thermal storage operate for approximately 4,550 hours annually with 3,500 full load hours [37].

Concrete storage materials testing the mathematical modeling of concrete storage system and experiment of the concrete storage model were studied in the concrete storage system for solar thermal power plant research in Thailand. Local materials were used for mixing concrete: water, cement, sand and rock. Four different compositions of samples were investigated. The compositions of concrete were sample 1 is 1:1:1.5: rock 3, sample 2 is 1:1:2:4, sample 3 is 1:1:2.5:4, sample 4 is 1:1:3:5. Thermo-physical properties such as density, specific heat capacity, volumetric heat capacity and thermal conductivity, the amount of energy stored were tested. According

to the experiment, sample 1 is the most suitable for investment because thermal conductivity, volumetric heat capacity and the amount of energy stored is higher than others. For temperature distribution, sample 1 is also better than others in both experiment and simulation. The price is also more suitable than other samples. Therefore, sample 1 is suitable for using concrete storage system for solar thermal power plant in Thailand [38].

In the model of predicting the specific heat capacity of fly-ash concrete, the specific heat of fly-ash cement pastes and mortars were tested experimentally. Mortars were organized differently the sand type and content. The result showed that the specific heat depends of the amount of water such as specific heat decreases as the amount of free water decreases. When cement is replaced by fly ash, the specific heat is high at young age but decrease in the long term due to the pozzolanic reaction. Cement pastes had higher specific heat than mortar. The model for predicting the specific heat of concrete depended on the time, material and mix proportion. The model was verified with many experimental results and simulated results [39].

In the model for predicting thermal conductivity of concrete, experiments were doing in order to know the thermal conductivity of fly ash using mortars and cement paste. Mortar specimens were arranged in order to study the effect of aggregates and pastes were arranged for studying the effect of fly ash. The experiments showed replacing fly ash instead of cement has low thermal conductivity. So, thermal conductivity of cement paste is lower than the mortar. A model for predicting the thermal conductivity of fly ash concrete based on time, material, and mix proportion. Thermal conductivity of concrete was calculated by multiplying the volumetric fractions and respective thermal conductivity values of each ingredient that are composed in the concrete. The model was acceptably proved by various experimental results for pastes, mortars and concretes [40].

Coefficient of thermal expansion (CTE) of cementations paste depends on the ages of paste. Pastes were made with different ratios of fly ash in the binder and water to binder. Especially at the early age, coefficient of thermal expansion of pastes rises with age, and is reduced with fly ash content. The model of coefficient of thermal expansion of paste is developed based on the time, material, and mix proportion. The



effect of water weight on CTE of paste is not significant. The CTE of cement-fly ash paste was rising day by day because of the long term continuing pozzolanic reaction. In this paper, it indicated that experiment and predicted results of CTE of the past are not too different. The model was acceptably proven by various experimental results [41].

Efficient energy storage is important to get the achievement of solar thermal power generation and industrial waste heat recovery. In the German Aerospace Center (DLR), sensible heat storage systems like concrete storage system have been developed to describe the design of the test module and results focusing the cost reduction in the heat exchanger using the high temperature concrete storage material. A 20 m<sup>3</sup> solid media storage test module is connected to an electrically heated thermal oil loop was built for testing and further improvements. That concrete block is successfully tested five months of operation in the temperature range between 300°C and 400°C and almost 100 thermal cycles with a temperature difference of 40K. With the successful start-up and thermal cycling operations of the concrete storage test module, it is proven that concrete storage technology is a suitable option for storing sensible heat storage for solar thermal power plants [4].

A three-dimensional finite element analysis is used for computing the temperature effect and restrained strain in mass concrete. The model was calculated based on the time, material properties, and mix proportion of concrete. Different size of mass concrete was analyzed; especially thickness and the casting method in order to know temperature effect and restrained strain effect in mass concrete. Continuous and discontinuous castings are used for casting methods. Layer casting and block casting were made for the discontinuous casting. For studying restrained strain in mass concrete and the effect of thermal properties of aggregate on temperature, different types of aggregate were using in the analysis process. Different conditions of curing; insulation and normal curing were also studied and compared. It was found from the analysis that the maximum temperature increases with the increase of the thickness of the structure. The use of layer casting is more effective for thermal cracking control of mass concrete. The insulation curing method is preferable for mass concrete. Aggregate with low coefficient of thermal expansion is beneficial to reduce the restrained strain [42].



For solar thermal power plants, a new type of heat storage material is developed by using basalt, bauxite, calcium aluminate cement, graphite, silica micro powder and steel fiber. The result showed that adding graphite increased the thermal conductivity, but decreased compressive strength and flexural strength of high temperature concrete storage material. The value of thermal conductivity is nearly 2.34 W/mK which showed that it has excellent properties. Therefore, it is the proper storage material for solar thermal power plants [43].

For thermal energy storage, experimental analysis of thermo-physical of concrete is done to determine suitable concrete composition for thermal energy storage. Thermal conductivity, resistivity, thermal diffusivity, compressive strength and thermal storage capacity are tested with different composition of concretes. They found that concrete mix of ratio 1:2:0 (cement: sand: gravel) has the highest compressive strength of 2.74 N mm<sup>-2</sup>, lowest thermal conductivity of 0.51 W/m°C, thermal storage capacity of 2.74 J m<sup>-3</sup>K. And another mix of ratio 1:1.2:1.1 has the highest thermal storage capacity of 3.22 J m<sup>-3</sup>K. The next mix of ratio 1: 1.9:1.7 has the highest thermal conductivity of 1.7 W/m°C [44].

Sensible and latent heat storage is combined as the storage unit of the total storage capacity. 1MWh storage system was studied using two-phase heat transfer fluid: PCM storage unit and concrete storage unit. That system contains three-part storage systems: preheating, evaporation and superheating. PCM storage is for evaporating and concrete storage is for preheating water and superheating steam. PCM storage unit use 140 kg sodium nitrate (NaOH3) with a melting temperature of 306°C. Concrete was made by mineral aggregates and hardened cement paste. After testing the concrete for several hours in the oven, the mass loss of the aggregates and concrete stabilizes at 500°C. This storage system was successfully tested in the laboratory for direct steam generation power plant [36].

Thermal and mechanical properties of iron and steel-fiber-reinforced concrete were researched for thermal energy storage application. Thermal conductivity, thermal resistivity, thermal diffusivity, density and compressive strength were tested and calculated the energy storage capacity. The linear heat source theory was used as the testing method. Three different sizes of concrete with same amount of steel fiber were

used for experimentation. The results indicated that the properties of concrete increased by adding both steel fibers and iron. Therefore, concrete with iron or steel fibers is more effective than plain concrete [45].

Concrete is used for the high-temperature heat storage state and steam accumulator is used for low temperature heat storage state in the two-stage thermal energy storage system. A numerical simulation model is used to analyze the thermal characteristics of the two-stage thermal storage system. The result indicated that thermal conductivity and the distance between the neighboring steel tubes are main parameters for the high-temperature concrete storage unit. The steam discharging speed of steam accumulator directly affects the heat behavior inside the concrete block [5].

Concrete mechanical properties are tested in the laboratory in order to know the mechanical properties of different environmental conditions. Temperature and humidity is controlled when curing. The air temperature is at 20 and 22°C and humidity is between 40 % and 60 %. Modulus of elasticity, compressive strength, and split tensile strength of concrete were tested. The experiment result showed that moisture content, modulus of elasticity and strength of concrete is conversely related to the temperature and modulus of elasticity was directly related to the compressive strength of concrete in both testing of moisture and temperature [46].

A three component thermal storage system combining sensible and latent heat storage is a promising option for application in DSG power plants. Both storage principles have been proven suitable for the relevant temperature level. A latent heat storage test module with 140 kg  $\text{NaNO}_3$  has demonstrated the sandwich concept for enhancement of heat transfer as a possibility to achieve high discharge power even at elevated temperatures in spite of the low heat conductivity of PCMs. After 172 cycles about the melting temperature of 306°C, no degradation was detected. Investigations of concrete aggregates and oven experiments carried out on concrete cubes followed by strength measurements support the hypothesis that high-temperature concrete could serve as sensible heat storage up to 500°C. A storage system was designed with a total storage capacity of approximately 1 MW h combining a PCM storage module for evaporating water and a concrete storage module for superheating steam. Both storage modules were prefabricated and only the storage material (concrete and sodium



nitrate) was poured/filled on site. The system underwent a comprehensive testing in a test-loop specially erected at the power plant Litoral of Endesa in Carboneras, Spain; the commission started in May 2010, where the concrete storage module was successfully heated over 350°C for the first time, expelling the excess water of the concrete. For large scale implementation of such a storage system, the single storage modules will be increased to a feasible module size. Then, the required storage capacity for the evaporation section will be built by connecting PCM modules in parallel. For the superheating section, the required number of concrete modules can be set up in series and in parallel. This modular set up makes it easy to deliver any required storage capacity [47].

The environmental impact was compared among the three different thermal energy storage (TES) systems for the comparative life cycle assessment of thermal energy storage systems for solar power plants research. Sensible heat storage both in solid (high temperature concrete) and liquid (molten salts) thermal storage media, and latent heat storage which uses a phase change material (PCM) were developed using the life cycle assessment (LCA) methodology. This research focused on the analyzing the global environmental impact produced during the manufacturing and operation phase of each storage system. Some hypothetical scenarios are studied to point out the differences between each TES system using the LCA methodology. The system based on solid media storage shows the lowest environmental impact per kWh stored on all three systems compared because of the simplicity design. In addition, the two tank molten salt storage system (liquid media) shows the highest environmental impact per kWh stored because it needs more material and complex equipment [48].

The effect of different water/cement (w/c) ratio and graphite content were investigated by testing compressive strength and thermo physical properties of hardened aluminate cement pastes and calculate the volume heat capacity. It was found that thermal conductivity and volume heat capacity were improved with the decrease of w/c and the increase of graphite content. Water/cement ratio (w/c) is a key factor affecting thermal properties of pastes. After the pure aluminate cement pastes and graphite composite pastes is heated treatment at 350°C for 6 h, compressive strength and thermal properties descended in a certain extent. At the German



Aerospace Center in Spain, high temperature concrete storage was tested at the temperature 200°C, 350°C and 370°C respectively. At 200°C, volume heat capacity is 2457 kJ/m<sup>3</sup>K, thermal conductivity is 1 W/m K and coefficient of thermal expansion is 9.3x10<sup>-6</sup>/K. At 350°C, volume heat capacity is 2519 kJ/m<sup>3</sup>K, thermal conductivity is 1 W/m K and coefficient of thermal expansion is 9.3x10<sup>-6</sup>/K. And at 370°C, volume heat capacity is 2475 kJ/m<sup>3</sup>K, thermal conductivity is 1.3 W/m K and coefficient of thermal expansion is 11.6x10<sup>-6</sup>/K [49].

In the experimental study on the thermal conductivity of concrete research, moisture content and aggregate volume fraction affected greatly on the thermal conductivity of concrete. Thermal conductivity decreases when the samples' temperature increases. And also water/cement ratio and types of admixture strongly affected thermal conductivity of mortar and cement paste. Morabito [40] also found that thermal conductivity decline linearly with temperature [50].

Al-Ostaz studied the effect of moisture content on the coefficient of thermal expansion of concrete. The results show that humidity and accumulated moisture content did not have a great impact on the coefficient of thermal expansion. The major control element is the aggregate type because concrete has approximately 70 % of aggregate in the mixture [51]. McHale also found the same concept by the study of the measurement and significance of the coefficient of thermal expansion of concrete in Rigid Pavement Design. Besides, he reported that related humidity, degree of paste, water-cement ratio, porosity and cement content are related function of the coefficient of thermal expansion [52]. Rein also found that the coefficient of thermal expansion of concrete depends on the concrete mix proportion and the degree of saturation [53].

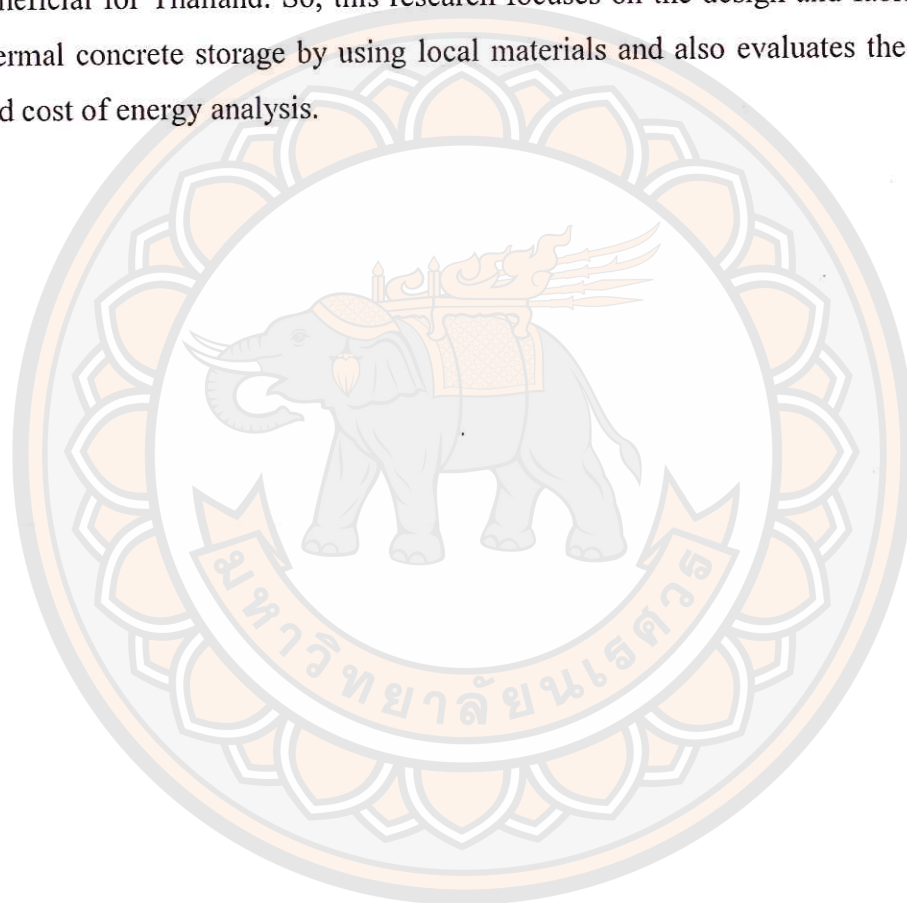
The impact of concrete structure on the thermal performance of the dual-media thermocline thermal energy storage (TES) tank which is very promising to be applied in concentrating solar power (CSP) systems is investigated. The lumped capacitance method is used since the introduction of corrected heat transfer coefficients between solid and fluid extends the validity of this method to large Biot numbers. The discharging performance of four typical concrete structures including the channel-embedded structure, the parallel-plate structure, the rod-bundle structure and the packed-bed structure is studied. The thermocline behaviors during the

discharging process for the four structures with the influences of feature size and fluid inlet velocity are analyzed, and the corresponding effective discharging efficiency and discharging time are reported. The results show that the concrete TES tanks with four different structures show different thermocline behaviors during the discharging process. The packed-bed structure gives the best discharging performance, followed by the rod-bundle structure, the parallel-plate structure and the channel-embedded structure sequentially. The discharging performance is also found to be influenced to some extent by the feature size and the fluid inlet velocity for all four different structures [54]. The research program economical concrete mixtures were developed that resisted temperatures up to 600°C. This temperature level represents a 50% increase over the operating temperature of current systems, which is approximately 400 °C. However, long-term testing of concrete is required to validate its use. At this temperature, the unit cost of energy stored in concrete (the thermal energy storage medium) is estimated at 0.88 - 1.00 \$/kW h thermal. These concrete mixtures, used as a thermal energy storage medium, can potentially change solar electric power output allowing production through periods of low to no insolation at lower unit costs [55].

The first concrete storage system was successfully tested by the German Aerospace Center (DLR) in Spain. But the cost of the storage system is still expensive. It is not suitable for Thailand. Therefore, Watchara Wongpanyo [38] studied concrete storage system for a parabolic trough power plant in Thailand. But the main parameters of storage system such as thermal conductivity and volume heat capacity were too low. For that reason, this research has been investigated the properties of concrete composition in order to improve the properties of the concrete storage which made of the local materials in Thailand. In addition, steel fiber or iron filling was significant included as the special material because especially, it improves the mechanical properties such as cracking. And it is also relatively influence on the thermal properties. Hence, it is expected to be valuable for the concrete thermal storage media.



This literature review argues that concrete storage for parabolic trough in power plant has potential because the cost is low. It is also available worldwide and is uncomplicated to process. Although inexpensive aggregates to the concrete are widely available, the concrete storage has not been developed easily as expected. To develop the large scale thermal concrete storage for parabolic trough power plant is very interesting. Especially, the development by using the local material materials could be beneficial for Thailand. So, this research focuses on the design and fabrication of the thermal concrete storage by using local materials and also evaluates the performance and cost of energy analysis.





## CHAPTER III

### RESEARCH METHODOLOGY

There are four main parts in the methodology of this research. Firstly, tests of the thermophysical properties of the concrete materials. Secondly, design of a thermal energy storage prototype. Thirdly, the experimental setup and testing of the thermal energy storage prototype. Finally, performance evaluation of the prototype, and cost of energy analysis. The detail of each topic is described as follows.

#### **Thermophysical properties of concrete materials test**

The thermo-physical properties of the material are important elements in the performance of concrete medium and its use for energy storage. These properties include density ( $\rho$ ), specific heat capacity ( $C_p$ ), thermal conductivity ( $k$ ) and the coefficient of thermal expansion ( $CTE$ ) [3]. The mechanical properties of the concrete material, such as compressive strength and permeability, must also be considered to ensure the long life time of the concrete storage unit. Overall, the primary feature of importance in the storage medium is the volumetric heat capacity [56]. A high volumetric heat capacity ( $\rho \times C_p$ ) reduces the required volume of the storage system and the number of metallic pipes inside the storage system. High thermal conductivity reduces the cost of the heat exchanger and insulation requirements and increases the dynamics in the system [45, 57]. Material with low thermal diffusivity will have a slow heat transfer rate and can store a larger amount of heat [54]. Accordingly, low thermal diffusivity is a necessary factor to achieve the high quality operation of the storage unit. CTE of the metallic pipes embedded in the storage system should match the CTE of the storage material and the value should not be too high. High CTE encourages cracking inside the concrete. Consequently, permeability testing was done as an indicator of the propensity for cracking inside the concrete.

### Materials selection

The concrete mixture used in this research was composed of water, sand, Portland cement, granite and iron filings. While there are many types of cement available, such as Portland cement, blended cement and expansive cement, Portland cement type I was used because it is commonly available cement with the compound composition range of C<sub>3</sub>S 45-55%, C<sub>2</sub>S 20-30%, C<sub>3</sub>A 8-12% and C<sub>4</sub>AF 6-10%. It is therefore readily available in the market at a reasonable price. There are many types of coarse aggregate used in concrete, as illustrated in Table 4. Among them, granite was selected because it has highest heat conductivity. Iron filings were included as a special material, for the purpose of decreasing the permeability of the concrete [35, 45, 58]. Adding iron filings has a considerable impact on the mechanical properties of concrete by reducing expansion and contracting of the concrete, reducing the propensity for cracking [55]. Iron filings also increase the thermal conductivity of the concrete. The water/cement ratio is also an essential factor affecting both the thermal and mechanical properties of concrete. By including the granite, and the iron filings, and carefully calculating the water/cement ratio, the optimum concrete composition was attained. This was essential for testing the efficiency and effectiveness of the thermal energy storage media. The quality of concrete is decided according to the strength of concrete, which is directly related to the structure of the hydrated cement paste.

The predicted thermal conductivity ( $k$ ) of concrete is assumed to be derivable based on the volumetric ratio and the thermal conductivity ( $k$ ) of the ingredients included in the hydrated product [40, 42]. The calculation function for  $k$  is:

$$k(t) = n_r k_r + n_s k_s + n_{sf} k_{sf} + n_w(t) k_w + n_c(t) k_c + n_{hp}(t) k_{hp} \quad (47)$$

$k$  = Thermal conductivity of concrete (W/m °C)

$k_r$  = Thermal conductivity of coarse aggregate (W/m °C)

$k_s$  = Thermal conductivity of sand (W/m °C)

$k_{sf}$  = Thermal conductivity of steel fiber (W/m °C)

$k_w$  = Thermal conductivity of water (W/m °C)



- $k_c$  = Thermal conductivity of cement (W/m °C)  
 $k_{hp}$  = Thermal conductivity of hydrated product (W/m °C)  
 $n$  = volumetric of concrete  
 $n_r$  = volumetric ratio of coarse aggregate  
 $n_s$  = volumetric ratio of sand  
 $n_{sf}$  = volumetric ratio of steel fiber  
 $n_w$  = volumetric ratio of water  
 $n_c$  = volumetric ratio of cement  
 $n_{hp}$  = volumetric ratio of hydrated product

The prediction of specific heat ( $C_p$ ) of concrete is assumed to be derivable based on the mass ratio and specific heat ( $C_p$ ) of the ingredients included in the hydrated product [39, 42]. The calculation function for  $C_p$  is:

$$C_p(t) = m_r c_r + m_s c_s + m_{sf} c_{sf} + m_w(t) c_w + m_c(t) c_c + m_{hp}(t) c_{hp} \quad (48)$$

- $C_p$  = Specific heat of concrete (J/kg °C)  
 $c_r$  = Specific heat of coarse aggregate (J/kg °C)  
 $c_s$  = Specific heat of sand (J/kg °C)  
 $c_{sf}$  = Specific heat of steel fiber (J/kg °C)  
 $c_w$  = Specific heat of water (J/kg °C)  
 $c_c$  = Specific heat of cement (J/kg °C)  
 $c_{hp}$  = Specific heat of hydrated product (J/kg °C)  
 $m$  = mass of concrete (kg)  
 $m_r$  = mass ratio of coarse aggregate (kg/kg of concrete)  
 $m_s$  = mass ratio of sand (kg/kg of concrete)  
 $m_{sf}$  = mass ratio of steel fiber (kg/kg of concrete)  
 $m_w$  = mass ratio of water (kg/kg of concrete)  
 $m_c$  = mass ratio of cement (kg/kg of concrete)  
 $m_{hp}$  = mass ratio of hydrated product (kg/kg of concrete)



The prediction of the coefficient of thermal expansion (*CTE*) of concrete is assumed to be derivable based on the volumetric ratio, modulus of elasticity and coefficient of thermal expansion (*CTE*) of ingredients including the hydrated product [41, 42]. The calculation function for *CTE* is:

$$CTE(t) = \frac{n_p CTE_p(t) E_p(t) + n_r CTE_r(t) E_r + n_s CTE_s(t) E_s + n_{sf} CTE_{sf}(t) E_{sf}}{n_p E_p(t) + n_r E_r + n_s E_s + n_{sf} E_{sf}} \quad (49)$$

$CTE(t)$	=	coefficient of thermal expansion of concrete (micron/°C)
$CTE_p$	=	coefficient of thermal expansion of paste (micron/ °C)
$CTE_r$	=	coefficient of thermal expansion of coarse aggregate (micron/°C)
$CTE_s$	=	coefficient of thermal expansion of sand (micron/°C)
$CTE_{sf}$	=	coefficient of thermal expansion of steel fiber (micron/°C)
$n_p$	=	volumetric ratios of paste (m <sup>3</sup> / m <sup>3</sup> of concrete)
$n_r$	=	volumetric ratios of coarse aggregate (m <sup>3</sup> / m <sup>3</sup> of concrete)
$n_s$	=	volumetric ratios of sand (m <sup>3</sup> / m <sup>3</sup> of concrete)
$n_{sf}$	=	volumetric ratios of steel fiber (m <sup>3</sup> / m <sup>3</sup> of concrete)
$E_p$	=	modulus of elasticity of paste (MPa)
$E_r$	=	modulus of elasticity of coarse aggregate (MPa)
$E_s$	=	modulus of elasticity of sand (MPa)
$E_{sf}$	=	modulus of elasticity of steel fiber (MPa)

As mentioned before, sand and rock are two of the parameters most influencing the concrete properties. For the composition of the concrete used in the prototype, four different compositions were selected based on the sand/rock ratio. These four different compositions of concrete by mass are shown in Table 8. Ratio 1,2 and 3,4 is same amount of sand. While ratio 2 is higher of rock than ratio 1 and ratio 4 is higher of rock than ratio 3.

**Table 8 Four different compositions of concrete**

Ratio	Water/cement	Cement	Sand	Rock	Iron filing
1	0.6	1	1.5	2	0.05
2	0.6	1	1.5	3	0.05
3	0.6	1	2	3	0.05
4	0.6	1	2	4	0.05

**Table 9 Prediction of thermal properties of concrete**

Ratio	Thermal Conductivity (W/m °C)	Specific heat (J/kg °C)	Coefficient of thermal expansion (micron/°C)
1	2.17	1,097	9.695
2	2.34	1,041	9.207
3	2.23	1,027	9.207
4	2.35	994	8.937

The prediction show in Table 9, indicate that the best thermal properties were achieved using Ratio 2. This is due to the high thermal conductivity and good specific heat capacity achieved by this combination of materials.

#### **Test of concrete properties**

Concrete was being mixed for the samples to test the properties by the following steps.

1. Sand, water, cement, granite and iron fillings were mixed with related mass ratio inside the concrete mixture bowl, as illustrated in Figure 14.
2. Samples of the concrete mixture were then placed into molds of different sizes.
3. The concrete samples were dried at room temperature.
4. After drying, samples of the concrete in the blocks were cured, immersed in water at room temperature for 28 days.

5. The dried, cured concrete samples were then tested for their thermo-physical characteristics.



**Figure 14 Concrete aggregates**

The samples of concrete cubes were selected for testing. The sample concrete cubes were 0.03 m x 0.005 m x 0.005 m rectangular shape, 0.02 m x 0.02 m x 0.02 m cubic shape in Figure 15 were prepared for the coefficient of thermal expansion, thermal conductivity, specific heat, thermal diffusivity, and density testing. And volume heat capacity is calculated. After curing for 28 days, concrete samples were taken out from the water and put immediately into the plastic bags in order to send to the testing laboratory.





**Figure 15 Configuration of tested samples of concrete**

Thermal conductivity, specific heat and thermal diffusivity were tested by Hot Disk Thermal Constant Analyzer (Hot disk AB) at the National Metal and Materials Technology Center (MTEC), National Science and Technology Development Agency, Thailand. Two pieces of one ratio were used for testing per time. A 2.00 mm radius probe was used and a power of 0.15 W was applied to the specimens for 2 second.

Coefficient of thermal expansion was tested by Dilatometer (Netzsch: DIL 402) and at density was tested by bulk density method (ASTM C373-88) at Thailand Institute of Scientific and Technological Research (TISTR), Thailand. Two pieces of one ratio were used for testing per time.

#### **Design of thermal energy storage prototype**

Several facets have to be considered when deciding on the type and design of any thermal storage system. A key issue in the design of a thermal energy storage system is its thermal capacity. However, selection of the appropriate system also depends on many cost-benefit considerations, technical criteria and environmental criteria. When considering a TES with an embedded pipe, the cost of a system depends mainly on the following items: the storage material itself, the embedded pipes (heat exchanger) for charging and discharging the system and the cost of the space and/or enclosure for the TES. [13]

From a technical point of view, the most important requirements are: high energy density in the storage material (storage capacity), good heat transfer between heat transfer fluid (HTF) and storage medium (efficiency), mechanical and chemical stability of the storage material (must support several charging/discharging cycles), compatibility between HTF, heat exchanger and/or storage medium (safety), complete reversibility of a number of charging/ discharging cycles (lifetime), low thermal losses, and ease of control. And finally, the most important design criteria from the point of view of technology are: operation strategy, maximum load, nominal temperatures and specific enthalpy drop in loads, and integration into the power plant. All these criteria have to be considered when deciding on the type and design of thermal storage [1].

A thermal energy storage is composed of parallel pipes for the heat transfer fluid (HTF) passing through the unit. The pipes are separated and surrounded by the storage material. For the physical model, identical conditions in all pipes are assumed. For the model, the storage unit is composed of so called storage pipes. A storage tube is a hollow cylinder of storage material with diameter  $D_a$ , in contact with a tube of outer diameter  $d_a$ . The storage material is characterized by the thermal conductivity, the mass specific heat capacity and the density. Steel pipes are used for carrying the heat transfer fluid, transmission and distribution of heat and to withstand the pressure. In the model, the pipe is discretized in the axial direction in pipe elements. The pipe element is surrounded by the storage material in the form of a hollow cylinder. This hollow cylinder is also discretized in the radial direction.

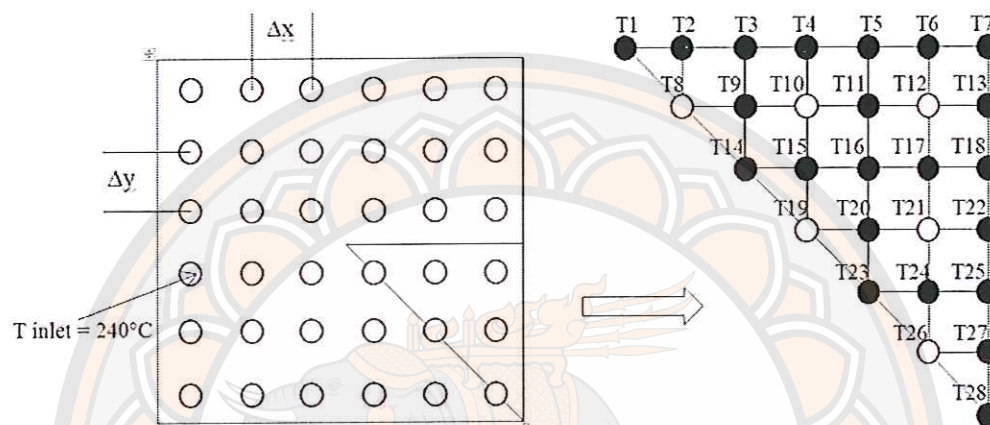
The modeling design of concrete storage system will calculate the heat transfer by using finite element procedure of two-dimensional, time-dependent heat conduction in Equation 36

$$\frac{\partial T}{\partial \tau} = \alpha \left\{ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right\} \quad (36)$$

Where  $\alpha$  is thermal diffusivity,  $m^2/s$ ,  $\alpha = \frac{k}{\rho C_p}$



To find the solution for equation 36, the finite element method will be applied for the temperature node as in Figure 17, there are the nodes of the temperature distribution of thermal energy storage and the nodes of constant at heat transfer fluid temperature.



**Figure 16 Two-dimensional model networks of unknown temperatures within the concrete storage system**

From figure 16, there are 28 equations to find the temperature distribution of concrete storage with describes as follow.

The interior nodes are T9, T11, T15, T16, T17, T20 and T24

The insulate boundary nodes are T2, T3, T4, T5 and T6

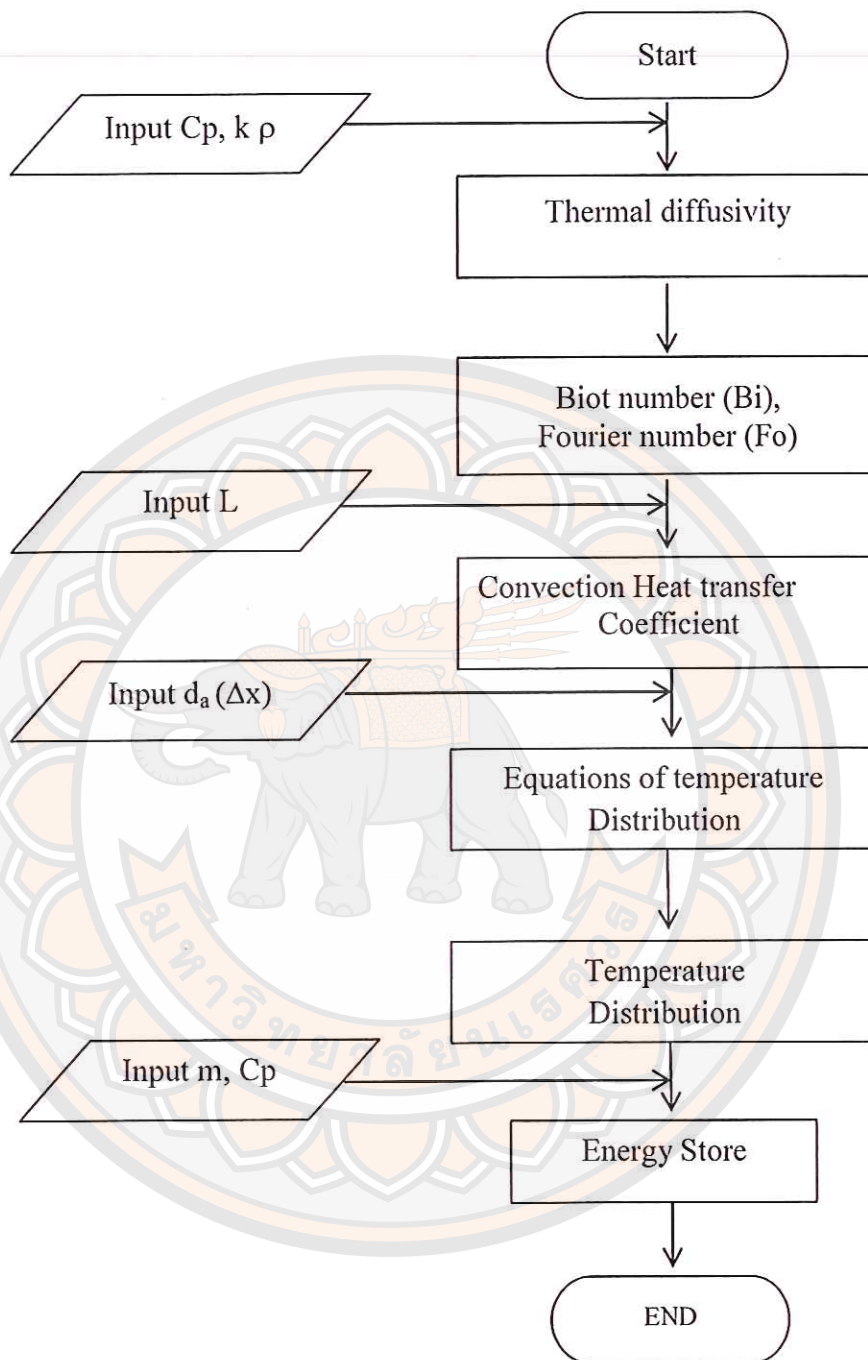
The convection boundary nodes are T13, T18, T22, T25 and T27

The condition boundary nodes are T1, T28, T14, T23 and T7

The steam inlet nodes are T8, T10, T12, T19, T21, T26 and which are considered to be constant at steam temperature.

The solution can be accomplished by 28 equations solving step by step as showing in the Figure 17.





**Figure 17** The modeling design flow chart

The simulation tool Matlab<sup>®</sup> was used to solve the numerical integration method and intensive differential equation for the development of the storage model (Appendix D). Beginning at the initial time and with initial conditions, they stepped

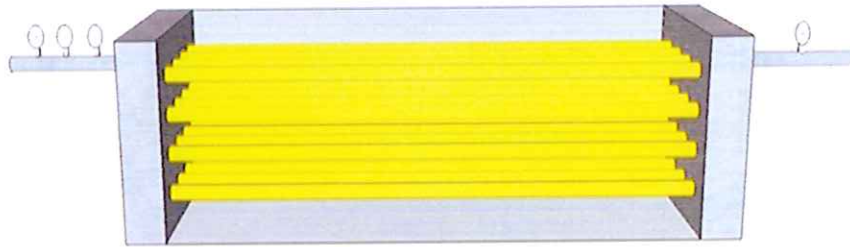
through the time interval and computed a solution at each time step. The user-friendly interface is developed in Matlab<sup>®</sup> using a tool called the Graphical User Interface Development Environment-GUIDE. This graphical user interface environment provides a reliable link between the user and the program itself. The Graphical User Interface (GUI) in this work call “NSimulation”.

### **Experimental setup and procedures**

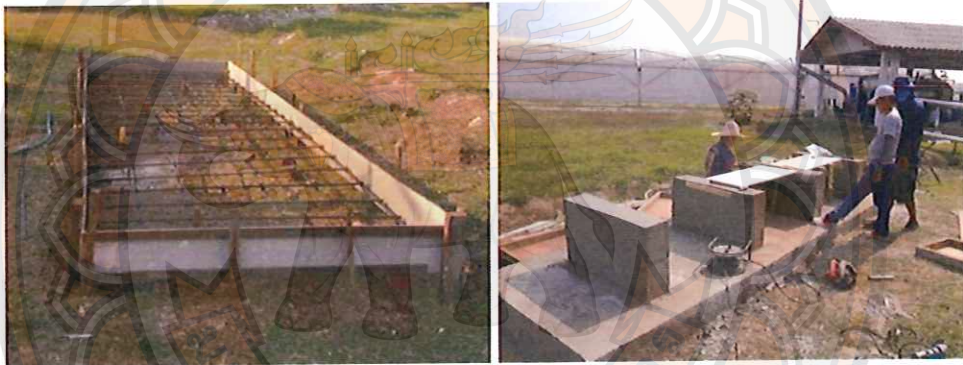
The thermal energy storage prototype was composed of pipes embedded in a concrete storage block. The embedded pipes are used for transporting and distributing the heat transfer medium while sustaining the pressure. The concrete storage stores the thermal energy as sensible heat. A special interface material was installed to reduce the friction between the concrete and the pipes due to the mismatch of thermal expansion.

The heat exchanger was composed of 16 pipes of high-temperature steel with the inner diameter of 12 mm and wall thickness of 7 mm. They were distributed in a square arrangement of 4 by 4 pipes with a separation of 80 mm. The storage prototype had the dimensions of 0.5 x 0.5 x 4 m. The heat exchanger structure of thermal energy storage prototype is shown in Figure 18 and the drawing is shown in Appendix F.

The construction began with the foundation settled as shown in Figure 19 and followed by putting up the piping system as shown in Figure 20. In order to record data for energy balances, the piping system was equipped with numerous of sensors. The mass flow as well as the saturated steam temperature and pressure were measured. The prototype was equipped with 56 thermocouples distributed within the storage material, on the embedded pipes and the header pipe. After the installation of additional reinforcement and measuring equipment, a formwork was installed and the storage space was filled with the thermal storage concrete as shown in Figure 22. The storage prototype was then covered by insulation on all sides and top and bottom as in Figure 23. The P&ID diagram of the process is shown in Figure 24.



**Figure 18** The heat exchanger structure of thermal energy storage prototype



**Figure 19** Foundation



**Figure 20** Piping system





**Figure 21 Install sensors and concrete materials**



**Figure 22 Thermal Energy Storage Prototype with insulation**

The experimental investigation was conducted at the Energy Park of the School of Renewable Energy Technology (SERT) at Naresuan University, Phitsanulok, Thailand. As the startup procedure, prior to the experimental processes proper, most of the water contained in the concrete was expelled by heating the concrete storage prototype from ambient temperature to 180 °C. During the process, the water evaporates and there was a buildup of vapor pressure within the concrete. During this process there was a buildup of vapor pressure within the concrete which needed to be carefully monitored to avoid damage to the concrete.

The subsequent operating conditions of the concrete storage prototype were:

1. Heat transfer fluid (Saturated steam)
2. Maximum internal pressure (10 bar)
3. Maximum temperature: up to 180 °C
4. Test temperature range between 110-180 °C
5. Mass flow rate: 0.009, 0.012 and 0.014 kg/s



**Figure 23 Data collecting of Thermal Energy Storage Prototype**

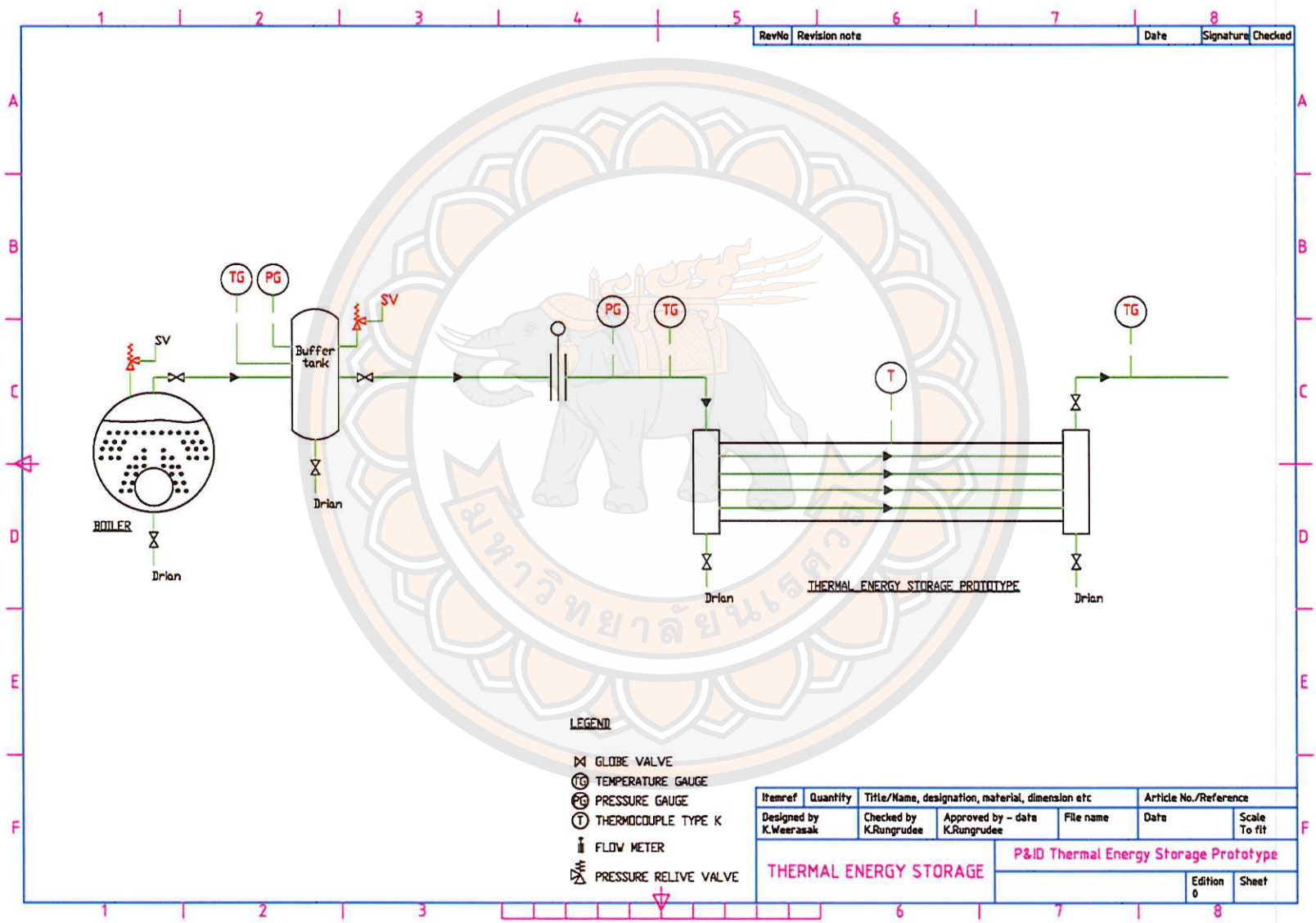
### **Data collecting**

A thermal energy storage unit is used to store the heat energy. There are two processes of thermal energy storage; the charging process is to store heat in a storage prototype that supplies the heat source and the discharging process is to extract heat from storage prototype.

The saturated steam from the boiler will work as the heat transfer fluid and will move through the pipes immersed in the storage prototype. The saturated steam inlet temperature was maintained about 180 °C. The storage prototype was covered all side by insulation. The K type thermocouples were embedded in the concrete to measure the temperature distribution. The recorder was collected data from all instruments as well as pressure and flow rate.



Figure 24 P&ID diagram





### Performance evaluation

Performance of thermal energy storage can be evaluated by energy and exergy analysis.

Energy analysis is based on the first law of thermodynamics, which is related to the conservation of energy. It can explain very well in quantity of energy. But exergy analysis under the second of thermodynamic can explain clearly in quality of energy. Second law analysis is a method that uses the conservation of mass and conservation of energy principles together with the entropy for the analysis, design and improvement of energy systems. Second law analysis is a useful method; to complement not to replace energy analysis [56].

For the first law and the second law of thermodynamics, the efficiencies based on energy and exergy can be defined for charging and discharging periods.

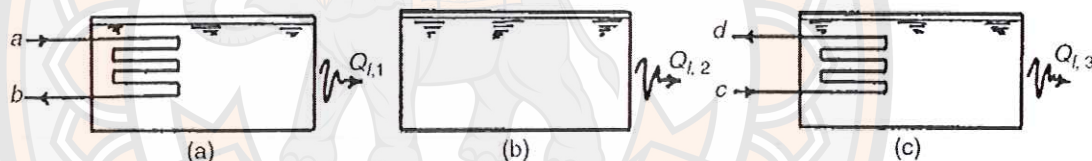


Figure 25 The three stage in a simple heat storage process: charging period (a), Storage period (b), and discharging period (c) [13]

#### 1. Conceptual Balance Equations of Energy and Exergy

An energy balance for the overall storage process can be written as [13]

$$\text{Energy input} - (\text{Energy recovered} + \text{Energy loss}) = \text{Energy accumulation} \quad (50)$$

$$\text{or} \quad (H_a - H_b) - [(H_d - H_c) + Q_l] = \Delta E$$

$$\dot{E}_{in} - [\dot{E}_{out} + \dot{Q}_l] = \frac{dE_{useful}}{dt}$$

$$\text{When} \quad \dot{E}_{in} = \sum_{i=1}^f \dot{m}_i (h_{a,i} - h_{b,i})$$

$$\dot{E}_{out} = \sum_{e=1}^f \dot{m}_e (h_{d,e} - h_{c,e})$$

$$\dot{Q}_l = \sum_{i=1}^3 \dot{Q}_{i,j} = \sum_{j=1}^3 \int_{T_j=1}^{T_j=3} d\dot{Q}_j$$

where  $h_a, h_b, h_c$  and  $h_d$  are the total enthalpies of the flows at states  $a, b, c$ , and  $d$ , respectively; and  $Q_l$  denotes the heat losses during the process and  $\Delta E$  the accumulation of energy in the TES.  $(h_a - h_b)$  represents the net heat delivered to the TES and  $(h_d - h_c)$  the net heat recovered from the TES.

An overall exergy balance can be written as

$$\text{Exergy input} - (\text{Exergy recovered} + \text{Exergy loss}) = \text{Exergy accumulation} \quad (51)$$

or 
$$[(\epsilon_a - \epsilon_b) - [(\epsilon_d - \epsilon_c) + X_l]] = \Delta \Xi$$

$$\dot{\epsilon}_{in} - [\dot{\epsilon}_{out} + \dot{X}_l] = \frac{d\epsilon_{useful}}{dt}$$

When

$$\dot{\epsilon}_{in} = \sum_{i=1}^f \dot{m}_i (\epsilon_{a,i} - \epsilon_{b,i})$$

$$\dot{\epsilon}_{out} = \sum_{e=1}^f \dot{m}_e (\epsilon_{d,e} - \epsilon_{c,e})$$

$$\dot{X}_l = \sum_{i=1}^3 \left(1 - \frac{T_0}{T_j}\right) \dot{Q}_{i,j} = \sum_{j=1}^3 \int_{T_j=1}^{T_j=3} \left(1 - \frac{T_0}{T_j}\right) d\dot{Q}_j$$

where  $\epsilon_a, \epsilon_b, \epsilon_c$  and  $\epsilon_d$  are the exergies of the flows at states  $a, b, c$ , and  $d$ , respectively; and  $X_l$  denotes the exergy loss associated with  $Q_l$ ; and  $\Delta \Xi$  is the exergy accumulation.  $(\epsilon_a - \epsilon_b)$  represents the net exergy input and  $(\epsilon_d - \epsilon_c)$  is the net

exergy recovered. The exergy content of the flow at the states  $k = a, b, c, d$  is evaluated as

$$\epsilon_k = [(H_k - H_0) - T_0(s_k - s_0)] \quad (52)$$

where  $\epsilon_k$ ,  $h_k$ , and  $s_k$  denote the exergy, enthalpy, and entropy of state  $k$ , respectively, and  $h_0$  and  $s_0$  the enthalpy and the entropy at the temperature  $T_0$  and pressure  $P_0$  of the reference environment.

$$X_{l,j} = \sum_{j=1}^f \left(1 - \frac{T_0}{T_j}\right) Q_{l,j} \quad (53)$$

where  $T_j$  represents a mean temperature within the storage prototype for period  $j$  and  $X_l$  denotes the exergy loss associated with  $Q_l$  [8].

## 2. Charging

An energy balance for the charging period can be written as follows [13]

$$\text{Energy input} - \text{Energy loss} = \text{Energy accumulation} \quad (54)$$

$$\text{or } (H_a - H_b) - Q_{l,1} = \Delta E_1 \quad (54a)$$

$$\Delta E_1 = E_{f,1} - E_{i,1} \quad (55)$$

$E_{i,1}$  and  $E_{f,1}$  denote the initial and the final energy of the TES for the charging period and  $Q_{l,1}$  denotes the heat losses during the period.  $h_a$  and  $h_b$  denote the enthalpy of state  $a$  and  $b$ .

An exergy balance for the charging period can be written as

$$\text{Exergy input} - \text{Exergy loss} = \text{Exergy accumulation} \quad (56)$$

$$(\epsilon_a - \epsilon_b) - X_{l,1} = \Delta \Xi_1 \quad (56a)$$

$$\Delta \Xi_1 = \Xi_{f,1} - \Xi_{i,1} \quad (57)$$



And  $\Xi_{i,1}$  and  $\Xi_{f,1}$  are the initial and the final exergy of the TES for the charging period.

### 3. Discharging

An energy balance for the discharging period can be written as [7]

$$-(\text{Energy recovered} + \text{Energy loss}) = \text{Energy accumulation} \quad (58)$$

$$-[(H_d - H_c) + Q_{l,3}] = \Delta E_3 \quad (58a)$$

$$\Delta E_3 = E_{f,3} - E_{i,3} \quad (59)$$

$E_{i,3}(=E_{f,2})$  and  $E_{f,3}$  denote the initial and final energies of the store for the discharging period. The quantity in square brackets represents the energy output during discharging.

An exergy balance for the discharging period can be written as follows:

$$-(\text{Exergy recovered} + \text{Exergy loss}) = \text{Exergy accumulation} \quad (60)$$

$$-[(\epsilon_d - \epsilon_c) + X_{l,3}] = \Delta \Xi_3 \quad (60a)$$

$$\Delta \Xi_3 = \Xi_{f,3} - \Xi_{i,3} \quad (61)$$

and  $\Xi_{i,3}(=\Xi_{f,2})$  and  $\Xi_{f,3}$  denote the initial and final exergies of the store for the discharging period.

### 4. Energy and Exergy Efficiency

Energy efficiency is the ratio of energy recovered from the thermal energy storage during discharging to the total energy input during charging. The energy efficiency ( $\eta$ ) can be defined as [7];

$$\begin{aligned} \eta &= \frac{\text{Energy recovered from TES during discharging}}{\text{Energy input to TES during charging}} \\ &= \frac{h_d - h_c}{h_a - h_b} = 1 - \frac{Q_l}{h_a - h_b} \end{aligned} \quad (62)$$

Exergy efficiency is the ratio of exergy recovered from the thermal energy storage during discharging to the total exergy input during charging. The exergy efficiency ( $\psi$ ) can be present as;

$$\begin{aligned}\psi &= \frac{\text{Exergy recovered from TES during discharging}}{\text{Exergy input to TES during charging}} \\ &= \frac{\epsilon_d - \epsilon_c}{\epsilon_a - \epsilon_b} = 1 - \frac{X_l}{\epsilon_a - \epsilon_b}\end{aligned}\quad (63)$$

### Cost of Energy

The economics of thermal energy storage system of solar thermal power plant are particularly complex with much inevitable uncertainty due to several factors. The principal reason for using solar energy for heating is to reduce the energy cost. Therefore, an economic analysis must be carried out to determine whether a particular system is economically advantageous for a particular project. In order to decide the acceptability of this thesis, the economic analysis as a cost of energy is considered.

The cost of thermal energy (typically baht/kWh<sub>t</sub>, Euro/kWh<sub>t</sub>, or \$/MWh<sub>t</sub>) is a calculation of the cost of thermal energy production. It includes the initial capital, discount rate, as well as the costs of continuous operation and maintenance. This type of calculation assists policy makers, researchers and others to guide discussions and decision making.

## CHAPTER IV

### RESULTS AND DISCUSSION

In this chapter, the results will be mentioned in four main parts. Firstly, properties of concrete. Secondly, design of thermal energy storage prototype. Thirdly, performance evaluation of the thermal energy storage prototype. Finally, cost of energy analysis. The detail of each topic is described as follows.

#### Properties of concrete

The thermophysical properties of the concrete samples have shown in Table 10.

**Table 10 The thermo-physical properties of the tested concrete materials in this research and the concrete storage materials developed by other researcher**

properties	Sample 1	Sample 2	Nhine (2014) [57]	Watchara (2009) [38]	DLR (2008) [37]
Density(kg/m <sup>3</sup> )	1820.00	1870.00	2557.00	1798.70	2750.00
Specific heat capacity (J/kg K)	1538.00	1430.00	1062.00	1015.00	916.00
Volume heat capacity (kJ/m <sup>3</sup> K)	2799.16	2674.10	2715.53	1825.68	2519.00
Thermal conductivity (W/m K)	1.03	0.97	1.93	0.40	1.00
CTE (1/K,10 <sup>-6</sup> )	8.21	10.39	6.67	N/A	N/A

Regarding the volumetric ratio of concrete mix materials, the proportion of cements and water are the same for all samples while the proportion of sand and rock are different, this affects the thermal properties, mass, and density of each sample. Due to the different proportion of sand and rock this has an effect on the mass of the concrete. This also affects inconstant density of the concrete. The testing result of the thermal conductivity, specific heat and thermal diffusivity of the concrete at National Metal and Materials Technology Center (MTEC), National Science and



Technology Development Agency, Thailand is shown in Appendix B. Testing result of the coefficient of thermal expansion and density of concrete at Thailand Institute of Scientific and Technological Research (TISTR), Thailand is shown in Appendix C.

The result displayed in Table 10 shows the thermo-physical properties of two concrete samples tested in this work, the concrete tested by Nhine (2014), the concrete tested by Watchara (2009) and the concrete tested by DLR Laboratory (2008). The interesting properties are the density, the specific heat capacity, the thermal conductivity and the volumetric heat capacity which is the product of the density and the specific heat. It is seen that although the density and thermal conductivity of both tested samples are not significantly different, the specific heat of sample 1 are clearly higher than that of the remaining samples. Especially its volumetric heat capacity which is the important parameter of solid thermal energy storage is also distinctly higher. When sample 1 is compared with the tested concrete from other researchers, the results show that the specific heat and the volumetric heat capacity are higher than other researchers, the density of both samples is lower than the concrete from Nhine and DLR Laboratory, and the thermal conductivity is similar with the concrete of DLR Laboratory but lower than the concrete from Nhine. Consequently, we used the result of sample 1 to design a thermal energy storage prototype.

### **Design of thermal energy storage prototype**

The simulation results are based on a variation of pipe spacing for the rough evaluation of the storage size.

#### **Amount of Stored Energy**

Sensible heat storage systems use the heat capacity and the change in temperature of material during the process of charging and discharging. The temperature of the storage material rises when energy is absorbed and drops when energy is withdrawn. Sensible heat,  $Q$  is stored in material of mass ( $m$ ) and specific heat ( $C_p$ ) by raising the temperature of the storage material from  $T_1$  to  $T_2$  and is expressed by Equation 1:

$$E = m \int_{T_1}^{T_2} C_p dT$$

$$E = v \int_{T_1}^{T_2} \rho C_p dT$$

Where  $\rho$  and  $v$  are density and a volume of the storage material.

**Table 11 The amount of energy stored in thermal energy storage material**

Distance	30 min (MJ)	60 min (MJ)	90 min (MJ)	120 min (MJ)	150 min (MJ)	180 min (MJ)
8 cm	36,900	50,100	56,600	60,650	63,450	65,550
10 cm	27,456	41,872	49,408	54,176	57,552	60,096
12 cm	16,002	27,900	34,560	38,916	42,030	44,406

Table 11 shows the result of energy stored in the thermal energy storage with variable of distance between centerline of flow channels ( $d_a$ ) of 8 cm, 10 cm and 12 cm. The sample model 8 cm has the highest energy storage with is slightly more than the energy storage by the sample with 10 cm and 12 cm. All samples display this trend in variables of time. However, the storage system with 10 cm requires the number of HTF pipes less than the unit with 8 cm so the price of storage system with 10 cm will be significantly cheaper than with 8 cm. The storage system with 12 cm is lower than others because the distance is a little far so the heat can't have proper distribution. Therefore, the sample model with the distance between centerline of 10 cm is the appropriate option for the Thermal Energy Storage Prototype in this work.

#### **Temperature distribution in thermal energy storage material of the distance between centerline of 10 cm**

Figure 26 shows the prediction of the mathematic model of the temperature distribution in all nodes (except HTF nodes) of the thermal energy storage when the HTF inlet temperature for charging was 180 °C and initial temperature was 90 °C with variable of time.



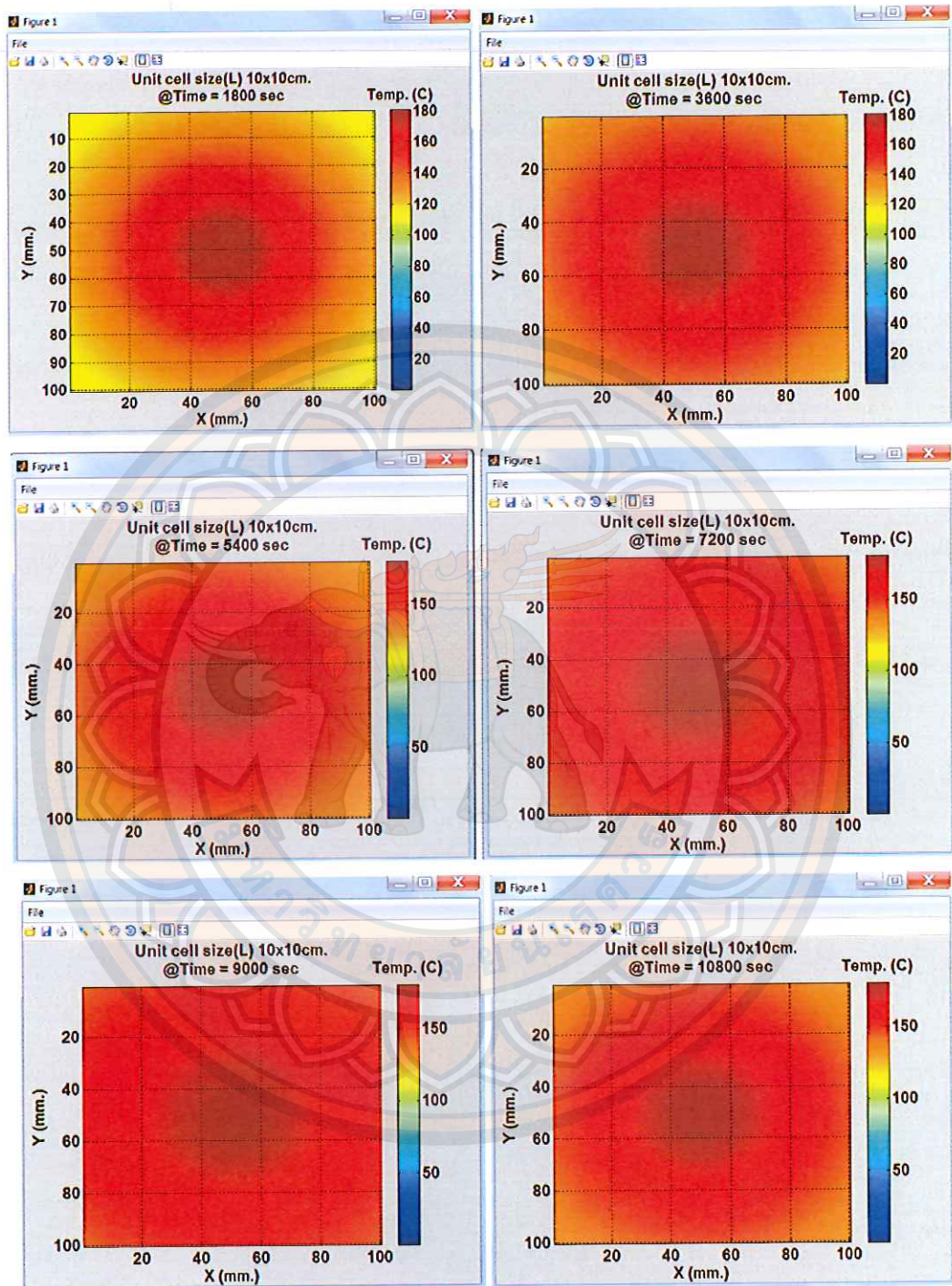


Figure 26 The temperature distribution all nodes in 30, 60, 90, 120, 150 and 180 minutes



## Performance test of the thermal energy storage prototype

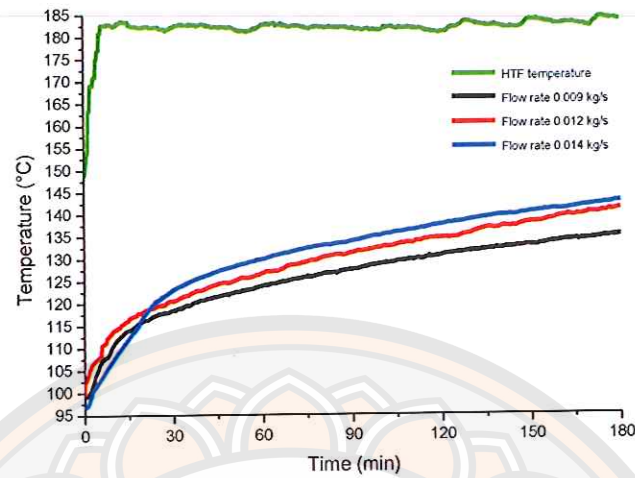
### Temperature profile charging and discharging

#### 1. Charging time

At the first time for startup, the concrete storage prototype is heat from the initial temperature to 180° C, some of water contain in the concrete is expelled. During this process, water evaporates and builds up a vapor pressure within the concrete. If the vapor pressure exceeds a critical value, serious damage may occur. The experiment of storage prototype was therefore closely monitored during initial heating up.

Having completed the startup procedure, the first charging experiment was commenced. As shown in Figure 27, the HTF inlet temperature and the storage prototype temperature were plotted for comparison of the HTF temperature and the average temperature of the storage prototype during this first charging experiment.

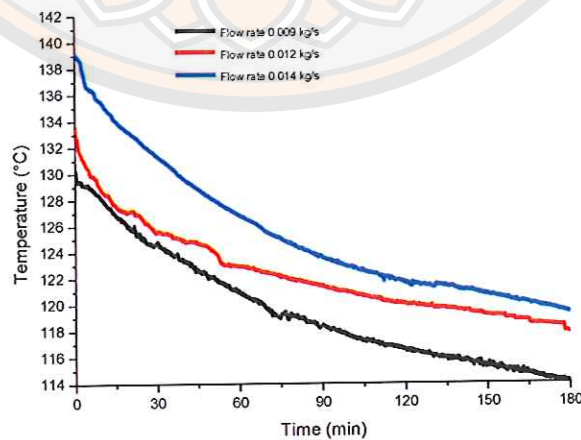
During the tests, HTF inlet mass flow rates were 0.009, 0.012 and 0.014 kg/s. The HTF inlet temperature to the storage prototype was manually increased very quickly to about 180 °C and then maintained at an almost constant level at about 180 °C for most of the charging process. Also, as shown in Figure 27, the storage prototype temperature slightly increased. This is shown at the different flow rates in Figure 27. Initially the volume average temperature of the storage bed rose rapidly and then rose slowly over the subsequent time of up to 180 minutes. This is due to the initial potential for heat conduction in the concrete. The heat conduction potential decreases with time as the storage bed gains heat of the HTF.



**Figure 27 Temperature of HTF inlet and storage prototype in charging experiment at 0.009, 0.012 and 0.014 kg/s.**

## 2. Discharging time

During the discharge process, various thermocouples were fixed inside the storage material in order to measure the temperature distribution. Figure 28 presents that the average temperature of the storage material inside the thermal storage prototype decreased over time. The decrease of storage bed temperature is due to the release of heat stored in storage prototype.

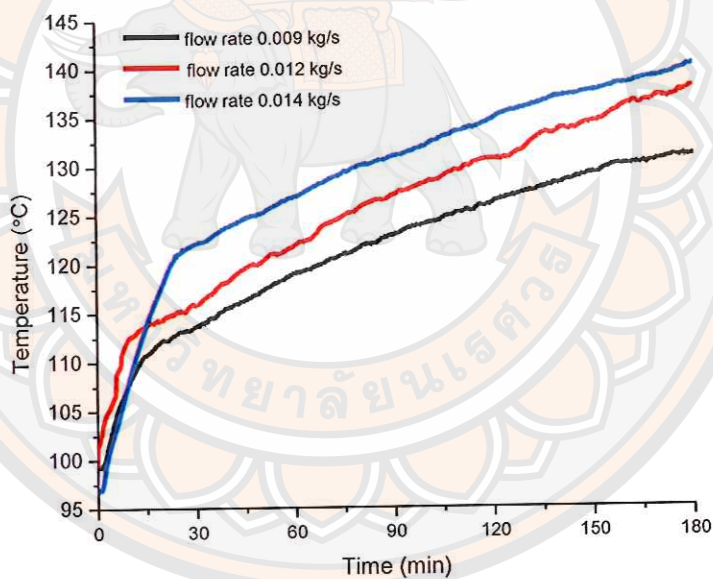


**Figure 28 Temperature of concrete storage prototype in discharging period at 0.009, 0.012 and 0.014 kg/s.**

## Temperature Distribution in Radius

### Temperature distribution on charging time

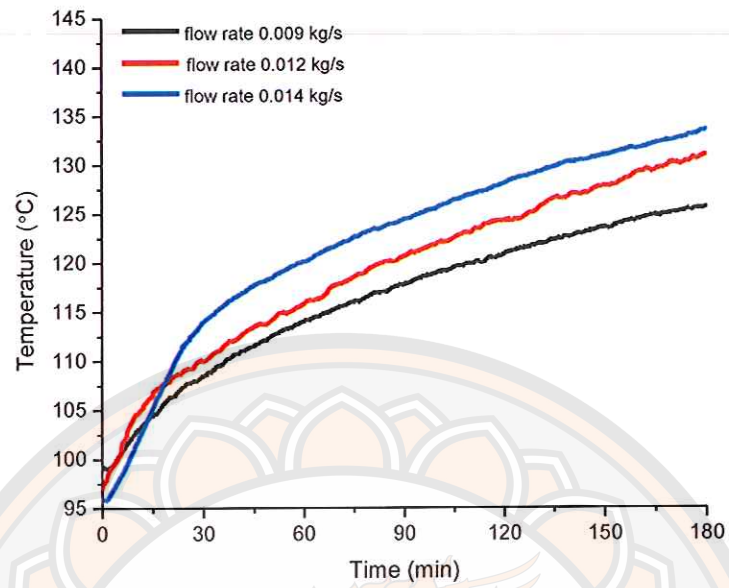
The radiant temperature distribution for thermal energy storage over charging time is shown in Figure 29 (a-c). Increasing the HTF flow rate increases the overall heat transfer coefficient enabling a faster exchange of heat which reduces the charging time. At higher HTF flow rates the time required to achieve a certain temperature decreased. At the HTF flow rate of 0.014 kg/s the temperature increase over time was greatest, followed by the flow rate of 0.012 kg/s with the flow rate of 0.009 kg/s being the slowest. Figure 29 (a-c) shows the comparisons of thermal distribution of temperature by thermal radiation for the three flow rates through the 1 cm, 2 cm and 3 cm from HTF pipe respectively.



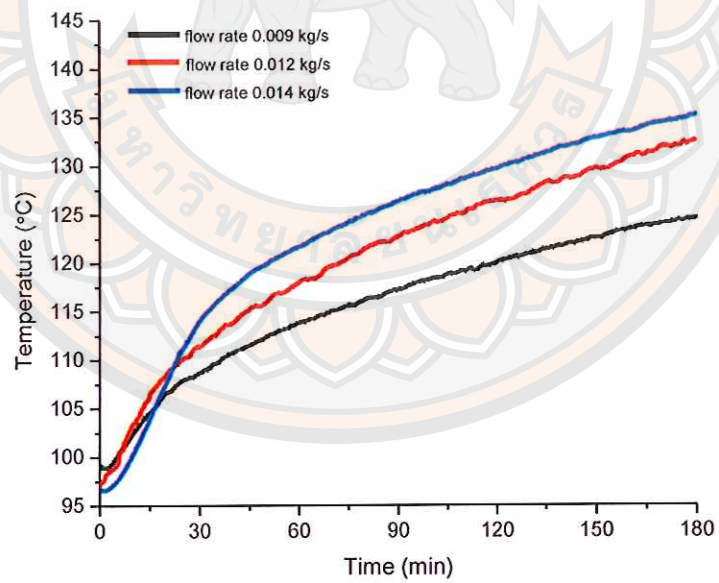
(a) Sensor is at 1 cm from HTF pipe

**Figure 29 Temperature distribution and flow rate on charging time of storage bed**





(b) Sensor is at 2 cm from HTF pipe

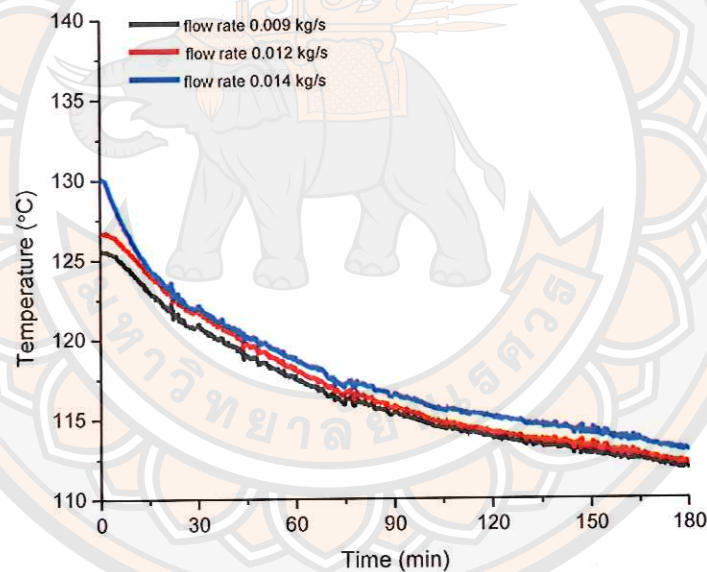


(c) Sensor is at 3 cm from HTF pipe

Figure 29 (cont.)

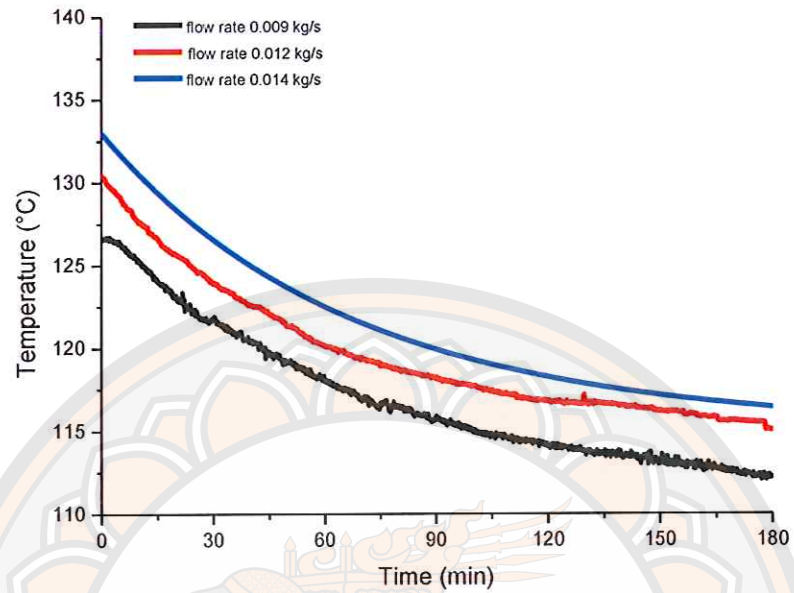
### Temperature distribution on discharging time

Heat discharge of the charged storage bed was initiated by passing HTF at a lower temperature ( $T_{inlet}$ ); The HTF receives the heat from the charged storage bed which decreases the storage bed temperature and also causes a rise in the HTF temperature along the bed. The radiant thermal distribution for thermal energy storage on discharging time is shown in Figure 31 (a-c), which shows the comparison of the thermal distribution of temperature of thermal radiation for the 1 cm, 2 cm and 3 cm HTF pipes. It can be seen that the decrease in discharging time of the storage bed with HTF at the flow rate of 0.014 kg/s was faster than that of 0.012 kg/s with 0.009 kg/s being the lowest value.

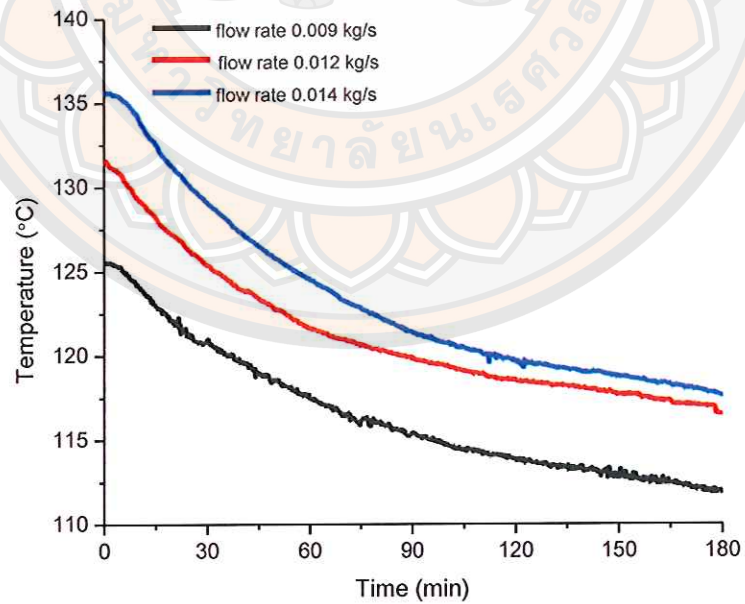


(a) Sensor is at 1 cm from HTF pipe

**Figure 30 Temperature distribution and flow rate on discharging time of storage bed**



(b) Sensor is at 2 cm from HTF pipe



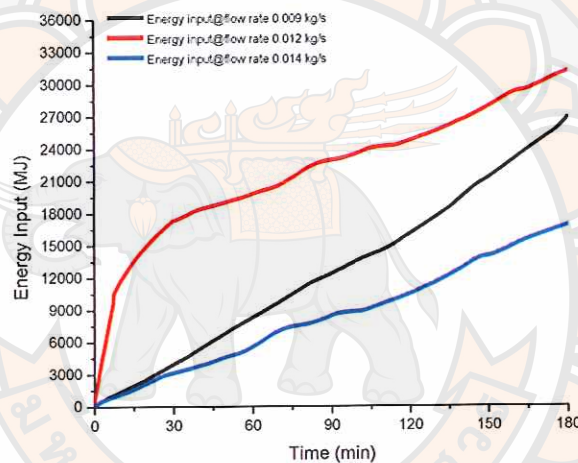
(c) Sensor is at 3 cm from HTF pipe

Figure 30 (cont.)



### Energy input

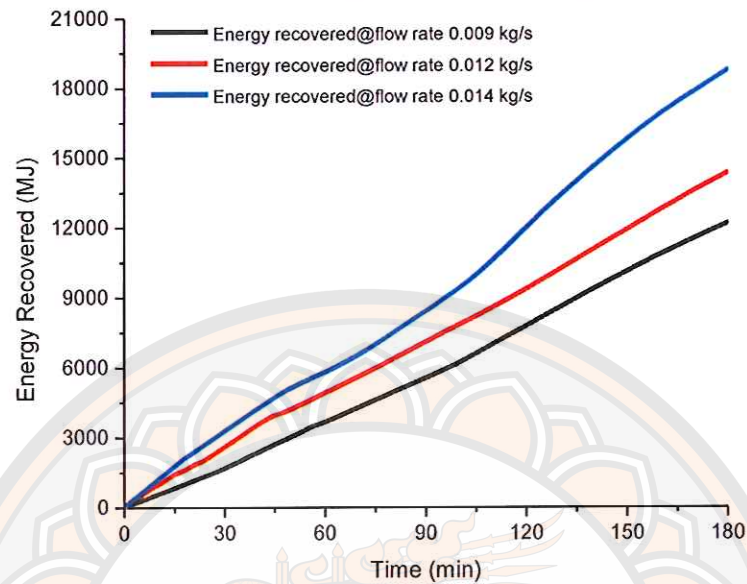
The thermal energy rates at the storage bed of the storage prototype are shown in Figure 31. The amount of thermal energy input in the storage materials at their respective charging times was calculated using Equation 54. The flow rate of 0.012 kg/s resulted in the fastest heat transfer from the pipe to the concrete, followed by the flow rate of 0.009 kg/s with 0.014 kg/s the lowest heat transfer value. The energy input at the various flow rates were 0.012 kg/s, 31,119 MJ, 0.009 kg/s, 26,891 MJ and 0.014 kg/s, 16,840 MJ



**Figure 31 Energy input during charging process at 0.009, 0.012 and 0.014 kg/s.**

### Energy recovered

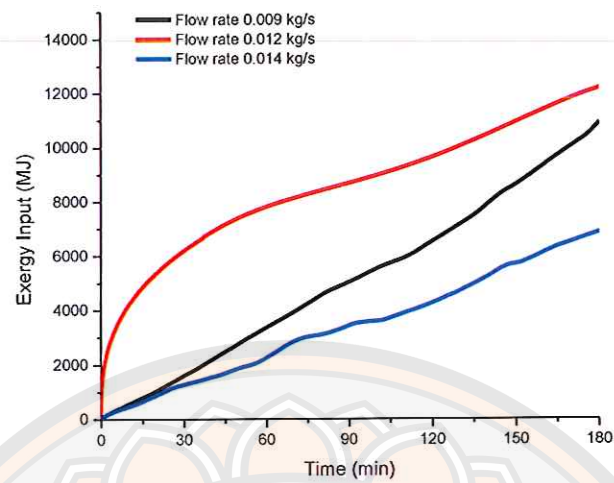
The thermal energy recovery rates of the storage bed are shown in Figure 32. The amount of thermal energy recovered was calculated using Equation 58. The calculations showed that at the flow rate of 0.014 kg/s the heat transfer from the concrete into the pipe was faster than at the flow rates of 0.012 and 0.009 kg/s. The energy recovered at each flow rate was, at 0.014 kg/s, 18,796 MJ, at 0.012 kg/s, 14,363 MJ and at 0.009 kg/s, 12,173 MJ.



**Figure 32 Energy recovered during discharging period at 0.009, 0.012 and 0.014 kg/s**

### Exergy input

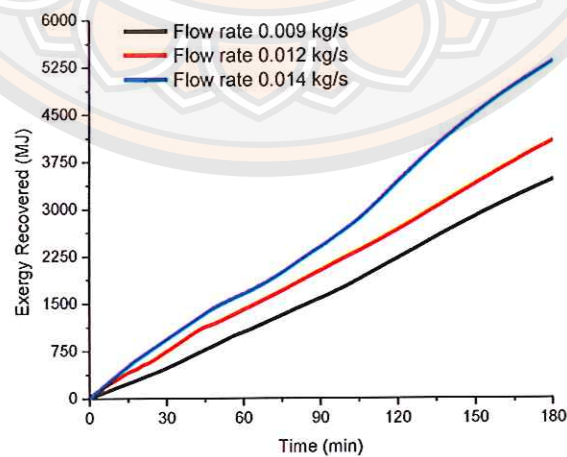
The exergy rates at the storage bed of the storage prototype were shown in Figure 33. The amount of exergy input in the storage materials at their respective charging times was calculated using Equation 56. The flow rate of 0.012 kg/s resulted in the fastest heat transfer from the pipe to the concrete, followed by the flow rate of 0.009 kg/s with 0.014 kg/s the lowest heat transfer value. The exergy input at the various flow rates were 0.012 kg/s, 12,244 MJ, 0.009 kg/s, 10,892 MJ and 0.014 kg/s, 6,882 MJ.



**Figure 33 Exergy input during charging process at 0.009, 0.012 and 0.014 kg/s.**

### Exergy recovered

The exergy recovery rates of the storage bed are shown in Figure 34. The amount of exergy recovered was calculated using Equation 60. The calculations showed that at the flow rate of 0.014 kg/s the heat transfer from the concrete into the pipe was faster than at the flow rates of 0.012 and 0.009 kg/s. The energy recovered at each flow rate was, at 0.014 kg/s, 5,342 MJ, at 0.012 kg/s, 4,071 MJ and at 0.009 kg/s, 3,462 MJ.



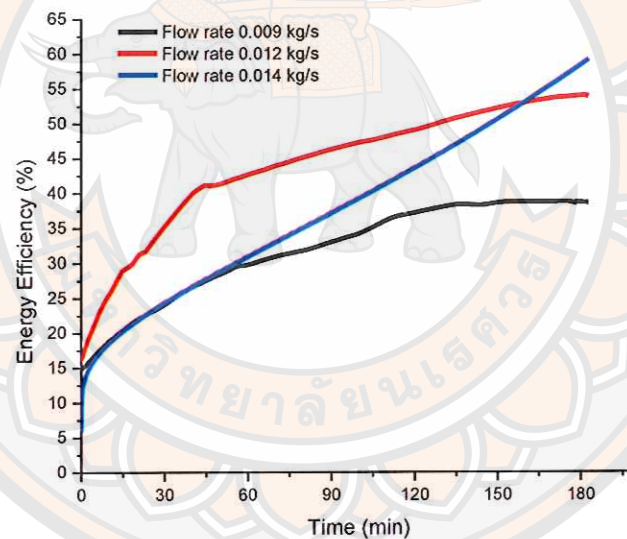
**Figure 34 Exergy recovered during discharging period at 0.009, 0.012 and 0.014 kg/s**



### Energy efficiency of Thermal Energy Storage prototype

The energy was degraded in the process of storage since it was extracted at a temperature lower than that at which it was previously stored. The energy efficiency of the storage bed was evaluated using Equation 62. Figure 35 shows the energy efficiency of the thermal energy storage prototype. The flow rate of 0.012 kg/s dramatically increased in the first 45 minute after which time it increased gradually while the efficiency at the flow rate of 0.014 kg/s increased sharply and at the flow rate of 0.009 kg/s increased slightly, while seeming to stabilize at a later time.

For 135 minutes of operation, the energy efficiency was 52% at the flow rate of 0.012 kg/s while the flow rate of 0.014 gave 47% energy efficiency. The flow rate of 0.009 kg/s gave 37% for 150 minutes operational time.

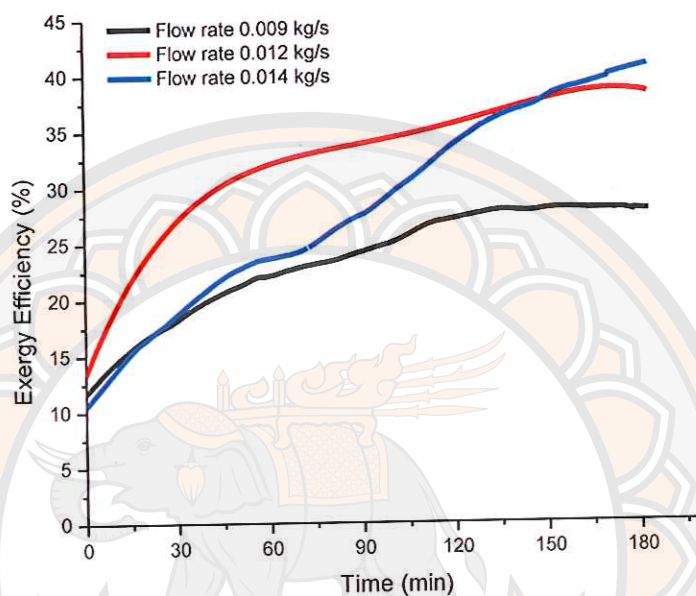


**Figure 35 Energy efficiency of Thermal Energy Storage Prototype at 0.009, 0.012 and 0.014 kg/s.**

### Exergy efficiency of Thermal Energy Storage prototype

The exergy efficiency of the storage bed was evaluated using Equation 63. Figure 36 shows the exergy efficiency of the thermal energy storage prototype. The flow rate of 0.012 kg/s dramatically increased in the first 45 minute after which time it increased gradually while the efficiency at the flow rate of 0.014 kg/s increased sharply and at the flow rate of 0.009 kg/s increased slightly, while seeming to stabilize

at a later time. For 120 minutes of operation, the exergy efficiency was 35% at the flow rate of 0.012 kg/s while the flow rate of 0.014 gave 32% exergy efficiency. The flow rate of 0.009 kg/s gave 27% for 135 minutes operational time.



**Figure 36 Exergy efficiency of Thermal Energy Storage Prototype at 0.009, 0.012 and 0.014 kg/s.**

### Cost of Energy

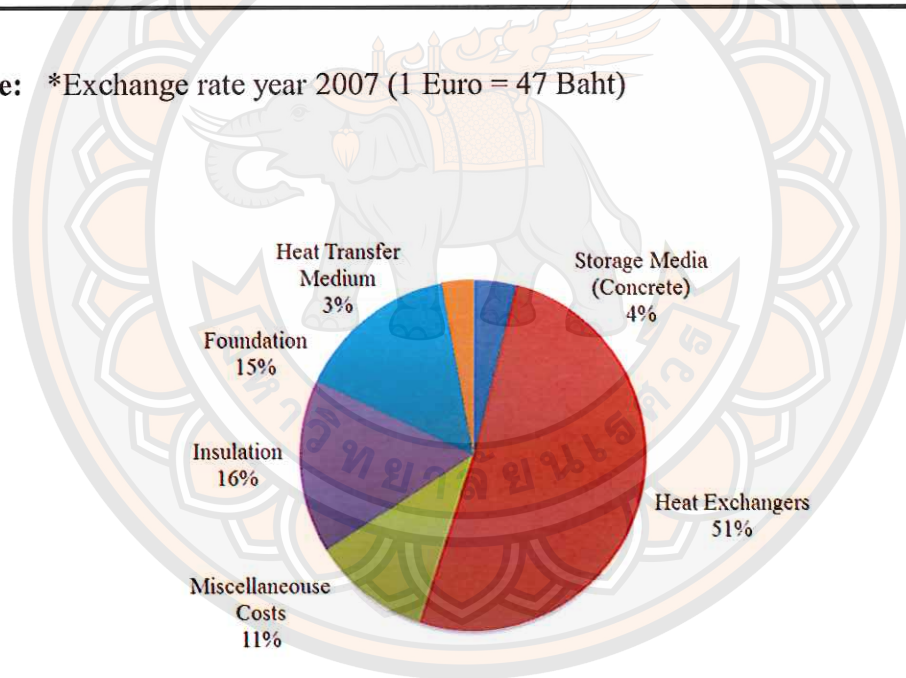
Table represents the comparisons of the capital cost structure of the thermal energy storage by using concrete material. The results show that the cost of the heat exchanger is more than 50% of total cost. The storage medium (concrete) in Thailand is inexpensive while the insulation and foundation is quite more expensive than the DLR cost.

The concrete storage technology is in a pre-commercial state from DLR with investment costs about 1,645 baht/kWh while the thermal energy storage prototype in this research with investment costs about 1,096 baht/kWh.

**Table 12 The Capital cost structure of thermal energy storage system**

Capital cost structure	TES prototype (%)	DLR (%)
Heat exchanger	51.68	57
storage media (concrete)	3.85	13
Miscellaneous cost	10.71	10
Insulation	15.77	8
Foundation	15.17	5
heat transfer medium	2.82	7
*Investment Cost (Baht/kWh)	1,096	1,645

**Note:** \*Exchange rate year 2007 (1 Euro = 47 Baht)

**Figure 37 Capital cost structure for Thermal Energy storage prototype**



## CHAPTER V

### CONCLUSION AND RECOMMENDATION

#### Conclusion

The conclusions are as follow

1. The investigations of the thermophysical properties of concrete aggregates of sample 1 in which the volumetric ratios are water(1): cement(1): sand(1.5): rock(3) with a density of  $1,820 \text{ kg/m}^3$ , specific heat of  $1,538 \text{ J/kg K}$ , thermal conductivity of  $1.03 \text{ W/m K}$  and coefficient thermal expansion of  $8.21 \text{ micron/K}$  is suitable for use in the thermal energy storage prototype because the specific heat and volumetric heat capacity are clearly higher than other samples investigated in this research. In comparison with the tested concrete from other researchers, the results show that the specific heat and the volumetric heat capacity are higher than Nhine, Watchara and DLR; the density is lower than the concrete from Nhine and DLR; and the thermal conductivity is similar with the concrete of DLR but lower than the concrete from Nhine.

2. The thermal energy storage was composed of 16 pipes of high-temperature steel with the inner diameter of 12 mm and wall thickness of 7 mm. They were distributed in a square arrangement of 4 by 4 pipes with a separation of 80 mm. The storage prototype had the dimensions of  $0.5 \times 0.5 \times 4 \text{ m}$ . The design of thermal energy storage varies the distance between centerline of flow channels ( $d_a$ ). The sample with  $d_a$  8 cm has the highest energy storage which is slightly more than the energy storage by the sample with  $d_a$  10 cm and  $d_a$  12 cm. Although the storage system with 10 cm requires the number of HTF pipes less than the unit with 8 cm but the price of storage system with 10 cm will be certainly cheaper than with 8 cm. The storage system with 12 cm is lower than others because the distance is too far so that the heat can't be distributed properly.

3. The performance evaluation of a thermal energy storage prototype is investigated in the part of charging/discharging. The experiment found that the increase or decrease in storage temperature depends on the HTF temperature, flow

rates, and initial temperature. The results showed that increasing the HTF flow rate increases the overall heat transfer coefficient, thereby enabling faster exchange of heat and reduces charging time. In charging period, the flow rate of 0.012 kg/s was the fastest heat transfer rate, the energy input to the concrete storage medium was 31,119 MJ and the exergy input was 12,244 MJ. In discharging period, the flow rate of 0.014 kg/s was the fastest heat transfer rate, the energy recovered over the discharging period was 18,796 MJ and the exergy recovered was 5,342 MJ. The energy efficiency of the thermal energy storage prototype at the flow rate of 0.012 kg/s was the best because it dramatically increased and gave 41% of energy efficiency in the first 45 minutes after which it continued to rise yet only gradually. Over 180 minutes of operation, the energy efficiency was 53% and the exergy efficiency was 38%.

4. From the cost of energy analysis, it is found that the capital cost structure of the thermal energy storage by using concrete material resulted that the cost of heat exchanger is more than 50% of total cost. The storage medium (concrete) in Thailand is inexpensive while the insulation and foundation is quite more expensive than the DLR cost. The concrete storage technology is in a pre-commercial state from DLR with investment costs about 1,645 baht/kWh while the thermal energy storage prototype in this research is cheaper with investment costs about 1,096 baht/kWh

### **Recommendation**

1. Since the concrete thermal storage system depends on the thermophysical properties, the significant points that should be considered are replacing cement with fly ash to improve workability, adding ratio of iron filling to improve thermal conductivity, putting silica fume to increase specific heat capacity.

2. The thermal cycles should be investigated more, when the thermal energy storage is operated for long period as the thermophysical properties of concrete may charge.

3. The thermal energy storage prototype should be tested with the parabolic trough plant in the experiment.

4. The results from this research can be a guideline for thermal storage system design for Commercial Solar Thermal Power Plant in Thailand.







**REFERENCES**

## REFERENCES

- [1] Gil, A., Medrano, Marc., Martorell, Ingrid., Lázaro, Ana., Dolado, Pablo., Zalba, Belén., et, al. (2010). State of the art on high temperature thermal energy storage for power generation. Part 1—Concepts, materials and modellization. **Renewable and Sustainable Energy Reviews**, 14(1), 31-55.
- [2] Sarı, A. and A. Karaipekli. (2007). Thermal conductivity and latent heat thermal energy storage characteristics of paraffin/expanded graphite composite as phase change material. **Applied Thermal Engineering**, 27(8–9), 1271-1277.
- [3] Laing, D., Steinmann, Wolf-Dieter., Tamme, Rainer., Richter and Christoph. (2006). Solid media thermal storage for parabolic trough power plants. **Solar Energy**, 80(10), 1283-1289.
- [4] Laing, D., Bahl, C., Bauer, T., Lehmann, D. and Steinmann, W. (2009). Test Results of Concrete Thermal Energy Storage for Parabolic Trough Power Plants. **Solar Energy Engineering-transactions of The Asme**, 131(4), 155-165.
- [5] Bai, F. and C. Xu. (2011). Performance analysis of a two-stage thermal energy storage system using concrete and steam accumulator. **Applied Thermal Engineering**, 31(14–15), 2764-2771.
- [6] Fernandez, A.I., Martí'nez, M., Segarra, M., Martorell, I. and Cabeza, L.F. (2010). Selection of materials with potential in sensible thermal energy storage. **Solar Energy Materials & Solar Cells**, 94, 1723-1729.
- [7] Sragovich, D. (1989). Transient analysis for designing and predicting operational performance of a high temperature sensible thermal energy storage system. **Solar Energy**, 43(1), 7-16.
- [8] Khare, S., Dell Amico, M., Knight, C., McGarry, S.(2013). Selection of materials for high temperature sensible energy storage. **Solar Energy Materials and Solar Cells**, 115(0), 114-122.

- [9] Tamme, R., D. Laing and W.-D. Steinmann. (2004). Advanced Thermal Energy Storage Technology for Parabolic Trough. **Journal of Solar Energy Engineering**, 126(2), 794-800.
- [10] Nandi, B.R., S. Bandyopadhyay and R., Banerjee. (2012). Analysis of high temperature thermal energy storage for solar power plant. In **The 2012 IEEE Third International Conference on Sustainable Energy Technologies (ICSET)** (pp. 438-444). Khamandu, Nepal: n.p.
- [11] Emerson E. John, W. Micah Hale and R. Panneer Selvam. (2011). Development of a High-Performance Concrete to Store Thermal Energy for Concentrating Solar Power Plants. In **ASME 2011 5<sup>th</sup> International Conference on Energy Sustainability** (pp. 523-529). Washington, DC, USA.: The American Society of Mechanical Engineers.
- [12] Huggins, R.A. (2010). **Energy storage**. New York, USA: Springer.
- [13] Ibrahim Dincer and Marc A. Rosen. (2011). **Thermal Energy Storage, systems and applications**. New York, USA: Springer.
- [14] German Aerospace Center (DLR). (2005). **Concentrating solar power for Mediterranean region**. N.P.: n.p.
- [15] International Energy Agency (IEA). (2010). **Technology roadmap of concentrating solar power**. N.P.: n.p.
- [16] Sukruedee Sukchai, W.C.-a., Sorawit Sonsaree, Rungrudee Boonsu, and Y.P. Joachim Krueger. (2015). Direct Steam Generation (DSG) Solar Thermal Power Plant in Thailand. In **The 19<sup>th</sup> Energy Symposium on Renewable Energy Technology Stralsund**. Germany: University of Applied science Stralsund.
- [17] Doerte Laing, C.B. and Michael Fiß. (2010). **Commissioning of a thermal energy storage system for direct steam generation in SolarPACES 2010**. Frankreich: Perpignan.
- [18] Doerte Laing, C.B., Thomas Bauer, Dorothea Lehmann and Wolf-Dieter Steinmann. (2011). Thermal energy storage for direct steam generation. **Solar Energy**, 85(4), 627-633.



- [19] Frank S. Barnes, J.G.L. (2011). **Large energy storage system handbook**.  
N.P.: n.p.
- [20] Wolf-Dieter Steinmann, Doerte Laing, Christian Odenthal.(2014). Development of the cellflux storage concept for sensible heat. **Journal of Solar Energy Engineering**,132(1), 1-8.
- [21] Gambhir, M.L. (2006). **Concrete technology** (3<sup>rd</sup> ed.). New Delhi:  
Tata Mcgraw-Hill.
- [22] Nijaguna, B.T. (2005). **Thermal science data book**. New Delhi:  
Tata Mcgraw-Hill.
- [23] Popovics, S. (1992). **Concrete materials**. New Jetsey: Noyes Publications.
- [24] Neville, A.M. (1995). **Properties of concrete** (4<sup>th</sup> ed.). London: Longman.
- [25] Janna, W.S. (2011). **Engineering heat transfer** (3<sup>rd</sup> ed.). Stillwater: CRC Press.
- [26] Holman, J.P. (2010). **Heat transfer** (10<sup>th</sup> ed.). New York, USA: McGraw-Hill.
- [27] Agency (IEA), I.E. (2010). **Technology roadmap concentrating solar power**.  
Paris, France: n.p.
- [28] Kelly, B. and Kearney, D. (2006). Thermal storage commercial plant design study for 2-tank indirect molten salt system, In F. Report, (Ed.), **NREL**.  
N.P.: n.p.
- [29] Züblin, D. (n.d.). **Flagsol, Schlussbericht Forschungsvorhaben WANDA, Pre-kommerzielle Entwicklung der WESPE Speicher-Technologie für den Einsatz in ANDASOL Kraftwerken**. Report, DLR. N.P.: n.p.
- [30] Xu, Y. and D.D.L. Chung. (2000). Effect of sand addition on the specific heat and thermal conductivity of cement. **Cement and Concrete Research**, 30(1), 59-61.
- [31] Herrmann, U.a.D.W.K., (2002). Survey of thermal energy storage for parabolic trough power plants. **Solar Energy Engineering**, 124(2), 145-152.
- [32] Dong, Z., Li, ZJ., Zhou, HM. and Wu, K. (2004). Development of thermal energy storage concrete. **Cement and concrete research**, 36(4), 927-934.

- [33] Doete Laing, W.-D.S., Michael Fiß, Rainer Tamme and Carsten Bahl. (2008). Solid Media Thermal Storage Development and analysis of modular storage operation concepts for parabolic trough power plants. **Solar Energy Engineering**, 130(1), 011006-011006-5.
- [34] Karaipekli, A., A. San and K. Kaygusuz. (2007). Thermal conductivity improvement of stearic acid using expanded graphite and carbon fiber for energy storage applications. **Renewable Energy**, 32(13), 2201-2210.
- [35] Aydm, S.a.B.B. (2008). High temperature resistance of normal strength and autoclaved high strength mortars incorporated polypropylene and steel fibers. **Construction and Building Materials**, 22(4), 504-512.
- [36] Doete Laing, T.B., Dorothea Lehmann and Carsten Bahl. (2010). Development the thermal energy storage system for parabolic trough power plants with direct steam generation. **Solar Energy Engineering**, 132(2), 1-8.
- [37] Doete Laing, D.L. (2008). Concrete storage for solar thermal power plants and industrial process heat. In **3<sup>rd</sup> International Renewable Energy Storage Conference**. Berlin, Germany: Proceedings.
- [38] Wongpanyo, W. (2009). **Concrete storage system for solar thermal power plant**. Doctoral dissertation, Ph.D., Naresuan University, Phisanulok.
- [39] Choktaweekam, P., W. Saengsoy and Tangtermsirikul, S. (2009). A model for predicting the specific heat capacity of fly-ash concrete. **ScienceAsia**, 35(2), 178-182.
- [40] Choktaweekarn, P., W. Saengsoy and S. Tangtermsirikul, (2009). A model for predicting thermal conductivity of concrete. **Magazine of Concrete Research**, 61(4), 271-280.
- [41] Choktaweekam, P.a.S.T. (2009). A Model for Predicting Coefficient of Thermal Expansion of Cementitious Paste. **ScienceAsia**, 35(1), 57-63.
- [42] Choktaweekam, P.a.S.T. (2010). Effect of aggregate type, casting, thickness and curing condition on restrained strain of mass concrete. **Sonklanakarini Journal of Science and Technology**, 32(4), 391.



- [43] Zhu, J., W. Zhou and W. Chen. (2010). Fabrication and thermal properties of a new heat storage concrete material. **Journal of Wuhan University of Technology-Mater. Sci. Ed**, 25(4), 628-630.
- [44] Adeyanju, A., T.O. Ani and D.O. Aldndele. (2010). Experimental determination of thermophysical properties of concrete for Thermal energy storage. **Journal of Engineering and Applied Sciences**, 5(4), 282-289.
- [45] Adeyanju, A.a.K.M. (2011). Effects of Steel Fibers and Iron Filings on Thermal and Mechanical Properties of Concrete for Energy Storage Application. **Journal of Minerals & Materials Characterization & Engineering**, 10(15), 1429-1448.
- [46] Samir N. Shoukrya, G.W.W., Brian Downiea and Mourad Y. Riadb. (2011). Effect of moisture and temperature on the mechanical properties of concrete. **Construction and Building Materials**, 25(2), 688-696.
- [47] Doerte Laing, C.B., Michael Fiß, Matthias Hempel, Mirko Meyer-Grünefeldt, Martin Eickhoff and Andreas Stückle. (2011). Test and evaluation of a Thermal energy storage system for direct steam generation. In **SolarPACES Conference** (pp.21-24). Granada, Spain: CNRS-PROMES.
- [48] Eduard Oró, Antoni Gil, Alvaro de Gracia, Dieter Boer and Luisa F. Cabeza,(2012). Comparative life cycle assessment of thermal energy storage systems for solar power plants. **Renewable energy**, 44(0), 166-173.
- [49] Hui-Wen Yuan, C.-H.L., Zhong-Zi Xu, Ya-Ru Ni and Xiang-Hui Lan. (2012). Mechanical and thermal properties of cement composite graphite for solar thermal storage materials. **Solar energy**, 86(11), 3227-3233.
- [50] Kook-Han Kim, S.-E.J., Jin-Keun Kim and Sungchul Yang. (2003). An experimental study on thermal conductivity of concrete. **Cement and concrete research**, 33(3), 363-371.
- [51] Al-Ostaz, A. (2007). **Effect of moisture content on the coefficient of thermal expansion of concrete**. USA.: Department of Civil Engineering, University of Mississippi.



- [52] Jagannath Mallela, A.A., Tom Harman, Chetana Rao, Rongfang Liu, Michae and Darter, L. (2005). Measurement and significance of the coefficient of thermal expansion of concrete in rigid pavement design. **Journal of the Transportation Research Board**, 19(1), 38-46.
- [53] Hein, D.K.a.S.S. (2012). Concrete coefficient of thermal expansion (CTE) and Its significance in mechanistic-empirical pavement design. In **Conference and Exhibition of the Transportation Association of Canada Transportation: Innovations and Opportunities in Canada**. Canada: n.p.
- [54] Association, N.C.M. (2012). **R-values of multi-wythe concrete masonry walls**. USA: National Concrete Masonry Association.
- [55] Lie, T.a.V.K. (1996). Thermal and mechanical properties of steel-fibre reinforced concrete at elevated temperatures. **Canadian Journal of Civil Engineering**, 23(2), 511-517.
- [56] Vundela Siva Reddy, S.C.K., Sudhir Kumar Tyagi and Naraya Lal Ponwar. (2010). An Approach to Analyse energy and exergy analysis of Thermal Power Plants. **A Review, Smart Grid and Renewable Energy**, 1, 143-152.
- [57] Htun, N.N. (2014). **Properties of concrete materials for thermal energy storage in School of Renewable Energy Technology**. Master thesis, M.S., Naresuan University, Phisanulok.



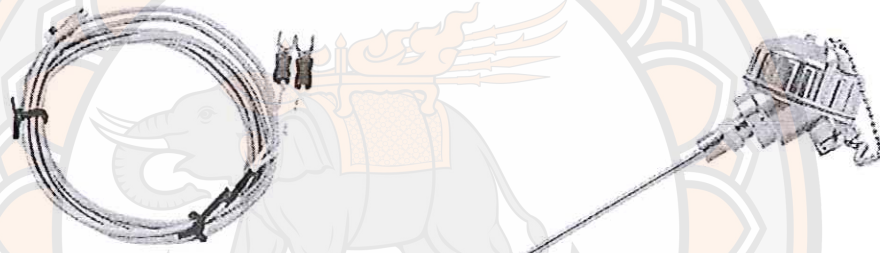
## APPENDIX A INSTRUMENTS



Pressure gauge (0-10 bar, 0-240 °C)



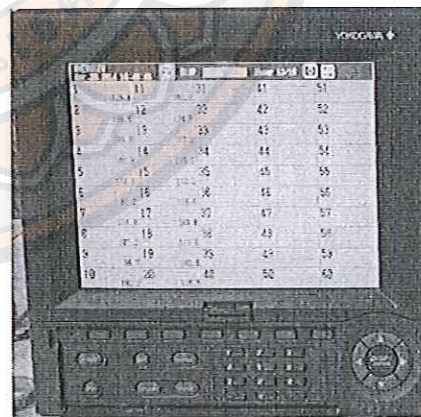
Oscilloscope (60-150 MHz, 2.5 k point)



Thermocouple type k (-200 °C to 1250 °C)



Vortex flow meter (versa flow vortex 1000, 34-VF-03-05)



Data recorder (Yokogawa DX1000)

Figure 38 Instruments



**APPENDIX B THERMAL CONDUCTIVITY, SPECIFIC HEAT CAPACITY  
AND THERMAL CONDUCTIVITY WERE TESTED AT  
NATIONAL METAL AND MATERIALS TECHNOLOGY  
CENTER (MTEC), THAILAND**



MTEC No. 0732/56

Report of Sample Analysis

Issued Date	:	11 December 2012
Customer	:	Ms Rungralee Boonsu School of Renewable Energy Technology, Naresuan University Thapao, Muang, Phitsanulok: 45000
Serviced by	:	Physical Characterization Laboratory, Materials Characterization Unit, National Metal and Materials Technology Center
Date received	:	3 December 2012
Date analyzed	:	11 December 2012
Sample	:	Concrete Sample as specified by customer


Sample identification : N/A

Objective : To determine the thermal conductivity of the sample.

Instrument used : Hot Disk Thermal Constant Analyser (Hot Disk AB)

Technical used : Thermal Constant Analysis (TCA)

Condition : Room temperature

Disk type: Kapton Insulation (Sensor No. C7577, Radius =2.031 mm)

MTEC No.0732/56



**Results and Interpretation:**


Thermal conductivities of the sample are reported in the table below

Sample	Condition Used		Thermal Properties	Measurement Results	Average	S.D.
	Out Put of Power (W)	Measuring Time (s)				
"Concrete Sample (Test1)"	0.06	5	Thermal Conductivity (W / m K)	1.0370	1.0353	0.0015
			1.0350			
			1.0340			
			Thermal Diffusivity (mm <sup>2</sup> /s)	0.6810	0.6734	0.0070
			0.6719			
			0.6672			
			Specific Heat (MJ/m <sup>3</sup> K)	1.5230	1.5380	0.0137
			1.5410			
			1.5500			

**Remarks:**

1. Thermal conductivity is a measure of the ability to transmit heat through the material.
2. Thermal diffusivity is a measure of transient heat flow and is defined as the thermal conductivity divided by the product of specific heat times density.
3. Specific heat is the quantity of heat needed to raise the temperature of a unit mass of the substance 1 degree of temperature.
4. Hot Disk Thermal Constant Analyser (Hot Disk AB)
 

Reproducibility	- Thermal Conductivity	±2%
	- Thermal Diffusivity	±5%
	- Specific Heat	±7%

Work performed by:   
(Miss Utaiwan Watcharosin)

Approved by:   
(Ms. Piyawan Panitanta)

Remarks:

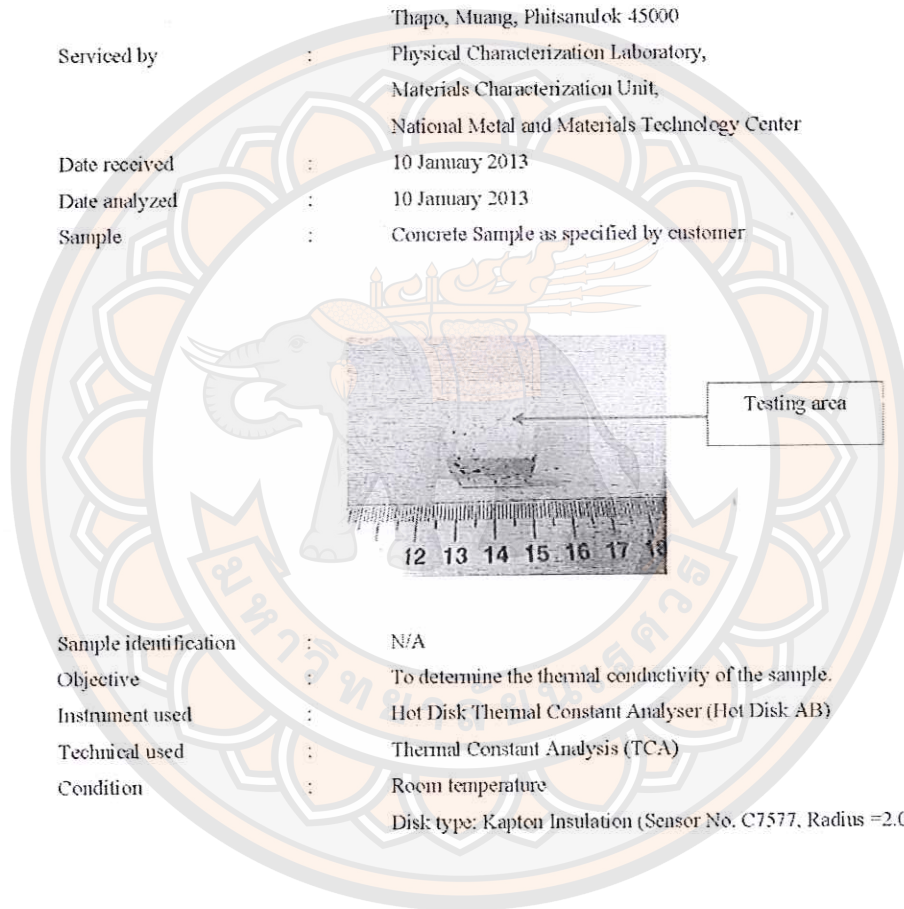
4. MTEC does not allow any alteration or modification of this report, or any part of this report, without prior formal written permission from MTEC.
5. MTEC will not accept liability for any damage whatsoever, resulting directly, from using data, results, conclusions or recommendations in this report for the purpose of designing, manufacturing or for other purpose
6. Experimental results are only valid for the specimens tested.



TA 0035/56

## Report of Sample Analysis

Issued Date : 10 January 2013  
 Customer : Ms. Rungudee Boonsu  
 School of Renewable Energy Technology, Naresuan University  
 Thapo, Muang, Phitsanulok 45000  
 Serviced by : Physical Characterization Laboratory,  
 Materials Characterization Unit,  
 National Metal and Materials Technology Center  
 Date received : 10 January 2013  
 Date analyzed : 10 January 2013  
 Sample : Concrete Sample as specified by customer



Sample identification : N/A  
 Objective : To determine the thermal conductivity of the sample.  
 Instrument used : Hot Disk Thermal Constant Analyser (Hot Disk AB)  
 Technical used : Thermal Constant Analysis (TCA)  
 Condition : Room temperature  
 Disk type: Kapton Insulation (Sensor No. C7577, Radius =2.001 mm)

TA 0035/56

1/3

**Results and Interpretation:**

Thermal conductivities of the sample are reported in the table below

Sample	Condition Used		Thermal Properties	Measurement Results	Average	S.D.
	Out Put of Power (W)	Measuring Time (s)				
"Concrete Sample (Test3)"	0.06	5	Thermal Conductivity (W / m K)	0.9744	0.9741	0.0003
			0.9741			
			0.9738			
			Thermal Diffusivity (mm <sup>2</sup> /s)	0.6815	0.6811	0.0048
			0.6857			
			0.6761			
			Specific Heat (MJ/m <sup>3</sup> K)	1.4300	1.4303	0.0095
			1.4210			
			1.4400			

**Remarks:**

1. Thermal conductivity is a measure of the ability to transmit heat through the material.
2. Thermal diffusivity is a measure of transient heat flow and is defined as the thermal conductivity divided by the product of specific heat times density.
3. Specific heat is the quantity of heat needed to raise the temperature of a unit mass of the substance 1 degree of temperature.
4. Hot Disk Thermal Constant Analyser (Hot Disk AB)
 

Reproducibility	- Thermal Conductivity	±2%
	- Thermal Diffusivity	±5%
	- Specific Heat	±7%

Work performed by: *อุทัยวัน วัชรสารสิน*  
(Miss Utaiwan Watcharesin)

Approved by: *ปิยawan ปานิตานตา*  
(Ms.Piyawan Panitanta)

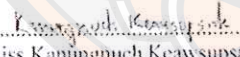



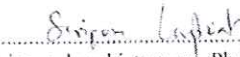
Remarks:

1. MTEC does not allow any alteration or modification of this report, or any part of this report, without prior formal written permission from MTEC.
2. MTEC will not accept liability for any damage whatsoever, resulting directly, from using data, results, conclusions or recommendations in this report for the purpose of designing, manufacturing or for other purpose
3. Experimental results are only valid for the specimens tested.



APPENDIX C COEFFICIENT OF THERMAL EXPANSION AND DENSITY  
WERE THAILAND INSTITUTE OF SCIENTIFIC AND  
TECHNOLOGICAL RESEARCH (TISTR), THAILAND



Request No.15/56	MATERIAL INNOVATION DEPARTMENT (MID)	Lab No. 18/56
Date September 1, 2012		Page 1 of 2
<b>REPORT ON TESTING AND ANALYSIS</b> FOR <b>NARESUAN UNIVERSITY</b> 99, Thapo, Mueang, Phitsanulok 65000		
Name Code of Sample	:- Concrete sample 1	
Characteristic of Sample	:- Rectangular and Cubic	
Method of testing/analysis	:- Coefficient of expansion (COE) by Dilatometer (Netzsch:DIL 402) and Density	
Result of testing/analysis	:-	
The coefficient of expansion (COE) and density of Concrete sample 1 are shown on page 2.		
Tested/analyzed by	Approved by	
 (Miss Kantingmueh Keawsupsak)	 (Siriporn Larpiattaworn, Ph.D.) Acting Director Material Innovation Department	
 (Ms. Panida Thaveethavorn)	 <b>TISTR</b>	
Examined by		
 (Siriporn Larpiattaworn, Ph.D.)		
FM-MID-GEN 02-02 Rev.2		
Remark: The above results are valid exclusively for tested/analysis sample as mentioned in the report. Publication of the results or testing and analysis is prohibited unless written permission is obtained from the governing of TISTR.		



TISTR

Request No.15/56 MATERIAL INNOVATION DEPARTMENT (MID)

Lab No. 18/56

Date September 1, 2012

Page 2 of 2

Sample name	coefficient of expansion (25-300°C) (1/K.10 <sup>-6</sup> )		Density (g/cm <sup>3</sup> )
	1	2	
Concrete sample 1	7.99	8.21	1.82

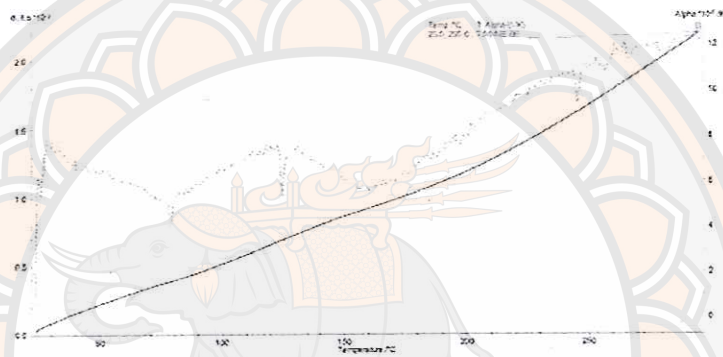


Fig 1. coefficient of expansion of Concrete sample 1 (No.1)

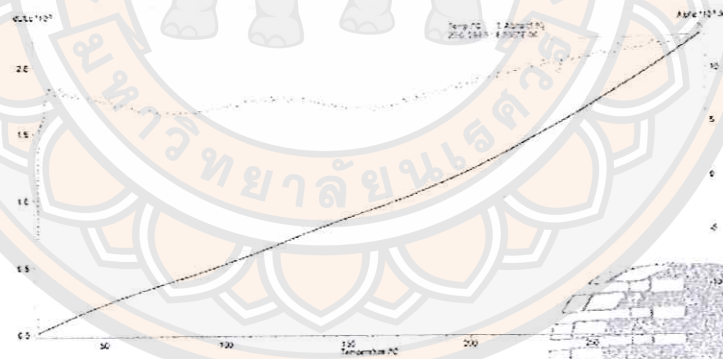


Fig 2. coefficient of expansion of Concrete sample 1 (No.2)

TISTR



TISTR

Request No.44/56

MATERIAL INNOVATION DEPARTMENT (MID)

Lab No. 33-56

Date December 3, 2012

Page 1 of 2

## REPORT ON TESTING AND ANALYSIS

FOR

NARESUAN UNIVERSITY

99, Thapo, Mueang, Phitsanulok 65000

Name/Code of Sample :- Concrete sample  
 Characteristic of Sample :- Rectangular and Cubic  
 Method of testing/analysis :- Coefficient of expansion (COE) by Dilatometer  
 (Netzsch:DIL 402) and Bulk density (ASTM C373-88)  
 Result of testing/analysis :-

Thermal expansion coefficient and bulk density of Concrete sample are shown on page 2.

Tested/analyzed by

Approved by

*Kanungnueh Keawsupsak*  
 (Miss Kanungnueh Keawsupsak)

*Siriporn Larpiattaworn*  
 (Siriporn Larpiattaworn, Ph.D.)  
 Acting Director

*Panida Thaveethavorn*  
 (Ms. Panida Thaveethavorn)

Material Innovation Department

TISTR

Examined by

*Siriporn Larpiattaworn*  
 (Siriporn Larpiattaworn, Ph.D.)

MI-MID-GEN-02-02 Rev.2

Remark: The above results are valid exclusively for tested analysis sample as mentioned in the report.  
 Publicity of the results on testing and analysis is prohibited unless written permission is obtained from the governor of TISTR.





TISTR

Request No.44/56 MATERIAL INNOVATION DEPARTMENT (MID)

Lab No. 33/56

Date December 3, 2012

Page 2 of 2

Sample name	COE (25-300°C) ( $10^{-6}, K^{-1}$ )	Bulk density ( $g/cm^3$ )
Concrete sample1	9.79	1.87

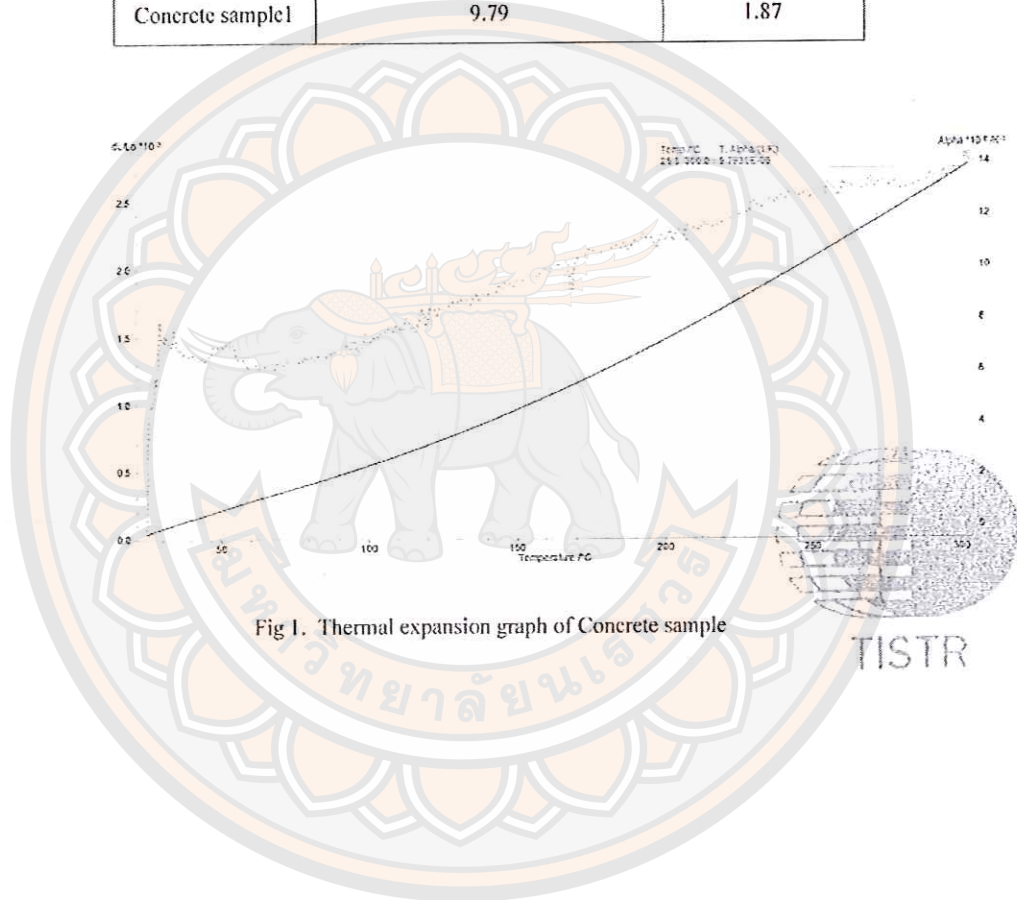


Fig 1. Thermal expansion graph of Concrete sample

## APPENDIX D PROGRAM AND FUNCTION OF NSIMULATION PROGRAM

**Table 13 The program list of NSimulation**

Program	Work
Memovie, RunMovie	Create and display movie of the distribution of heat in 1 unit cell at different time. The schedule is less than 2 seconds.
HeatPower, HeatEnergy	Calculate and display of heat (Q) at different times of 1 unit cell.
HeatTrans, GHeat	Calculate and display the temperature (T) at different stages.
LTSolution, LTG	Calculate and display the spread of heat at any time displaying in 2 different styles: 1 unit cell and multi-cell. (Which shows a 3x3cells only).
MainProgram, NumericalHeat	Calculate and display the spread of heat in less than or equal to 10 seconds.
NSimulation	A Graphic User Interface (GUI) of all applications, there are two files are the extensions. M is a source code file that interacts with the GUI. Fig file layout of the GUI.

**Table 14 The function list of the NSimulation**

Function	Input Parameter	Output	Work
Merotate	( $\alpha$ , time, size per unit cell, room temperature, pipe temperature).	Array 2D of temperature	Create 1 unit cell with the temperature distribution of the parameter set.
defindB	(Row, column, the size of the Array).	boundaries (B)	Find out what kind of scope for Rows and columns entered. The return value is the number Which have the

Table 14 (cont.)

Function	Input Parameter	Output	Work
			following meanings: 1 inside 21 On the edge 22 The right edge 23 The lower edge 24 The left edge 31 The upper left corner 32 The top right corner 33 The bottom right corner 34 The bottom left corner
diffu	(T1,T2,T3,T4,Tmn)	0/1	T1 to T4 and T(m,n) are any number. If T(m,n) is the most, it will return 0 but the rest is 1
gendata	(Length, temperature, tube, dx, room temperature)	Array 2D temperature	Create a 1/4 unit cell with pipe temperature but does not include the distribution of heat in which a distance dx is step.
Meboundary	(Row, column, Array 2D temperature, F0, B)	Array 2D temperature	Calculate the temperature at the edge of the row and column in which F0 is Fourier and B is the number of edges which have the function defined B.
Mecorner	(Row, column, Array 2D temperature, F0, B)	Array 2D temperature	Calculate the temperature over the row and column in which F0 is the Fourier number and B is a function of the angle of defined B.
Meinterior	(Row, column, Array 2D temperature, F0)	Array 2D temperature	Calculate the temperature inside the cell, the row and column in which F0 is the Fourier number.



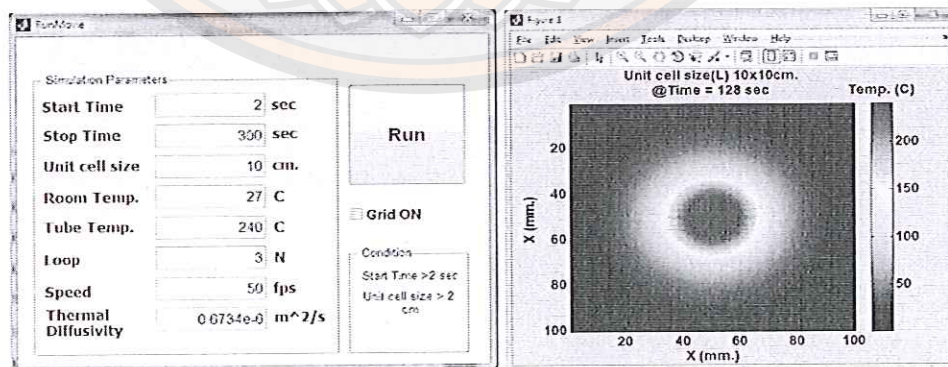
## The operation of each program in NSimulation

### Memovie (Runmovie)

The determinable value for Memovie program show in Table 15. Before running this program, the users need to be configured to run as in Figure 39. When running, the show on the first round will be much slower than the latter. The program must compute and complete storage. In this graph, the display will include the title, which is at the top. This will tell you the size of the unit cell and time. The right-hand column of the chart is to show the temperature in degrees Celsius (C) (the type of color stripe is 'jet').

**Table 15 The determinable value for Memovie program**

Determinable value	Note
Start time	Unit in seconds (sec)
end time	Unit in seconds (sec)
One unit cell size	Unit in centimeters (cm)
room temperature	Unit in degree Celsius ( $^{\circ}\text{C}$ )
pipe temperature	Unit in degree Celsius ( $^{\circ}\text{C}$ )
number of cycles	Unit in round(N)
playback speed	Unit in frames per second (fps)
Thermal Diffusivity ( $\alpha$ )	Unit in square meter per second ( $\text{m}^2 / \text{s}$ )



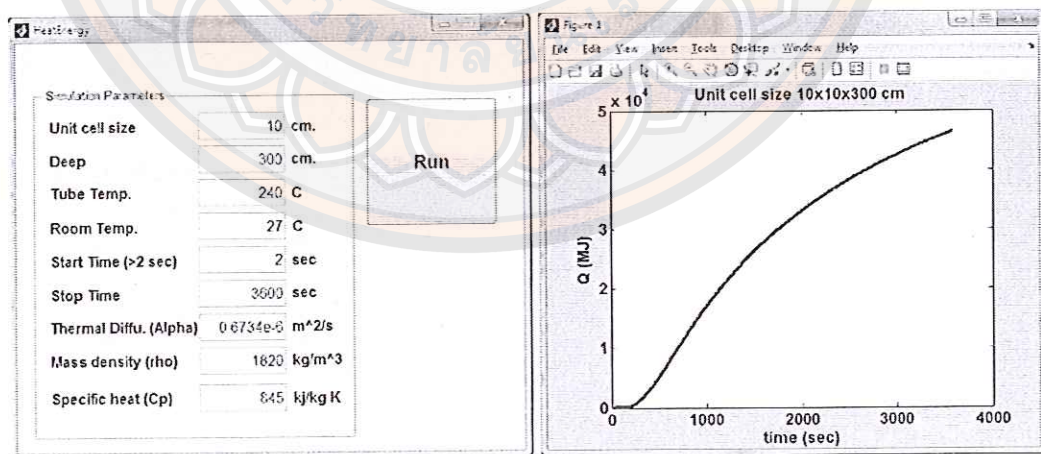
**Figure 39 The user interface of Memovie program**

### HeatPower (HeatEnergy)

The determinable value for HeatPower program show in Table 16. Before running this program, users need to be configured to run as in Figure 40.

**Table 16 The determinable value for HeatPower program**

Determinable value	Note
One unit cell size (width = length)	Unit in Centimeters (cm.)
depth	Unit in Centimeters (cm.)
pipe temperature	Unit in Degree Celsius ( $^{\circ}\text{C}$ )
room temperature	Unit in Degree Celsius ( $^{\circ}\text{C}$ )
Start time	Unit in Seconds (sec)
end time	Unit in Seconds (sec)
Thermal Diffusivity ( $\alpha$ )	Unit in square meters per second ( $\text{m}^2 / \text{s}$ )
Mass density ( $\rho$ )	Unit in kilograms per cubic meter ( $\text{kg} / \text{m}^3$ )
Specific Heat ( $C_p$ )	Unit in kilojoules per kilogram kelvin ( $\text{kJ} / \text{kg}$ K)



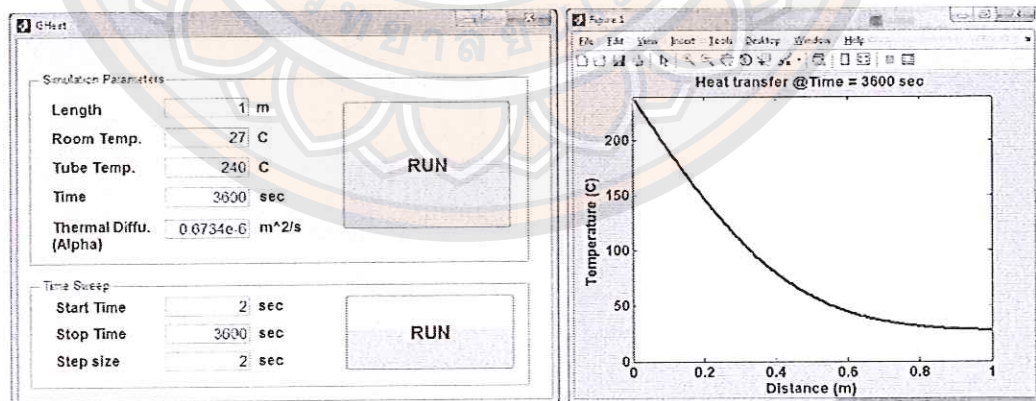
**Figure 40 The user interface of HeatPower program**

### HeatTrans (GHeat)

The determinable value for HeatTrans program is shown in table 17. Before running this program, users need to be configured to run as in Figure 41. The program can be displayed in two different types: graph temperature with distance at any given time and the movie in which it changes from time to time in the defining moment.

**Table 17** The determinable value for HeatTrans program

Determining value	Note
Distance	Unit in meters (m.)
pipe temperature	Unit in degree Celsius (°C)
room temperature	Unit in degrees Celsius (°C)
Thermal Diffusivity ( $\alpha$ )	Unit in square meters per second ( $m^2 / s$ )
Time	Unit in seconds (sec)
Start time	Unit in seconds (sec)
End time	Unit in seconds (sec)
Step size	Unit in seconds (sec)



**Figure 41** The user interface of HeatTrans program

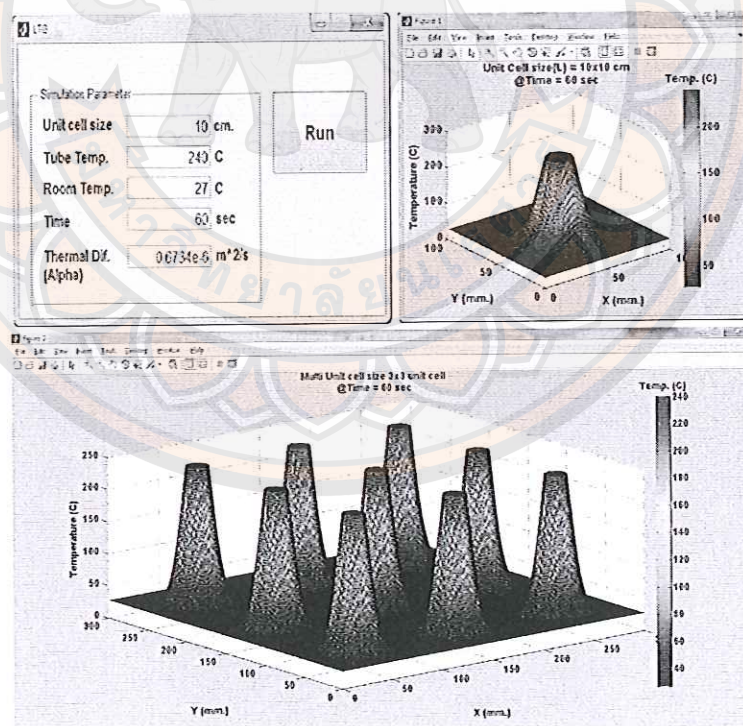


### LTSolution (LTG)

The determinable value for LTSolution program is shown in Table 18. Before running this program, users need to be configured to run as in Figure 42. There are two types of result in running the program: the first represents just one unit cell and the second is to show several unit cells.

**Table 18** The determinable value for LTSolution program

Determinable value	Note
One unit cell size	Measured in Centimeters (cm.)
pipe temperature	Measured in Degrees (C)
room temperature	Measured in Degrees (C)
time	Measured in seconds (sec)
Thermal Diffusivity ( $\alpha$ )	Measured in square meters per second ( $m^2/s$ )



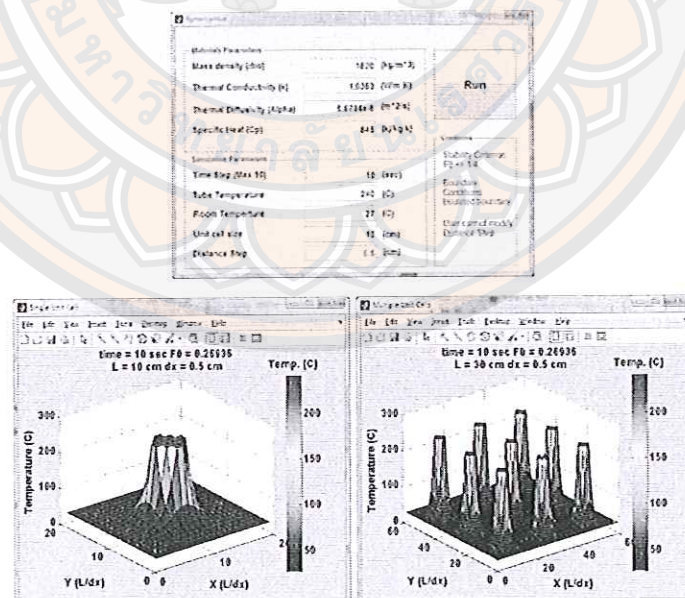
**Figure 42** The user interface of LTSolution program

### MainProgram

The determinable value for MainProgram is shown in Table 19. Before running this program, the users need to configure to run as in Figure 43. The program will show in two different types of result: one unit cell and multi-cell in which the program will determine the distance dx is 0.5 cm. automatically.

**Table 19 The determinable value for MainProgram**

Determinable value	Note
Mass density ( $\rho$ )	Unit in kilograms per cubic meter ( $\text{kg} / \text{m}^3$ )
Thermal conductivity ( $k$ )	Unit in watts per meter Kelvin ( $\text{W} / \text{m K}$ ).
Thermal Diffusivity ( $\alpha$ )	Unit in square meters per second ( $\text{m}^2 / \text{s}$ )
Specific Heat ( $Cp$ )	Unit in kilojoules per kilogram kelvin ( $\text{kJ} / \text{kg K}$ )
Time step (max 10 sec)	Unit in seconds (sec)
Pipe temperature	Unit in Degree Celsius ( $^{\circ}\text{C}$ )
Room temperature	Unit in Degree Celsius ( $^{\circ}\text{C}$ )
One unit cell size	Unit in Centimeters (cm.)



**Figure 43 The user interface of Main Program**

## APPENDIX E SOURCE CODE PROGRAM AND FUNCTION OF MATHEMATIC DESIGNING MODEL

### E.1 : Calculation and display movie of the distribution of heat in 1 unit cell at the different time

#### Memovie

```
clc
clear
Alpha = 0.6734e-6;
te=5*60;
ts=2;
YL=10;          · %Desire length (cm)
tm=ts:2:te;
Tr=27;
YT=240;
ntime=3;
Eps=50;
Line=0; %zeros mean not have grid
%F1=nan*zeros(1,length(tm));
%scrsz = get(0,'ScreenSize');
%figure('Position',[col row width high]) left button is origin
%figure('Position',[200 100 800 500])
h=figure;
for m=1:1:length(tm)
    T1=Merotate(Alpha,tm(m),YL,Tr,YT);
    image(T1);
    colormap(jet(YT));
    title(['Unit cell size(L) ' num2str(YL) 'x' num2str(YL) 'cm.'];...
        ['@Time = ' num2str(tm(m)) ' sec'],'FontSize',14,'fontweight','b');
    xlabel('X (mm.)','FontSize',14,'fontweight','b');
    ylabel('Y (mm.)','FontSize',14,'fontweight','b');
    ht=colorbar;
    title(ht,'Temp. (C)','FontSize',14,'fontweight','b');
    if(Line>0)
        grid on
    end
    set(gca,'FontSize',14,'fontweight','b');
    F1(m)=getframe;
end
```



```

movie(F1,ntime-1,Fps);

%save figure to image file with true color display
%h=figure
%image(T1);
%colormap(jet(240))
%colorbar
%print(h,'-djpeg ','metest')

```

## Runmovie

```

function varargout = RunMovie(varargin)
% RUNMOVIE MATLAB code for RunMovie.fig
%   RUNMOVIE, by itself, creates a new RUNMOVIE or raises the existing
%   singleton*.
%
%   H = RUNMOVIE returns the handle to a new RUNMOVIE or the handle to
%   the existing singleton*.
%
%   RUNMOVIE('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in RUNMOVIE.M with the given input arguments.
%
%   RUNMOVIE('Property','Value',...) creates a new RUNMOVIE or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before RunMovie_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to RunMovie_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help RunMovie

% Last Modified by GUIDE v2.5 29-Jul-2013 16:05:58

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @RunMovie_OpeningFcn, ...

```

```

        'gui_OutputFcn', @RunMovie_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before RunMovie is made visible.
function RunMovie_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to RunMovie (see VARARGIN)

% Choose default command line output for RunMovie
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes RunMovie wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = RunMovie_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

```

```

% --- Executes on button press in cbGrid.
function cbGrid_Callback(hObject, eventdata, handles)
% hObject    handle to cbGrid (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of cbGrid

% --- Executes on button press in btRun.
function btRun_Callback(hObject, eventdata, handles)
% hObject    handle to btRun (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Alpha = str2double(get(handles.tbAlpha,'string'));
te=str2double(get(handles.tbTe,'string'));
ts=str2double(get(handles.tbTs,'string'));
YL=str2double(get(handles.tbSize,'string'));           %Desire length
(cm)
tm=ts:2:te;
Tr=str2double(get(handles.tbTr,'string'));
YT=str2double(get(handles.tbYT,'string'));
ntime=str2double(get(handles.tbLoop,'string'));
Fps=str2double(get(handles.tbSpeed,'string'));
Line=get(handles.cbGrid,'value'); %zeros mean not have grid
%F1=nan*zeros(1,length(tm));
%scrsz = get(0,'ScreenSize');
%figure('Position',[col row width high]) left button is origin
%figure('Position',[200 100 800 500])
h=figure;
for m=1:1:length(tm)
    T1=Merotate(Alpha,tm(m),YL,Tr,YT);
    image(T1);
    colormap(jet(YT));
    title(['Unit cell size(L) ' num2str(YL) 'x' num2str(YL) 'cm.'];...
        ['@Time = ' num2str(tm(m)) ' sec'],'FontSize',14,'fontweight','b');
    xlabel('X (mm.)','FontSize',14,'fontweight','b');
    ylabel('Y (mm.)','FontSize',14,'fontweight','b');
    ht=colorbar;
    title(ht,'Temp. (C)','FontSize',14,'fontweight','b');
    if(Line>0)
        grid on
    end
end

```



```

    set(gca,'FontSize',14,'fontweight','b');
    F1(m)=getframe;
end
movie(F1,ntime-1,Fps);
clear all

function tbTs_Callback(hObject, eventdata, handles)
% hObject    handle to tbTs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTs as text
%        str2double(get(hObject,'String')) returns contents of tbTs as a
double

% --- Executes during object creation, after setting all properties.
function tbTs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbTe_Callback(hObject, eventdata, handles)
% hObject    handle to tbTe (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTe as text
%        str2double(get(hObject,'String')) returns contents of tbTe as a
double

% --- Executes during object creation, after setting all properties.
function tbTe_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTe (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end
```

```
function tbSize_Callback(hObject, eventdata, handles)
```

```
% hObject handle to tbSize (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of tbSize as text
% str2double(get(hObject,'String')) returns contents of tbSize as a
double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function tbSize_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to tbSize (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end
```

```
function tbTr_Callback(hObject, eventdata, handles)
```

```
% hObject handle to tbTr (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of tbTr as text
% str2double(get(hObject,'String')) returns contents of tbTr as a
double
```

```

% --- Executes during object creation, after setting all properties.
function tbTr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbYT_Callback(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYT as text
%         str2double(get(hObject,'String')) returns contents of tbYT as a
double

% --- Executes during object creation, after setting all properties.
function tbYT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbLoop_Callback(hObject, eventdata, handles)
% hObject    handle to tbLoop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbLoop as text
%         str2double(get(hObject,'String')) returns contents of tbLoop as a
double

```



```

% --- Executes during object creation, after setting all properties.
function tbLoop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbLoop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbSpeed_Callback(hObject, eventdata, handles)
% hObject    handle to tbSpeed (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbSpeed as text
%         str2double(get(hObject,'String')) returns contents of tbSpeed as a
double

% --- Executes during object creation, after setting all properties.
function tbSpeed_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbSpeed (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbAlpha_Callback(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of tbAlpha as text
%         str2double(get(hObject,'String')) returns contents of tbAlpha as a
double

% --- Executes during object creation, after setting all properties.
function tbAlpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

## E.2 : Calculate and display of heat (Q) at different times of 1 unit cell.

### HeatPower

```

clc
clear

YL=6;           %unit cell size (cm)
YZ=300;        %Desire Deep (cm)
YT=240;        %Desire Temperature (c)
Ytime=900;     %Watch on this rang of time (min)
Tr=27;         %Room temperature (c)
Ts=2;

V=YL*YL*YZ*0.01;
rho=1820;      %*****
mass=rho*V;
k=1.0353;
Alpha = 0.6734e-6; %*****
Cp = k/(rho*Alpha); %*****
t=Ts:2:Ytime;
R=(YL*0.01)/2; %in meter
%Temperature at distance YL/2
T=YT+(Tr-YT).*erf(R./(2.*sqrt(Alpha.*t)));
Q=ones(1,length(t));
for m=1:1:length(t)
    Q(1,m)=mass*Cp*(T(1,m)-Tr);

```

```
end
```

```
figure
plot(t,Q./10^6,'linewidth',3)
xlabel('time (sec)','FontSize',14,'fontweight','b');
ylabel('Q (MJ)','FontSize',14,'fontweight','b');
title(['Unit cell size ' num2str(YL) 'x' num2str(YL) 'x' num2str(YZ) '
cm']...
, 'FontSize',14,'fontweight','b');
set(gca,'FontSize',14,'fontweight','b');
```

## HeatEnergy

```
function varargout = HeatEnergy(varargin)
% HEATENERGY MATLAB code for HeatEnergy.fig
% HEATENERGY, by itself, creates a new HEATENERGY or raises the existing
% singleton*.
%
% H = HEATENERGY returns the handle to a new HEATENERGY or the handle to
% the existing singleton*.
%
% HEATENERGY('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in HEATENERGY.M with the given input
arguments.
%
% HEATENERGY('Property','Value',...) creates a new HEATENERGY or raises
the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before HeatEnergy_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to HeatEnergy_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help HeatEnergy

% Last Modified by GUIDE v2.5 29-Jul-2013 17:22:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
```



```

gui_State = struct('gui_Name',      mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @HeatEnergy_OpeningFcn, ...
                  'gui_OutputFcn',  @HeatEnergy_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before HeatEnergy is made visible.
function HeatEnergy_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to HeatEnergy (see VARARGIN)

% Choose default command line output for HeatEnergy
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes HeatEnergy wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = HeatEnergy_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Get default command line output from handles structure
varargout{1} = handles.output;

function tbYL_Callback(hObject, eventdata, handles)
% hObject    handle to tbYL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYL as text
%        str2double(get(hObject,'String')) returns contents of tbYL as a
double

% --- Executes during object creation, after setting all properties.
function tbYL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbYZ_Callback(hObject, eventdata, handles)
% hObject    handle to tbYZ (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYZ as text
%        str2double(get(hObject,'String')) returns contents of tbYZ as a
double

% --- Executes during object creation, after setting all properties.
function tbYZ_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYZ (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbYT_Callback(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYT as text
%         str2double(get(hObject,'String')) returns contents of tbYT as a
double

% --- Executes during object creation, after setting all properties.
function tbYT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbTr_Callback(hObject, eventdata, handles)
% hObject    handle to tbTr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTr as text
%         str2double(get(hObject,'String')) returns contents of tbTr as a
double

% --- Executes during object creation, after setting all properties.
function tbTr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```



```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbTs_Callback(hObject, eventdata, handles)
% hObject    handle to tbTs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTs as text
%       str2double(get(hObject,'String')) returns contents of tbTs as a
double

% --- Executes during object creation, after setting all properties.
function tbTs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbYtime_Callback(hObject, eventdata, handles)
% hObject    handle to tbYtime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYtime as text
%       str2double(get(hObject,'String')) returns contents of tbYtime as a
double

% --- Executes during object creation, after setting all properties.
function tbYtime_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYtime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```

```

% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbAlpha_Callback(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbAlpha as text
%         str2double(get(hObject,'String')) returns contents of tbAlpha as a
double

% --- Executes during object creation, after setting all properties.
function tbAlpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbRho_Callback(hObject, eventdata, handles)
% hObject    handle to tbRho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbRho as text
%         str2double(get(hObject,'String')) returns contents of tbRho as a
double

% --- Executes during object creation, after setting all properties.
function tbRho_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbRho (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbCp_Callback(hObject, eventdata, handles)
% hObject handle to tbCp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbCp as text
% str2double(get(hObject,'String')) returns contents of tbCp as a
double

% --- Executes during object creation, after setting all properties.
function tbCp_CreateFcn(hObject, eventdata, handles)
% hObject handle to tbCp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btRun.
function btRun_Callback(hObject, eventdata, handles)
% hObject handle to btRun (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
YL=str2double(get(handles.tbYL,'string')); %unit cell size (cm)
YZ=str2double(get(handles.tbYZ,'string')); %Desire Deep (cm)
YT=str2double(get(handles.tbYT,'string')); %Desire Temperature (c)
Ytime=str2double(get(handles.tbYtime,'string')); %Watch on this rang of
time (min)
Tr=str2double(get(handles.tbTr,'string')); %Room temperature (c)
Ts=str2double(get(handles.tbTs,'string'));

```



```

V=YL*YL*YZ*0.01;
rho=str2double(get(handles.tbRho,'string')); %*****
mass=rho*V;
Alpha = str2double(get(handles.tbAlpha,'string')); %*****
Cp = str2double(get(handles.tbCp,'string')); %*****
t=Ts:2:Ytime;
R=(YL*0.01)/2; %in meter
%Temperature at distance YL/2
T=YT+(Tr-YT).*erf(R./(2.*sqrt(Alpha.*t)));
Q=ones(1,length(t));
for m=1:1:length(t)
    Q(1,m)=mass*Cp*(T(1,m)-Tr);
end
figure
plot(t,Q./10^6,'linewidth',3)
xlabel('time (sec)','FontSize',14,'fontweight','b');
ylabel('Q (MJ)','FontSize',14,'fontweight','b');
title(['Unit cell size ' num2str(YL) 'x' num2str(YL) 'x' num2str(YZ) '
cm']...
, 'FontSize',14,'fontweight','b');
set(gca,'FontSize',14,'fontweight','b');

```

### E.3 : Calculate and display the temperature (T) at different stages.

#### HeatTrans

```

clc
clear

Alpha = 0.6734e-6;
Ls=1; %meter
YT=240;
Tr=27;
t=3600; % lmin

X1=0.001:0.001:Ls; %meter
% X2=zeros(1,10);
% Xk=[X2 X1];
% X=zeros(1,length(X1));
% for u=1:1:length(X1)
%     X(1,u)=Xk(1,u);
% end
L=length(X1);
Xp=0.005:0.005:0.005*L;

```

```

T1=YT+(Tr-YT)*erf(X1./(2*sqrt(Alpha*t)));
figure
plot(Xp,T1,'linewidth',3)
title(['Heat transfer @Time = ' num2str(t) '
sec'], 'FontSize',14,'fontweight','b')
ylabel('Temperature (C)', 'FontSize',14,'fontweight','b')
xlabel('Distance (m)', 'FontSize',14,'fontweight','b')
axis([0 Ls 0 YT])
set(gca, 'FontSize',14,'fontweight','b');

Ts=10;
Te=10000;
step=10;
figure
for t=Ts:step:Te
    T1=YT+(Tr-YT)*erf(X1./(2*sqrt(Alpha*t)));
    plot(Xp,T1,'linewidth',3)
    title(['Heat transfer @Time = ' num2str(t) '
sec'], 'FontSize',14,'fontweight','b')
    ylabel('Temperature (C)', 'FontSize',14,'fontweight','b')
    xlabel('Distance (m)', 'FontSize',14,'fontweight','b')
    axis([0 Ls 0 YT])
    set(gca, 'FontSize',14,'fontweight','b');
    F1(t)=getframe;
end
% movie(F1,loop-1,fps)
% Numerical Solution max 10 sec
% t=10;
% T7(:,1)=27.*ones(length(X),1);
% x=1:1:10;
% T7(x,1)=240;
% F0=Alpha*t/(0.005^2);
% for m=10:1:length(X)-1
%     T7(m,1)=T7(m,1)+F0*(T7(m+1,1)+T7(m-1,1)-2*T7(m,1));
% end
%
% figure
% plot(Xp,T7)
% title(['F0 = ' num2str(F0) ' time = ' num2str(t) 'sec'])

```

## GHeat

```

function varargout = GHeat(varargin)
% GHEAT MATLAB code for GHeat.fig
%
%   GHEAT, by itself, creates a new GHEAT or raises the existing
%   singleton*.
%
%
%   H = GHEAT returns the handle to a new GHEAT or the handle to
%   the existing singleton*.
%
%
%   GHEAT('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GHEAT.M with the given input arguments.
%
%
%   GHEAT('Property','Value',...) creates a new GHEAT or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before GHeat_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to GHeat_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help GHeat
%
% Last Modified by GUIDE v2.5 29-Jul-2013 19:00:22
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GHeat_OpeningFcn, ...
                  'gui_OutputFcn',  @GHeat_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else

```



```

    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GHeat is made visible.
function GHeat_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GHeat (see VARARGIN)

% Choose default command line output for GHeat
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GHeat wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GHeat_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function tbTs_Callback(hObject, eventdata, handles)
% hObject    handle to tbTs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTs as text
%        str2double(get(hObject,'String')) returns contents of tbTs as a
double

% --- Executes during object creation, after setting all properties.
function tbTs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTs (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbTe_Callback(hObject, eventdata, handles)
% hObject handle to tbTe (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTe as text
% str2double(get(hObject,'String')) returns contents of tbTe as a
double

% --- Executes during object creation, after setting all properties.
function tbTe_CreateFcn(hObject, eventdata, handles)
% hObject handle to tbTe (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbStep_Callback(hObject, eventdata, handles)
% hObject handle to tbStep (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbStep as text
% str2double(get(hObject,'String')) returns contents of tbStep as a
double

```

```

% --- Executes during object creation, after setting all properties.
function tbStep_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbStep (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btRunMovie.
function btRunMovie_Callback(hObject, eventdata, handles)
% hObject    handle to btRunMovie (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Alpha = str2double(get(handles.tbAlpha,'string'));
Ls=str2double(get(handles.tbLs,'string'));    %meter
YT=str2double(get(handles.tbYT,'string'));
Tr=str2double(get(handles.tbTr,'string'));
X1=0.001:0.001:Ls;    %meter
L=length(X1);
Xp=0.005:0.005:0.005*L;
Ts=str2double(get(handles.tbTs,'string'));
Te=str2double(get(handles.tbTe,'string'));
step=str2double(get(handles.tbStep,'string'));
figure
try
for t=Ts:step:Te
    T1=YT+(Tr-YT)*erf(X1./(2*sqrt(Alpha*t)));
    plot(Xp,T1,'linewidth',3)
    title(['Heat transfer @Time = ' num2str(t) ' sec']...
        , 'FontSize',14,'fontweight','b')
    ylabel('Temperature (C)','FontSize',14,'fontweight','b')
    xlabel('Distance (m)','FontSize',14,'fontweight','b')
    axis([0 Ls 0 YT])
    set(gca,'FontSize',14,'fontweight','b');
    F1(t)=getframe;
end
catch
    [user,sys] = memory;

```



```

    txt=user.MemAvailableAllArrays;
    msgbox(['Available Memory ' num2str(txt) ' byte'],'Memory');
    A='Out of memory';
    B='Please decrease stop time';
    C='Or';
    D='Increase step size';
    errordlg({A;B;C;D},'Error Report')
end
clear all;

function tbLs_Callback(hObject, eventdata, handles)
% hObject    handle to tbLs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbLs as text
%         str2double(get(hObject,'String')) returns contents of tbLs as a
double

% --- Executes during object creation, after setting all properties.
function tbLs_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbLs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbTr_Callback(hObject, eventdata, handles)
% hObject    handle to tbTr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTr as text
%         str2double(get(hObject,'String')) returns contents of tbTr as a
double

```

```

% --- Executes during object creation, after setting all properties.
function tbTr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function tbYT_Callback(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYT as text
%         str2double(get(hObject,'String')) returns contents of tbYT as a
double

% --- Executes during object creation, after setting all properties.
function tbYT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbTime_Callback(hObject, eventdata, handles)
% hObject    handle to tbTime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTime as text
%         str2double(get(hObject,'String')) returns contents of tbTime as a
double

```

```

% --- Executes during object creation, after setting all properties.
function tbTime_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbAlpha_Callback(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbAlpha as text
%         str2double(get(hObject,'String')) returns contents of tbAlpha as a
double
% --- Executes during object creation, after setting all properties.
function tbAlpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btRun.
function btRun_Callback(hObject, eventdata, handles)
% hObject    handle to btRun (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Alpha = str2double(get(handles.tbAlpha,'string'));
Ls=str2double(get(handles.tbLs,'string'));           %meter
YT=str2double(get(handles.tbYT,'string'));
Tr=str2double(get(handles.tbTr,'string'));

```



```

t=str2double(get(handles.tbTime,'string')); % lmin

X1=0.001:0.001:Ls; %meter
% X2=zeros(1,10);
% Xk=[X2 X1];
% X=zeros(1,length(X1));
% for u=1:1:length(X1)
%     X(1,u)=Xk(1,u);
% end
L=length(X1);
Xp=0.005:0.005:0.005*L;

T1=YT+(Tr-YT)*erf(X1./(2*sqrt(Alpha*t)));
figure
plot(Xp,T1,'linewidth',3)
title(['Heat transfer @Time = ' num2str(t) '
sec'],'FontSize',14,'fontWeight','b')
ylabel('Temperature (C)','FontSize',14,'fontWeight','b')
xlabel('Distance (m)','FontSize',14,'fontWeight','b')
axis([0 Ls 0 YT])
set(gca,'FontSize',14,'fontWeight','b');

```

**E.4 : Calculate and display the spread of heat at any time displaying in 2 different styles: 1 unit cell and multi-cell.**

#### **LTSolution**

```

clc
clear

Alpha = 0.6734e-6;
t=10;
YL=10; %Desire length (cm)
Tr=27;
YT=240;

figure
T1=Merotate(Alpha,t,YL,Tr,YT);
surf(T1);
title(['Unit Cell size(L) = ' num2str(YL) 'x' num2str(YL)...
' cm @Time = ' num2str(t) ' sec'],'FontSize',14,'fontWeight','b');
colormap(jet);

```

```

xlabel('X (mm.)','FontSize',14,'fontweight','b');
ylabel('Y (mm.)','FontSize',14,'fontweight','b');
zlabel('Temperature (C)','FontSize',14,'fontweight','b');
colorbar
set(gca,'FontSize',14,'fontweight','b');

figure
Ta=[T1,T1,T1;T1,T1,T1;T1,T1,T1];
surf(Ta);
title(['Multi Unit cell size 3x3 unit cell '...
      '@Time = ' num2str(t) ' sec'],'FontSize',14,'fontweight','b');
colormap(jet);
xlabel('X (mm.)','FontSize',14,'fontweight','b');
ylabel('Y (mm.)','FontSize',14,'fontweight','b');
zlabel('Temperature (C)','FontSize',14,'fontweight','b');
colorbar
set(gca,'FontSize',14,'fontweight','b');

```

## LTG

```

function varargout = LTG(varargin)
% LTG MATLAB code for LTG.fig
%   LTG, by itself, creates a new LTG or raises the existing
%   singleton*.
%
%   H = LTG returns the handle to a new LTG or the handle to
%   the existing singleton*.
%
%   LTG('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in LTG.M with the given input arguments.
%
%   LTG('Property','Value',...) creates a new LTG or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before LTG_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to LTG_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help LTG

```

```

% Last Modified by GUIDE v2.5 29-Jul-2013 21:55:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @LTG_OpeningFcn, ...
                  'gui_OutputFcn',  @LTG_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before LTG is made visible.
function LTG_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to LTG (see VARARGIN)

% Choose default command line output for LTG
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes LTG wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = LTG_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB

```



```

% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in btRun.
function btRun_Callback(hObject, eventdata, handles)
% hObject      handle to btRun (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
Alpha =str2double(get(handles.tbAlpha,'string'));
t=str2double(get(handles.tbTime,'string'));
YL=str2double(get(handles.tbYL,'string'));           %Desire length (cm)
Tr=str2double(get(handles.tbTr,'string'));
YT=str2double(get(handles.tbYT,'string'));

figure
T1=Merotate(Alpha,t,YL,Tr,YT);
surf(T1);
title(['Unit Cell size(L) = ' num2str(YL) 'x' num2str(YL)...
      ' cm'];['@Time = ' num2str(t) ' sec'],'FontSize',14,'fontweight','b');
colormap(jet);
xlabel('X (mm.)','FontSize',14,'fontweight','b');
ylabel('Y (mm.)','FontSize',14,'fontweight','b');
zlabel('Temperature (C)','FontSize',14,'fontweight','b');
ht=colorbar;
title(ht,'Temp. (C)','FontSize',14,'fontweight','b');
set(gca,'FontSize',14,'fontweight','b');

figure
Ta=[T1,T1,T1;T1,T1,T1;T1,T1,T1];
surf(Ta);
title('Multi Unit cell size 3x3 unit cell ');...
      ['@Time = ' num2str(t) ' sec'],'FontSize',14,'fontweight','b');
colormap(jet);
xlabel('X (mm.)','FontSize',14,'fontweight','b');
ylabel('Y (mm.)','FontSize',14,'fontweight','b');
zlabel('Temperature (C)','FontSize',14,'fontweight','b');
ht=colorbar;
title(ht,'Temp. (C)','FontSize',14,'fontweight','b');
set(gca,'FontSize',14,'fontweight','b');

```

```

function tbYL_Callback(hObject, eventdata, handles)
% hObject    handle to tbYL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYL as text
%        str2double(get(hObject,'String')) returns contents of tbYL as a
double

% --- Executes during object creation, after setting all properties.
function tbYL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbYT_Callback(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbYT as text
%        str2double(get(hObject,'String')) returns contents of tbYT as a
double

% --- Executes during object creation, after setting all properties.
function tbYT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbYT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function tbTr_Callback(hObject, eventdata, handles)
% hObject    handle to tbTr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTr as text
%        str2double(get(hObject,'String')) returns contents of tbTr as a
double

% --- Executes during object creation, after setting all properties.
function tbTr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function tbTime_Callback(hObject, eventdata, handles)
% hObject    handle to tbTime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbTime as text
%        str2double(get(hObject,'String')) returns contents of tbTime as a
double

% --- Executes during object creation, after setting all properties.
function tbTime_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbTime (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

function tbAlpha_Callback(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tbAlpha as text
%        str2double(get(hObject,'String')) returns contents of tbAlpha as a
double

% --- Executes during object creation, after setting all properties.
function tbAlpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tbAlpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

### **E.5: Calculate and display the spread of heat in less than or equal to 10 seconds.**

#### **MainProgram**

```

clc
clear
%Constance paramters
rho=1820;      %mass density (kg/m^3)
k=1.0353;     %thermal Conductivity (W/m K)
%Alpha from measurement = 0.6734
Alpha = 0.6734e-6; %Thermal diffusivity (m^2/s)
Cp = k/(rho*Alpha); %specific heat (kj/kg k)
%Am=Alpha*60;   %Thermal diffusivity (m^2/min)

%Heat Equation
%Force Temperture at this point
Tr=27;        %Room temperature
L=10;         %unit cell size in cm
YT=240;       %True temperature
dt=10;        %time max 1.5 sec
dx=0.005;     %meter
F0=Alpha*dt/(dx^2);

```

```

%unit cell plot
T=gendata(L,YT,dx,Tr); %output of gendata is unit cell

[ST,~]=size(T);
m=1;
n=1;
B=0;
while(m<=ST)
    while(n<=ST)
        B=defindB(m,n,ST);
        %disp(B)
        if(B>=30)
            T=Mecorner(m,n,T,F0,B);
        end
        if(B>=20&&B<30)
            T=Meboundary(m,n,T,F0,B);
        end
        if(B==1)
            T=Meinterior(m,n,T,F0);
        end
        n=n+1;
    end
    m=m+1;
    n=1;
end
T2=fliplr(T);
T3=[T2 T];
T4=flipud(T3);
T1=[T4;T3];

figure('name','Single Unit Cell','NumberTitle','off');
surf(T1);
title(['time = ' num2str(dt) ' sec F0 = ' num2str(F0) ' L = ' num2str(L)...
      ' cm dx = ' num2str(dx*100) ' cm'],'FontSize',14,'fontweight','b');
colormap(jet);
xlabel('X (L/dx)','FontSize',14,'fontweight','b');
ylabel('Y (L/dx)','FontSize',14,'fontweight','b');
zlabel('Temperature (C)','FontSize',14,'fontweight','b');
colorbar
set(gca,'FontSize',14,'fontweight','b');

%Multiple Unit Cells
%3x3 unit cells

```

```

Tcell=[T1,T1,T1;T1,T1,T1;T1,T1,T1];
figure('name','Multiple Unit Cells','NumberTitle','off');
surf(Tcell);
title(['time = ' num2str(dt) ' sec F0 = ' num2str(F0) ' L = ' num2str(L*3)...
      ' cm dx = ' num2str(dx*100) ' cm'],'FontSize',14,'fontweight','b');
colormap(jet);
xlabel('X (L/dx)','FontSize',14,'fontweight','b');
ylabel('Y (L/dx)','FontSize',14,'fontweight','b');
zlabel('Temperature (C)','FontSize',14,'fontweight','b');
colorbar
set(gca,'FontSize',14,'fontweight','b');

```

### NumericalHeat

```

function varargout = NumericalHeat(varargin)
% NUMERICALHEAT MATLAB code for NumericalHeat.fig
%
%   NUMERICALHEAT, by itself, creates a new NUMERICALHEAT or raises the
existing
%   singleton*.
%
%
%   H = NUMERICALHEAT returns the handle to a new NUMERICALHEAT or the
handle to
%   the existing singleton*.
%
%
%   NUMERICALHEAT('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in NUMERICALHEAT.M with the given input
arguments.
%
%
%   NUMERICALHEAT('Property','Value',...) creates a new NUMERICALHEAT or
raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before NumericalHeat_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to NumericalHeat_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help NumericalHeat

```



```

% Last Modified by GUIDE v2.5 28-Jul-2013 22:39:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @NumericalHeat_OpeningFcn, ...
                  'gui_OutputFcn',  @NumericalHeat_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before NumericalHeat is made visible.
function NumericalHeat_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to NumericalHeat (see VARARGIN)

% Choose default command line output for NumericalHeat
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes NumericalHeat wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = NumericalHeat_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in btRun.
function btRun_Callback(hObject, eventdata, handles)
% hObject    handle to btRun (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

rho=str2double(get(handles.rhoIn,'string'));    %mass density (kg/m^3)
k=str2double(get(handles.kIn,'string'));    %thermal Conductivity (W/m K)
%Alpha from measurement = 0.6734
Alpha = str2double(get(handles.alphaIn,'string')); %Thermal diffusivity
(m^2/s)
Cp = str2double(get(handles.cpIn,'string')); %specific heat (kJ/kg k)
%Am=Alpha*60;    %Thermal diffusivity (m^2/min)

%Heat Equation
%Force Temperture at this point
Tr=str2double(get(handles.TrIn,'string'));    %Room temperature
L=str2double(get(handles.LIn,'string'));    %unit cell size in cm
YT=str2double(get(handles.YTIn,'string'));    %True temperature
dt=str2double(get(handles.dtIn,'string'));    %time max 1.5 sec
dx=0.005;    %meter
F0=Alpha*dt/(dx^2);
%unit cell plot
T=gendata(L,YT,dx,Tr); %output of gendata is unit cell

[ST,~]=size(T);
m=1;
n=1;
B=0;
while(m<=ST)
    while(n<=ST)
        B=defindB(m,n,ST);
        %disp(B)
        if(B>=30)
            T=Mecorner(m,n,T,F0,B);
        end
        if(B>=20&&B<30)
            T=Meboundary(m,n,T,F0,B);
        end
        if(B==1)

```

```

        T=Meinterior(m,n,T,F0);
    end
    n=n+1;
end
m=m+1;
n=1;
end
T2=fliplr(T);
T3=[T2 T];
T4=flipud(T3);
T1=[T4;T3];

figure('name','Single Unit Cell','NumberTitle','off');
surf(T1);
title(['time = ' num2str(dt) ' sec F0 = ' num2str(F0)];[' L = '
num2str(L)...
' cm dx = ' num2str(dx*100) ' cm]], 'FontSize',14,'fontweight','b');
colormap(jet);
xlabel('X (L/dx)', 'FontSize',14,'fontweight','b');
ylabel('Y (L/dx)', 'FontSize',14,'fontweight','b');
zlabel('Temperature (C)', 'FontSize',14,'fontweight','b');
ht=colorbar;
title(ht, 'Temp. (C)', 'FontSize',14,'fontweight','b');
set(gca, 'FontSize',14,'fontweight','b');

%Multiple Unit Cells
%3x3 unit cells
Tcell=[T1,T1,T1;T1,T1,T1;T1,T1,T1];
figure('name','Multiple Unit Cells','NumberTitle','off');
surf(Tcell);
title(['time = ' num2str(dt) ' sec F0 = ' num2str(F0)];[' L = '
num2str(L*3)...
' cm dx = ' num2str(dx*100) ' cm]], 'FontSize',14,'fontweight','b');
colormap(jet);
xlabel('X (L/dx)', 'FontSize',14,'fontweight','b');
ylabel('Y (L/dx)', 'FontSize',14,'fontweight','b');
zlabel('Temperature (C)', 'FontSize',14,'fontweight','b');
ht=colorbar;
title(ht, 'Temp. (C)', 'FontSize',14,'fontweight','b');
set(gca, 'FontSize',14,'fontweight','b');

function dtIn_Callback(hObject, eventdata, handles)
% hObject    handle to dtIn (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dtIn as text
% str2double(get(hObject,'String')) returns contents of dtIn as a
double

% --- Executes during object creation, after setting all properties.
function dtIn_CreateFcn(hObject, eventdata, handles)
% hObject handle to dtIn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
function YTin_Callback(hObject, eventdata, handles)
% hObject handle to YTin (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of YTin as text
% str2double(get(hObject,'String')) returns contents of YTin as a
double

% --- Executes during object creation, after setting all properties.
function YTin_CreateFcn(hObject, eventdata, handles)
% hObject handle to YTin (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
function TrIn_Callback(hObject, eventdata, handles)
% hObject handle to TrIn (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TrIn as text
%         str2double(get(hObject,'String')) returns contents of TrIn as a
double

% --- Executes during object creation, after setting all properties.
function TrIn_CreateFcn(hObject, eventdata, handles)
% hObject      handle to TrIn (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function LIn_Callback(hObject, eventdata, handles)
% hObject      handle to LIn (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LIn as text
%         str2double(get(hObject,'String')) returns contents of LIn as a
double

% --- Executes during object creation, after setting all properties.
function LIn_CreateFcn(hObject, eventdata, handles)
% hObject      handle to LIn (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dxIn_Callback(hObject, eventdata, handles)
% hObject      handle to dxIn (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dxIn as text
%          str2double(get(hObject,'String')) returns contents of dxIn as a
double

% --- Executes during object creation, after setting all properties.
function dxIn_CreateFcn(hObject, eventdata, handles)
% hObject      handle to dxIn (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function rhoIn_Callback(hObject, eventdata, handles)
% hObject      handle to rhoIn (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of rhoIn as text
%          str2double(get(hObject,'String')) returns contents of rhoIn as a
double

% --- Executes during object creation, after setting all properties.
function rhoIn_CreateFcn(hObject, eventdata, handles)
% hObject      handle to rhoIn (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```



```

function kIn_Callback(hObject, eventdata, handles)
% hObject    handle to kIn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of kIn as text
%        str2double(get(hObject,'String')) returns contents of kIn as a
double

% --- Executes during object creation, after setting all properties.
function kIn_CreateFcn(hObject, eventdata, handles)
% hObject    handle to kIn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function alphaIn_Callback(hObject, eventdata, handles)
% hObject    handle to alphaIn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of alphaIn as text
%        str2double(get(hObject,'String')) returns contents of alphaIn as a
double

% --- Executes during object creation, after setting all properties.
function alphaIn_CreateFcn(hObject, eventdata, handles)
% hObject    handle to alphaIn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function cpIn_Callback(hObject, eventdata, handles)
% hObject    handle to cpIn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cpIn as text
%        str2double(get(hObject,'String')) returns contents of cpIn as a
double

% --- Executes during object creation, after setting all properties.
function cpIn_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cpIn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

## E.6: Function

### Merotate

```

function [T1]=Merotate(Alpha,t,YL,Tr,YT)
% Alpha = 0.6734e-6;
% t=1.5*60*60;
%
% YL=6;           %Desire length (cm)
X=0.001:0.001:0.01*YL/2; %meter
L=length(X);

T=Tr.*ones(L,L);
for m=1:1:L
    for n=1:1:L
        C=sqrt(X(m)^2+X(n)^2);
        if (C<=0.0101)
            T(m,n)=YT;
            Cold=C;
        else
            T(m,n)=YT+(Tr-YT)*erf((C-Cold)/(2*sqrt(Alpha*t)));
        end
    end
end

```

```

end
end

T2=fliplr(T);
T3=[T2 T];
T4=flipud(T3);
T1=[T4;T3];

% figure
% surf(T1)
end

% surf(T1);
% title(['Unit Cell size(L) = ' num2str(YL) 'x' num2str(YL)...
%       ' cm @Time = ' num2str(tm(m)) ' sec'],'FontSize',14,'fontweight','b');
% colormap(jet);
% xlabel('X (mm.)','FontSize',14,'fontweight','b');
% ylabel('Y (mm.)','FontSize',14,'fontweight','b');
% zlabel('Temperature (C)','FontSize',14,'fontweight','b');
% colorbar
% set(gca,'FontSize',14,'fontweight','b');

% Ta=[T1,T1,T1;T1,T1,T1;T1,T1,T1;];
% surf(Ta);
% title(['Multi Unit cell size 3x3 unit cell '...
%       '@Time = ' num2str(t(m)) ' sec'],'FontSize',14,'fontweight','b');
% colormap(jet);
% xlabel('X (mm.)','FontSize',14,'fontweight','b');
% ylabel('Y (mm.)','FontSize',14,'fontweight','b');
% zlabel('Temperature (C)','FontSize',14,'fontweight','b');
% colorbar
% set(gca,'FontSize',14,'fontweight','b');

```

### defindB

```

function [B]=defindB(m,n,ST)
%interior B=1x;
%Boundary B=2x;
%corner B=3x;
%ST = size of T
B=0;
%Corner *****
if (m==1&n==1)

```



```

        B=31;    %C1
    end
    if (m==1&&n==ST)
        B=32;    %C2
    end
    if (m==ST&&n==ST)
        B=33;    %C3
    end
    if (m==ST&&n==1)
        B=34;    %C4
    end
    %Boundary *****
    if (m==1&&B==0)
        B=21;    %B1
    end
    if (n==ST&&B==0)
        B=22;    %B2
    end
    if (m==ST&&B==0)
        B=23;    %B3
    end
    if (n==1&&B==0)
        B=24;    %B4
    end
    %Interior *****
    if (B==0)
        B=1;
    end
end
end

```

### diffu

```

function [out]=diffu(T1,T2,T3,T4,Tmn)
A=[T1 T2 T3 T4];
B=max(A);
if (Tmn<B)
    out=1;
else
    out=0;
end
end

```

**gendata**

```

function [T]=gendata(L,YT,dx,Tr)
Lm=L*0.01;           %in meter
S=0:dx:Lm;          %S in meter
Sa=length(S)-1;
T=Tr.*ones(Sa/2,Sa/2);
for m=1:1:Sa/2
    for n=1:1:Sa/2
        A=S(m); %in meter
        B=S(n); %in meter
        C=sqrt(A^2+B^2);
        if(C<=0.01)
            %disp(['C=' num2str(C) ' m=' num2str(m) ' n=' num2str(n)])
            T(m,n)=YT;
        end
    end
end
end

```

**Meboundary**

```

function [T]=Meboundary(m,n,T,F0,B)
%Top
if(B==21)
    T1=T(m,n-1);
    T2=T(m+1,n);
    T3=T(m,n+1);
    T4=0;
end
%right
if(B==22)
    T1=T(m-1,n);
    T2=T(m,n-1);
    T3=T(m+1,n);
    T4=0;
end
%button
if(B==23)
    T1=T(m,n+1);
    T2=T(m-1,n);
    T3=T(m,n-1);
    T4=0;
end
%left

```

```

if (B==24)
    T1=T(m+1,n);
    T2=T(m,n+1);
    T3=T(m-1,n);
    T4=0;
end

Tmn=T(m,n);
R=diffu(T1,T2,T3,T4,Tmn);
if (R>0)
    T(m,n)=F0*(2*T2+T1+T3)+(1-4*F0)*Tmn;
end
end

```

### Mecornor

```

function [T]=Mecorner(m,n,T,F0,B)
%top left
if (B==31)
    T1=T(m,n+1);
    T2=T(m+1,n);
end
%top right
if (B==32)
    T1=T(m,n-1);
    T2=T(m+1,n);
end
%button right
if (B==33)
    T1=T(m-1,n);
    T2=T(m,n-1);
end
%button left
if (B==34)
    T1=T(m-1,n);
    T2=T(m,n+1);
end
Tmn=T(m,n);
R=diffu(T1,T2,0,0,Tmn);
if (R>0)
    T(m,n)=2*F0*(T1+T2)+(1-4*F0)*Tmn;
end
end

```



**Meinterior**

```
function [T]=Meinterior(m,n,T,F0)
T1=T(m,n-1);
T2=T(m-1,n);
T3=T(m,n+1);
T4=T(m+1,n);
Tmn=T(m,n);
R=diffu(T1,T2,T3,T4,Tmn);
if(R>0)
    T(m,n)=F0*(T(m-1,n)+T(m,n+1)+T(m+1,n)+T(m,n-1))+(1-4*F0)*T(m,n);
end
end
```



APPENDIX F THE DESIGN OF THERMAL ENERGY STORAGE  
PROTOTYPE

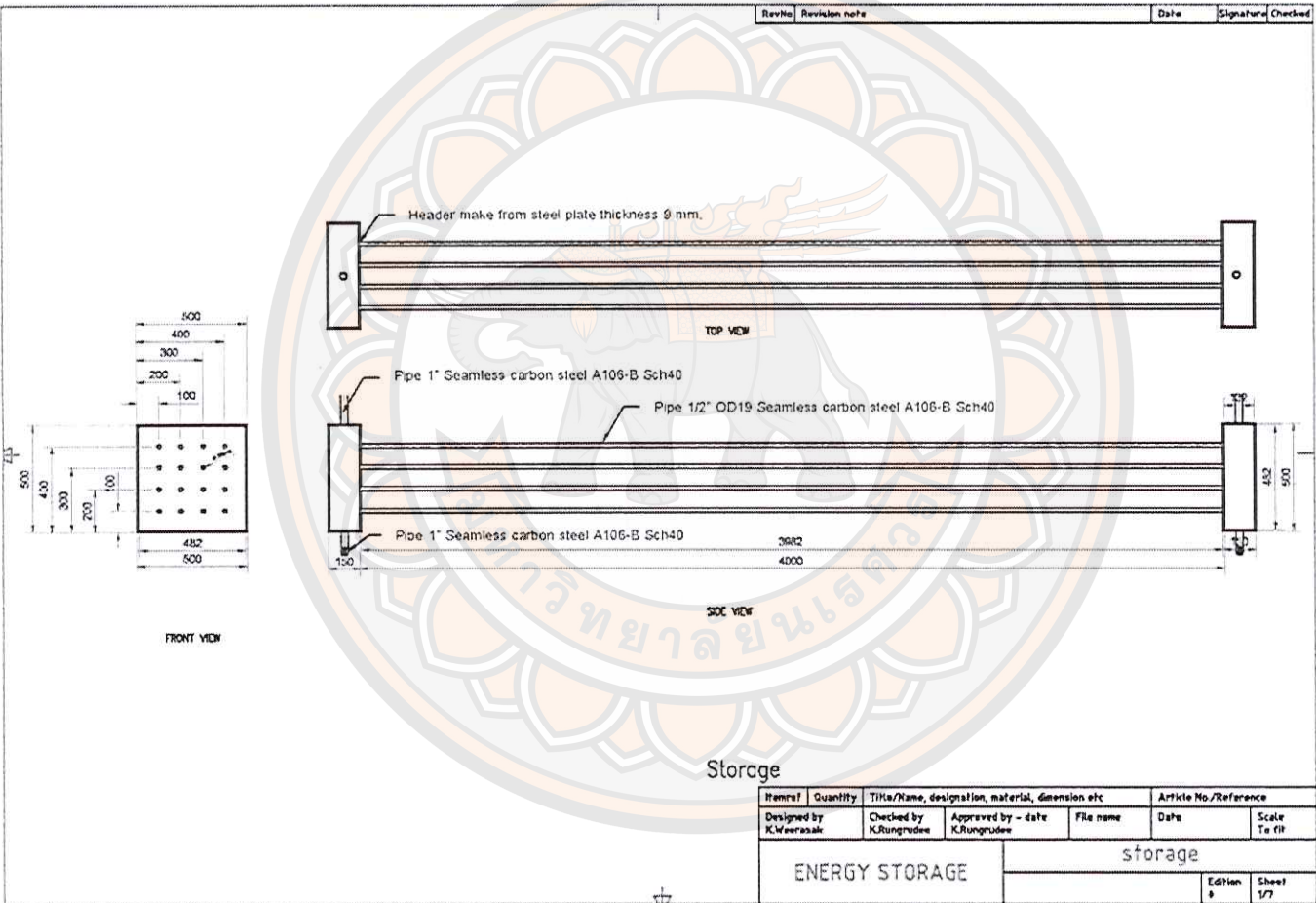


Figure 44 Front view of the thermal energy storage prototype

Figure 45 Concrete foundation

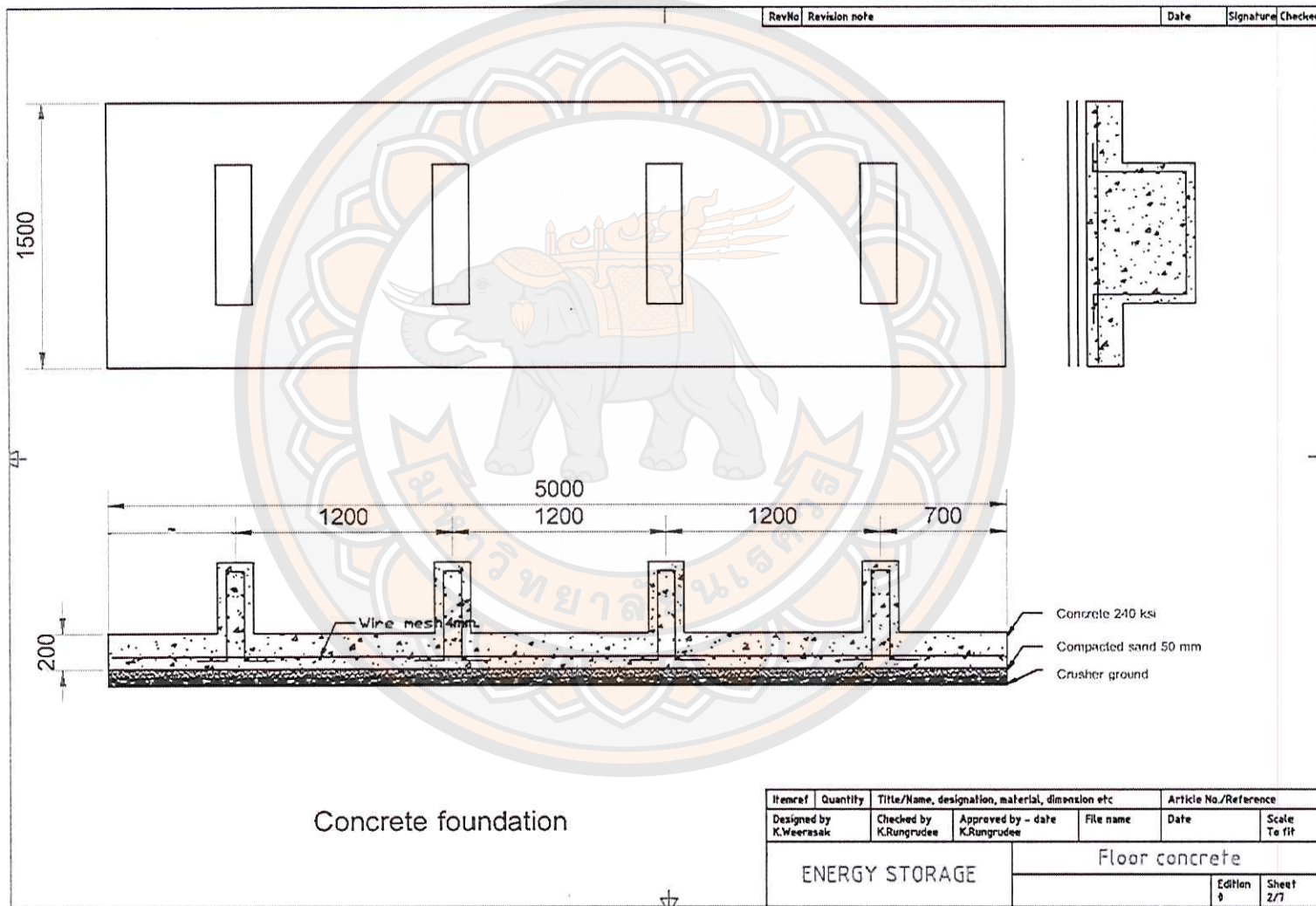




Figure 46 Structural components (A)

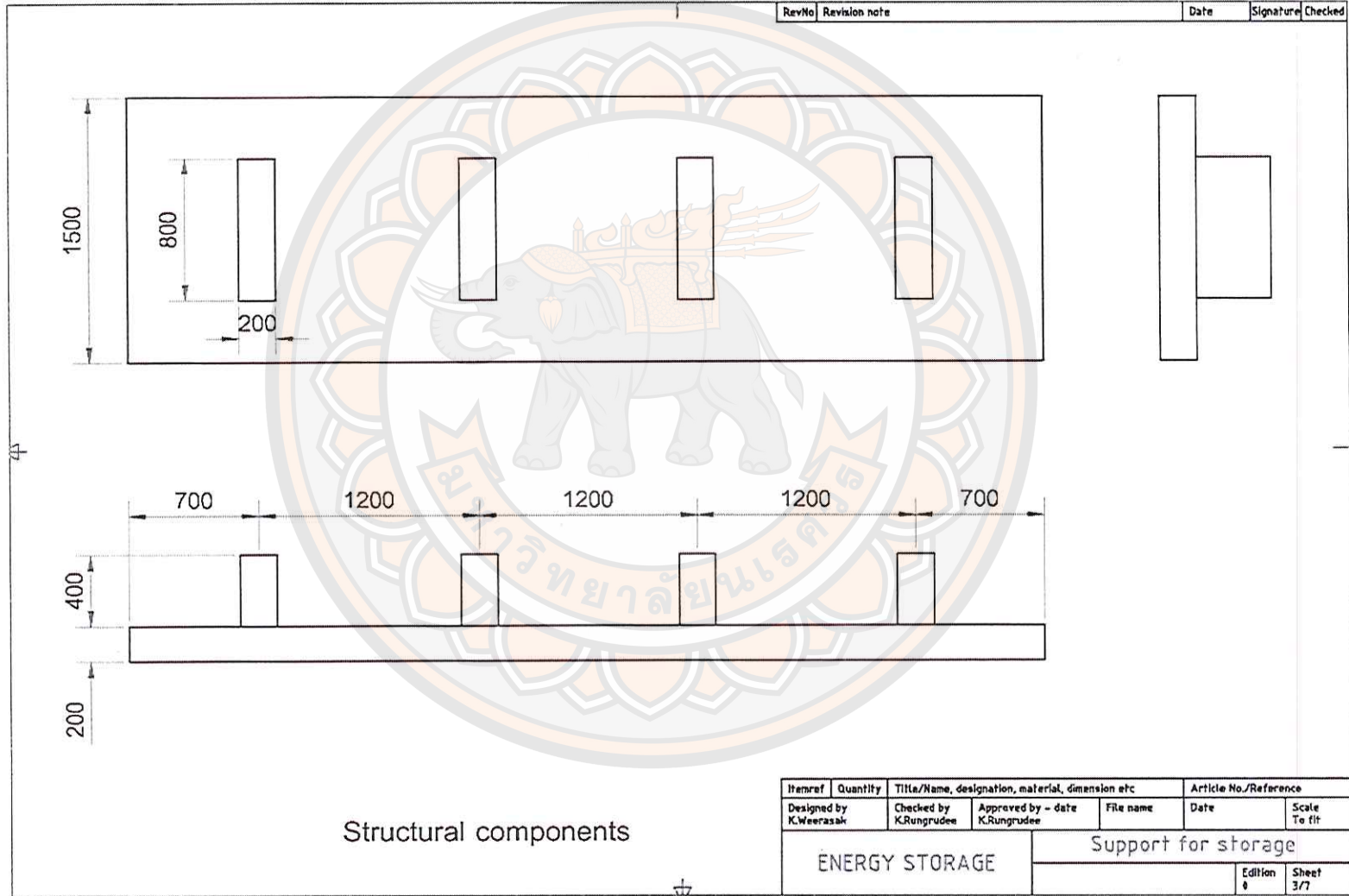


Figure 47 Structural components (B)

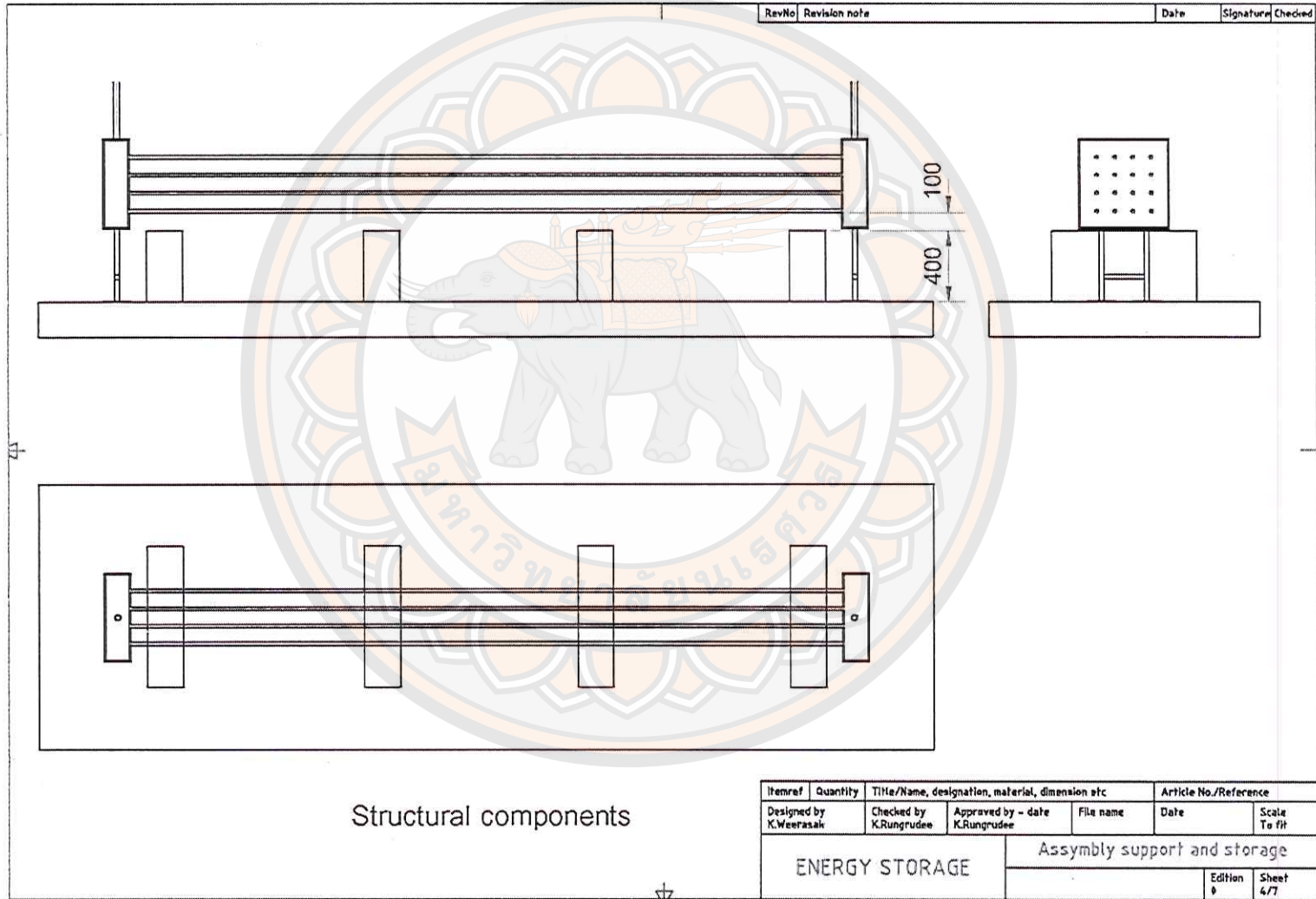


Figure 48 Reinforced Concrete (A-B)

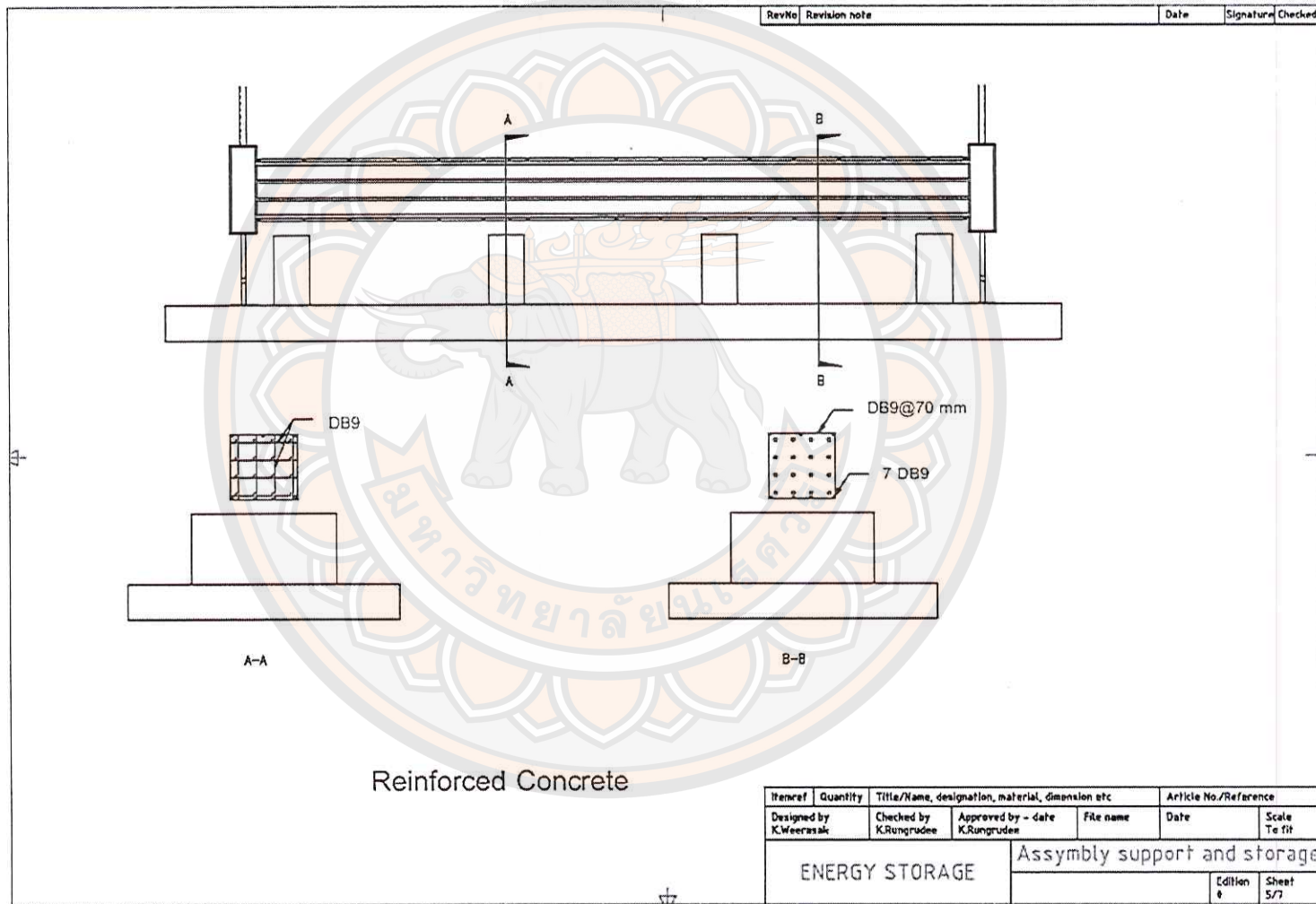




Figure 49 Reinforced Concrete (C-D)

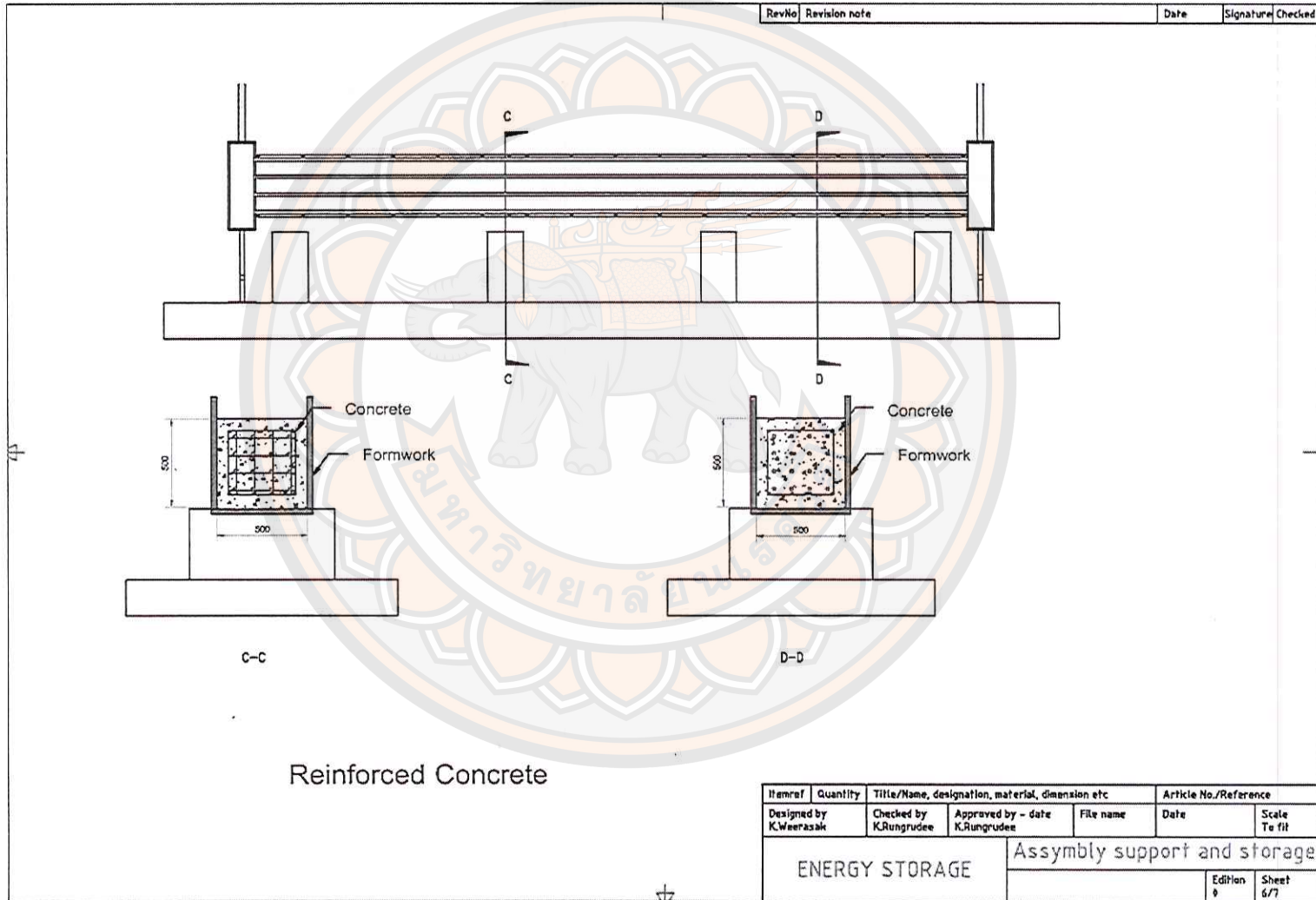
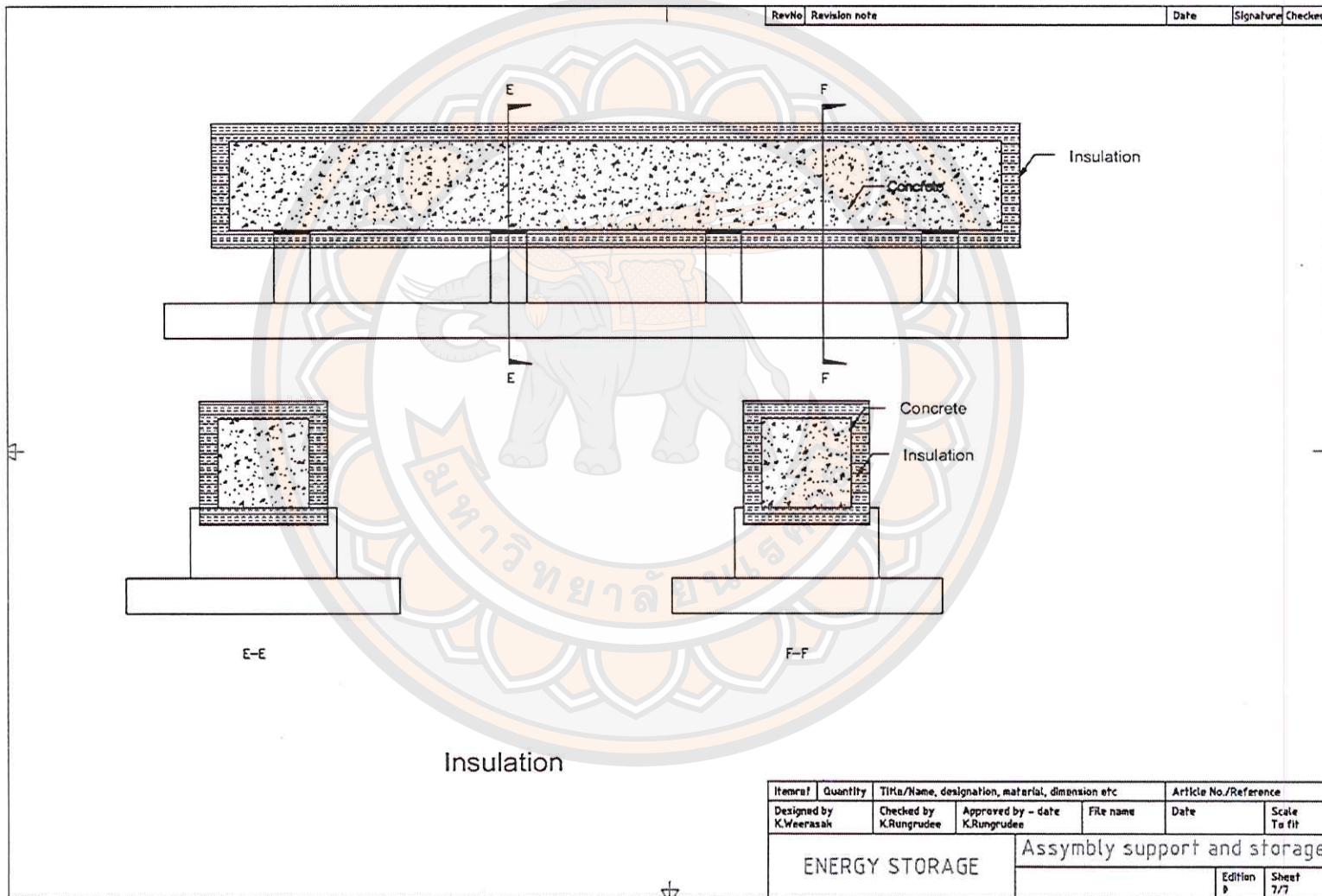


Figure 50 Insulation



**APPENDIX G SOLVING FOR ENERGY INPUT, ENERGY RECOVERED,  
EXERGY INPUT, EXERGY RECOVERED, ENERGY  
EFFICIENCY AND EXERGY EFFICIENCY**

The calculation of Energy and Exergy.

**For charging period**

At time = 0 minute  $\dot{m} = 0.009 \text{ kg/s}$ ,

Temperature input = 183.2 °C      Temperature output = 169.7 °C

Enthalpy input ( $h_a$ ) = 2782.4800 kJ/kg      Enthalpy output ( $h_b$ ) = 2236.0800 kJ/kg

Entropy input ( $S_a$ ) = 6.5645 kJ/kg K      Entropy output ( $S_b$ ) = 5.4673 kJ/kg K

1. Energy input =  $\dot{m}(h_a - h_b)$

$$= 0.009 \text{ kg/s} \times (2782.4800 \text{ kJ/kg} - 2236.0800 \text{ kJ/kg}) \times 10 \text{ s}$$

$$= 49.18 \text{ J}$$

2. Exergy input =  $\dot{m}[(h_a - h_b) - T_0(s_a - s_b)]$

$$= [0.009 \text{ kg/s} \times (2782.4800 \text{ kJ/kg} - 2236.0800 \text{ kJ/kg})] - [298.15 \text{ K}$$

×

$$(6.5645 \text{ kJ/kg K} - 5.4673 \text{ kJ/kg K})] \times 10 \text{ s}$$

$$= 19.73 \text{ J}$$

**For discharging period**

At time = 0 minute  $\dot{m} = 0.009 \text{ kg/s}$ ,

Temperature input = 112.2 °C      Temperature output = 118.9 °C

Enthalpy input ( $h_c$ ) = 824.8109 kJ/kg      Enthalpy output ( $h_d$ ) = 930.2050 kJ/kg

Entropy input ( $S_c$ ) = 2.3619 kJ/kg K      Entropy output ( $S_d$ ) = 2.6156 kJ/kg K



$$1. \text{ Energy recovered} = \dot{m}(h_d - h_c)$$

$$= 0.009 \text{ kg/s} \times (930.2050 \text{ kJ/kg} - 824.8109 \text{ kJ/kg}) \times 10 \text{ s}$$

$$= 9.49 \text{ J}$$

$$2. \text{ Exergy recovered} = \dot{m}(\epsilon_d - \epsilon_c)$$

$$= \dot{m}[(h_d - h_c) - T_0(s_d - s_c)]$$

$$= [0.009 \text{ kg/s} \times (930.2050 \text{ kJ/kg} - 824.8109 \text{ kJ/kg})]$$

$$- [298.15 \text{ K} \times (2.6156 \text{ kJ/kg K} - 2.3619 \text{ kJ/kg K})] \times 10 \text{ s}$$

$$= 2.68 \text{ J}$$

**Energy Efficiency;**

$$\eta = \frac{\text{Energy recovered from TES during discharging}}{\text{Energy input to TES during charging}}$$

$$= \frac{\dot{m}(h_d - h_c)}{\dot{m}(h_a - h_b)} = 1 - \frac{Q_l}{h_a - h_b}$$

$$= 19.29 \%$$

**Exergy Efficiency ;**

$$\psi = \frac{\text{Exergy recovered from TES during discharging}}{\text{Exergy input to TES during charging}}$$

$$= \frac{\dot{m}(\epsilon_d - \epsilon_c)}{\dot{m}(\epsilon_a - \epsilon_b)} = 1 - \frac{X_l}{\epsilon_a - \epsilon_b}$$

$$= 13.57 \%$$

The results of all calculation have shown in the following table

Table 20 The results of all calculation at flow rate 0.009 kg/s

Times (min)	Energy input (J)	Energy recovered (J)	Exergy input (J)	Exergy recovered (J)	Energy Efficiency (%)	Exergy Efficiency (%)
0	49.18	9.49	19.73	2.68	19.29	13.57
3	538.97	184.76	218.64	52.22	34.28	23.88
6	911.29	355.09	369.86	100.46	38.97	27.16
9	1270.91	520.32	515.82	147.26	40.94	28.55
12	1624.30	683.41	659.28	193.49	42.07	29.35
15	1974.09	844.76	801.16	239.25	42.79	29.86
18	2327.01	1012.63	944.21	286.90	43.52	30.38
21	2726.02	1183.20	1106.22	335.34	43.40	30.31
24	3130.40	1350.33	1270.45	382.83	43.14	30.13
27	3539.44	1515.61	1436.52	429.83	42.82	29.92
30	3957.54	1698.36	1606.25	481.97	42.91	30.01
33	4356.36	1896.90	1768.02	538.78	43.54	30.47
36	4783.74	2101.74	1941.21	597.50	43.94	30.78
39	5238.86	2308.91	2125.44	656.98	44.07	30.91
42	5700.23	2512.75	2311.59	715.48	44.08	30.95
45	6142.44	2714.05	2490.12	773.23	44.19	31.05
48	6577.28	2913.47	2666.24	830.40	44.30	31.15
51	7029.18	3114.66	2849.60	888.08	44.31	31.17
54	7464.67	3331.29	3026.37	950.38	44.63	31.40
57	7887.71	3526.63	3198.00	1006.39	44.71	31.47
60	8310.41	3677.66	3369.40	1049.38	44.25	31.14
63	8730.07	3857.45	3539.47	1100.75	44.19	31.10
66	9145.23	4047.65	3707.63	1155.16	44.26	31.16
69	9577.46	4240.32	3882.75	1210.29	44.27	31.17
72	10019.55	4430.69	4061.76	1264.71	44.22	31.14
75	10469.87	4617.08	4244.07	1317.91	44.10	31.05

Table 20 (cont.)

Times (min)	Energy input (J)	Energy recovered (J)	Exery input (J)	Exergy recovered (J)	Energy Efficiency (%)	Exergy Efficiency (%)
78	10930.82	4805.94	4430.58	1371.79	43.97	30.96
81	11379.19	4996.15	4611.86	1426.09	43.91	30.92
84	11746.05	5183.70	4760.35	1479.58	44.13	31.08
87	12060.44	5365.29	4887.68	1531.36	44.49	31.33
90	12397.55	5549.92	5024.28	1584.00	44.77	31.53
93	12748.82	5742.12	5166.73	1638.91	45.04	31.72
96	13116.81	5939.34	5315.82	1695.44	45.28	31.89
99	13502.26	6141.34	5471.95	1753.46	45.48	32.04
102	13842.37	6366.99	5609.76	1818.17	46.00	32.41
105	14136.12	6597.59	5728.66	1884.23	46.67	32.89
108	14411.45	6832.91	5839.99	1951.58	47.41	33.42
111	14730.93	7072.03	5969.26	2019.99	48.01	33.84
114	15126.51	7309.70	6129.58	2087.92	48.32	34.06
117	15610.25	7541.98	6325.89	2154.23	48.31	34.05
120	16092.76	7777.36	6521.71	2221.35	48.33	34.06
123	16554.35	8013.30	6708.83	2288.59	48.41	34.11
126	17025.49	8249.79	6899.70	2355.91	48.46	34.15
129	17500.18	8487.87	7091.90	2423.65	48.50	34.17
132	17971.50	8727.40	7282.52	2491.79	48.56	34.22
135	18474.92	8967.30	7485.84	2559.96	48.54	34.20
138	19075.15	9203.35	7728.66	2626.94	48.25	33.99
141	19728.41	9434.83	7992.98	2692.53	47.82	33.69
144	20338.69	9663.44	8239.96	2757.21	47.51	33.46
147	20833.33	9889.54	8440.54	2821.06	47.47	33.42
150	21270.45	10114.36	8617.94	2884.48	47.55	33.47
153	21769.79	10337.87	8820.42	2947.46	47.49	33.42



Table 21 (cont.)

Times (min)	Energy input (J)	Energy recovered (J)	Exery input (J)	Exergy recovered (J)	Energy Efficiency (%)	Exergy Efficiency (%)
174	30523.33	13911.89	11999.98	45.58	32.88	38.56
177	30853.34	14138.76	12135.32	45.83	33.04	38.60
180	31119.98	14363.05	12244.63	46.15	33.25	38.65

Table 22 The results of all calculation at flow rate 0.014 kg/s

Times (min)	Energy input (J)	Energy recovered (J)	Exery input (J)	Exergy recovered (J)	Energy Efficiency (%)	Exergy Efficiency (%)
0	68.63	9.15	27.51	0.61	13.33	10.57
3	526.78	76.79	213.69	12.44	14.58	11.51
6	831.91	131.95	338.22	26.12	15.86	12.49
9	1066.83	183.02	434.15	41.74	17.16	13.47
12	1323.18	243.68	539.16	59.31	18.42	14.43
15	1623.87	318.39	662.33	78.44	19.61	15.33
18	1958.67	404.79	799.27	97.85	20.67	16.12
21	2296.00	493.86	937.22	115.21	21.51	16.74
24	2647.65	592.25	1080.71	134.33	22.37	17.37
27	2950.52	685.73	1204.39	154.53	23.24	18.02
30	3141.27	759.46	1282.58	176.11	24.18	18.72
33	3334.65	836.98	1361.61	198.78	25.10	19.41
36	3528.57	917.45	1440.67	222.31	26.00	20.08
39	3744.86	1005.95	1529.07	246.48	26.86	20.72
42	3943.92	1093.30	1610.70	271.55	27.72	21.36

Table 22 (cont.)

Times (min)	Energy input (J)	Energy recovered (J)	Exergy input (J)	Exergy recovered (J)	Energy Efficiency (%)	Exergy Efficiency (%)
45	4184.63	1192.52	1709.40	296.31	28.50	21.93
48	4456.12	1300.83	1820.69	320.36	29.19	22.44
51	4699.19	1397.41	1920.30	340.42	29.74	22.83
54	4889.01	1480.05	1998.12	359.84	30.27	23.22
57	5162.24	1584.59	2109.63	377.93	30.70	23.51
60	5565.88	1720.70	2273.89	392.59	30.92	23.65
63	6008.92	1872.76	2453.92	409.35	31.17	23.80
66	6490.69	2039.74	2649.51	427.55	31.43	23.96
69	6918.19	2197.13	2823.23	447.57	31.76	24.18
72	7241.07	2336.23	2954.59	472.01	32.26	24.54
75	7434.52	2449.10	3033.64	500.78	32.94	25.03
78	7560.92	2549.52	3085.44	532.77	33.72	25.61
81	7689.14	2653.65	3137.99	566.45	34.51	26.19
84	7900.18	2781.37	3224.44	599.43	35.21	26.70
87	8148.48	2919.95	3325.82	631.64	35.83	27.15
90	8451.65	3072.70	3449.34	661.83	36.36	27.53
93	8672.52	3208.98	3539.56	695.93	37.00	27.99
96	8765.42	3317.98	3577.63	736.56	37.85	28.62
99	8836.05	3422.68	3606.57	779.16	38.74	29.28
102	8910.64	3530.98	3637.13	823.53	39.63	29.94
105	9146.30	3696.66	3733.44	869.69	40.42	30.51
108	9401.45	3878.73	3837.95	920.66	41.26	31.13
111	9665.67	4073.77	3946.31	976.40	42.15	31.78
114	9915.02	4272.58	4048.53	1036.55	43.09	32.47



fluid. The present work seeks to remedy this perceived lack by investigating the thermal storage performance of solid state SHS systems, using local concrete material, in Thailand, under the various climatic conditions to be found in Thailand.

## II. THEORY OF THERMAL ENERGY STORAGE PERFORMANCE

The overall storage process of the thermal energy storage process is shown in Fig 1.



Fig 1. The three stage in a simple heat storage, charging period (a), Storage period (b), and discharging period (c) [8].

### A. Storage Equation

The storage system is the sensible heat in solid storage medium. An energy balance for the overall storage process can be written as [8]

$$\text{Energy input} - (\text{Energy recovered} + \text{Energy loss}) = \text{Energy accumulation} \quad (1)$$

$$\text{or} \quad \dot{m}(h_a - h_b) - [\dot{m}(h_d - h_c) + Q_l] = \Delta E \quad (1a)$$

where  $h_a, h_b, h_c$  and  $h_d$  are the total enthalpies of the flows at states  $a, b, c$ , and  $d$ , respectively, and  $Q_l$  denotes the heat loss during the process and  $\Delta E$  is the accumulation of energy in the TES.  $(h_a - h_b)$  represents the net heat delivered to the TES and  $(h_c - h_d)$  the net heat recovered from the TES.

### B. Charging period

Charging time is the time taken for the stored bed's volume average temperature to reach a specified rise in temperature  $\Delta T$ . An energy balance for the charging period can be written as follows: [8]

$$\text{Energy input} - \text{Energy loss} = \text{Energy accumulation} \quad (2)$$

$$\text{or} \quad \dot{m}(h_a - h_b) - Q_{l,1} = \Delta E_1 \quad (2a)$$

$$\Delta E_1 = E_{f,1} - E_{i,1} \quad (3)$$

$E_{i,1}$  and  $E_{f,1}$  denote the initial and the final energy of the TES for the charging period and  $Q_{l,1}$  denotes the heat loss during the period.  $h_a$  and  $h_b$  denote the enthalpy of state  $a$  and  $b$ .

### C. Discharging period

Discharging time is the time taken for the storage bed to attain a volume average temperature of  $T_{\text{ind}}$ . An energy balance for the discharging period can be written as [8]

$$-(\text{Energy input} + \text{Energy loss}) = \text{Energy accumulation} \quad (4)$$

$$-[\dot{m}(h_d - h_c) + Q_{l,3}] = \Delta E_3 \quad (4a)$$

$$\Delta E_3 = E_{f,3} - E_{i,3} \quad (5)$$

$E_{i,3}(=E_{f,2})$  and  $E_{f,3}$  denote the initial and final energies of the storage for the discharging period. The quantity in square brackets represents the energy output during discharging.

### D. Energy efficiency

Energy efficiency is the ratio of energy recovered from the thermal energy storage during discharging to the total energy input during charging. The energy efficiency ( $\eta$ ) can be defined as [8]

$$\eta = \frac{\text{Energy recovered from TES during discharging}}{\text{Energy input to TES during charging}} \quad (6)$$

$$\eta = \frac{h_d - h_c}{h_a - h_b} \quad (6a)$$

## III. EXPERIMENTAL PROCESS

### A. Experimental instruments

The concrete storage prototype was composed of pipes embedded in a concrete storage block. The embedded pipes are used for transporting and distributing the heat transfer medium while sustaining the pressure. The concrete storage stores the thermal energy as sensible heat. A special interface material was installed to reduce the friction between the concrete and the pipes due to the mismatch of thermal expansion.

The heat exchanger was composed of 16 pipes of high-temperature steel with the inner diameter of 12 mm and wall thickness of 7 mm. They were distributed in a square arrangement of 4 by 4 pipes with a separation of 82 mm. The storage prototype had the dimensions of 0.5 x 0.5 x 4 m.

In order to record data for energy balances, the piping system was equipped with numerous of sensors. The mass flow as well as the water/steam temperature and pressure were measured. The prototype was equipped with 56 thermocouples distributed within the storage material, on the embedded pipes and the header pipe. After the installation of additional reinforcement and measuring equipment, a formwork was installed and the storage space was filled with the thermal storage concrete. The storage prototype was then covered by insulation on all sides and top and bottom.



Fig 2. Schematic of thermal storage



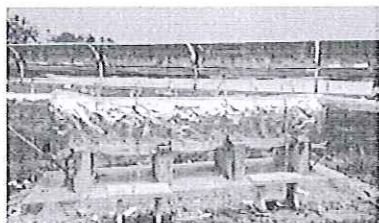


Fig 3. Thermal storage prototype

TABLE I: THERMOPHYSICAL PROPERTIES OF STORAGE PROTOTYPE FROM THE LABORATORY

Parameter	Values	Unit
Density of the concrete	1820	kg/m <sup>3</sup>
Specific heat capacity of the concrete	1538	J/kg-K
Thermal conductivity of the concrete	1.0352	W/m-K
Coefficient thermal expansion of the concrete	8.21	10 <sup>-6</sup> /K
Coefficient thermal expansion of the tubes	12.3	10 <sup>-6</sup> /K
Density of the HTF	5.14	kg/m <sup>3</sup>
Specific heat capacity of the HTF	2556	J/kg-K
Thermal conductivity of the HTF	36.61x10 <sup>-3</sup>	W/m-K

#### B. Experimental procedure

The experimental investigation was carried out at the Energy Park of the School of Renewable Energy Technology (SERT) at Naresuan University, Phitsanulok, Thailand.

As the startup procedure, prior to the experimental processes proper, most of the water contained in the concrete was expelled by heating the concrete storage prototype from ambient temperature to 180°C. During the process the water evaporates and there is a buildup of vapor pressure within the concrete. During this process there was a buildup of vapor pressure within the concrete which needed to be carefully monitored to avoid damage to the concrete.

The subsequent operating conditions of the concrete storage prototype were:

- Heat transfer fluid (water/steam)
- Maximum internal pressure (10 bar)
- Maximum temperature: up to 180 °C
- Test temperature range between 110-180 C
- Mass flow rate: 0.009, 0.012 and 0.014 kg/s

## IV. RESULT AND DISCUSSION

#### A. Charging time

Having completed the startup procedure, the first charging experiment was commenced. As shown in Fig 4, the HTF inlet temperature and the storage prototype temperature were plotted for comparison of the HTF temperature and the average temperature of the storage prototype during this first charging experiment.

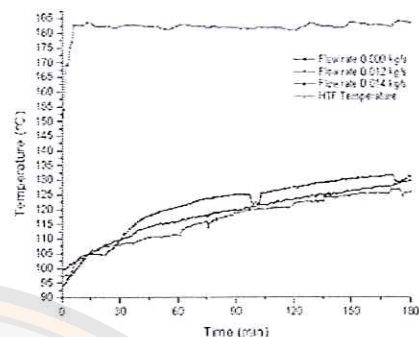


Fig 4. Temperature of HTF inlet and storage prototype in charging experiment at 0.009, 0.012 and 0.014 kg/s.

During the tests, HTF inlet mass flow rates were 0.009, 0.012 and 0.014 kg/s. The HTF inlet temperature to the storage prototype was manually increased very quickly to about 180 °C and then maintained at an almost constant level at about 180 °C for most of the charging process. Also, as shown in Fig.4, the storage prototype temperature slightly increased. This is shown at the different flow rates in Fig.4 Initially the volume average temperature of the storage bed rose rapidly and then rose slowly over the subsequent time up to 180 minutes. This is due to the initial potential for heat conduction in the concrete. The heat conduction potential decreases with time as the storage bed gains heat of the HTF.

#### B. Energy input

The thermal energy rates at the storage bed of the storage prototype were shown in Fig 5.

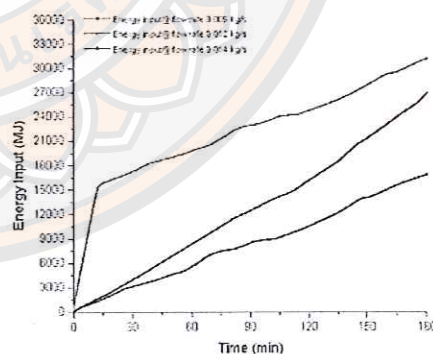


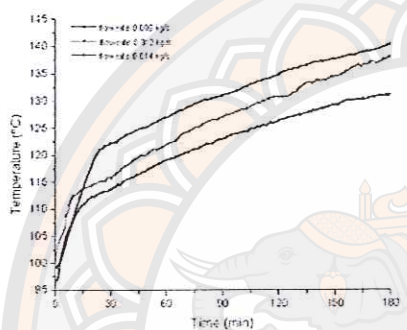
Fig 5. Energy input during charging process at 0.009, 0.012 and 0.014 kg/s.

The amount of thermal energy input in the storage materials at their respective charging times was calculated using Equation (2). The flow rate of 0.012 kg/s resulted in the fastest heat transfer from the pipe to the concrete, followed by the flow rate of 0.009 kg/s with 0.014 kg/s the lowest heat transfer value. The energy input at the various flow rates were 0.012 kg/s, 31,119 MJ, 0.009 kg/s, 26,891 MJ and 0.014 kg/s, 16,840 MJ.

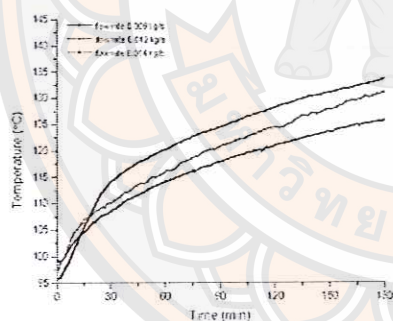
### C. Radiant Thermal distribution on charging time

The radiant thermal distribution for thermal energy storage over charging time is shown in Fig's. 6(a), 6(b) and 6(c). Increasing the HTF flow rate increases the overall heat transfer coefficient enabling faster exchange of heat which reduces the charging time. At higher HTF flow rates the time required to achieve a certain temperature decreased. At the HTF flow rate of 0.014 kg/s the temperature increase over time was greatest, followed by the flow rate of 0.012 kg/s with the flow rate of 0.009 kg/s being the slowest.

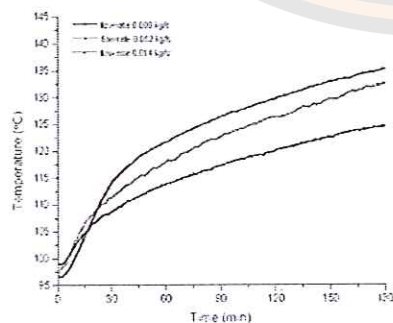
Figs 6(a-c) show the comparisons of thermal distribution of temperature by thermal radiation for the three flow rates through the 1 cm, 2 cm and 3 cm from HTF pipe respectively.



(a)



(b)



(c)

Fig 6. Radial thermal distribution and flow rate on charging time of storage bed.

### D. Discharging time

During the discharge process, various thermocouples were fixed inside the storage material in order to measure the temperature distribution. The average temperature of the storage material inside the thermal storage prototype decreased over time (Fig 7).

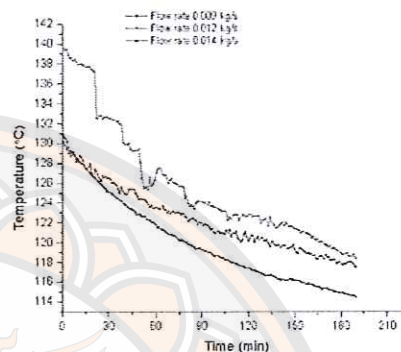


Fig 7. Temperature of concrete storage prototype in discharging period at 0.009, 0.012 and 0.014 kg/s.

### E. Energy recovered

The thermal energy recovery rates of the storage bed are shown in Fig 8. The amount of thermal energy recovered was calculated using Equation (4). The calculations showed that at the flow rate of 0.014 kg/s the heat transfer from the concrete into the pipe was faster than at the flow rates of 0.012 and 0.009 kg/s. The energy recovered at each flow rate was, at 0.014 kg/s, 18,796 MJ, at 0.012 kg/s, 14,363 MJ and at 0.009 kg/s, 12,173 MJ.

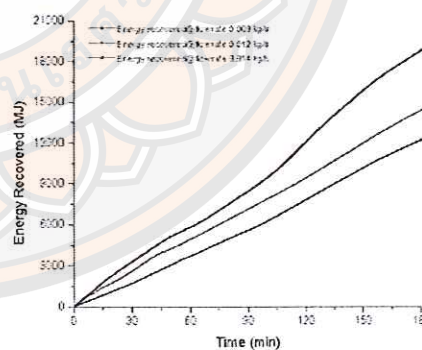


Fig 8. Energy recovered during discharging period at 0.009, 0.012 and 0.014 kg/s

### F. Radial Thermal distribution on discharging time

Heat discharge of the charged storage bed was initiated by passing HTF at a lower temperature ( $T_{inlet}$ ); The HTF receives the heat from the charged storage bed which decreases the storage bed temperature and also causes a rise in the HTF temperature along the bed. The radiant thermal distribution for thermal energy storage on discharging time is shown in Figs. 9(a), 9(b) and 9(c), which show comparison of the thermal distribution of temperature of thermal radiation for the 1 cm, 2 cm and 3 cm HTF pipes. It can be seen that the