THERMAL ANALYSIS
OF POWER CABLES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OYTUN GUVEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2007

Approval of the thesis:

THERMAL ANALYSIS
OF POWER CABLES

submitted by **OYTUN GUVEN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by

Prof. Dr. Canan Özgen ________
Dean, Graduate School of **Natural and Applied Science**

Prof. Dr. İsmet Erkmen ________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Arif Ertaş ________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Nevzat Özay ________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Arif Ertaş ________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Ahmet Rumeli ________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr.Osman Sevaioglu ________________
Electrical and Electronics Engineering Dept., METU

Cem Ozgur Gercek(M.s) ________________
Tubitak Uzay,

Date: 25/12/2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : OYTUN GUVEN

Signature :

ABSTRACT

TERMAL ANALYSIS OF POWER CABLES

GUVEN, Oytun.
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Arif ERTAŞ
December 2007, 43 pages

This thesis investigates temperature distribution and hence heat dissipation of buried power cables. Heat dissipation analysis of a simple practical application and the parameters that affect the heat dissipation are discussed.

In analyzing temperature distribution in the surrounding medium , a computer program is developed which is based on gauss-seidel iteration technique. This method is applied to a sample test system and heat dissipation curves for several parameters are obtained. Also, current carrying capacities of various types of cables are determined using dissipated heat values.

Keywords: Temperature distribution, heat transfer.

ÖZ

GÜÇ KABLOLARININ ISI ANALİZİ

GUVEN, Oytun.
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Arif ERTAŞ
Aralık 2007, 43 sayfa

Bu çalışma, toprak altına döşenmiş güç kabloların ısı analizlerini incelemektedir. Isı analizini etkileyen temel faktörler tartışılmaktadır.
Kabloyu çevreleyen ortamın sıcaklık ve ısı dağılım analizinde, gauss-seidel iterasyon metodunu baz alan bir bilgisayar programı geliştirilmekte ve kullanılmaktadır. Bu metod örnek bir test sistemine uygulanarak ısı dağılım eğrileri elde edilmiştir.Bu ısı değerleri kullanılarak çeşitli tipte kabloların akım taşıma kapasiteleri hesaplanmıştır.

Anahtar sözcükler: Sıcaklık dağılımı,ısı transferi.

ACKNOWLEDGEMENTS

I would like to thank Prof. Dr. Arif Ertaş for his valuable guidance and support.

I would like to extend my gratitude also to Can Atılır for his help in developing computer program.

I am grateful to my coordinators Ihsan Alp- Tamer Akkur and my company NEXANS for their tolerance during my graduate education.

I wish to express my special thanks to my parents Mehmet Ali and Mualla Guven and to my sister Pelin Guven for their support and understanding during this study.

TABLE OF CONTENTS

PLAGIARISM iii

ABSTRACT..iv

ÖZ……………… v

ACKNOWLEDGEMENTS vi

TABLE OF CONTENTS..vii

CHAPTER

1. INTRODUCTION 1

2. CALCULATION OF HEAT DISSIPATION ……………………………..5

2.1 Mathematical Modeling …...5

2.1.1 Finite difference equations for interior and boundary points**..............**8

2.2 Heat Dissipation from the grid boundaries**..**10

2.2.1. Determination of suitable number of grid points..............................12

2.2.1.2 Temperature distribution in the surrounding medium....................13

2.2.1.3**.** Three Cables in a Trench…………………………………………15

2.2.1.4. Cable under a concrete plaque……………………………………15

3. FACTORS AFFECTING HEAT DISSIPATION AND CURRENT CARRYING CAPACITIES OF VARIOUS CABLES…………………17

3.1. Trench Width...17

3.2. Trench Depth...18

3.3. Cable Location...18

3.4. Effect of thermal conductivities...20

3.6. Effect of ambient temperature...20

3.7. Effect of wind velocity..22

3.8. Conclusion...23

3.9. Determining Current Carrying Capacities of Various Cables.............23

4.CONCLUSION..26

5.REFERENCES...28

6.APPENDIX 30

LIST OF FIGURES

1 Schematic diagram of a buried cable 5

2 Node representation of the geometry 6

3 Schematic Diagram for a buried Cable.. ...10

4 Heat Dissipation versus Drichlet distance from the cable.....................11

5 Heat Dissipation versus number of grid points....................................13

6 Temperature distribution in the surrounding medium…….……..…...14

7 Heat Dissipation versus trench width... 18

8 Heat Dissipation versus trench depth… ……………………………...19

9 Heat Dissipation versus cable Location ………………………………19

10 Heat Dissipation versus K_T…..………..……………………………...20

11 Heat Dissipation versus wind velocity……....………………………..22

12 Current Carrying Capacities of Cables………………………………..25

CHAPTER 1

INTRODUCTION

Power cables are widely used in power transmission and distribution. Cable voltages up to 500 kV are in operation and researches are continuing for voltage levels of 750 and 1000 kV. Further developments occurred in the insulating materials, as a result voltage withstand level is increased.

Voltage level and the temperature of a cable determine its maximum permissible power transmission. Heating of cable insulation is caused by the heat generation in the conductor. Heat generation in the conductor is proportional to its resistance and the current through it; i.e, I^2R. On the other hand, heat transfer from the cable depends on the thermal properties of its insulation and the other surrounding medium. Thermal instability occurs and causes the cable to burn, if the generated heat is not transferred into the surrounding medium of the cable. Therefore, the maximum permissible temperature of cable insulation limits the current carrying capacity of the cable. The accurate prediction of thermal conditions in the underground cable system is essential for economical cable system design. Because of high capital costs of underground transmission systems, small differences in predicting cable ratings can represent considerable saving in investment. Over the last decade, many works have been devoted to developing an accurate method for calculating the thermal distribution. [1-4]

Pioneer work was done by Neher [4]. An analytical solution was found and known as Kennelly's equation [4].

$$\Delta T_p = \frac{W}{2\pi K}^{\ln}\left(\frac{d'}{d}\right)$$

where

W: is the input rate per unit length of the cable

K: is the thermal conductivity of the medium.

$\Delta T_p = T_p - T_{amb}$, T_p : temperature at point p, T_{amb} : Ambient temperature

d : distance from the cable center to point p

d' : distance from the center of the cable image to point p

Recently more reliable works based on numerical techniques have dealt with the calculation of the thermal fields of underground cables . These works based on Kennelly's equation. Due to the assumptions used in Kenelly's equation, these methods are restricted to ideal cable / trench conditions, where soil is homogeneous with single valued thermal conductivity and the ground surface is isothermal plane. For instance, Gela [3] used boundary element method for the solution of heat dissipation. Boundary element method has the advantage of considering the region boundaries rather that it's interior which reduces the problem to two dimensional. However in this work, there are some unjustified constraints such as neglecting the convective heat transfer into the atmosphere.

Purpose of this work is to develop a more accurate method for calculating the thermal fields. Problem is complex because many parameters such as installation geometry (dimensions of trench, mother soil and the cable location)

[5-6], thermal and physical properties of the types of the soils involved [7], seasonal variation and effect of ambient temperature.

In this work, finite difference method is used which divides the geometry into nodes and sets the heat balance equation at each node. The heat balance equation equates the heat sum coming from the four surrounding nodes to the heat generated or absorbed at this node. After setting the heat balance equations for all different types of nodes, temperature values of all nodes are determined. By introducing a new constraint, which is based on energy conservation law, heat dissipation from the cable is calculated. Energy conservation law equates the dissipated heat from the cable to the radiated heat from the region boundaries. Model proposed in this work can handle the underground system considering the seasonal variation, soil properties and installation geometry in comparison with the other studies.

While choosing appropriate cable for a transmission system, there are three important parameters to be considered.

1) Maximum short circuit withstand current: A cable is desired to carry short circuit current until the relay cuts the power in medium voltage transmission system. Mainly, cross sectional areas of medium voltage cables are determined considering the relay response time and short circuit withstand current, i.e, 1x95 mm^2 XLPE insulated medium voltage cable can withstand a short circuit current of 13.8 kA for one second. For 0.6 second this value is found to be 18.4 kA.

2) Voltage drop calculations: In low voltage applications, voltage drop calculations are also held considering the resistance and reactance values of a cable. Appropriate cable for application is determined considering the voltage drop.

3) Rated current carrying capacities at full load: At steady state, maximum current carrying capacities of cables are taken into account in continuous loading conditions. Cross sectional areas of power cables are determined considering these parameters. In this thesis, heat dissipation analysis is studied considering the steady state maximum current carrying capacities of cables.

In this work, heat dissipation for underground cable systems will be determined, initially. Following this, factors affecting the heat dissipation are investigated. Also, values found in this work are compared to proposed catalogue values of the producers.

CHAPTER 2

CALCULATION OF HEAT DISSIPATION

In this chapter, finite difference method is used to determine the heat distribution in the surrounding medium (soil). A numerical technique is proposed to calculate the dissipated heat from the cable. Suitable values of parameters in proposed numerical technique are also found.

2.1. Mathematical Modeling

Finite difference method is used to determine the heat dissipation from the buried cable. Figure 1 shows an ordinary geometry of a buried cable.

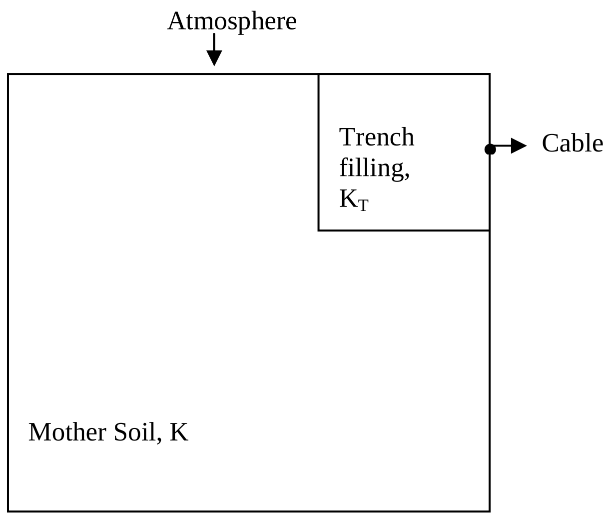

Figure 1: Schematic diagram of a buried cable

The area surrounding the cable having different thermal property than the mother soil is called trench filling. Trench filling thermal conductivity is K_T (W/C°.m) and the mother soil thermal conductivity is K (W/C°.m)

Finite difference method suggests to divide the geometry shown in figure1 into nodes and a grid is formed.

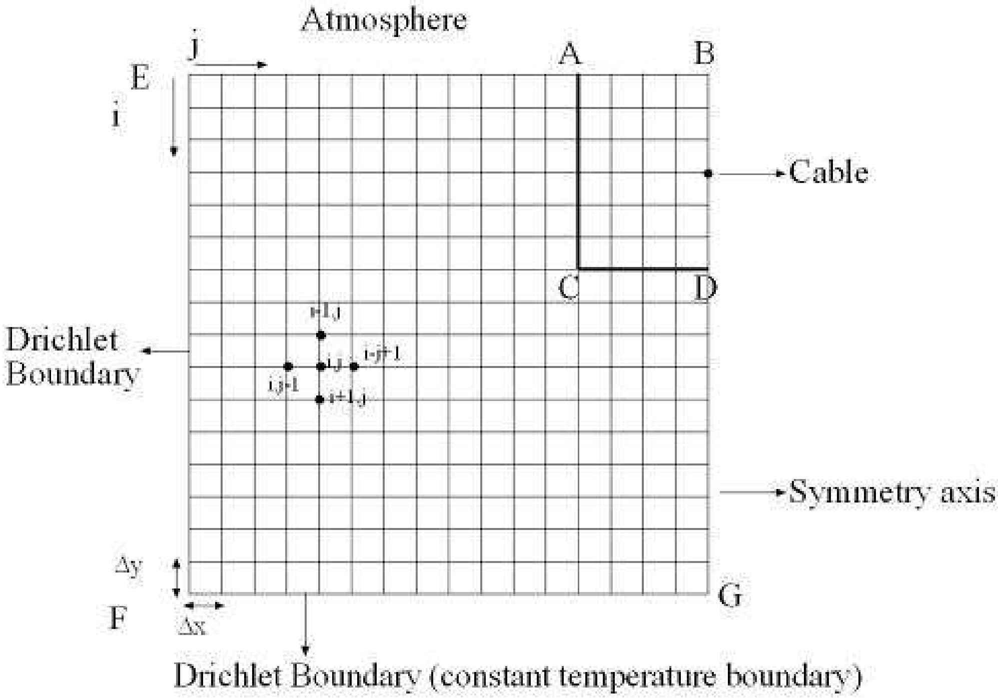

Figure 2: Node representation of the geometry

At each nodal point of the grid surrounding the buried cable, a heat balance equation is used to determine the heat flow.

For any nodal point (i,j) summation of the heat coming from the four surrounding nodes is equal to the heat generated or received at the node itself.[12]

$$Q_{ij-ij+1}+Q_{ij-ij-1}+Q_{ij-i+1j}+Q_{ij-i-1j}=Q_{ij} \quad (1)$$

The heat dissipation between the earth surface and the atmosphere is transferred by convection, while inside the grid heat is transferred by conduction.

$Q_{conduction}$: K A grad(T) (2)

where K is the thermal conductivity of the soil [W/m°C]

A = area [m^2]

$Q_{convection}$: h A ΔT (3)

where h = convective heat transfer coefficient [W/m^2 °C]

Substituting equations 3 and 2 into equation 1 and taking Q_{ij} as 0 because any node in the grid is neither a heat generator nor a heat sink, temperature of a node is found to be in general form.

$$T_{ij} = \frac{\left(aT_{ij+1} + bT_{ij-1} + cT_{i+1j} + dT_{i-1j}\right)}{\left(a+b+c+d\right)} \quad (4)$$

Constants a,b,c,d have different values depending on the location of the node in the grid. Following example explains the method of obtaining these constants for a node next to the cable.

Heat from all four surrounding nodes (i,j+1), (i,j-1), (i+1,j) and (i-1,j) into node i,j can be calculated as follows, respectively

$$Q_{ij+1} \text{ (heat coming from the cable) } = K_T \Delta y \frac{\left(T_{ij+1} - T_{ij}\right)}{\Delta x} \quad (5)$$

$$Q_{ij-1} = K_T \Delta y \frac{\left(T_{ij-1} - T_{ij}\right)}{\Delta x} \quad (6)$$

$$Q_{i+1j} = K_T \Delta x \frac{\left(T_{i+1j} - T_{ij}\right)}{\Delta y} \quad (7)$$

$$Q_{i-1j} = K_T \Delta x \frac{(T_{i-1j} - T_{ij})}{\Delta y} \quad (8)$$

By substituting equations 5,6,7,8 into equation 1 and considering that Q_{ij}= 0 since the node is not a heat sink or heat source, temperature equation at node i,j in a mesh with Δx=Δy is found to be

$$T_{ij} = \frac{\left(T_{ij-1} + T_{ij+1} + T_{i-1j} + T_{i+1j}\right)}{4} \quad (9)$$

Similarly, finite difference equations for the temperature at nodes on the interface lines (boundary points) and inside the grid are determined in the coming section.

2.1.1. Finite difference equations for interior and boundary points

Finite difference equations for determining the temperature inside both the mother soil and the trench soil is given by;

$$T_{ij} = (T_{ij+1} + T_{ij-1} + T_{i+1j} + T_{i-1j}) / 4 \quad (10)$$

At vertical interface between the trench soil and mother soil:

$$T_{ij} = \frac{\left(K + K_T\right)}{2} \; [KT_{ij-1} + K_T T_{ij+1} + \frac{\left(K + K_T\right)}{2} (T_{i+1j} + T_{i-1j})] \quad (11)$$

At horizontal interface between the trench soil and mother soil;

$$T_{ij} = \frac{\left(K + K_T\right)}{2} \; [\frac{\left(K_T + K\right)}{2} \; (T_{ij+1} + T_{ij-1}) + K\, T_{i+1j} + K_T\, T_{i-1j}] \quad (12)$$

For the point at the corner of vertical and horizontal interface:

$$T_{ij} = \frac{1}{\left(K_T + 3K\right)} \left[\frac{K_T + K}{2} . \left(T_{ij+1} + T_{i-1j}\right) + K\left(T_{i+1j} + T_{ij-1}\right) \right] \quad (13)$$

On the symmetry axis

$$T_{ij} = \frac{\left(2T_{ij-1} + T_{i+1j} + T_{i-1j}\right)}{4} \tag{14}$$

For points on the interface between the mother soil and the atmosphere:

While calculating the Q flow from the atmosphere, heat received from the top soil by radiation from the sun has to be considered. This heat is represented as q

From left neighbour, heat flow is

$$Q = \frac{K\left(T_{ij-1} - T_{ij}\right)}{2} + \frac{h\Delta x\Delta y\left(T_A - T_{ij}\right)}{2}$$

from right neighbour, heat flow is

$$Q = \frac{K\left(T_{ij+1} - T_{ij}\right)}{2} + \frac{h\Delta x\Delta y\left(T_A - T_{ij}\right)}{2}$$

from upper neighbour, heat flow is

$Q=h\Delta x\Delta y(T_A-T_{ij}) + q$

from lower neighbour, heat flow is

$Q=K\ (T_{i+1j} - T_{ij})$

Finally,the temperature at the node i,j is found to be

$$T_{ij} = \frac{\frac{K}{2}\left(T_{i,j-1} + T_{ij+1}\right) + 2h\Delta x\Delta y T_A + T_{i+1j}K + q}{\frac{(3k + 2h\Delta x\Delta y)}{2}} \tag{15}$$

For the points on the interface between the trench soil and the atmosphere:

Eqn. 15 holds also for this boundary by replacing K with K_T

Using these finite difference equations, temperature values of all nodes and heat dissipation from grid boundaries are determined.

2.2. Heat Dissipation from the grid boundaries

In the finite difference method, the semi infinite region around the cable is truncated by placing an artificial boundary sufficiently for away from the cable. This boundary is called Drichlet Boundary, also the constant temperature boundary.

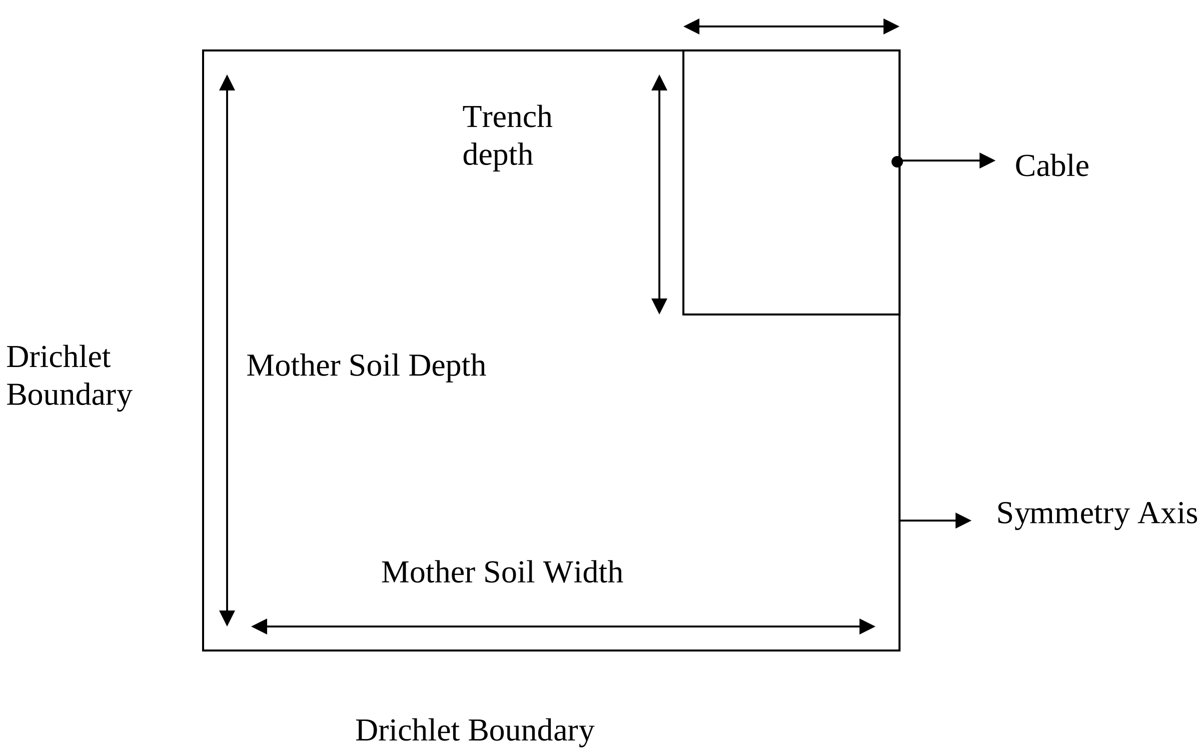

Figure 3: Schematic Diagram for a buried Cable

Distance from the cable for placing the constant temperature boundary is determined with the help of a developed computer program. If the distance from the cable where the region is truncated is small, it introduces an error in the calculation of heat dissipation. For different types of cable configuration (varying K_T, K, geometry and atmosphere temperature), most appropriate value of this distance is obtained. Drichlet boundary must be located to a distance where the dissipated heat from the cable saturates. (where the solution converges).

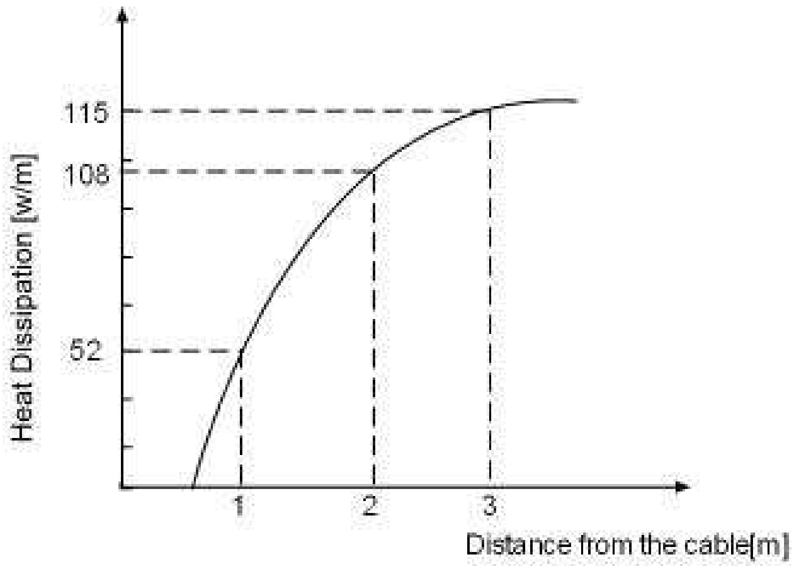

Figure 4 : Heat Dissipation versus Drichlet distance from the cable

Dissipated heat value converges at a distance of 3 meters. Drichlet boundary must be located to a distance equal to or above this value. This value is same for different number of grid points. Dissipated heat value saturates at 3 meter distance from the cable for every number of grid points. Figure 4 is representation for 120x120 square grid.

After locating the drichlet boundary, an additional constraint is introduced based on the energy conservation principle. When the temperatures of all nodes in the grid are determined by numerical method which will be explained in the next section, heat at each boundary can be calculated as a heat loss in a form of convective heat transfer at the atmosphere boundary and conductive heat transfer at the rest of the boundaries. This heat loss is equal to the heat generated by the cable. In the following section, suitable number of grid points to calculate the temperature distribution and hence the heat dissipation is explained.

2.2.1. Determination of suitable number of grid points

Finite Difference equations for the underground cable system are arranged to suit the Gauss – seidel iteration technique. To start with, initial temperatures of all nodes in the grid are assigned. For given values of installation geometry, K_T and K, temperature of the cable and the atmosphere; new temperature values of all nodes are calculated by Gauss Seidel iteration technique. If the difference in temperature in two successive iterations is within the absolute error value for any node, value found at last iteration is assigned as the temperature of the node. Codes for the program for calculating the temperature values of all nodes are given in Appendix . The most important factor in this iteration technique is the number of nodes. For a geometry given with the following reference values;

Mother Soil depth and width	:	6.1	m
Trench depth and width	:	1.22	m and 0.61 m
Trench thermal conductivity	:	2	W / m°C
Mother soil thermal conductivity	:	0.67	W / m°C
Cable depth	:	0.91	m
Cable temperature	:	90	°C in steady state
Atmosphere temperature	:	20	°C

Number of nodes has been varied from 20 x 20 to 200 x 200 (dimensions of $\Delta x = \Delta y$ changes from 0.305 to 0.0305). Variation of the heat dissipation with the mesh points is given in figure 5 for an absolute error of 10^{-4} and initial temperature of 25°C.

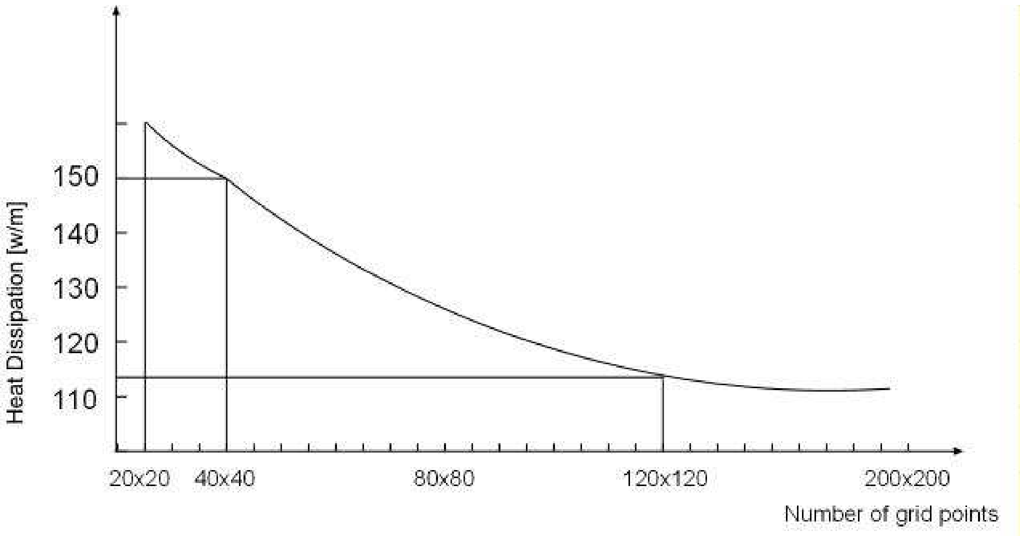

Figure 5: Heat Dissipation versus number of grid points

As seen in figure 5, as the number of nodes increases, heat dissipation converges to a fixed value. So minimum number of nodes for calculating the heat dissipation is the value where the dissipated heat starts to saturate. Minimum 120 x 120 square grid can be chosen. Initial temperature set value and absolute error value only effects the number of iterations. For a practical situation given with the reference values, dissipated heat is found to be

115 W /m. This is consistent with the work done by Hanna and Chikhani [8].

2.2.1.2. Temperature distribution in the surrounding medium

In the previous section, suitable values for two important parameters; drichlet boundary distance and number of grid points are determined. It is observed that for achieving correct results, drichlet boundary must be set to a distance of 3 meters. Number of grid points do not have any effect on placing the drichlet boundary. Dissipated heat value converges at 3 meters every time for different values of grid points. After locating the drichlet boundary, suitable number of grid points for successive results is found as 120x120.Now, geometry is ready for calculating the temperature distribution in the surrounding medium using Gauss-Seidel iteration technique. Using the developed computer programme, temperature distribution is found to be as in figure 6.There is heat transfer from three of the boundaries. One of them is atmosphere boundary, others are drichlet boundaries. Using convective and conductive heat transfer equations for these three boundaries and summing up heat losses gives the dissipated heat from the cable. This value is found to be 115 W/m for our reference values.

Ambient Temperature 20°

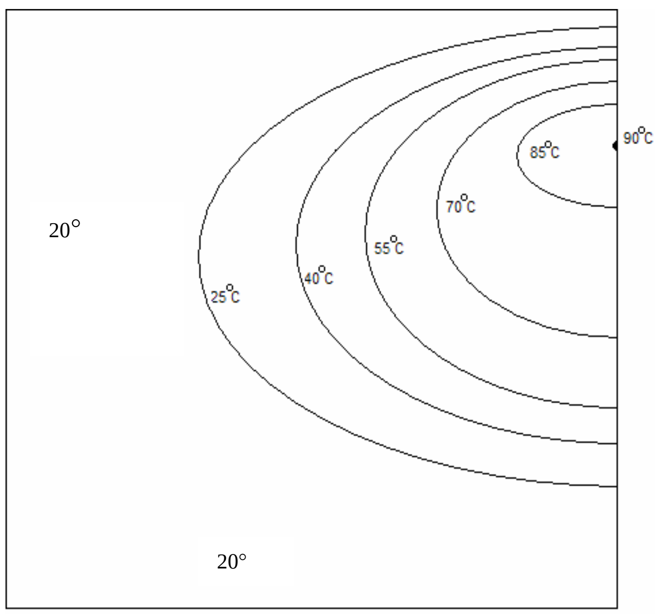

Figure 6: Temperature distribution in the surrounding medium

2.2.1.3. Three Cables in a Trench

In previous application, only one power cable is considered to be installed in the trench. Therefore, in the grid representation of the geometry, there is only one heat generator. However, in practical applications, power is transmitted in three phases and therefore three same type power cables are used. In this situation, three of the nodes are represented as heat generators and their temperature value are set to 90°C. At this time, it is trivial that heat dissipated from the grid boundaries is increased because of the fact that the number of heat generators is increased.

In the previous application, total heat loss from the grid boundaries is found to be 115 W/m. From now on, when three cables are considered, total heat loss from the grid boundaries is found to be 214 W/m which means that for each cable 107 W/m. This is an expected solution because in single cable application, trench filling and mother soil is absorbing only the heat coming from a single cable. This result shows that, in multi-cable applications, current carrying capacities of power cables is decreased by %3.5.

2.2.1.4. Cable under a concrete plaque

One common application is installing power cables under a concrete plaque. In this application, cable is not directly installed in the trench. There is a plaque under the atmosphere boundary. Under this plaque, trench filling is applied. So with this application, an additional thermal conductivity is introduced. This is concrete plaques thermal conductivity which is smaller than mother soil thermal conductivity. Plaques width is taken as 0.61 m and depth is taken as 0.4 m. Thermal conductivity of the plaque is considered to be

0.5 W/C°.m

Due to the fact that thermal conductivity of plaque is much less than the trench filling, new application allows less heat to be dissipated from the cable. For single cable application, dissipated heat value is found to be 103 W/m. Dissipated heat value is decreased by %5.6.

CHAPTER 3

FACTORS AFFECTING HEAT DISSIPATION AND CURRENT CARRYING CAPACITIES OF VARIOUS CABLES

The complexity of determining the heat dissipated from buried cable is due to the involvement of many parameters. These parameters are the thermal properties of the mother and trench soils, environmental conditions and the geometry. In this chapter, effects of these parameters are investigated for single cable application. Several conclusions are achieved in order to increase the heat dissipation. Current carrying capacities of various cables are calculated using these values.

3.1. Trench Width

Effect of trench width is investigated by varying 0.61 m to 6.1 m for the cable system with the following parameters:

K_T	:	2 W / °C.m
Trench depth	:	1.22 m
Cable location	:	0.91 m
Cable temp	:	90 °C

T atmosphere : 20 °C

Mother soil depth and width : 6.1 m

Figure 7 shows the variation of heat dissipation versus trench width. It is seen that the heat dissipation increases with the increase of trench width up to a certain value. Beyond this value, the heat dissipation becomes constant.

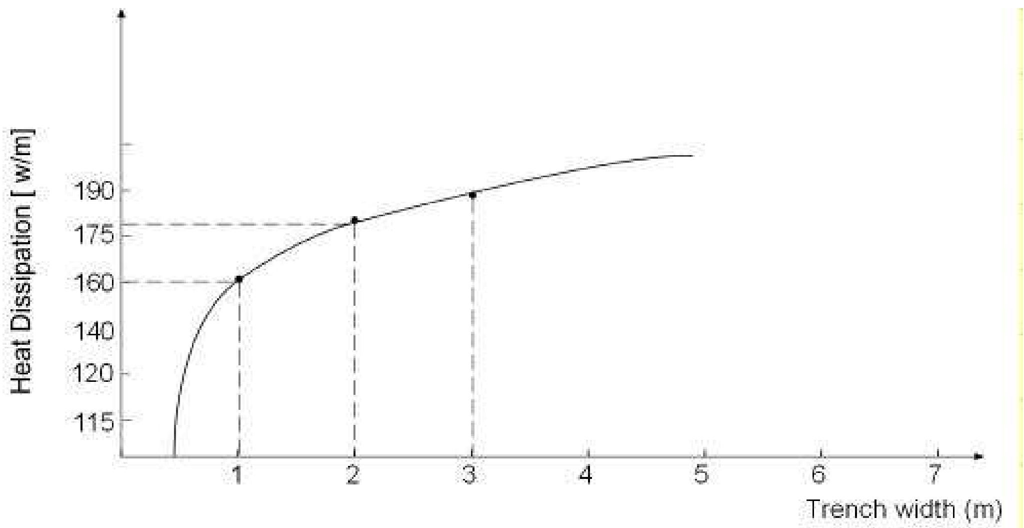

Figure 7: Heat Dissipation versus trench width

3.2. Trench Depth

Trench depth is varied from 1.22 m – to 4.88 m while the other parameters are kept constant. The calculation shows a slight increase in the heat dissipation. The maximum value of the increase in the heat dissipation gained by the depth increase is approximately %6 relative to the original depth of 1.22 m.Figure 8 shows the variation of dissipated heat with the trench depth.

3.3. Cable Location

Figure 9 shows the behavior of the heat dissipation with the cable location. It is obvious that as the cable location becomes closer to the surface, heat dissipation increases. This is because of the fact that heat transfer between the atmosphere and the cable increases as the cable comes closer to the surface.

Cable location and width of the trench have major effect in heat dissipation in comparison with the trench depth.

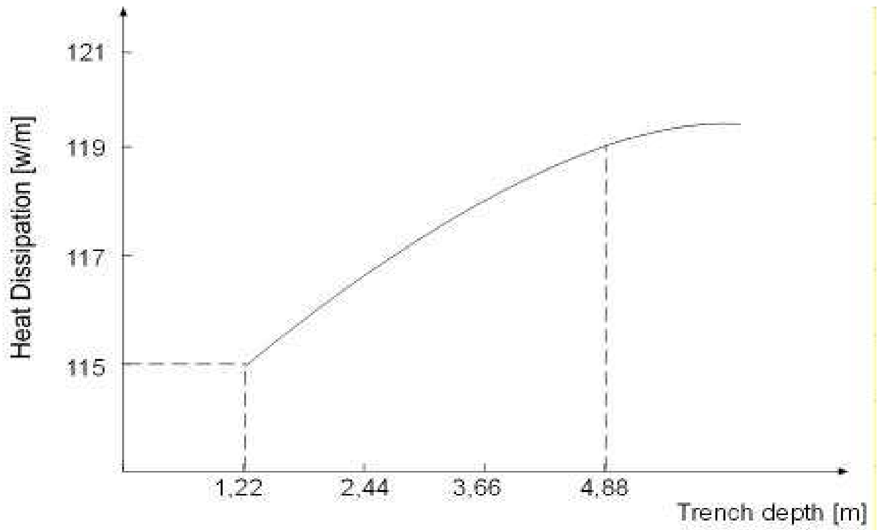

Figure 8: Heat Dissipation versus trench depth

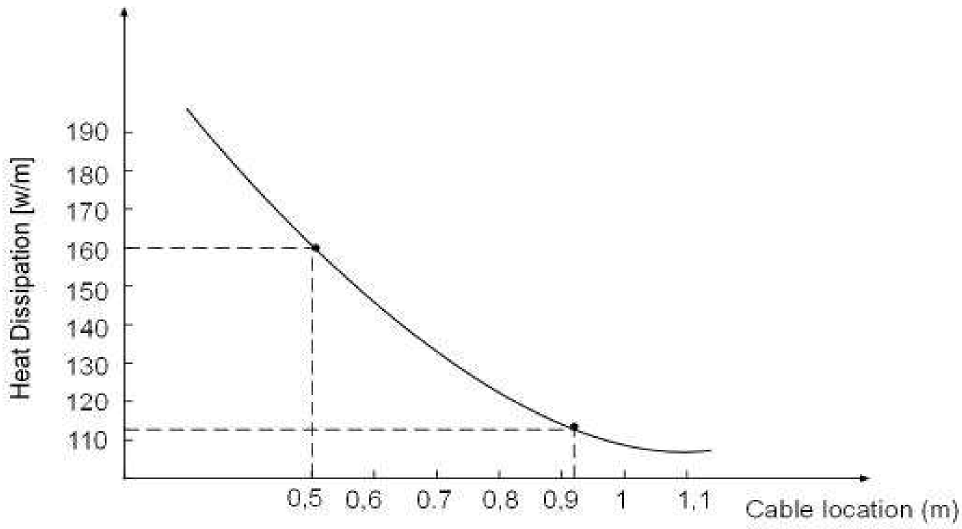

Figure 9: Heat Dissipation versus Cable Location

3.4. Effect of thermal conductivities

In this part, effect of thermal conductivities of trench filling and mother soil is considered. Thermal conductivity of trench filling is varied from 0.67 W/m°C to 3.5W/m°C. Heat dissipation increases with increasing the thermal conductivity of the trench filling. Mother soil thermal conductivity has a slight effect in comparison with the trench filling. Figure 10 shows the variation of heat dissipation with the trench filling thermal conductivity.

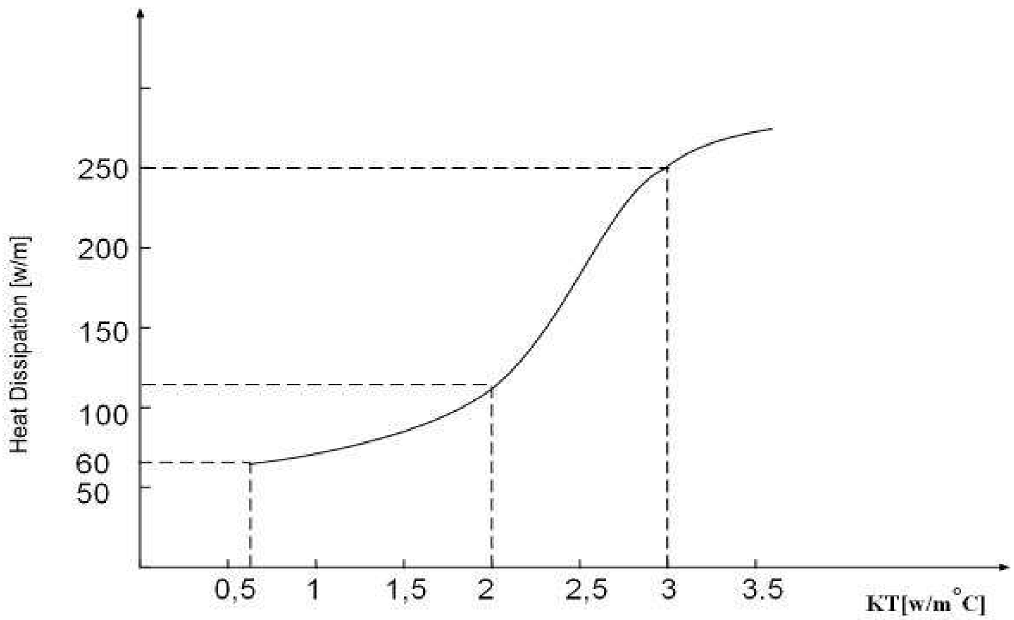

Figure 10: Heat Dissipation versus K_T

3.6. Effect of ambient temperature

Effects of seasonal changes are investigated in this section. Rain fall, amount of heat received from sun and ambient temperature are certain factors affecting the heat dissipation. If these factors are desired to be less effective, cable shall be located to a suitable depth from the surface. This depth is found by the following formula theoretically [9]

$$Z = 1.6\sqrt{p\sigma\pi} \qquad (16)$$

where

Z : the depth of the soil(m)

σ : the thermal diffusivity of the soil (K/dc) (m^2/s)

K : thermal conductivity of the soil (W/C°.m)

d : soil density (kg/m^3)

c : soil specific heat (joule / kg°C)

p : time period (hours /day)

Above equation indicates that the stable depth depends mainly on the thermal diffusivity and considered time. Thermal diffusivity changes with the moisture content.

It is obvious that the calculated heat is responding to the change in weather. The increase in heat dissipation due to the ambient temperature is minimal in summer compared to the increase in winter. Average ambient temperature values are given below.

August : 35 °C , 38 °C

September : 20 °C , 22 °C

December : - 5 °C , 3 °C

May : 29 °C , 32 °C

And dissipated heat values are found as follows:

August : 105 W/m

September: 112.5 W/m

December: 121 W/m

May : 110 W/m

These results are expected because as the ambient temperature falls, atmosphere absorbs more heat from the soil surface.

3.7. Effect of wind velocity

As expected, wind has a cooling effect. Wind velocity effects the heat transfer coefficient and hence the dissipated heat. Heat transfer coefficient can be calculated from the following equation [10].

$$h = 7.371 + 6.43\ v^{0.75} \tag{17}$$

where h is the convective heat transfer coefficient and v is the wind velocity. Figure 10 shows the variation of heat dissipation with the wind velocity. As the wind velocity increases, heat dissipation increases. Nevertheless, increase is %3.5 at most.

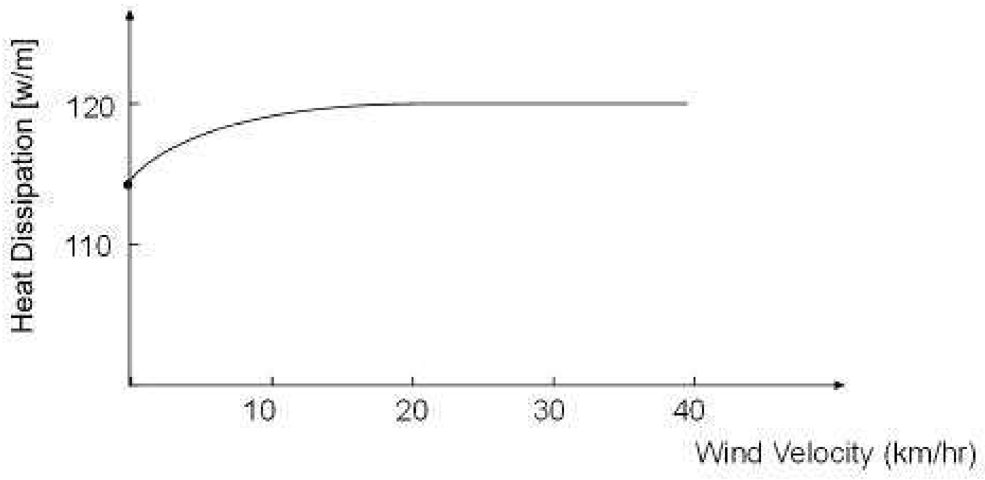

Figure 11: Heat Dissipation versus wind velocity

3.8. Conclusion

Following conclusions which are achieved in chapter three can guide power engineers who work in cable installation.

i) Heat Dissipation increases with the increase of trench width up to a certain value.

ii) Trench depth has a minor effect on heat dissipation

iii) Location of the cable is an important factor in heat dissipation. As the cable comes closer to the surface, heat dissipation increases.

iv) As the trench thermal conductivity increases, heat dissipation increases

v) Effect of seasonal variation can be as high as %15

vi) Wind has a cooling effect on heat dissipation

vii) Current carrying capacities of cables are increased by increasing the heat dissipation.

3.9. Determining Current Carrying Capacities of Various Cables

Cables having voltage withstand levels 34.5 kV 154 kV and 380 kV are used in transmission and distribution. These cables are cross linked polyethylene insulated cables (XLPE). XLPE insulation can heat up to 90°C. Polyvinlychloride and polyethylene (PVC-PE) insulation can heat up to 70°C. Because of this fact these cables are XLPE insulated cables.

Usually, cross sectional areas of these cables are 25,50,70,95,120,150,185 and 240 mm^2. Construction of these cables is copper conductor-inner semiconductive layer-insulation-outer semiconductive layersemiconductive tape-screen and outer sheath . In this section, current carrying capacities of these cables are determined and compared with the catalogue values supplied by the producers.

For a practical situation given with the reference values, dissipated heat is found to be 115 w/m in chapter 2. So, maximum current carrying capacities of those cables are limited to a point where the dissipated heat reaches 115 w/m. This heat is equal to I^2R losses in the conductor. Hence, when resistance values of cables at 90°C are found, current carrying capacities can easily be determined.

$$R = \frac{l}{A}\delta$$

where l is length in meters, A is the cross sectional area in m^2 and δ is self resistance.

$$\delta(T) = \delta o(1+\sigma\Delta T)$$

where δo self resistance value at 0°C (Ω m)

σ self resistance temperature coefficient

ΔT difference in temperature (°C)

For $\delta o = 1.7 \times 10^{-8}$ and $\sigma = 0.0395$, current carrying capacities are calculated and shown in figure 12 together with the catalogue values.[11] There is approximately %3.5 difference with the catalogue values. The reason is that, catalogue values are not obtained under site conditions.

	Calculated(A)	Catalogue(A)
1x 25	192.9	
1x 35	228.11	214
1x 50	272.4	251
1x 70	323.33	306
1x 95	375.6	363
1x 120	420.62	410
1x 150	472.08	449
1x 185	529.6	503
1x 240	597.6	576

Figure 12: Current Carrying Capacities of Cables

At the same time, current carrying capacities in the catalogue values are not obtained considering the seasonal variation. For instance, in august, heat dissipation is found to be 105 w/m. For 1x95 mm^2 cable in August, current carrying capacity is found to be 358.93 Amperes. This value is less than the given catalogue value. To conclude, for a system designer, it would be useful to consider the current carrying capacities of cables are less than the values suggested in catalogues. It is seen that, in summer it may be a risk to trust the catalogue values.

CHAPTER 4

CONCLUSION

The heat generated by the cable in service conditions is absorbed by the surrounding medium. There is a certain max temperature that the cable insulation can heat up to. This temperature and hence the dissipated heat is the basic limitation on the ampacity of the cable. Therefore, it is essential to determine the dissipated heat and find the methods for increasing the heat dissipation from the cable for a designer. Current carrying capacities are determined from the dissipated heat and small predictions may result in considerable saving.

In this thesis, a computer program is developed using Visual Studio.net and executed to find the heat dissipation from the cable. Temperature distribution around the cable is determined, and heat dissipation from the cable into the surrounding medium is calculated. Dissipated heat depends on many parameters such as installation geometry, environmental conditions and thermal properties of the soil. While calculating the heat dissipation, temperature of the cable is taken as 90°C, which is the maximum withstand temperature for XLPE insulated cables.

Parameters affecting the heat dissipation have been investigated. Methods for increasing the heat dissipation and hence the current carrying capacities of these cables have been proposed. Increasing the trench width increases the heat dissipation up to a certain value. Trench depth has a very slight effect in comparison with the trench width. Getting cable closer to the surface increases the convective heat transfer with the atmosphere and hence the heat dissipation. Increasing the thermal conductivity of the soil increases the heat dissipation significantly (i.e in August 105 w/m, in December 121 w/m).

Current carrying capacities of various types of cables are calculated with the proposed method. These values are compared with the catalogue values supplied by the producers. It is observed that it is unsafe to trust the catalogue values in every season because in summer time, current carrying capacities of cables are less than the suggested values in catalogues. Therefore, it is logical to consider the worst situation for a designer.

As a future work, this study can be applied to multicore cable systems. Also, this study may guide the power engineers for the projects of replacing the overhead lines with the underground power cables.

REFERENCES

[1] H.N. Cox and R.Coates ‘’Thermal Analysis of Power Cable in Soils of Temperature-Responsive Thermal Resistivity’’Proc.IEEE,Vol.112,pp.2275-2283,1965.

[2] J.K. Mitchell and O.N Abdel-Hadi, ’’Temperature Distributions Around Buried Cables ‘’, IEEE Transactions,Vol.PAS-98,No.4,pp.1158-1166,1979.

[3] G.Gela and J.J. Dai, ‘’Calculation of Thermal Field of Underground Cables Using the Boundary Element Method’’ IEEE Transactions on Power Delivery,Vol.3,No. 4,pp.1280-1288,1988

[4] J.H. Neher, ‘’The Temperature Rise of Buried Cables and Pipes’’, AIEE Transactions, Vol.68, pp.9-21,1949.

[5] H.N. Cox, W.W. Holdup and D.J.Skipper, ’’Development in UK Cable-Installation Techniques to Take Account of Environmental Thermal Resistivities’’,Proc.IEEE,Vol.122,No.11,pp.1253-1259,1975

[6] T.H.Carr, Mains Practice, Transmission and Distribution of Electrical Energy,George Newnes LTD,Chapter 9 , 1956

[7] W.Woodside and J.N.Messmer, ’’Thermal Conductivity of Porous Media’’,J.Appl.Phys 32 (9),pp.1688-1699,1961

[8] M.A. Hanna and A.Y. Chikhani, ’’Thermal Analysis of Power Cables in Multi-Layered Soil, Part 1 : Theoretical Model’’ submitted to the IEEE PES. Summer Meeting,1992.

[9] M.A. Hanna and A.Y. Chikhani, ’’Thermal Analysis of Power Cables in Multi-Layered Soil, Part 2 : Application’’ IEEE Transactions on Volume 8, Issue 3, Jul 1993 Page(s):772 - 778

[10] V.J. Lunardini, Heat Transfer in Cold Climate, Van Nostrand Reinhold Company,1981.

[11] Nexans Cables Product Catalogue

[12] F.Kreith, Principles of Heat Transfer, 3rd Edition, New York: Harper& Row Publishers, 1973.

APPENDIX

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Windows.Forms;

namespace HeatDissipiationAnalysis
{
	/// <summary>
	/// Summary description for Node.
	/// </summary>
	public class Node
	{
		private DispGrid OwnerGrid;

		public System.Int32 NumberOfModificationIterations = 0;

		public System.Int32 Left = 0;
		public System.Int32 Top = 0;
		public System.Int32 Width = 5;
		public System.Int32 Height = 5;
		public System.Drawing.Color Color = System.Drawing.Color.Black;

		public Node LefternNeighbour = null;
		public Node RighternNeighbour = null;
		public Node UpperNeighbour = null;
		public Node LowerNeighbour = null;

		private System.Drawing.Color ColorOfLine = System.Drawing.Color.Gray;
		private System.Drawing.Color ColorOfTrenchLine =
System.Drawing.Color.Red;
```

```
private System.Object _NonCertainTemperature;

private System.Object _Temperature;

protected decimal LefternTemperature()
{
	if (this.LefternNeighbour == null)
	{
		return this.OwnerGrid.TD;
	}
	else
	{
		return this.LefternNeighbour.Temperature(true);
	}
}

protected decimal LowerTemperature()
{
	if (this.LowerNeighbour == null)
	{
		return this.OwnerGrid.TD;
	}
	else
	{
		return this.LowerNeighbour.Temperature(true);
	}
}

		protected decimal RighternTemperature()
{
	if (this.RighternNeighbour == null)
	{
		return this.LefternNeighbour.Temperature(true);
	}
	else
	{
		return this.RighternNeighbour.Temperature(true);
	}
}

		protected decimal UpperTemperature()
{
	if (this.UpperNeighbour == null)
	{
		return this.OwnerGrid.TA;
	}
	else
	{
		return this.UpperNeighbour.Temperature(true);
	}
```

```
}

protected decimal TemperatureOnDirichlet()
{
	return this.OwnerGrid.TD;
}

protected decimal TemperatureOnSymetryAxis()
{
	return ((this.LefternTemperature()*2) + this.UpperTemperature() +
this.LowerTemperature())/4;
}

protected decimal TemperatureBetweenMotherSoilAndAtmosphere()
{
	return (this.LefternTemperature() + this.RighternTemperature() +
2*this.LowerTemperature() +
2*this.OwnerGrid.TCA*this.OwnerGrid.TA/this.OwnerGrid.TCMS)/(4+2*this.OwnerGrid.TC
A/this.OwnerGrid.TCMS);
}

protected decimal TemperatureInsideTrenchSoilAndMotherSoil()
{
	return (this.LefternTemperature() + this.RighternTemperature() +
this.UpperTemperature() + this.LowerTemperature())/4;
}

protected decimal TemperatureOnVerticalTrenchLine()
{
	return
		(
			Convert.ToDecimal(0.5)
			/
			(this.OwnerGrid.TCMS + this.OwnerGrid.TCT)
		)
		*
		(
			this.OwnerGrid.TCMS *
this.LefternTemperature()
			+
			this.OwnerGrid.TCT * this.RighternTemperature()
			+
			Convert.ToDecimal(0.5) * (this.OwnerGrid.TCMS
+ this.OwnerGrid.TCT) * (this.UpperTemperature() + this.LowerTemperature())
		);
}

protected decimal TemperatureOnHorizontalTrenchLine()
{
	return
```

```
                        Convert.ToDecimal(0.5) /
                        (this.OwnerGrid.TCT + this.OwnerGrid.TCMS)
                        *
                        (Convert.ToDecimal(0.5) *
                            (this.OwnerGrid.TCMS + this.OwnerGrid.TCT) *
                            (
                                this.LefternTemperature()
                                +
                                this.RighternTemperature()
                            )
                            +
                            this.OwnerGrid.TCMS * this.LowerTemperature()
                            +
                            this.OwnerGrid.TCT * this.UpperTemperature()
                        ) ;
                }

                protected decimal
TemperatureOnIntersectionOfHorizontalAndVerticalTrenchLines()
                {
                    return TemperatureOnVerticalTrenchLine();
                }

                protected decimal TemperatureBetweenTrenchAndAthmosphere()
                {
                    return (this.LefternTemperature() + this.RighternTemperature() +
2*this.LowerTemperature() +
2*this.OwnerGrid.TCA*this.OwnerGrid.TA/this.OwnerGrid.TCT)/(4+2*this.OwnerGrid.TCA/
this.OwnerGrid.TCT);
                }

                private void OncekiSimdikiGoster(System.Decimal Onceki, System.Decimal
Simdiki, System.Decimal FarkOrani)
                {
                    System.Windows.Forms.MessageBox.Show("Önceki: " +
Onceki.ToString() + ", " + "Şimdiki: " + Simdiki.ToString() + ", " + "Fark Oranı: " +
FarkOrani.ToString());
                }

                public System.Decimal Temperature(System.Boolean DonotReCalculate)
                {
                    System.Boolean TemperatureWasUndefined = (this._Temperature
== null);
                    if (this._Temperature == null)
                    {
                        if (!DonotReCalculate)
                        {
                            this.NumberOfModificationIterations ++;
                        }
                        if (this.IsOnDirichlet)
```

```
{
    this._Temperature =
this.TemperatureOnDirichlet();
}
else if (this.IsOnSymetryAxis)
{
    if(!DonotReCalculate)
    {
        System.Decimal
CurrentNonCertainTemperature = this.TemperatureOnSymetryAxis();
        if (this._NonCertainTemperature != null
&&
System.Decimal.Compare(System.Decimal.Divide(System.Decimal.Subtract(CurrentNonCerta
inTemperature, Convert.ToDecimal(this._NonCertainTemperature)),
Convert.ToDecimal(this._NonCertainTemperature)), this.OwnerGrid.AE) <= 0)
        {
            this._Temperature =
CurrentNonCertainTemperature;
//
    OncekiSimdikiGoster(Convert.ToDecimal(this._NonCertainTemperature),
CurrentNonCertainTemperature, (CurrentNonCertainTemperature-
Convert.ToDecimal(this._NonCertainTemperature))/Convert.ToDecimal(this._NonCertainTem
perature));
        }
        this._NonCertainTemperature =
CurrentNonCertainTemperature;
    }
    else if (this._NonCertainTemperature == null &&
!(this.Y == this.OwnerGrid.CY && this.X == this.OwnerGrid.CX-1))
    {
        return this.OwnerGrid.ITN;
    }
}
else if (this.IsBetweenMotherSoilAndAtmosphere)
{
    if(!DonotReCalculate)
    {
        System.Decimal
CurrentNonCertainTemperature = this.TemperatureBetweenMotherSoilAndAtmosphere();
        if (this._NonCertainTemperature != null
&&
System.Decimal.Compare(System.Decimal.Divide(System.Decimal.Subtract(CurrentNonCerta
inTemperature, Convert.ToDecimal(this._NonCertainTemperature)),
Convert.ToDecimal(this._NonCertainTemperature)), this.OwnerGrid.AE) <= 0)
        {
            this._Temperature =
CurrentNonCertainTemperature;
//
    OncekiSimdikiGoster(Convert.ToDecimal(this._NonCertainTemperature),
CurrentNonCertainTemperature, (CurrentNonCertainTemperature-
```

```
Convert.ToDecimal(this._NonCertainTemperature))/Convert.ToDecimal(this._NonCertainTem
perature));
                    }
                    this._NonCertainTemperature =
CurrentNonCertainTemperature;
                }
                else if (this._NonCertainTemperature == null &&
!(this.Y == this.OwnerGrid.CY && this.X == this.OwnerGrid.CX-1))
                {
                    return this.OwnerGrid.ITN;
                }
            }
            else if (this.IsCableNode)
            {
                this._Temperature = this.OwnerGrid.TC;
            }
            else if (this.IsInTrench || this.IsInMotherSoil)
            {
                if(!DonotReCalculate)
                {
                    System.Decimal
CurrentNonCertainTemperature = this.TemperatureInsideTrenchSoilAndMotherSoil();
                    if (this._NonCertainTemperature != null
&&
System.Decimal.Compare(System.Decimal.Divide(System.Decimal.Subtract(CurrentNonCerta
inTemperature, Convert.ToDecimal(this._NonCertainTemperature)),
Convert.ToDecimal(this._NonCertainTemperature)), this.OwnerGrid.AE) <= 0)
                    {
                        this._Temperature =
CurrentNonCertainTemperature;
//
    OncekiSimdikiGoster(Convert.ToDecimal(this._NonCertainTemperature),
CurrentNonCertainTemperature, (CurrentNonCertainTemperature-
Convert.ToDecimal(this._NonCertainTemperature))/Convert.ToDecimal(this._NonCertainTem
perature));
                    }
                    this._NonCertainTemperature =
CurrentNonCertainTemperature;
                }
                else if (this._NonCertainTemperature == null &&
!(this.Y == this.OwnerGrid.CY && this.X == this.OwnerGrid.CX-1))
                {
                    return this.OwnerGrid.ITN;
                }
            }
            else if (this.IsOnVerticalTrenchLine)
            {
                if(!DonotReCalculate)
                {
```

```
						System.Decimal
CurrentNonCertainTemperature = this.TemperatureOnVerticalTrenchLine();
						if (this._NonCertainTemperature != null
&&
System.Decimal.Compare(System.Decimal.Divide(System.Decimal.Subtract(CurrentNonCerta
inTemperature, Convert.ToDecimal(this._NonCertainTemperature)),
Convert.ToDecimal(this._NonCertainTemperature)), this.OwnerGrid.AE) <= 0)
						{
							this._Temperature =
CurrentNonCertainTemperature;
//
	OncekiSimdikiGoster(Convert.ToDecimal(this._NonCertainTemperature),
CurrentNonCertainTemperature, (CurrentNonCertainTemperature-
Convert.ToDecimal(this._NonCertainTemperature))/Convert.ToDecimal(this._NonCertainTem
perature));
						}
						this._NonCertainTemperature =
CurrentNonCertainTemperature;
					}
					else if (this._NonCertainTemperature == null &&
!(this.Y == this.OwnerGrid.CY && this.X == this.OwnerGrid.CX-1))
					{
						return this.OwnerGrid.ITN;
					}
				}
				else if (this.IsOnHorizontalTrenchLine)
				{
					if(!DonotReCalculate)
					{
						System.Decimal
CurrentNonCertainTemperature = this.TemperatureOnHorizontalTrenchLine();
						if (this._NonCertainTemperature != null
&&
System.Decimal.Compare(System.Decimal.Divide(System.Decimal.Subtract(CurrentNonCerta
inTemperature, Convert.ToDecimal(this._NonCertainTemperature)),
Convert.ToDecimal(this._NonCertainTemperature)), this.OwnerGrid.AE) <= 0)
						{
							this._Temperature =
CurrentNonCertainTemperature;
//
	OncekiSimdikiGoster(Convert.ToDecimal(this._NonCertainTemperature),
CurrentNonCertainTemperature, (CurrentNonCertainTemperature-
Convert.ToDecimal(this._NonCertainTemperature))/Convert.ToDecimal(this._NonCertainTem
perature));
						}
						this._NonCertainTemperature =
CurrentNonCertainTemperature;
					}
					else if (this._NonCertainTemperature == null &&
!(this.Y == this.OwnerGrid.CY && this.X == this.OwnerGrid.CX-1))
```

```
{
return this.OwnerGrid.ITN;
}
}
else if
(this.IsOnIntersectionOfHorizontalAndVerticalTrenchLines)
{
if(!DonotReCalculate)
{
System.Decimal
CurrentNonCertainTemperature =
this.TemperatureOnIntersectionOfHorizontalAndVerticalTrenchLines();
if (this._NonCertainTemperature != null
&&
System.Decimal.Compare(System.Decimal.Divide(System.Decimal.Subtract(CurrentNonCerta
inTemperature, Convert.ToDecimal(this._NonCertainTemperature)),
Convert.ToDecimal(this._NonCertainTemperature)), this.OwnerGrid.AE) <= 0)
{
this._Temperature =
CurrentNonCertainTemperature;
//

OncekiSimdikiGoster(Convert.ToDecimal(this._NonCertainTemperature),
CurrentNonCertainTemperature, (CurrentNonCertainTemperature-
Convert.ToDecimal(this._NonCertainTemperature))/Convert.ToDecimal(this._NonCertainTem
perature));
}
this._NonCertainTemperature =
CurrentNonCertainTemperature;
}
else if (this._NonCertainTemperature == null &&
!(this.Y == this.OwnerGrid.CY && this.X == this.OwnerGrid.CX-1))
{
return this.OwnerGrid.ITN;
}
}
else if (this.IsBetweenTrenchAndAthmosphere)
{
if(!DonotReCalculate)
{
System.Decimal
CurrentNonCertainTemperature = this.TemperatureBetweenTrenchAndAthmosphere();
if (this._NonCertainTemperature != null
&&
System.Decimal.Compare(System.Decimal.Divide(System.Decimal.Subtract(CurrentNonCerta
inTemperature, Convert.ToDecimal(this._NonCertainTemperature)),
Convert.ToDecimal(this._NonCertainTemperature)), this.OwnerGrid.AE) <= 0)
{
this._Temperature =
CurrentNonCertainTemperature;
```

```
//
	OncekiSimdikiGoster(Convert.ToDecimal(this._NonCertainTemperature),
CurrentNonCertainTemperature, (CurrentNonCertainTemperature-
Convert.ToDecimal(this._NonCertainTemperature))/Convert.ToDecimal(this._NonCertainTem
perature));
						}
						this._NonCertainTemperature =
CurrentNonCertainTemperature;
					}
					else if (this._NonCertainTemperature == null &&
!(this.Y == this.OwnerGrid.CY && this.X == this.OwnerGrid.CX-1))
					{
						return this.OwnerGrid.ITN;
					}
				}
				else
				{
					this._NonCertainTemperature =
this.OwnerGrid.ITN;
				}
				if (this._NonCertainTemperature == null)
				{
					this._NonCertainTemperature =
this._Temperature;
				}
			}

			if(TemperatureWasUndefined && this._Temperature != null)
			{

	this.OwnerGrid.NumberOfNodeswithUnknownTemperature -= 1;
			}

			return Convert.ToDecimal(this._NonCertainTemperature);
		}

		public Node(DispGrid OwnerGrid, Node LefternNeighbour, Node
RighternNeighbour, Node UpperNeighbour, Node LowerNeighbour)
		{
			try
			{
				this.OwnerGrid = OwnerGrid;

				this.LefternNeighbour = LefternNeighbour;
				if (this.LefternNeighbour != null)
				{
					this.LefternNeighbour.RighternNeighbour = this;
				}

				this.RighternNeighbour = RighternNeighbour;
```

```
				if (this.RighternNeighbour != null)
				{
					this.RighternNeighbour.LefternNeighbour = this;
				}

				this.UpperNeighbour = UpperNeighbour;
				if (this.UpperNeighbour != null)
				{
					this.UpperNeighbour.LowerNeighbour = this;
				}

				this.LowerNeighbour = LowerNeighbour;
				if (this.LowerNeighbour!= null)
				{
					this.LowerNeighbour.UpperNeighbour = this;
				}

				this.RefreshNode();

				if (this.LowerNeighbour != null)
				{
					this.LowerNeighbour.RefreshNode();
				}

				if (this.LefternNeighbour != null)
				{
					this.LefternNeighbour.RefreshNode();
				}
			}
			catch
			{
				throw;
			}
		}

		public System.Boolean IsCableNode
		{
			get
			{
				return (this.X == this.OwnerGrid.CX && this.Y ==
this.OwnerGrid.CY);
			}
		}

		public System.Boolean IsInTrench
		{
			get
			{
				return ((this.X > this.OwnerGrid.TX && this.Y >
this.OwnerGrid.TY) && this.UpperNeighbour != null && this.RighternNeighbour != null);
```

```
			}
		}

		public System.Boolean IsOnVerticalTrenchLine
		{
			get
			{
				return (this.X == this.OwnerGrid.TX && this.Y >
this.OwnerGrid.TY && this.UpperNeighbour != null);
			}
		}

		public System.Boolean IsOnHorizontalTrenchLine
		{
			get
			{
				return (this.Y == this.OwnerGrid.TY && this.X >
this.OwnerGrid.TX && this.RighternNeighbour != null);
			}
		}

		public System.Boolean
IsOnIntersectionOfHorizontalAndVerticalTrenchLines
		{
			get
			{
				return (this.X == this.OwnerGrid.TX && this.Y ==
this.OwnerGrid.TY);
			}
		}

		public System.Boolean IsBetweenTrenchAndAthmosphere
		{
			get
			{
				return ((this.X >= this.OwnerGrid.TX && this.Y >
this.OwnerGrid.TY) && this.UpperNeighbour == null);
			}
		}

		public System.Boolean IsOnDirichlet
		{
			get
			{
				return ((this.LefternNeighbour == null ||
this.LowerNeighbour == null) && this.UpperNeighbour != null && this.RighternNeighbour !=
null);
			}
		}
```

```
		public System.Boolean IsInMotherSoil
		{
			get
			{
				return ((this.X < this.OwnerGrid.TX || this.Y <
this.OwnerGrid.TY) && !this.IsOnDirichlet && this.UpperNeighbour != null &&
this.RighternNeighbour != null);
			}
		}

		public System.Boolean IsBetweenMotherSoilAndAtmosphere
		{
			get
			{
				return (this.X < this.OwnerGrid.TX &&
this.UpperNeighbour == null);
			}
		}

		public System.Boolean IsOnSymetryAxis
		{
			get
			{
				return (!this.IsCableNode &&
!this.IsBetweenTrenchAndAthmosphere && this.RighternNeighbour == null);
			}
		}

		public System.Int32 X
		{
			get
			{
				Node NodeIterator = this;
				System.Int32 iterator1 = 0;
				while (NodeIterator.LefternNeighbour != null)
				{
					iterator1++;
					NodeIterator = NodeIterator.LefternNeighbour;
				}
				return iterator1;
			}
		}

		public System.Int32 Y
		{
			get
			{
				Node NodeIterator = this;
				System.Int32 iterator1 = 0;
				while (NodeIterator.LowerNeighbour != null)
```

```
				{
					iterator1++;
					NodeIterator = NodeIterator.LowerNeighbour;
				}
				return iterator1;
			}
		}

		public void RefreshNode()
		{
			try
			{
				if (this.IsOnDirichlet)
				{
					this.Color = System.Drawing.Color.Orange;
//					this._Temperature = this.OwnerGrid.TD;
				}
				else if (this.IsOnSymetryAxis)
				{
					this.Color = System.Drawing.Color.Fuchsia;
				}
				else if (this.IsBetweenMotherSoilAndAtmosphere)
				{
					this.Color = System.Drawing.Color.Black;
				}
				else if (this.IsInMotherSoil)
				{
					this.Color = System.Drawing.Color.Gray;
				}
				else if (this.IsCableNode)
				{
					this.Color = System.Drawing.Color.Red;
//					this._Temperature = this.OwnerGrid.TC;
				}
				else if (this.IsBetweenTrenchAndAthmosphere)
				{
					this.Color = System.Drawing.Color.LightBlue;
				}
				else if
(this.IsOnIntersectionOfHorizontalAndVerticalTrenchLines)
				{
					this.Color = System.Drawing.Color.Yellow;
				}
				else if (this.IsOnVerticalTrenchLine)
				{
					this.Color = System.Drawing.Color.Green;
				}
				else if (this.IsOnHorizontalTrenchLine)
				{
					this.Color = System.Drawing.Color.Blue;
```

```
					}
					else if (this.IsInTrench)
					{
						this.Color = System.Drawing.Color.Brown;
					}
					this.OwnerGrid.Invalidate();


//					if
(this.IsOnIntersectionOfHorizontalAndVerticalTrenchLines)
//					{
//						this.pRighternLine.BackColor =
ColorOfTrenchLine;
//						this.pLefterntLine.BackColor = ColorOfLine;
//						this.pUpperLine.BackColor = ColorOfTrenchLine;
//						this.pLowerLine.BackColor = ColorOfLine;
//					}
//					else if (this.IsOnHorizontalTrenchLine)
//					{
//						this.pRighternLine.BackColor =
ColorOfTrenchLine;
//						this.pLefterntLine.BackColor =
ColorOfTrenchLine;
//						this.pUpperLine.BackColor = ColorOfLine;
//						this.pLowerLine.BackColor = ColorOfLine;
//					}
//					else if (this.IsOnVerticalTrenchLine)
//					{
//						this.pUpperLine.BackColor = ColorOfTrenchLine;
//						this.pLowerLine.BackColor = ColorOfTrenchLine;
//						this.pRighternLine.BackColor = ColorOfLine;
//						this.pLefterntLine.BackColor = ColorOfLine;
//					}
//					else
//					{
//						this.pRighternLine.BackColor = ColorOfLine;
//						this.pLefterntLine.BackColor = ColorOfLine;
//						this.pUpperLine.BackColor = ColorOfLine;
//						this.pLowerLine.BackColor = ColorOfLine;
//					}

//
	this.OwnerGrid.OwnerForm.toolTip1.SetToolTip(this.pPoint, "Temperature: " +
this.Temperature.ToString());
				}
				catch
				{
					throw;
				}
			}
```