



Optimization of the number of elements for thermal maps utilizing Radtherm/RT
by Allison Elizabeth Bristow

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in
Mechanical Engineering
Montana State University
© Copyright by Allison Elizabeth Bristow (2003)

Abstract:

In this thesis an analytical approach was used to optimize the number of elements required to produce accurate thermal maps utilizing RadTherm/RT. RadTherm/RT is a thermal model that uses all modes of heat transfer to predict road temperatures. The goal is to maximize the length of highway modeled and minimize the computation time. A methodology is implemented using a 4-part C/C++ algorithm to select particular elements to delete. The section of code that specifically determines the elements to delete is referred to as the view factor trim routine.

Initial steps are performed in order to obtain the information needed to determine the elements to delete. First, two files are necessary to create a new project in RadTherm/RT. Montana State University's Geographic Information and Analysis Center produced the ASCII files for elevation, road and vegetation maps using ARC/INFO GIS software. These files are loaded into RadTherm/RT to create a new project (.tdf). The project is executed to produce the view factor file (.vfs). Geometric view factors are used as the basis to eliminate elements. Radiation view factors are used in ThermoAnalytics' software to determine the intensity of radiation of each element to every other element in the model and to the environment. This allows the terrain model to include full effects of solar shadowing, multiple reflections, and reradiation of geometric objects. The geometric view factors of each element are calculated using ThermoAnalytics Ray Trace Algorithm and the results are stored in the view factor file (.vfs).

Two text files are required as input into the view factor trim routine. The first file contains the view factor information. This file is obtained by applying an algorithm written by ThermoAnalytics. The routine converts the view factor file (.vfs), which is in HDF format, to a text file. A second file is obtained by applying a routine that retrieves each element and its attributes, which includes the element number, part number, UTM coordinates, and elevation.

The view factor trim routine uses the element attributes and view factor information text files to determine which elements to delete from the model. The routine was developed to have the capability of designating any part(s) as the elements of interest, which are the elements of primary interest for temperature. Furthermore, the values used for the criteria to determine which elements to delete can be modified. The first criterion is the distance from the elements of interest, where all elements that meet this criterion are kept. The next criterion determines the minimum view factor value to keep an element that directly exchanges radiation with an element of interest. Similarly, a minimum view factor criterion is set for elements that emit to or receive energy from elements that have a direct radiation exchange to the elements of interest. The result from the view factor trim routine is a text file that contains the elements to delete.

The final part of the code uses the text file that contains the elements to delete and creates a new project in RadTherm/RT. The primary difference in the new project is that it contains a new part composed of the elements to delete. This allows a user to view the elements, before deleting them.

OPTIMIZATION OF THE NUMBER OF ELEMENTS FOR THERMAL MAPS
UTILIZING RADTHERM/RT

by

Allison Elizabeth Bristow

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Mechanical Engineering

MONTANA STATE UNIVERSITY – BOZEMAN
Bozeman, Montana

September 2003

© COPYRIGHT

by

Allison Elizabeth Bristow

2003

All Rights Reserved

N378
B773

iii

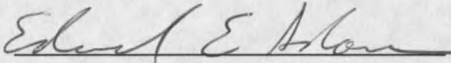
APPROVAL

of a thesis submitted by

Allison Elizabeth Bristow

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

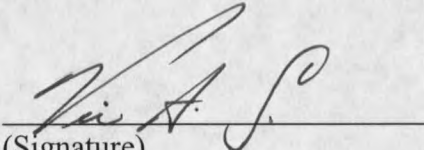
Dr. Edward E. Adams


(Signature)

3-SEPT-2003
Date

Approved for the Department of Mechanical and Industrial
Engineering

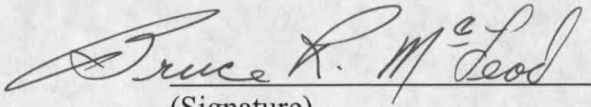
Dr. Vic A. Cundy


(Signature)

9/3/03
Date

Approved for the College of Graduate Studies

Dr. Bruce R. McLeod


(Signature)

9-8-03
Date

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University – Bozeman, I agree that the library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis (paper) in whole or in parts may be granted only by the copyright holder.

Signature 

Date 9/2/03

TABLE OF CONTENTS

1. CHAPTER 1 INTRODUCTION	1
2. CHAPTER 2 BACKGROUND	6
HEAT TRANSFER	6
Conduction	6
Convection	7
Radiation: Processes and Properties	7
Shortwave Radiation	9
Longwave Radiation	9
Radiation Exchange Between Surfaces	10
Surface Absorption, Reflection, and Transmission	10
Surface Emission	12
View Factor	12
RADTHERM/RT HEAT TRANSFER	23
Interior Layer Energy Balance	28
Surface Energy Balance	29
Energy Storage	30
Air Convection	30
Precipitation Convection	30
Ground Layer Conduction	31
Solar Load	31
Expanding Governing Equation	34
Net-radiation Method – Enclosure Theory	35
NUMERICAL METHOD OF SOLUTION	41
Transient Formulation	42
Computation of Temperature Distribution	44
3. CHAPTER 3 METHODOLOGY	46
RAY-TRACING ALGORITHM	46
View Factor Rays	49
View Factor Subdivisions	50
Apparent Area Resolution	50
Hand Calculation for View Factor	51
THE PROCESS TO DELETE ELEMENTS	52
Initial Setup	54
Step 1a - Digital Elevation Map Files (.asc)	54
Step 1b - Terrain File (.asc)	55
Step 2 - Trimming the Original Map	58

TABLE OF CONTENTS - CONTINUED

Step 3 - RadTherm/RT Project Files (.tdf).....	60
Step 4 - Element Attribute File	63
Step 5 - View Factor File (.vfs)	65
Step 6 – Code to Delete Elements.....	68
Step 6a – Separate Elements.....	69
Step 6b – Apply Distance Criteria	70
Step 6c – View Factors to Elements of Interest.....	71
Step 6d – View Factors to Elements with View Factors to Elements of Interest.....	73
Step 6e – Output File	75
Step 7 – Create a new part	76
Creating a RadTherm Project.....	76
 4. CHAPTER 4 FINDINGS/RESULTS	 80
RADIATION EXCHANGE – SIMPLE MODEL	80
Shadowing Effects	81
Shortwave Radiation Exchange	84
180° Flat Model.....	85
135° Model.....	87
90° Model.....	92
Longwave Radiation Exchange	98
MODEL OF BOZEMAN PASS	104
No View Factor or Distance Criteria	111
No View Factor and Distance Criteria Set.....	113
View Factor and Distance Criteria Set.....	115
Evaluate View Factor Criterion	115
Results.....	118
MODEL OF ROCKY CANYON	120
Coordinates for Rocky Canyon Model	120
Assign RadTherm/RT Part ID.	123
Location for Temperature Comparison.....	127
Summer Test Case – July 2 nd , 2003	129
Weather File - Summer.....	129
Temperature Comparison – Mile Marker 315	131
Temperature Comparison – Mile Marker 315½	132
Temperature Comparison – Mile Marker 316	132
Temperature Comparison – Mile Marker 316½	133
Temperature Comparison – Mile Marker 317	133
Temperature Comparison – Mile Marker 317½	135

TABLE OF CONTENTS - CONTINUED

Temperature Comparison – Mile Marker 318	135
Winter Test Case – February 2 nd , 2003.....	136
Weather File - Winter	136
Temperature Comparison – Mile Marker 315	137
Temperature Comparison – Mile Marker 315½	137
Temperature Comparison – Mile Marker 316	140
Temperature Comparison – Mile Marker 316½	140
Temperature Comparison – Mile Marker 317	141
Temperature Comparison – Mile Marker 317½	142
Temperature Comparison – Mile Marker 318	143
5. CHAPTER 5 SUMMARY/CONCLUSIONS.....	144
6. REFERENCES CITED.....	150
7. APPENDICES	152
APPENDIX A: C/C++ CODE – RETRIEVE ELEMENT ATTRIBUTES	153
APPENDIX B: C/C++ CODE – VIEW FACTOR TRIM ROUTINE	160
APPENDIX C: C/C++ CODE – UPDATE RADTHERM PROJECT FILE	179

LIST OF TABLES

Table	Page
1. View factor ray setting in RadTherm/RT.	50
2. View factor ray setting in RadTherm/RT.	52
3. Land cover classification system used by Montana Gap Analysis Project.....	56
4. Summary of the iterations for trimming a map of Bozeman Pass to a size that produces the view factor file (.vfs).	59
5. Solar Azimuth and Zenith angles at two-hour increments on July 22, 1993.....	83
6. (a) View factor values for Figure 48a. Receiver #17 is the sky node. (b) View factor values for Figure 48b. Receiver #26 is the sky node.....	87
7. The absorptivity values for the sloped terrain for each case.....	88
8. View factors for Element 10 labeled on Figure 50b with $\theta = 135^\circ$. (a) Shows the receiving elements that element 10 emits to. Note that receiver number 26 is the sky node. (b) The elements that emit to element 10.	89
9. View factors for Element 10 labeled for a rock wall at 135° , and for a rock wall at 90°	93
10. The percentage of the view factor values for element 40 in each model in Figure 60.	100
11. Results for the model of Bozeman Pass with the distance criterion and view factor criteria set to zero. For this case, 82,075 elements were deleted.....	112
12. Results for the model of Bozeman Pass with the distance criterion set to 250- meters and the view factor criteria set to zero. For this case, 81,448 elements are deleted.	113
13. The emissivity values for the terrain types in RadTherm/RT.....	116

LIST OF TABLES - CONTINUED

Table	Page
14. Results for the model of Bozeman Pass with the distance criterion set to 250-meters and the view factor criteria set to 0.0005. For this case, 288,308 elements are deleted.....	118
15. Results for Rocky Canyon using the view factor trim routine with the distance criteria set to 0 and the view factor criteria set to 0.0005.....	123
16. Land cover classifications mapped to RadTherm/RT part ID's.	125
17. Each RadTherm/RT part and the associated properties.....	127
18. The temperatures are compared at each ½ mile between mile marker 315 and 318. The UTM coordinates and element numbers at listed for each location.....	128

LIST OF FIGURES

Figure	Page
1. Electromagnetic Spectrum (After Incropera and DeWitt, 1996, Figure 12.3).....	8
2. Absorbed, reflected, and transmitted shortwave radiation with a semitransparent medium.	11
3. Radiation emitted by a surface. (a) Spectral distribution. (b) Directional distribution. (After Incropera and DeWitt, 1996, Figure 12.4).....	13
4. Radiation emitted from a point in differential area dA_i into a solid angle $d\omega$ subtended by differential area dA_n . (After Incropera and DeWitt, 1996, Figure 12.5).....	14
5. (a) Definition of a plane angle. (b) Definition of a solid angle. (After Incropera and DeWitt, 1996, Figure 12.6).....	15
6. In the spherical coordinate system, a solid angle subtended by dA_n from a point on dA_i . (After Incropera and DeWitt, 1996, Figure 12.7).....	16
7. Two surface normals of the differential areas dA_i and dA_j . θ_i and θ_j are the angles to the line connecting the two bodies.	17
8. The projection of dA_i and dA_j normal to the direction of radiation.	17
9. The differential solid angle subtended by dA_j from a point on dA_i	18
10. A diffuse reflector.	20
11. Surface radiosity.	21
12. A mesh created for RadTherm/RT for a region of interest.	24
13. A single soil element and the associated layers.	25
14. Heat transfer modes for each node.	26
15. Angle of incidence.	33

LIST OF FIGURES - CONTINUED

Figure	Page
16. Energy quantities incident upon and leaving a typical surface of an enclosure	37
17. The position of the sun is specified in terms of the zenith and azimuth angles, θ and ϕ , respectively.....	48
18. (a) A single ray cast from an element for ThermoAnalytics Ray Trace Algorithm. (b) Multiple rays cast from an element for ThermoAnalytics Ray Trace Algorithm. (Figures from ThermoAnalytics, Inc. website).....	49
19. (a) A simple model used for the hand calculation of the view factor. The model consists of two finite rectangles of the same length, having one common edge at an angle of 90° to each other. (b) The equation used to calculate the view factor from 1 to 2. (After Siegel and Howell, Figure 16 in the Appendix)	51
20. Example of ASCII input file for elevation data created from USGS 30 meter grid.....	55
21. Example of the ASCII input file for the terrain and road information. This file has been converted from the land classification to RadTherm/RT Part ID's. Note that 1 represents a part ID such as the road.....	57
22. Command line to execute the gis_manip PERL script.	59
23. (a) Set each part to type terrain. (b) Specify a weather file. (c) Set the same start and end solution times.....	61
24. Flow chart of Steps 1 through 3. Step 1 is generating the ASCII files. Step 2 determines the UTM coordinates for a map with less than 330,000 elements. Step 3 creates a new project in RadTherm/RT.	62
25. Example of a RadTherm/RT project composed of 2 parts.	63
26. Step 4 of the procedure. A C/C++ code is used to create a text file with the element attributes: Element ID, Part ID, UTM coordinates, and elevation.....	64
27. Example of the text file that contains the element attributes. This is the output file from Step 4.	65

LIST OF FIGURES - CONTINUED

Figure	Page
28. Flow chart for Step 5. This step converts the view factor file to an ASCII text file.....	66
29. The command line to execute the vf2asc utility	66
30. This is the output file from Step 5. Example of the view factor file produced from the vf2asc executable.	67
31. Flow chart for an overview of Step 6. The Element Attribute text file and View factor text file are used as input into Step 6. The output from Step 6 is a text file containing the elements to delete.	68
32. Flow chart for Step 6a. This step separates all the surface elements for the terrain map into two categories. The first category is the elements of interest with a rank of 1 and the second category is the surrounding elements with a rank of 2.....	69
33. A simple demonstration terrain model to describe the ranking after Step 6a.	70
34. Simple demonstration terrain model to describe the ranking after Step 6b.....	71
35. Illustrates the three options for Step 6c. The emitter is an element of interest, or the receiver is an element of interest, or neither the emitter nor receiver is an element of interest.	72
36. Simple demonstration terrain model to describe the ranking after Step 6c.	73
37. Simple demonstration terrain model to describe the ranking after Step 6d.....	74
38. Summary of the ranks applied to the elements in Steps 6a, 6b, 6c, and 6d.	75
39. An example of the output text file from Step 6e. The first row of the file contains the number of elements to delete. The subsequent rows are the element numbers to delete (R = 1).....	76
40. Summary of steps 6b through 6e.	77

LIST OF FIGURES - CONTINUED

Figure	Page
41. Flow chart for Step 7. The elements to delete, which are contained in the text file, are assigned to a new Part ID.	78
42. Example of the new update.tdf project created for RadTherm/RT. A new part has been created (Part 18) labeled "Elements to Delete".	79
43. Example of the update.tdf project after Part 18, which contains the elements to delete, has been removed.....	79
44. Simple model to demonstrate importance of shading. A represents the interstate and B represents a wall of rock.	81
45. A simple model that represents a flat highway.....	82
46. A temperature plot for element 6 of the interstate without a rock wall versus element 6 of the interstate with a rock wall. The plot is from 12:00 AM on July 22, 1993 to 12:00 PM on July 23, 1993.	83
47. The side view of a model with a sloped rock wall with a varying angle θ	84
48. (a) Simple model of the interstate with fifteen surface elements. (b) Simple model of the interstate with an adjoining terrain part composed of nine surface elements (16-24) and modeled as rock.....	85
49. Plot of the 24-hour temperature output for Element 10 for both models described in Figure 48b.....	86
50. (a) Side view of the model with the sloped wall at an angle of 135° . (b) A model composed of two parts. Part A is the interstate with fifteen elements. Part B is the sloped wall at 135° from the interstate.....	88
51. The temperature plot for Element number 10 for two cases. A rock slope with: 1) an absorptivity (α) of 0.05 and 2) $\alpha = 0.90$	91
52. The temperature plot for Element number 20 for both cases. For the first case, the rock slope has an absorptivity (α) of 0.05 and in the second case, $\alpha = 0.90$. Element 20 is the center element on the sloped terrain.....	92

LIST OF FIGURES - CONTINUED

Figure	Page
53. (a) Side view of the simple model consisting of two parts, the interstate and a rock wall at and angle of 90°. (b) A simple RadTherm/RT model consisting of two parts, the interstate and a rock wall at 90°. Fifteen elements are assigned to the interstate and nine elements are assigned to the rock wall at 90°.	92
54. The temperature plot for Element 10 for both cases. For the first case, the rock slope has an absorptivity of 0.05 and in the second case has an absorptivity of 0.90.	94
55. The temperature plot for 90 degree wall, Element 20 for two cases. With a rock slope that has 1) an absorptivity (α) of 0.05 and 2) $\alpha = 0.90$	95
56. The highway temperature of Element 10 for the model with a sloped terrain at 135° and 90° with an absorptivity value of 0.90.	96
57. The temperature plot of Element 10 for the model with terrain sloped at 135° and 90° with an absorptivity value of 0.05.	97
58. Side view of the models used to compare the longwave radiation effects. (a) Simple model without a left rock wall. (b) Simple model with a rock wall on the left side.	98
59. The model used to compare the longwave radiation effects with the pertinent elements labeled.	100
60. The temperature plot of interstate element 40 for both models. Note that the solar radiation is set to zero at 9:50 AM.	101
61. The temperature plot of Element 40 where the left rock wall increases in 10-meter segments.....	102
62. A plot of the temperature of interstate Element 40 to Element 112 on the left rock wall.....	103
63. A plot of the temperature of interstate Element 40 (with no left wall) to the sky temperature for the model.	104

LIST OF FIGURES - CONTINUED

Figure	Page
64. The UTM coordinates used to create the model for Bozeman Pass.	104
65. RadTherm/RT model for Bozeman Pass consisting of 327,610 surface elements and 17 parts.....	105
66. The Part ID that contains the elements of interest must be specified. The two lines of code that must be set for Step 6 are labeled A and B.....	107
67. A simple model to illustrate the distance criterion set at 100 meters.	108
68. The distance criterion in Step 6b is assigned in the code, which is labeled as C. In this case, the distance criterion is 250 meters.....	109
69. The view factor criteria must be set for lines D and E. In this case, the view factor criteria are 0.0005.....	109
70. This is the identical model as in Figure 65, but contains an additional part. This part (Part 18) contains the elements to delete.	110
71. A magnified picture of the interstate through Bozeman Pass. The empty spaces are a result of the deleted elements.....	112
72. The resulting model of Bozeman Pass with the distance criterion set to 250 meters and the view factor criteria set to zero. Only 82,000 surface elements have been deleted and 245,000 surface elements remain in the model.....	114
73. Plot of Equation (81), which is the energy lost or gained from surface element k to j . The x-axis is the radiation from surface element j and varies from 0 to 50,000 Watts. The graph plots the radiation term of surface k for four different temperature values of j	117
74. The resulting model of Bozeman Pass with the distance criterion set to 250 meters and the view factor criteria set to 0.0005. 288,308 surface elements have been deleted and 39,302 surface elements remain in the model.....	119

LIST OF FIGURES - CONTINUED

Figure	Page
75. The region selected for the model of Rocky Canyon. The interstate spans from mile marker 314½ to mile marker 318.....	121
76. A 3-dimensional map of Interstate 90 through Rocky Canyon from mile marker 314 heading east.	121
77. RadTherm/RT model for Rocky Canyon consisting of 33,096 surface elements.	122
78. RadTherm/RT model for Rocky Canyon after the view factor trim routine is applied. The distance criterion was set at zero and the view factor criteria were set at 0.0005. This model consists of 5,777 surface elements.....	124
79. An example of the RadTherm/RT GUI for a foliage terrain part.	126
80. Meteorological Plots from 12:00 am on July 2 nd , 2003 for (a) air temperature vs. time (b) solar radiation vs. time (c) cloud cover verse time of day.....	130
81. (a) Temperature comparison for element 17775 (original map) verse element 4545 (map with deleted elements) for July 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	131
82. (a) Temperature comparison for element 19009 (original map) verse element 4964 (map with deleted elements) for July 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	132
83. (a) Temperature comparison for element 14329 (original map) verse element 3310 (map with deleted elements) for July 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	133
84. (a) Temperature comparison for element 10832 (original map) verse element 1903 (map with deleted elements) for July 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	134
85. (a) Temperature comparison for element 10494 (original map) verse element 1813 (map with deleted elements) for July 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	134

LIST OF FIGURES - CONTINUED

Figure	Page
86. (a) Temperature comparison for element 7388 (original map) verse element 810 (map with deleted elements) for July 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	135
87. (a) Temperature comparison for element 5470 (original map) verse element 379 (map with deleted elements) for July 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	136
88. Meteorological Plots from 12:00 am on February 2 nd , 2003 for (a) air temperature vs. time (b) solar radiation vs. time (c) cloud cover verse time of day	138
89. (a) Temperature comparison for element 5470 (original map) verse element 379 (map with deleted elements) for February 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.	139
90. (a) Temperature comparison between element 19009 (original map) verse element 4964 (map with deleted element) for February 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.....	139
91. (a) Temperature comparison between element 14329 (original map) verse element 3310 (map with deleted element) for February 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.....	140
92. (a) Temperature comparison between element 10832 (original map) verse element 1903 (map with deleted element) for February 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.....	141
93. (a) Temperature comparison between element 10494 (original map) verse element 1813 (map with deleted element) for February 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.....	141

LIST OF FIGURES - CONTINUED

Figure	Page
94. (a) Temperature comparison between element 7388 (original map) verse element 810 (map with deleted element) for February 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.....	142
95. (a) Temperature comparison between element 5470 (original map) verse element 379 (map with deleted element) for February 2 nd , 2003. (b) Difference in Temperature between the original map and the trimmed map.....	143

NOMENCLATURE

A	Area (m^2)
C_{ak}	Air to ground convection conductor
C_{Ho}	Bare soil heat transfer coefficient
C_{kk}	Ground layer conduction
C_p	Specific heat at constant pressure (J/kg-K)
C_{Rain}	Rain convection conductor
CAP_{k_i}	Heat capacitance ($\text{J/ m}^2\text{-K}$)
d_i	Thickness of ground layers (m)
E	Emissive Power (W/m^2)
E_b	Emissive Power of Blackbody (W/m^2)
F_{i-j}	View Factor
G	Irradiation (W/m^2)
h_{fg}	Latent Heat of vaporization (J/kg)
$I_{\lambda,e}$	Emitted spectral intensity ($\text{W/m}^2\text{-K}$)
I_i	Intensity of radiation leaving surface i ($\text{W/m}^2\text{-K}$)
J_i	Radiosity (W/m^2)
k	Von Karmon constant
k	Thermal conductivity (W/m-K)
m	Mass (kg)

NOMENCLATURE - CONTINUED

n_i	Surface normal
n_j	Surface normal
N	Number of surfaces (or elements) in an enclosure
q	Heat transfer rate (W)
Q	Heat flux (W/m ²)
r	Distance between centers of differential areas (m).
t	Time (s)
T_a	Air temperature (K)
T_k	Temperature at the surface (K)
T_R	Rain temperature (K)
T_s	Absolute Temperature (K)
u	Specific internal energy (kJ/kg)
U	Total internal energy (kJ)
v	Average rain velocity (mm/hr)
v_a	Average Wind Velocity (m/s)
w	Soil moisture
$\dot{W}$	rate at which work is performed (W)
ω_g	Volume fraction of surface soil moisture
w_k	Volume fraction of surface soil moisture
α	Absorptivity

NOMENCLATURE – CONTINUED

ε	Emissivity
λ	Wavelength (m)
ϕ	Azimuth angle (rad)
ρ	Reflectivity (Albedo)
ρ_a	Density of air (kg/m ³)
τ	Transmissivity
θ	Zenith angle (rad)
σ	Stefan-Boltzmann Constant
ω	Solid angle (sr)

CHAPTER 1

INTRODUCTION

The Western Transportation Institute (WTI) is collaborating with the Montana Department of Transportation (MDT), Meridian Environmental Technology Incorporated, The Montana Department of Federal Highways Administration, Montana State University Civil Engineering, and ThermoAnalytics Incorporated (TAI) on a project known as the Greater Yellowstone Regional Traveler and Weather Information System (GYRTWIS). Part of this project is to develop and integrate a pavement temperature thermal model to provide a more detailed temperature forecast for two mountain passes: Lookout Pass on the Montana-Idaho border and Bozeman Pass [Ballard, 2001]. The goal is to accurately provide current and forecast weather and road condition information to highway operators and users. The prediction of pavement temperatures is currently being provided to help decision makers improve winter road maintenance operations. It could also be used in the future to provide forecasts of icy conditions for travelers.

The idea for modeling sections of highway is based on the adaptation of computational models used by the U.S military for the prediction of vehicle infrared images. Two programs were developed as tools to model the surface temperatures of vehicles for use in infrared imagery. The origin of the one dimensional, first principles heat transfer software for which the pavement model is derived is the Thermal Contrast Model (TCM), initially developed for the U.S. Air Force and PRISM (Physically Reasonable Infrared Signature Mode), which was produced at Michigan Technological

University's Keweenaw Research Center in conjunction with the U.S. Army Tank-Automotive Command (TACOM) [Adams, 1999]. Basically, the infrared signature of a vehicle exposed to a set of meteorological conditions indicates the surface temperature of the vehicle. The three-dimensional modeled vehicles were composed of a collection of flat plates termed "facets". The terrain in the model was considered as background and originally represented as an isothermal, flat plate. The facet concept used for vehicles was extended and used to represent the terrain presently used in the thermal modeling software [Adams and McDowell, 1991].

The current modeling tool used to predict these surface temperatures is a Radiation based Thermal Model for Road Temperature (RadTherm/RT). RadTherm/RT is a thermal modeling tool intended for comprehensive heat management design and analysis. ThermoAnalytics developed the software in conjunction with Montana State University to predict and graphically display road and surrounding terrain temperatures. The software provides a tool to analyze and model terrain temperature utilizing 3-D radiation, convection, and 1-D conduction. The input into the software includes a digital 3D-faceted terrain representation, meteorological data, and time/location information.

Developments in Geographic Information Systems (GIS), and the availability of Digital Elevation Maps (DEM) make it possible to produce files for elevation, road, and vegetation maps. The vegetation in this context includes grass, forests, water, rock, soil, etc. The elevation data was compiled from a United States Geological Survey (USGS) DEM's at a 30-meter resolution. The weather file includes standard meteorological parameters such as air temperature, wind velocity, precipitation, relative humidity, and

radiation data. From USGS DEM's, the mesh of the model was created in 30-meter square sections. Once the files were loaded into RadTherm/RT the sections were further divided into triangular elements. The output from RadTherm/RT is the temperature map for each node in the system.

The first step in constructing the thermal model is to fabricate maps for the region of interest [Adams, 1999]. Interstate 90 through Bozeman Pass will be the area of focus for this thesis. Geographic Information Systems (GIS) experts from Montana State University's Geographic Information and Analysis Center (GIAC) produced the ASCII files for the elevation, road and vegetation maps using ARC/INFO GIS software. The U.S. Geologic Survey provides the digital elevation maps as the source for the elevation file. The terrain file is simply the GIS vegetation ID's from the land cover class and is converted to RadTherm/RT part numbers used to group land covers with similar engineering properties, including emissivity, absorptivity, thermal conductivity, etc.

The elevation and terrain files are loaded into RadTherm/RT to create a new project. However, there is a limitation on the size of a thermal map that can be processed. As the boundaries of the thermal map increase, the number of elements also increases. At a certain point, the calculations for the thermal solution become too complex for computers to solve. Therefore, to optimize the models, a methodology to optimize the number of elements without changing the dimensions of the thermal map was considered beneficial.

One attempt to reduce the size of the thermal map was with the use of a topographic map. The goal was to eliminate the maximum number of elements while keeping all important terrain features that could cause temperature changes on the highway by shade,

radiation exchange, etc. The major concern for this approach is the lack of standards for reducing the boundaries of the map. For example, if a pertinent terrain attribute, such as a mountain peak, is deleted it could greatly influence the temperature output of the terrain or road.

Another method to increase the efficiency of the thermal maps combine the efforts of Montana State University and ThermoAnalytics, Inc. The DEM and vegetation file produced by GIAC for Bozeman Pass is sent to ThermoAnalytics, Inc. to be remeshed. This process includes combining terrain elements to reduce the number of nodes in the map. For instance, four 30-meter square elements are combined to create a single 60x60-meter square element. There are two disadvantages of this process. First, by combining elements, different thermal properties are lost and inaccuracies may be introduced. Second, coarsening the map can be a time-consuming task.

For this project, an analytical approach to optimize the elements used for thermal maps in RadTherm/RT was developed. Chapter 2 gives an overview of heat transfer principles and develops the equations used in the software. View factors are introduced and explained as the basis for eliminating elements for this methodology. A view factor is the fraction of energy that leaves one surface and is absorbed at a second surface. ThermoAnalytics' software determines the radiation relationships between elements that are visible to one another in the model and to the environment [www.thermoanalytics.com]. This algorithm is explored in detail in Chapter 3.

A simple model is introduced in Chapter 3 to describe the energy exchange between elements in different scenarios. The view factors are shown for different orientations and

resulting temperatures are compared. Conclusions from the example demonstrate that energy exchange is highly dependent on angle of slope, distance, and the material surface.

The next section of Chapter 3 describes a detailed methodology used to reduce the number of elements from a thermal map. A four-part C/C++ code was implemented to complete this task. This process is designed to be universal for any region of interest. The goal of this thesis is to significantly reduce the number of elements to improve the efficiency of the map without sacrificing accuracy. This would allow longer stretches of highway to be run with minimal computation time.

CHAPTER 2

BACKGROUND

Heat Transfer

Thermal energy is related to the temperature of matter. The higher the temperature of a material of given mass, the greater its thermal energy. Heat transfer is the exchange of thermal energy through a body or between bodies and occurs when there is a temperature difference. When a temperature gradient exists between two bodies, thermal energy transfers from the one with a higher temperature to the one with a lower temperature.

There are three modes of heat transfer: conduction, convection, and radiation. Any energy exchange between bodies occurs through one of these modes or a combination of them [Incropera and DeWitt, 1996]. The following section gives a brief introduction to each mode.

Conduction

The definition of conduction is the transfer of energy from the more energetic to the less energetic particles of a substance due to interactions between the particles. When there exists a temperature gradient, energy transfer by conduction will occur in the direction of decreasing temperature [Incropera and DeWitt, 1996]. The rate of heat flow due to conduction increases with the thermal conductivity of the material, the cross-sectional area available for it to pass through, and the temperature difference from one

side to the other. As material thickness increases, due to the increased path length, the longer it takes the heat to “flow” from one side to another.

Convection

Convection is heat transfer by mass motion of a fluid such as air or water when the heated fluid is caused to move away from the source of heat, carrying energy with it. Natural convection above a hot surface occurs when hot air expands it becomes less dense, and rises. On the other hand, forced convection is caused by something external such as atmospheric winds or a fan. RadTherm/RT uses the flat plate assumption to calculate the forced convective heat transfer. This is highly dependent on wind speed and temperature difference between the air and surface.

Radiation: Processes and Properties

Radiation is the transfer of energy in the form of electromagnetic waves. This radiation is emitted by objects in all directions without the need of a solid or fluid to transfer the heat [Incropera and DeWitt, 1996]. It uses electromagnetic radiation (photons), which travels at the speed of light and is emitted by any matter with temperature above zero degrees Kelvin (-273°C). The main types of radiation (from short to long wavelengths) are; gamma rays, X-rays, ultraviolet (UV), visible light, infrared (IR), microwaves, and radio waves. The electromagnetic spectrum classifies radiation according to wavelengths. High-energy physicists and nuclear engineers are primarily concerned with the short wavelength gamma rays, X rays, and UV radiation, while electrical engineers focus on the longer wavelength microwaves and radio waves

[Incropera and DeWitt, 1996]. Thermal radiation is given off by all surfaces that surround us and is in band wavelengths from 0.1 to 100 μm in the electromagnetic spectrum [Sparrow, 1970]. The complete electromagnetic spectrum is illustrated in Figure 1.

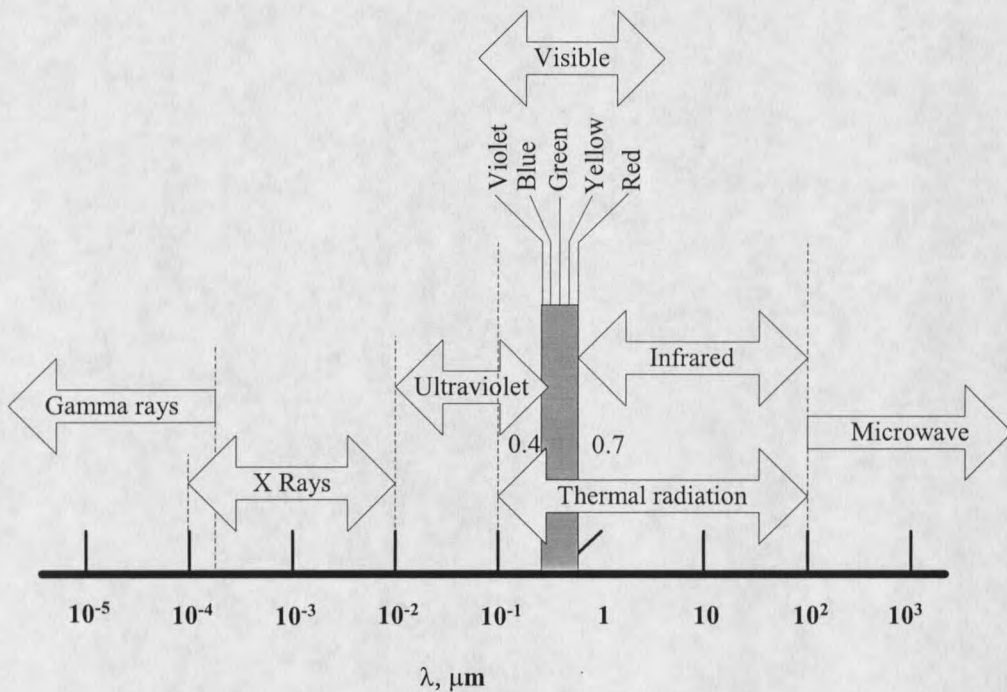


Figure 1. Electromagnetic Spectrum (After Incropera and DeWitt, 1996, Figure 12.3)

Radiation with shorter wavelengths is more energetic. The temperature of the body largely determines the type of radiation emitted. Very hot objects, such as the sun at ~ 5800 K, emit more energetic radiation including visible and UV. For engineering applications, radiated power is usually in infrared, and sometimes visible or UV. Radiation can be emitted, absorbed, transmitted and reflected by matter (solid, liquid and gas) and can be broken into two parts, shortwave and longwave [Sparrow, 1970].

Shortwave Radiation. The primary source of energy forcing atmospheric motion and many different processes in the atmosphere, at the Earth's surface and in the oceans, is the radiant energy from the sun [Pluss, 1997]. The radiation emitted by the sun is approximately equal to the emission of a blackbody at 5750 K. More than 99% of the energy received from the sun at the top of the earth's atmosphere is in the wavelength range 0.2 to 4.0 μm and known as shortwave radiation [Pluss, 1997]. Shortwave (or Solar) radiation reaches the earth's surface either by being transmitted directly through the atmosphere, direct solar radiation, or by being scattered or reflected to the surface, known as diffuse solar radiation. Diffuse solar radiation is scattered by air molecules, aerosol particles, cloud particles, or other particles.

About 50 percent of shortwave radiation is reflected back into space, while the remaining shortwave radiation at the top of the atmosphere is absorbed by the earth's surface and re-radiated as thermal infrared, or longwave radiation [Pluss, 1997].

Longwave Radiation. Hot objects like the sun emit radiation in shorter wavelengths, while colder bodies, such as the Earth or Earth's atmosphere, emit radiation in longer wavelengths ranging from 4.0 to 100.0 μm . Most longwave radiation reaching the earth surface is emitted from the lowest layers of the atmosphere. The main emitter of incoming longwave radiation is the greenhouse of the earth (e.g. water vapor, carbon dioxide, and ozone) [Pluss, 1997]. Longwave radiation is created from the temperature difference between the atmosphere and the earth's surface. On a clear day the atmosphere is much cooler than the earth's surface and energy is emitted from the surface to the sky.

Radiation Exchange Between Surfaces

All bodies above absolute zero Kelvin temperature radiate energy. Substances emit and absorb radiant energy at a rate depending on the absolute temperature and physical properties of the substance. Not only do they radiate or emit energy, but they also receive and absorb it from other sources [Beckwith, et al, 1993]. Two sets of criteria govern radiative heat transfer between surfaces. The first depends on the material properties, including emissivity, absorptivity, and reflectivity with each depending on temperature.

The second consists of the geometry and the transmission of radiation as a function of direction from the surface and the wavelength of the emitted radiation. This is calculated by the use of view factors. The study of radiation heat transfer in engineering is concerned with the heat loss by emission and heat gain by absorption.

Surface Absorption, Reflection, and Transmission. Incoming shortwave (or solar) radiation that strikes the earth's surface is partially reflected and partially absorbed, in proportion to the surface reflectivity. Figure 2 describes the absorbed, reflected, and transmitted radiation with a semitransparent medium. These incoming portions are divided into fractions of absorptivity (α), reflectivity (ρ), and transmissivity (τ), where

$$\alpha + \rho + \tau = 1. \quad (1)$$

Absorptivity is defined as the fraction of incident radiation (or irradiation) absorbed by an object in relation to a blackbody (range 0-1). Irradiance is the amount of electromagnetic energy incident on a surface per unit time per unit area. This quantity is often referred to as "flux" [Ozisik, 1977].

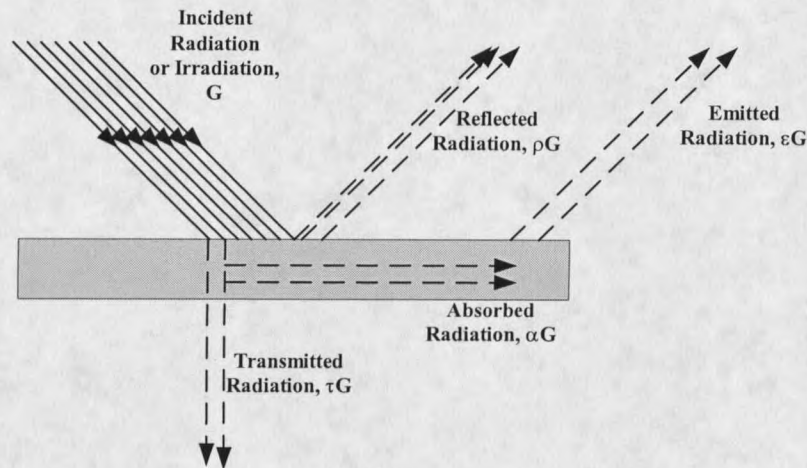


Figure 2. Absorbed, reflected, and transmitted shortwave radiation with a semitransparent medium.

A blackbody is any object that completely absorbs any heat or light radiation falling upon it. Reflectivity (or albedo) is the ratio of the reflected solar radiation divided by incoming solar radiation, incident upon a surface. Transmissivity is a property that is dependent on the temperature of the body and the wavelength of the incident radiation. It is a dimensionless value and measured as the fraction of incident radiation that is transmitted through the body [Ozisik, 1977].

A highly polished surface is referred to as an ideal reflector, where reflectivity (ρ) approaches 1 ($\rho \rightarrow 1$). Many gases represent substances of high transmissivity, for which $\tau \rightarrow 1$. An example of an idea absorber, or blackbody, is a small opening into a large cavity, for which $\alpha \rightarrow 1$ [Beckwith, et al, 1993]. For opaque solid bodies, such as soil, rock, or highway the transmissivity component can be ignored and Equation (1) is

written as,

$$\alpha + \rho = 1. \quad (2)$$

Surface Emission. The rate at which energy is released per unit area by a surface is known as the emissive power E [Incropera and DeWitt, 1996]. At a prescribed temperature and wavelength, no surface emits more than a blackbody. The total emissive power of a blackbody and the upper limit of emissive power is specified by the Stefan-Boltzmann law,

$$E_b = \sigma T_s^4, \quad (3)$$

where σ is the Stefan-Boltzmann constant ($\sigma = 5.67 \times 10^{-8} \text{ W / (m}^2 \text{K}^4)$) and T_s is the absolute temperature (K) of the surface. The heat flux emitted by a real surface is less than that of a blackbody and dependent on the surfaces' emissivity value,

$$E = \varepsilon \sigma T_s^4. \quad (4)$$

Emissivity is a characteristic of a surface that determines its ability to emit heat by radiation [Ozisik, 1977]. It is a ratio of the amount of energy an object absorbs and reradiates versus amount of energy a perfect emitter or black body would radiate under the same conditions (range 0-1).

View Factor. The Stefan-Boltzman Equation (3) provides a method of determining the total energy leaving a surface, but gives no indication of the direction in which it travels. Since the goal is to calculate how heat is distributed among various objects, a second phenomenon is introduced that governs radiative heat transfer between surfaces. The geometric effects are introduced and these are described by the view factor (also

called a configuration or shape factor), F_{ij} , and are therefore important in computing the radiation exchange between any two surfaces. By definition, the view factor F_{ij} is the fraction of the radiation leaving surface i that is intercepted by surface j .

To develop a general expression for F_{ij} , consider three fluxes of thermal radiation, called emissive power, irradiation, and radiosity. First consider radiative emission from a surface (emissivity), which has both directional and spectral wavelength dependencies, is schematically shown in Figure 3.

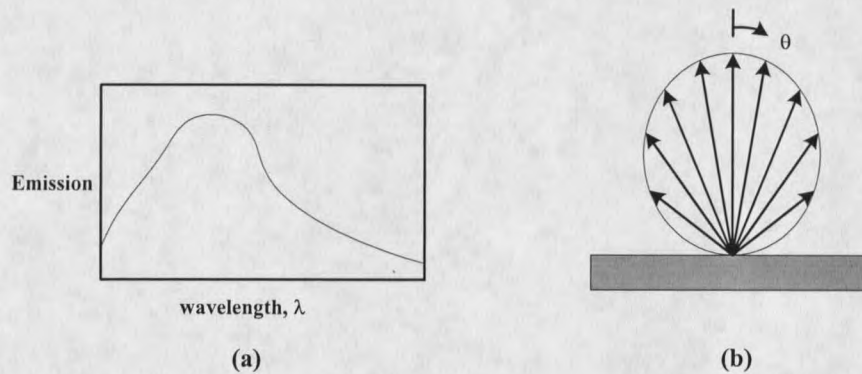


Figure 3. Radiation emitted by a surface. (a) Spectral distribution. (b) Directional distribution. (After Incropera and DeWitt, 1996, Figure 12.4)

Since radiation can potentially emit in all directions, it is most convenient to develop the mathematics in a spherical coordinate system, as illustrated in Figure 4. The differential element dA_1 on the surface of a material emits thermal radiation. The surface-normal direction is indicated by the unit vector n , and θ is the zenith angle from the surface normal.

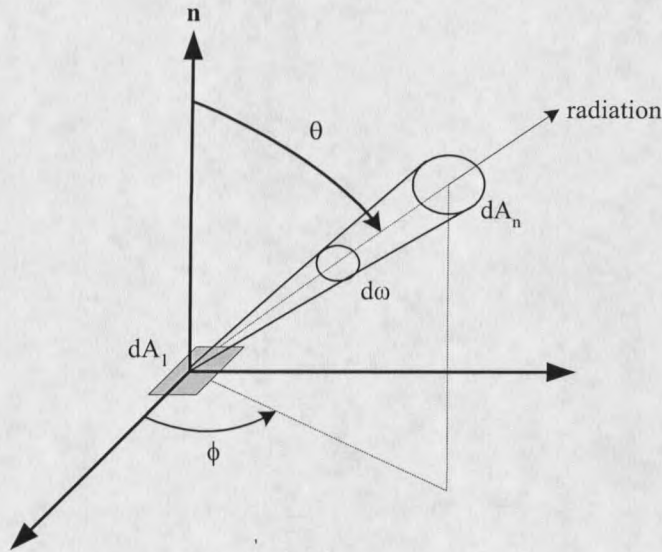


Figure 4. Radiation emitted from a point in differential area dA_1 into a solid angle $d\omega$ subtended by differential area dA_n . (After Incropera and DeWitt, 1996, Figure 12.5)

The radiation passes through a differential area dA_n that is normal to a ray emanating from the surface element dA_1 at angles θ and ϕ of the spherical coordinate system. The differential solid angle $d\omega$ is formed from a ray r connecting the center of dA_n to the center of dA_1 . The differential solid angle subtended by dA_n from a point on the surface dA_1 depends on two things. First, the distance r between centers of the differential areas and second, the area segment of a spherical surface dA_n that projects normal to the connecting ray between dA_1 and dA_n . As seen in Figure 5a, the differential plane angle $d\alpha$ is defined by a section between the rays of a circle and is measured in terms of arc length dl on the circle to the radius r of the circle [Incropera and DeWitt, 1996],

$$d\alpha = \frac{dl}{r}. \quad (5)$$

This is often used to find an arc length (dl), where

$$dl = r \cdot d\alpha. \quad (6)$$

The idea of an angle and arc length is taken to three dimensions to define a differential solid angle, $d\omega$. When the solid angle is multiplied by the radius squared, it has dimensions of length squared, or area, and will have the magnitude of the differential area (dA_n), where

$$dA_n = r^2 \cdot d\omega. \quad (7)$$

Solving Equation (7) for $d\omega$ yields

$$d\omega = \frac{dA_n}{r^2}. \quad (8)$$

A solid angle is shown in Figure 5b.

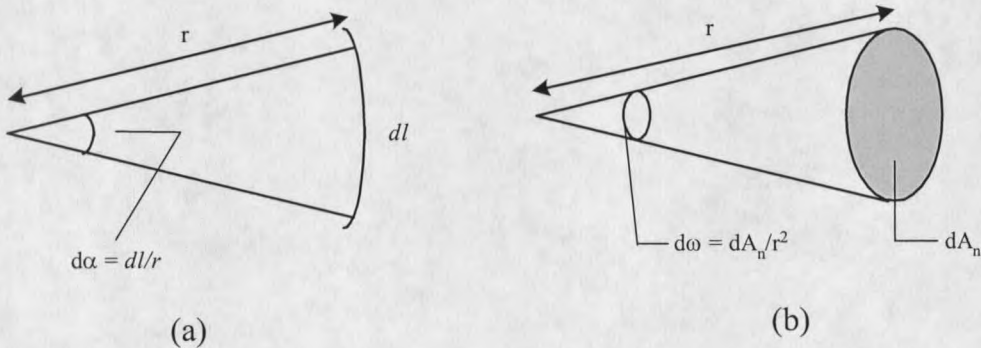


Figure 5. (a) Definition of a plane angle. (b) Definition of a solid angle. (After Incropera and DeWitt, 1996, Figure 12.6)

Since dA_n is a segment of a spherical surface, normal to the (θ, ϕ) direction, as shown in Figure 6, dA_n may be represented as the following

$$dA_n = r^2 \sin \theta \, d\theta \, d\phi. \quad (9)$$

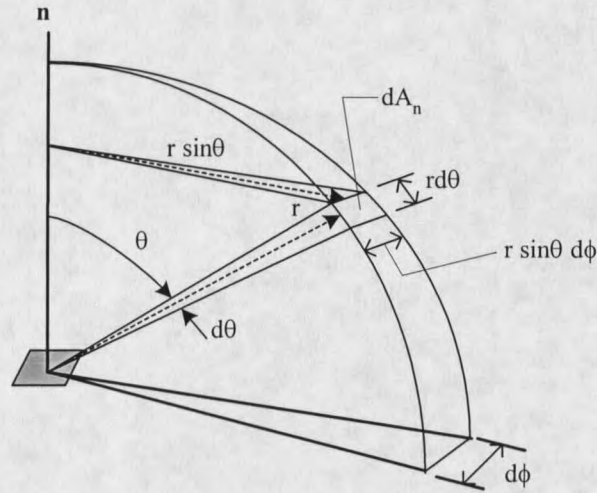


Figure 6. In the spherical coordinate system, a solid angle subtended by dA_n from a point on dA_i . (After Incropera and DeWitt, 1996, Figure 12.7)

Then, from the definition of $d\omega$, it is clear that

$$d\omega = \sin \theta \cdot d\theta \cdot d\phi \quad (10)$$

The above definitions are important when considering how energy radiates from a point source. Next, consider the rate at which emission from dA_i passes through dA_n as seen in Figure 4. Consider two infinitesimal elements of two surfaces dA_i and dA_j as seen in Figure 7. Each surface has a normal vector n_i and n_j , and the normals make angles θ_i and θ_j with the line connecting the two elements (Figure 7).

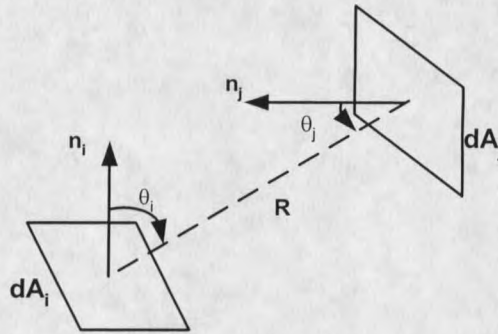


Figure 7. Two surface normals of the differential areas dA_i and dA_j . θ_i and θ_j are the angles to the line connecting the two bodies.

A variable called emitted spectral intensity, $I_{\lambda,e}(\lambda, \theta, \phi)$, is defined as the “rate at which radiant energy is emitted at the wavelength λ in the (θ, ϕ) direction, per unit area of the emitting surface normal to this direction, per unit solid angle around this direction, and per unit wavelength interval $d\lambda$ about λ [Incropera and DeWitt, 1996]. To illustrate “per unit area of emitting surface normal to this direction” refer to Figure 8.

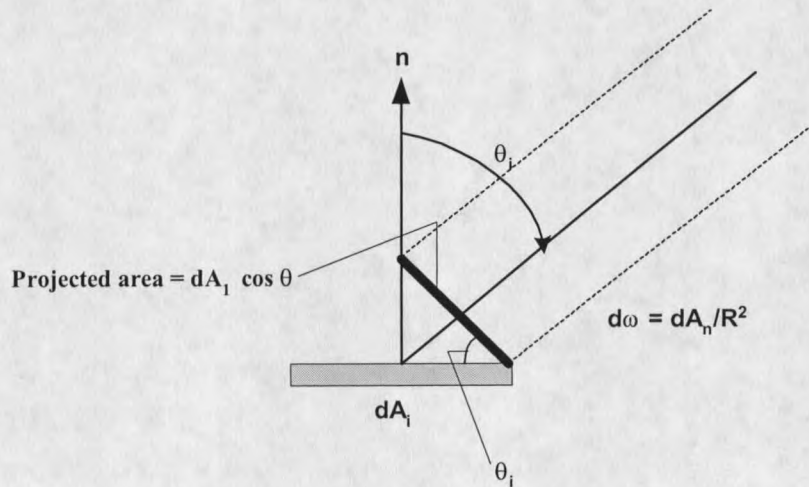


Figure 8. The projection of dA_i and dA_j normal to the direction of radiation.

The emitted spectral intensity is defined to account for the way in which an observer situated along an emitted ray would see dA_i . The differential area dA_n is normal to a ray emitting at an angle θ_i . The projected view of dA_i is $dA_i \cos \theta_i$. As θ_i approaches $\pi/2$, the view of dA_i is reduced until $\theta = \pi/2$ and at this point the projected view (dA_n) is zero. This is an important point because even if a surface is very hot, i.e., has a high emitted intensity, an observer viewing at a very low, shallow angle will receive very little energy from the surface. This reduced transfer is due to the small area through which the energy may be projected to the observer.

Figure 9 demonstrates the next part of the spectral intensity definition “per unit solid angle around this direction”.

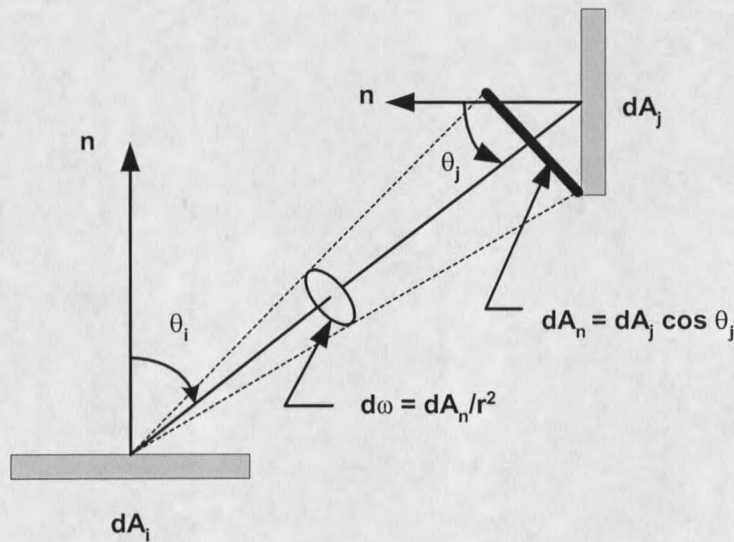


Figure 9. The differential solid angle subtended by dA_j from a point on dA_i .

The differential solid angle $d\omega_{j-i}$ subtended by dA_j relative to a point on dA_i depends on the distance between the centers of the differential areas r and the area dA_n that dA_j projects normal to the connecting ray between dA_i and dA_j .

$$d\omega_{j-i} = \frac{dA_n}{r^2} = \frac{dA_j \cos \theta_j}{r^2} \quad (11)$$

The emitted spectral intensity is defined as

$$I_\lambda(\lambda, \theta, \phi) = \frac{dq}{dA_i \cos \theta d\omega d\lambda}, \quad (12)$$

and

$$dq_\lambda = \frac{dq}{d\lambda}, \quad (13)$$

where dq is the rate at which radiation is emitted. Using the definition in Equation (13), Equation (12) is solved for dq_λ , which is the rate at which radiation of wavelength λ leaves dA_i and passes through dA_n ,

$$dq_\lambda = I_\lambda(\lambda, \theta, \phi) dA_i \cos \theta d\omega \quad (14)$$

This equation can be simplified if the emitting surface is a gray surface. A surface is gray if its emissivity and absorptivity are independent of wavelength, for the wavelengths at which irradiation and emission take place. Applying the assumption of a gray surface, Equation (14) can be written as the radiation from surface i and intercepted by j :

$$dq_{i-j} = I_i(\theta_i, \phi_i) (dA_i \cos \theta_i) d\omega_{j-i}, \quad (15)$$

where I_i is the intensity of radiation leaving surface i . Applying Equation (11), Equation (15) becomes

$$dq_{i-j} = I_i(\theta_i, \phi_i)(dA_i \cos \theta_i) \left(\frac{dA_j \cos \theta_j}{r^2} \right). \quad (16)$$

This shows that as an observer moves farther and farther from a given heat surface (i.e., increasingly large r), the amount of energy or heat received will be reduced because of the r^2 term in the denominator. Equation (16) can be reduced further if all visible light sources are diffuse emitters and all surfaces are diffuse reflectors. For a diffuse reflector an incoming ray of light is equally likely to be reflected in any direction over the hemisphere as in Figure 10.

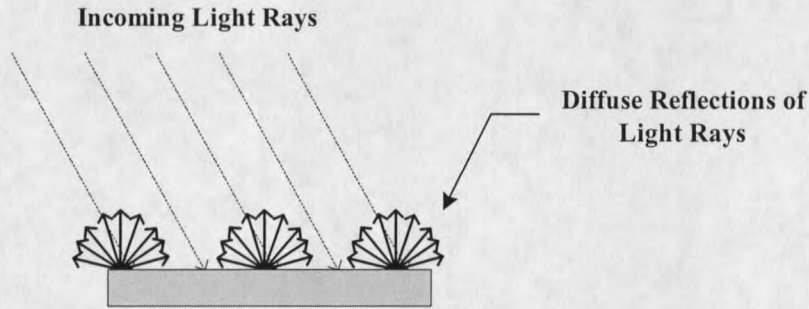


Figure 10. A diffuse reflector.

A diffuse emitter is a surface that emits radiation equally in all directions, analogous to reflections shown in Figure 10.

Therefore, the radiation for a diffuse surface, $I_i(\theta_i, \phi_i)$, is not dependent on direction (θ_i, ϕ_i) and gives a relationship between the radiation intensity, I_i , and the radiosity J_i ,

$$I_i = \frac{J_i}{\pi}. \quad (17)$$

Note that the radiation flux, in this case the radiosity, is based on the actual surface area, while the intensity is based on the projected area. The radiosity, J_i , from a surface (i) is the sum of the emissive power and the reflected part of the irradiation as shown in Figure 11.

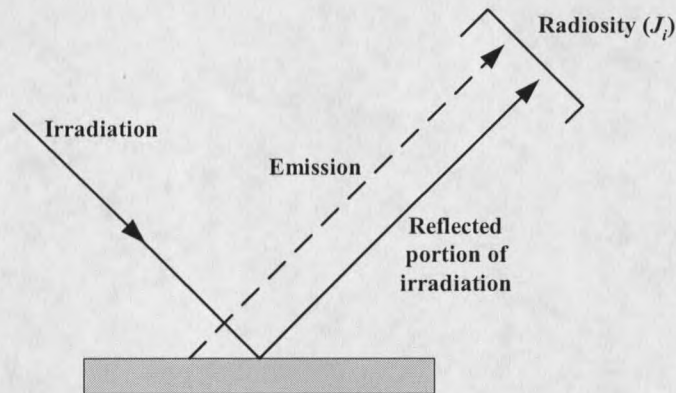


Figure 11. Surface radiosity.

Any body that has a temperature above absolute zero emits energy in the form of thermal radiation flux, which is called the Emissive Power, E . The irradiation G is defined as the flux incident on a surface, usually as the result of emissive power or radiosity from another surface.

Substituting Equation (17) into Equation (16) yields

$$dq_{i \rightarrow j} = J_i \frac{\cos \theta_i \cos \theta_j}{\pi r^2} dA_i dA_j. \quad (18)$$

The total rate at which radiation leaves surface i and is intercepted by j is obtained by integrating over the two surfaces. Note that the radiosity J_i is uniform over the surface A_i ,

$$q_{i \rightarrow j} = J_i \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} dA_i dA_j. \quad (19)$$

Using the definition of a view factor as the fraction of the radiation that leaves A_i and is intercepted by A_j ,

$$F_{i \rightarrow j} = \frac{q_{i \rightarrow j}}{A_i J_i}. \quad (20)$$

Substituting Equation (19) into Equation (20) gives us the view factor equation,

$$F_{i \rightarrow j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} dA_i dA_j. \quad (21)$$

In a similar manner the view factor from A_j to A_i is found to be

$$F_{j \rightarrow i} = \frac{1}{A_j} \int_{A_j} \int_{A_i} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} dA_i dA_j. \quad (22)$$

An important relation is the reciprocity for view factors between finite areas. The double integrals in Equation (21) and Equation (22) are identical and therefore gives us the reciprocity relation

$$A_i F_{ij} = A_j F_{ji}. \quad (23)$$

A second important relation relates to surfaces in an enclosure. From conservation of energy requirements, all radiation leaving any surface in the enclosure must be intercepted by a surface making up the enclosure. When this is applied to N surfaces in

an enclosure, the fraction of energy leaving one surface and reaching surfaces of the enclosure must total to unity [Incropera and DeWitt, 1996],

$$F_{i-1} + F_{i-2} + F_{i-3} + \cdots + F_{i-i} + \cdots + F_{i-N} = \sum_{j=1}^N F_{i-j} = 1. \quad (24)$$

The term F_{i-i} is included in the summation because when a surface is concave, it will intercept a part of its own emitted energy. On the other hand, for a plane or convex surface, F_{i-i} is zero.

RadTherm/RT Heat Transfer

RadTherm/RT is a thermal modeling program developed to predict surface terrain and road temperatures. Inputs into the software include a digital elevation map, a vegetation map, meteorological data, and time/location information [ThermoAnalytics, 1999]. The vegetation map includes different types of vegetation, rock, soil, and road information. A RadTherm/RT project includes a region of interest that is composed of a number of parts. Every part will have one or more elements associated with it and a thermal node is assigned to each element. Parts are collections of elements that have the same material properties. For example, short grass is considered a separate part from the interstate. The thermal results are solved for the individual elements, but all properties including part type, materials, thickness, convection coefficient, etc., are assigned at the part level. The properties applied to a part, are applied to all of the elements assigned to the part [ThermoAnalytics, 2000].

Every element has one or more thermal nodes, depending on the part type that has been chosen for the particular element. For this project, only terrain elements were considered. Therefore, each element in the mesh is modeled as having one surface (front) and a single thermal node is associated with the surface of each element. The thermal results are calculated for each of these nodes [ThermoAnalytics, 2000].

The geometry of a map is divided into a two-dimensional grid, where each rectangular elements of the grid is divided into two triangular terrain elements. The following example uses a quadrilateral element for simplicity of discussion. Figure 12 shows a map of Bozeman Pass and a section of the map broken into quadrilateral elements, each with a thermal node and an assigned part ID.

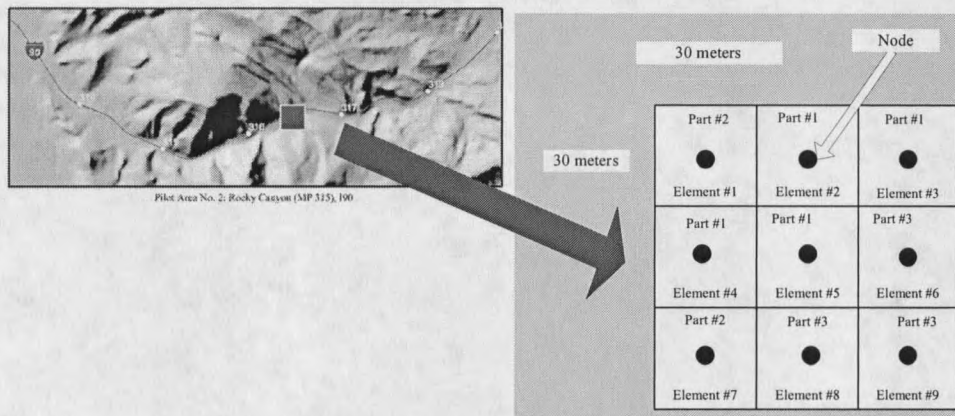


Figure 12. A mesh created for RadTherm/RT for a region of interest.

Terrain elements can be used to model several types of surfaces in RadTherm/RT including asphalt, concrete, foliage, soil, water, swamp, or snow. Each surface type is

divided into a more detailed part types, which include an interstate road, deciduous trees, short grass, rocky field, or fresh snow. These detailed part types are used to assign material properties for emissivity, absorptivity, conductivity, etc.

To develop the energy balance for a particular node, consider a single soil-type terrain element as in Figure 13.

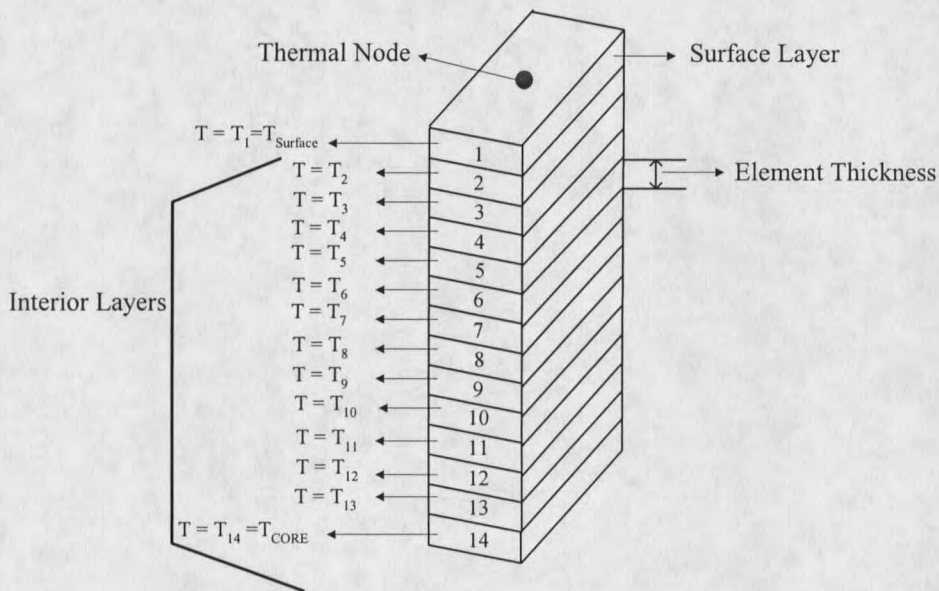


Figure 13. A single soil element and the associated layers.

The soil model is an energy balance solution for solving the surface temperature, as well as the interior layers, for seven different varieties of soil type under moisture conditions ranging from completely dry to saturated. The model consists of thirteen layers with varying properties dependent on soil type and water content [ThermoAnalytics, 1999]. The fourteenth soil layer has a known temperature called the core temperature and will be further described later in this section.

The governing equations are determined for each node. RadTherm/RT uses all modes of heat transfer including conduction, convection, and radiation in order to solve the temperature of a particular surface node. Figure 14 shows the modes for a particular surface element.

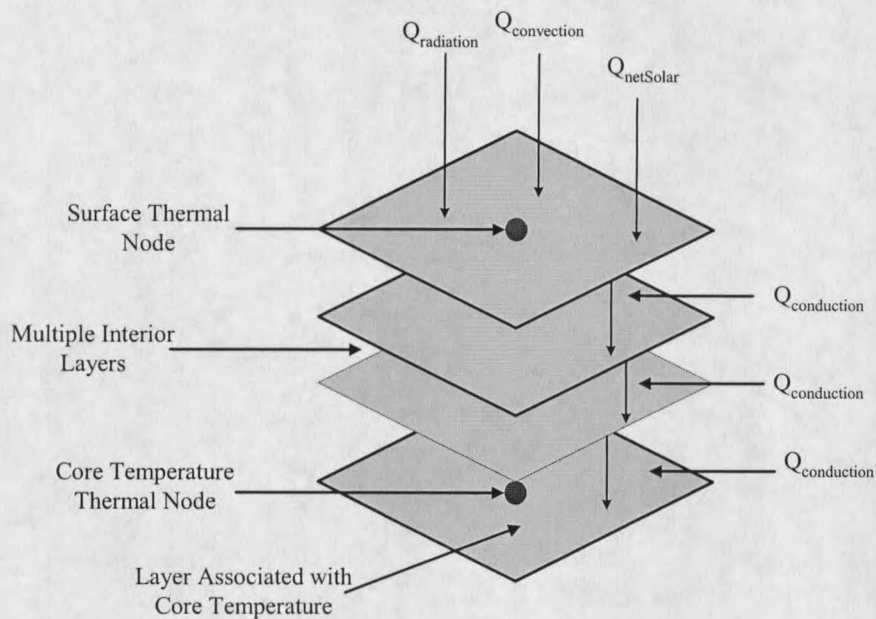


Figure 14. Heat transfer modes for each node.

The governing equations for the model are the transient energy equations and the net-radiation enclosure equation. The first law of thermodynamics is used to obtain the transient energy equation for Figure 14,

$$\frac{\partial}{\partial t} \Delta E = q - W . \quad (25)$$

The mechanical work rate (i.e. power) is denoted as $\dot{W}$, $\dot{q}$ is the net heat transfer rate across the system boundary (W), t is time, and ΔE is the change in the sum of potential, kinetic, and internal energy as

$$\frac{\partial}{\partial t} \Delta E = \frac{\partial}{\partial t} (\Delta PE + \Delta KE + \Delta U), \quad (26)$$

where PE is the potential energy, KE is the kinetic energy, U is the internal energy, and t is time. Mechanical work effects such as friction are neglected and therefore, $\dot{W} = 0$. Since the model forms a stationary closed system, the changes in kinetic and potential energies are negligible, so $\Delta PE = \Delta KE = 0$ [Cengel and Boles, 1994]. Therefore, the change in stored energy, ΔE , is only a function of total internal energy, U (kJ), and can be further simplified and expressed in terms of the specific internal energy, u (kJ/kg):

$$\frac{\partial}{\partial t} \Delta E = \frac{\partial}{\partial t} \Delta U = \frac{\partial}{\partial t} (m \Delta u) = \rho A d \frac{\partial}{\partial t} \Delta u, \quad (27)$$

where m is the mass, ρ is density, A is area, and d is the thickness of the material. Since only temperature is considered (i.e. not deformation, etc.), internal energy can be written as the specific heat, C_p (J/kg-K), times the temperature, T (K), yielding

$$\rho A d \frac{\partial}{\partial t} \Delta u = \rho A d C_p \frac{\partial T}{\partial t}. \quad (28)$$

Applying the simplifications above, the first-law in Equation (25) yields

$$\rho \cdot C_p \cdot \frac{\partial T}{\partial t} \cdot d \cdot A = \dot{q} \quad (29)$$

Equation (29) will be expanded to get the energy balance equations for the interior layers as well as the surface layer. An additional development of the view factor equation will

also be discussed in the following section because it is an important concept when dealing with radiation exchange between surfaces.

Interior Layer Energy Balance

The only heat transfer in the interior ground layers is conduction. Equation (29) becomes

$$CAP_{k_i} \cdot A_k \cdot \frac{\partial T}{\partial t} = q_{conduction} \quad (30)$$

$$CAP_{k_i} = (\rho C_p (w_i)) d_i, \quad (31)$$

for the interior layer energy balance, where CAP_{k_i} is the heat capacitance term, ρ is density (kg/m^3), the volumetric concentration of soil moisture is defined as w , A_k is the area of the k^{th} element (m^2), d_i is the thickness of each interior layer (m), and $q_{conduction}$ is the conduction heat rate (W). The notation for the subscript k_i is used to distinguish between the surface node and its associated layers. The k in subscript k_i denotes the surface element number and the subscript i in k_i indicates the interior node number in the k^{th} element. Throughout this thesis, the term T_k without the subscript i is used to represent T_{k_i} , which is the temperature of the surface node in element k . The conduction term in Equation (30) is further expanded to yield the transient energy equation for each interior thermal node of the k^{th} element. The energy balance in this case becomes

$$CAP_{k_i} \cdot A_k \cdot \frac{\partial T_{k_i}}{\partial t} = (T_{k_{i-1}} - T_{k_i}) \cdot A_k \cdot C_{kk_{i-1}} + (T_{k_{i+1}} - T_{k_i}) \cdot A_k \cdot C_{kk_i} \quad (32)$$

for $i = N_{Layers}$, $T_{N_{Layers}+1} = T_{coreTemp}$

$$C_{kk_i} = \frac{k(w_i)}{d_i} \quad (33)$$

is simple one-dimensional conduction relationship between the nodes above and below the current layer. In Equation (32), C_{kk_i} is the interior layer conduction where the thermal conductivity term k is a function of the soil moisture, w . N_{Layers} represents the number of layers for the particular terrain type. When $i=1$ the T_{k_i} term is the surface temperature of the k^{th} element. For the case of a soil element, N_{Layers} is thirteen. The model includes an additional node, $N_{Layers} + 1$, as seen in Figure 13 which is used to define a known core temperature, also known as the diurnal depth temperature. This diurnal depth is the depth at which the ground does not change over a 24-hour (diurnal) period [ThermoAnalytics, 1999].

Surface Energy Balance

The energy balance for the terrain surface layer consists of six terms: net sky radiation exchange, air convection, precipitation convection, solar load, and interior ground layer conduction. These terms are set equal to the internal energy storage term, which is the time rate of change of temperature times the heat capacitance [Terrain Manual].

Expanding the terms in Equation (29) for surface node k yields:

$$CAP_{k_1} A_k \frac{\partial T_k}{\partial t} = \left[q_{convection_k} + q_{conduction_k} + q_{radiation} + q_{netSolar_k} + q_{Rain_k} \right] \quad (34)$$

In Equation (34), CAP_{k_i} is the surface layer capacitance, T_k is the surface node temperature, A_k is the area of element k (m^2), and q represents the heat rate (W).

Energy Storage. The thermal capacitance is a function of the current moisture level and soil type, i.e.,

$$CAP_{k_i} = (\rho C_p (w_1)) d_1, \quad (35)$$

where $w_1 = w_g$. In Equation (35) the volumetric concentration of soil moisture is defined as w . The term w_g is the volume fraction of surface soil moisture, d_1 is the surface layer thickness, and ρ is the density.

Air Convection. The convection between the air and ground is represented as

$$q_{convection_k} = (T_a - T_k) \cdot A_k \cdot C_{ak}, \quad (36)$$

where T_a is the air temperature, A_k is the area of element k , and T_k is the temperature of the surface. The variable C_{ak} represents the air to ground convection conductor and defined as

$$C_{ak} = \rho_a C_{p,a} C_{Ho} v_a, \quad (37)$$

where ρ_a is the density of the air, $C_{p,a}$ is the specific heat of the air, C_{Ho} is the bare soil heat transfer coefficient, and v_a is the average wind velocity.

Precipitation Convection. The rain is handled separately with a simple advection formula given in Equation (38),

$$q_{Rain_k} = (T_{Rain} - T_k) \cdot A_k \cdot C_{Rain}. \quad (38)$$

C_{Rain} is the rain convection conductor and is defined as

$$C_{Rain} = \rho \cdot v \cdot C_p = 1.1611 \cdot v. \quad (39)$$

The average rain velocity (mm/hr) is expressed as v , ρ is the density of water, and C_p is the specific heat of water.

Ground Layer Conduction. Soil conductance is a function of the current moisture level and soil type. The conduction in the terrain nodes is only one dimensional through the layers and the elements do not transfer heat from neighboring transverse nodes.

$$q_{conduction_k} = (T_2 - T_k) \cdot A_k \cdot C_{kk_1} \quad (40)$$

$$C_{kk_1} = \frac{k(w_1)}{d_1} \quad (41)$$

In Equation (40), C_{kk_1} is the ground layer conduction and the thermal conductivity term, k , is a function of the soil moisture, w . The variable w_1 is the volume fraction of surface soil moisture, and d_1 is the surface layer thickness.

Solar Load. $q_{netSolar_k}$ is the amount of incident solar energy absorbed by each node in Watts. This is a known quantity in the governing equation because it is not temperature dependent. The magnitude of incident solar energy is the combination of the following terms [ThermoAnalytics, 2000]

$$q_{netSolar_k} = q_{directSolar_k} + q_{reflected_in_solar_k} - q_{reflected_out_solar_k}, \quad (42)$$

where $q_{directSolar_k}$ is the amount of incident solar energy absorbed from the measured direct solar irradiance (W), $q_{reflected_in_solar_k}$ is the sum of the reflected energy from other

surfaces (W), and $q_{reflected_out_solar_k}$ is the amount of energy not absorbed by the terrain elements, which is deflected to other enclosure elements (W).

The amount of incident solar energy absorbed from the measured direct solar irradiance, $Q_{direct_measured}$ (W/m²), is

$$q_{directSolar_k} = Q_{direct_measured} \cdot \alpha_k \cdot A_{view_k}, \quad (43)$$

where α_k is the absorptivity of the element k and A_{view_k} is the view area. The view area is the amount of the sun an element sees at a particular time and is defined as

$$A_{view_k} = A_{apparent_k} \cdot F_{visible_k}, \quad (44)$$

where $F_{visible_k}$ is the visibility factor. The apparent area, $A_{apparent_k}$, defines the unshadowed area of the element projected in the direction of the sun. This projection also depends on the angle of incidence

$$A_{apparent_k} = (A_k) \cdot \cos(\beta), \quad (45)$$

where A_k is the area of element k and β is the angle of incidence. The visibility factor is the fraction of the element that is not shadowed (by other elements) when looking at it from the position of the sun. Clearly an element receives more energy when the incoming rays from the sun are not obstructed by another element. The visibility factor is calculated by ThermoAnalytics Ray Tracing algorithm at every time step. Rays are cast from different locations on the element towards the sun. The visibility factor determines the portion of an element that receives direct solar radiation. The Ray Trace algorithm will be discussed in greater detail in the next chapter. The angle of incidence, β , as

shown in Figure 15 is the angle between the element normal and the incoming ray from the sun.

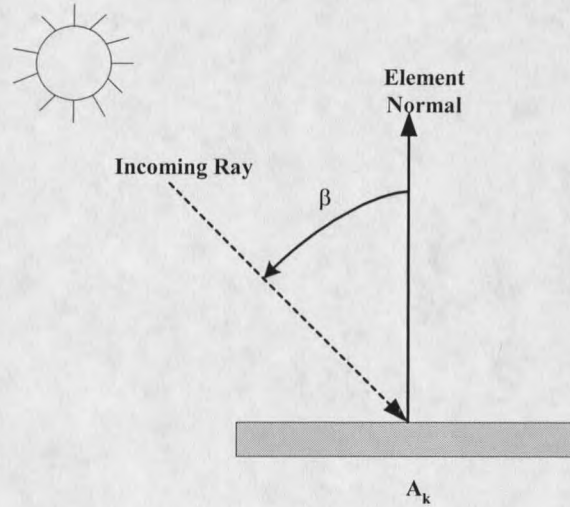


Figure 15. Angle of incidence.

The sum of the reflected energy is defined as

$$q_{\text{reflected_in_solar}_k} = q_{\text{refl_elements}_k} + q_{\text{diffuse_in}_k} + q_{\text{refl_default}_k} \quad (46)$$

The $q_{\text{refl_elements}_k}$ term includes the incoming reflected solar radiation from other elements.

$q_{\text{diffuse_in}_k}$ is the “first strike” diffuse solar irradiance (i.e. from the sky). This solar irradiance is not to be confused with the sky temperature, which is long wavelength radiation. Reflections from the default terrain onto the surface are accounted for in the term $q_{\text{refl_default}_k}$. The default terrain is included in RadTherm/RT when an element sees beyond the portion of terrain that is explicitly modeled with specified terrain elements.

The amount of energy not absorbed by an element and reflected to other elements is represented by $q_{reflected_out_solar_k}$. Assuming that the terrain surfaces are gray and diffuse emitters, the amount of reflected radiation would be

$$q_{reflected_out_solar_k} = \left[\begin{array}{l} Q_{direct_measured} \cdot A_{view} + q_{relected_in_solar_k} \\ -(q_{directSolar_k} + q_{reflected_in_solar_k} \cdot \alpha_k) \end{array} \right] \quad (47)$$

The solution of $q_{netSolar_k}$ is an iterative one. First, the sky assumed black and any solar radiation that is reflected back towards the sky does not get further reflected. An initial guess is made for $q_{reflected_out_solar_k}$ for each node k . Then, $q_{reflected_in_solar_k}$ is calculated for each node k (which depends on the guesses for $q_{reflected_out_solar_k}$ for the other elements), and a better estimate for $q_{reflected_out_solar_k}$ is calculated (i.e., it is just the reflected $q_{reflected_in_solar_k}$). Then this process is repeated iteratively until the values for $q_{reflected_in_solar_k} - q_{reflected_out_solar_k}$ converge to within some tolerance [Personal communication A. Curran].

Expanding Governing Equation. Expanding Equation (34) yields

$$CAP_{k_i} \cdot A_k \cdot \frac{\partial T}{\partial t} = \left[\begin{array}{l} (T_a - T_k) \cdot A_k \cdot C_{ak} + (T_{k_2} - T_k) \cdot A_k \cdot C_{kk_1} + q_{radiation_k} + q_{netSolar_k} \\ + (T_{Rain} - T_k) \cdot A_k \cdot C_{Rain} \end{array} \right] \quad (48)$$

For convenience, Equation (32) for the interior surface layers is repeated as

$$CAP_{k_i} \cdot A_k \cdot \frac{\partial T_{k_i}}{\partial t} = (T_{k_{i-1}} - T_{k_i}) \cdot A_k \cdot C_{kk_{i-1}} + (T_{k_{i+1}} - T_{k_i}) \cdot A_k \cdot C_{kk_i} \quad (49)$$

The only unknowns in Equation (48) are $q_{\text{radiation}_k}$, the surface temperature T_k , and the interior layer temperature T_{k_2} . When Equation (49) is written for each of the interior nodes of each element it gives $N_{\text{Layers}} - 1$ equations and N_{Layers} unknowns, T_k through $T_{k_{N_{\text{Layers}}}}$. Combining this with Equation (48) gives N_{Layers} equations and $N + 1$ unknowns. The additional unknown is the $q_{\text{radiation}_k}$ term. In order to solve this system of equations, the enclosure theory is introduced to solve for $q_{\text{radiation}_k}$.

Net-radiation Method – Enclosure Theory. The radiation term $q_{\text{radiation}_k}$ in Equation (47) is solved using the Net-radiation Method Enclosure Theory. An enclosure is an area that is completely surrounded by solid surfaces or open areas. Thus an enclosure accounts for all directions surrounding the terrain by defining a volume that includes the terrain, background terrain, and sky dome. The radiation exchange between diffuse-gray surfaces in an enclosure is determined by the net-radiation equation of the Enclosure Theory. The net-radiation equation is in terms of node emissivity, surface area, the Stephan-Boltzmann constant, the view factor from node k to node j , and the temperatures of nodes k and j [ThermoAnalytics, 2000].

The enclosure is composed of N discrete surfaces (or elements), which are taken as both diffuse and gray. The term diffuse indicates that the directional emissivity and directional absorptivity do not depend on direction. Therefore, similar to a blackbody, the emitted intensity will be uniform over all directions. Spectral emissivity and spectral absorptivity of a gray surface are independent of wavelength but do depend on temperature. Due to this assumption, at each surface temperature the emitted radiation

will be the same fraction of blackbody radiation for all wavelengths [Siegel and Howell, 1981]. To summarize, a diffuse-gray surface absorbs a fixed fraction of incident radiation from any direction and at any wavelength and emits radiation that is a fixed fraction of blackbody radiation for all directions and wavelengths [Siegel and Howell, 1981]. Therefore, for a diffuse-gray surface in an enclosure, the total (hemispherical) emissivity of the surface is equal to its total (hemispherical) absorptivity [Incropera and DeWitt, 1996]. This relationship is known as Kirchoff's law,

$$\varepsilon = \alpha . \quad (50)$$

Another condition that must be met is each surface or element of the enclosure must be at a uniform temperature. As a result, the emitted energy is taken to be uniform over the surface of each element of the enclosure [Siegel and Howell, 1981].

Part of the energy incident on the surface is reflected. Because the surface have been defined as gray, the first assumption made for the reflected energy is that it is diffuse. This assumption indicates that the reflected intensity at each position is uniform for all directions. The second assumption is the reflected energy is uniform over each element of the enclosure. Therefore, the reflected and emitted energy for each surface have the same diffuse and equally distributed characteristics [Siegel and Howell, 1981].

Energy leaving a surface does not vary with angular direction if the surface is both a diffuse emitter and diffuse reflector. View factors can be introduced for a non-black surface if the diffuse-uniform condition applies to both the reflected and emitted energy.

The net-radiation method will be used to simplify the complex radiative exchange that occurs inside an enclosure. To derive the net-radiation enclosure equation consider

the k^{th} surface area, A_k , of an enclosure (Figure 16), where Q_{in_k} is the rate of incoming radiant energy per unit inside area (irradiation) and Q_{out_k} is the rate of outgoing radiant energy per unit inside area (radiosity).

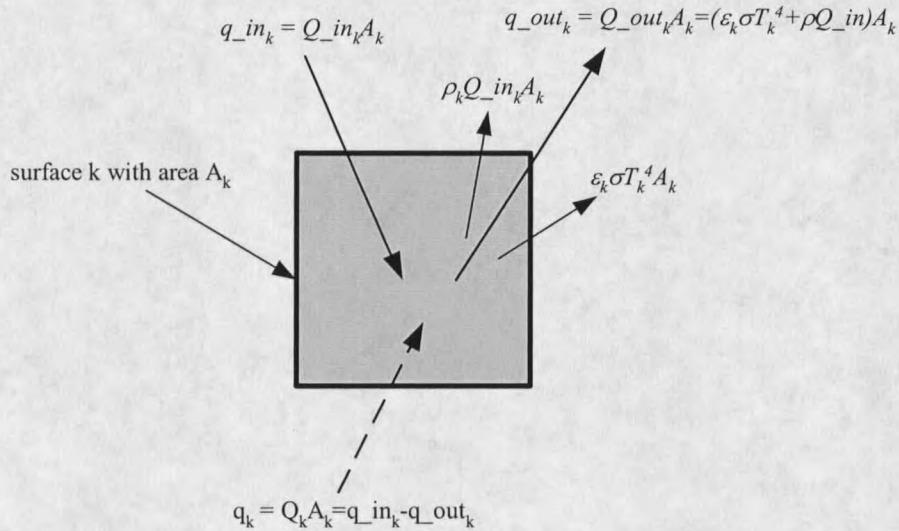


Figure 16. Energy quantities incident upon and leaving a typical surface of an enclosure.

Q_k is the energy flux that would have to be transferred to the surface by another mode other than radiation to maintain the specified surface temperature [Incropera and DeWitt, 1996],

$$Q_k = Q_{out_k} - Q_{in_k}. \quad (51)$$

The net radiation heat transfer rate from surface k is

$$q_k = A_k \cdot (Q_{out_k} - Q_{in_k}). \quad (52)$$

Note that in general, the radiosity, or energy flux leaving the surface, is composed of directly emitted plus reflected energy. By definition,

$$Q_{out_k} = \varepsilon_k \cdot \sigma \cdot T_k^4 + \rho_k \cdot Q_{in_k}. \quad (53)$$

Reflectivity may be written using Equation (2) and Equation (50) as follows

$$\rho_k = (1 - \alpha_k) = (1 - \varepsilon_k). \quad (54)$$

Therefore, Equation (53) can be written as

$$Q_{out_k} = \varepsilon_k \cdot \sigma \cdot T_k^4 + (1 - \varepsilon_k) \cdot Q_{in_k}, \quad (55)$$

where the relations in Equation (54) are for opaque gray surfaces. Equations (52) and (53) are the basic equations for enclosure analysis. The key to simplifying the algebraic problem is to recognize that the irradiation in Equation (52) can be evaluated in terms of the radiosity of neighboring surfaces in the enclosure. If the k^{th} surface can view itself (i.e., it is concave), a part of its outgoing flux will contribute directly to its incident flux.

That is,

$$A_k \cdot Q_{in_k} = \sum_{j=1}^N F_{j-k} \cdot A_j \cdot Q_{out_j}. \quad (56)$$

Using the reciprocity property of view factors,

$$A_j \cdot F_{j-k} = A_k \cdot F_{k-j}, \quad (57)$$

Equation (56) can then be written alternatively as

$$A_k \cdot Q_{in_k} = A_k \cdot \sum_{j=1}^N F_{k-j} \cdot Q_{out_j}, \quad (58)$$

and reduces to

$$Q_{in_k} = \sum_{j=1}^N F_{k-j} \cdot Q_{out_j}. \quad (59)$$

Equations (55) and (59) provide two expressions for Q_{in_k} . Each of these equations is substituted into Equation (52) separately to get two basic heat balance equations for q_k in terms of Q_{out_k} . Solving Equation (55) for Q_{in_k} yields

$$Q_{in_k} = A_k \cdot \left(\frac{Q_{out_k}}{1 - \varepsilon_k} - \frac{\varepsilon_k \cdot \sigma \cdot T_k^4}{1 - \varepsilon_k} \right). \quad (60)$$

Substituting Equation (55) into Equation (52) yields

$$q_k = A_k \cdot \frac{\varepsilon_k}{(1 - \varepsilon_k)} \cdot (\sigma \cdot T_k^4 - Q_{out_k}). \quad (61)$$

Repeating the same process and substituting Equation (59) into Equation (52) gives

$$q_k = A_k \cdot (Q_{out_k} - \sum_{j=1}^N F_{k-j} \cdot Q_{out_j}). \quad (62)$$

The heat transfer rate for each element (q_k) can then be defined as either the energy supplied to the surface k by nonradiative means (convection or conduction to A_k), or as the net radiative loss from surface k resulting from radiation in the enclosure [Siegel and Howell, 1981]. Equations (61) and (62) represent the balance between the energy supplied by modes other than radiation and the net radiative loss.

In Equations (61) and Equation (62) the unknowns are q_k , T_k , and Q_{out_k} . These equations can be written for each of the N surfaces in the enclosure. By combining these two equations, Q_{out_k} can be eliminated giving N equations relating the unknown q_k 's and T_k 's. To further explain this process, consider three surfaces in an enclosure. Using Equations (61) and (62), the pair of equations give

$$\frac{q_1}{A_1} = \frac{\varepsilon_1}{(1-\varepsilon_1)} \cdot (\sigma T_1^4 - Q_{-out_1}), \quad (63)$$

$$\frac{q_1}{A_1} = q_{o,k} - F_{1-1} \cdot Q_{-out_1} - F_{1-2} \cdot Q_{-out_2} - F_{1-3} \cdot Q_{-out_3}, \quad (64)$$

$$\frac{q_2}{A_2} = \frac{\varepsilon_2}{(1-\varepsilon_2)} \cdot (\sigma T_2^4 - Q_{-out_2}), \quad (65)$$

$$\frac{q_2}{A_2} = Q_{-out_2} - F_{2-1} \cdot q_{o,1} - F_{2-2} \cdot Q_{-out_2} - F_{2-3} \cdot Q_{-out_3}, \quad (66)$$

$$\frac{q_3}{A_3} = \frac{\varepsilon_3}{(1-\varepsilon_3)} \cdot (\sigma T_3^4 - Q_{-out_3}), \quad (67)$$

$$\frac{q_3}{A_3} = Q_{-out_3} - F_{3-1} \cdot Q_{-out_1} - F_{3-2} \cdot Q_{-out_2} - F_{3-3} \cdot Q_{-out_3}, \quad (68)$$

Equations (63), (65), and (67) are solved for Q_{-out_k} in terms of T_k and q_k . The resulting expressions for Q_{-out_k} are then substituted into equations (64), (66), and (68). This substitution eliminates the Q_{-out_k} terms and using summation notation, the resulting equations can be expressed as

$$\sum_{j=1}^N \left(\frac{\delta_{kj}}{\varepsilon_j} - F_{k-j} \cdot \frac{1-\varepsilon_j}{\varepsilon_j} \right) \cdot \frac{q_j}{A_j} = \sum_{j=1}^N (\delta_{kj} - F_{k-j}) \cdot \sigma T_j^4. \quad (69)$$

When writing Equation (69) for each surface, k takes on the values of $1, 2, 3 \dots N$, and δ_{kj} is the Kronecker delta defined as

$$\delta_{kj} = \begin{cases} 1 & \text{when } k = j, \\ 0 & \text{when } k \neq j. \end{cases} \quad (70)$$

In order to solve for q_k , Equation (69) can be expanded as

$$\begin{aligned}
& \frac{-q_1}{A_1} \cdot F_{k-1} \cdot \frac{1-\varepsilon_1}{\varepsilon_1} - \frac{q_2}{A_2} \cdot F_{k-2} \cdot \frac{1-\varepsilon_2}{\varepsilon_2} - \dots + \frac{q_k}{A_k} \left(\frac{1}{\varepsilon_k} - F_{k-k} \frac{1-\varepsilon_k}{\varepsilon_k} \right) - \dots - \frac{q_N}{A_N} F_{k-N} \frac{1-\varepsilon_N}{\varepsilon_N} \\
& = -F_{k-1} \cdot \sigma \cdot T_1^4 - F_{k-2} \cdot \sigma \cdot T_2^4 - \dots + (1-F_{k-k}) \cdot \sigma \cdot T_k^4 - \dots - F_{k-N} \cdot \sigma \cdot T_N^4
\end{aligned} \tag{71}$$

Solving Equation (71) for q_k yields:

$$q_k = \frac{\varepsilon_k \cdot A_k}{1 - F_{k-k} \cdot (1 - \varepsilon_k)} \left[\sum_{j=1}^N \sigma \cdot F_{k-j} \cdot (T_k^4 - T_j^4) + \sum_{j=1}^N \left((1 - \delta_{kj}) \cdot F_{k-j} \cdot \frac{1 - \varepsilon_j}{\varepsilon_j} \right) \cdot \frac{q_j}{A_j} \right] \tag{72}$$

The term q_k is the net radiative loss from the k^{th} surface and is equal to $-q_{\text{radiation}_k}$ in Equation (34). Replacing $q_{\text{radiation}_k}$ with $-q_k$ in Equation (48) and solving for q_k gives the following equation [Marttila, 1999]:

$$q_k = \left[\begin{aligned} & -CAP_{k_i} \cdot A_k \cdot \frac{\partial T_k}{\partial t} + (T_a - T_k) \cdot A_k \cdot C_{ak} + (T_2 - T_k) \cdot A_k \cdot C_{kk} + q_{\text{netSolar}_k} \\ & + (T_{\text{Rain}} - T_k) \cdot A_k \cdot C_{\text{Rain}} \end{aligned} \right] \tag{73}$$

Numerical Method of Solution

RadTherm solves the energy balance equation simultaneously for convection, conduction, and radiation. The time-averaging Crank-Nicholson implicit finite difference scheme is applied to the final governing Equations (32), (72), and (73) for the numerical solution. ThermoAnalytics utilizes the time-averaging Crank-Nicholson method because it is one of the most versatile numerical methods available and the method is unconditionally stable and second order accurate in time and space [Marttila, 1999]. The following section describes the development of the final governing equations for transient

analysis. The solution to the temperature distribution is then described. A majority of the mathematics for this section have been eliminated due to its complexity. The complete development of the numerical solution is shown in Marttila 1999.

Transient Formulation

The T_k^4 terms in Equation (72) are linearized to make the radiation term proportional to a temperature difference rather than to the difference between two temperatures to the fourth power [Incropera and DeWitt, 1996] as shown

$$(T_k^4 - T_j^4) = (T_k + T_j) \cdot (T_k^2 + T_j^2) \cdot (T_k - T_j) \quad (74)$$

The Crank-Nicholson implicit finite difference method is applied to Equation (72), the net-radiation equation, to yield an equation for the heating rate (q_k) for each surface,

$$q_k = \frac{\varepsilon_k A_k}{1 - F_{k-k}(1 - \varepsilon_k)} \left[\sum_{j=1}^N \sigma F_{k-j} (T_k^2 + T_j^2) (T_k + T_j) \left(\frac{T'_k + T_k}{2} - \frac{T'_j + T_j}{2} \right) + \sum_{j=1}^N \left((1 - \delta_{kj}) F_{k-j} \frac{1 - \varepsilon_j}{\varepsilon_j} \right) \frac{q_j}{A_j} \right], \quad (75)$$

where T'_k and T'_j represent values from the current time step. The non-primed terms refer to values from the previous time step. Again, the Crank-Nicholson method is applied to Equation (32) and solving for T'_{k_i} yields an equation for the interior surface nodes:

$$T'_{k_i} = \frac{2 \cdot \frac{CAP_{k_i} \cdot A_k}{\Delta t} \cdot T_{k_i} + (T'_{k_{i-1}} + T_{k_{i-1}} - T_{k_i}) \cdot A_k \cdot C_{kk_{i-1}} + (T'_{k_{i+1}} + T_{k_{i+1}} - T_{k_i}) \cdot A_k \cdot C_{kk_i}}{2 \cdot \frac{CAP_{k_i} \cdot A_k}{\Delta t} + C_{kk_{i-1}} \cdot A_k + C_{kk_i} \cdot A_k}. \quad (76)$$

Equation (76) is written for each interior node T'_{k_2} through $T'_{k_{N_{Layers}}}$ associated with every surface element k . Once more, the Crank-Nicholson method is used on Equation (73) and yields an equation for the surface node k ,

$$q_k = \left\{ \begin{aligned} & -\frac{CAP_k \cdot A_k}{\Delta t} \cdot (T'_k - T_k) + \left[\left(\frac{T'_a + T_a}{2} \right) - \left(\frac{T'_k + T_k}{2} \right) \right] \cdot A_k \cdot C_{ak} \\ & + \left[\left(\frac{T'_{k_2} + T_{k_2}}{2} \right) - \left(\frac{T'_k + T_k}{2} \right) \right] \cdot A_k \cdot C_{kk} \\ & + \left[\left(\frac{T'_{Rain} + T_{Rain}}{2} \right) - \left(\frac{T'_k + T_k}{2} \right) \right] \cdot A_k \cdot C_{Rain} + \frac{q'_{netSolar_k} + q_{netSolar_k}}{2} \end{aligned} \right\} \quad (77)$$

To eliminate the term q_k , Equation (75) and Equation (77) are combined

$$\begin{aligned} & \left[-2 \cdot \frac{CAP_k \cdot A_k}{\Delta t} \cdot (T'_k - T_k) + (T'_a + T_a - T'_k - T_k) \cdot A_k \cdot C_{ak} + (T'_{2_k} + T_{2_k} - T'_k - T_k) \cdot A_k \cdot C_{kk} \right. \\ & \left. + (T'_{Rain} + T_{Rain} - T'_k - T_k) \cdot A_k \cdot C_{Rain} + q'_{netSolar_k} + q_{netSolar_k} \right] \\ & = \frac{\varepsilon_k A_k}{1 - F_{k-k}(1 - \varepsilon_k)} \left[\sum_{j=1}^N \sigma F_{k-j} (T_k^2 + T_j^2) (T_k + T_j) (T'_k + T_k - T'_j - T_j) \right. \\ & \left. + 2 \sum_{j=1}^N \left((1 - \delta_{kj}) F_{k-j} \frac{1 - \varepsilon_j}{\varepsilon_j} \right) \frac{q_j}{A_j} \right] \quad (78) \end{aligned}$$

For the surface layers, solving Equation (78) for T'_k yields

$$T'_k = \frac{\left[2 \cdot \frac{CAP_k \cdot A_k}{\Delta t} \cdot T_k + \frac{\varepsilon_k A_k}{1 - F_{k-k}(1 - \varepsilon_k)} \sum_{j=1}^N \sigma F_{k-j} (T_k^2 + T_j^2) (T_k + T_j) (T'_j + T_j - T_k) \right.}{\left[2 \cdot \frac{CAP_k \cdot A_k}{\Delta t} + \frac{\varepsilon_k A_k}{1 - F_{k-k}(1 - \varepsilon_k)} \sum_{j=1}^N \sigma F_{k-j} (T_k^2 + T_j^2) (T_k + T_j) + \right.}, \quad (79)$$

$$\left. + (T'_a + T_a - T_k) \cdot A_k \cdot C_{ak} + (T'_{k_2} + T_{k_2} - T_k) \cdot A_k \cdot C_{kk} + \right. \\ \left. (T'_{Rain} + T_{Rain} - T_{Rain}) \cdot A_k \cdot C_{Rain} + Q'_{netSolar_k} + Q_{netSolar_k} \right. \\ \left. - \frac{2\varepsilon_k A_k}{1 - F_{k-k}(1 - \varepsilon_k)} \sum_{j=1}^N \left((1 - \delta_{kj}) F_{k-j} \frac{1 - \varepsilon_j}{\varepsilon_j} \right) \frac{q_j}{A_j} \right] \\ \left[C_{kk} \cdot A_k + C_{ak} \cdot A_k + C_{Rain} \cdot A_k \right]$$

which is the equation that is used in the numerical solution [Marttila, 1999].

Computation of Temperature Distribution

To determine the temperature distribution through each element, the discretized governing Equations (79) and (76) are applied to each thermal node. This application results in N equations and $2N$ unknowns for each T_k and q_{k_i} for all elements. For the initial time step the matrix is solved for steady state. A guess for T_k is made which leaves N equations and N unknowns. Gauss elimination with partial pivoting is used as a preconditioner to solve the resulting system of equations. The advantage of using Gauss elimination is the equations are solved simultaneously, which allows the node temperatures to converge quickly from initial values to the temperatures that satisfy the governing equations. After the direct solution (i.e. Gauss solution) is complete, the solution proceeds with an iterative method, Successive Overrelaxation (SOR). The governing Equations (75), (76), and (79) are evaluated for each node using the SOR iteration. SOR is a technique that can be used to accelerate any iterative procedure by

making corrections to particular values before the next iteration [Tannehill et al, 1997]. Once all node temperatures have converged, the iteration is complete. The solution then advances to the next time step and repeats the above procedure [Marttila, 1999].

CHAPTER 3

METHODOLOGY

One of the main factors that determine the radiation exchange between surfaces is the view factor. A view factor is defined as the fraction of radiation leaving a surface x , which is then intercepted by surface y . The view factor depends on three main factors: size of surface, shape of surface, and orientation of surface. ThermoAnalytics has developed a ray-tracing algorithm that computes the view factors for each element and is discussed in the following section.

In this chapter a simple model is introduced to explain the motivation for deleting elements by the use of view factors. The effects of shading and a look at longwave and shortwave radiation exchange between surfaces is investigated. By using simple models these areas are explored and the use of view factors is discussed. The last part of the chapter describes the analytical approach used to delete elements by the use of view factors.

Ray-Tracing Algorithm

Radiation view factors are used in ThermoAnalytics' software to model the radiation heat rate from each element to every other element in the model and to the environment [ThermoAnalytics, 2000]. This allows the terrain model to include the full effects of solar shadowing, multiple reflections, and re-radiation of geometric features. Each triangular surface element (or facet) of the terrain has a view factor to the background,

sky, and other elements [www.thermoanalytics.com]. Note that each elements surface is planar. ThermoAnalytics computes both radiation view factors and solar projected (apparent) areas by the use of a ray tracing algorithm which casts rays from each element.

The geometric view factors are calculated initially at the start of a RadTherm/RT run using the ray trace technique. This is done first and only once for a given terrain model. These view factors are based only on the geometry of the terrain file and are calculated prior to the thermal solution. A view factor file (.vfs) is created, which is then used for the calculation of the thermal solution.

The ray trace algorithm is then used during the thermal solution, at each time step, to calculate the view area for each surface element (Equation(44)). Rays are cast from the centroid of each element to the sun, which accounts for the amount of direct solar radiation that reaches each surface element. At the specified geographic location and time, the positions of the primary source of irradiation (the sun) is defined by two angles, the solar azimuth and zenith angles, that are updated at each time step. The angle from the vertical is the zenith angle, and the angle measured counter-clockwise in the horizontal plane from the reference direction is the azimuth angle as shown in Figure 17.

To better explain the ray tracing technique consider two parallel plates (Figure 18a). A unit hemisphere is created at the centroid of each element. Figure 18a illustrates a single ray being cast and intercepted by another element.

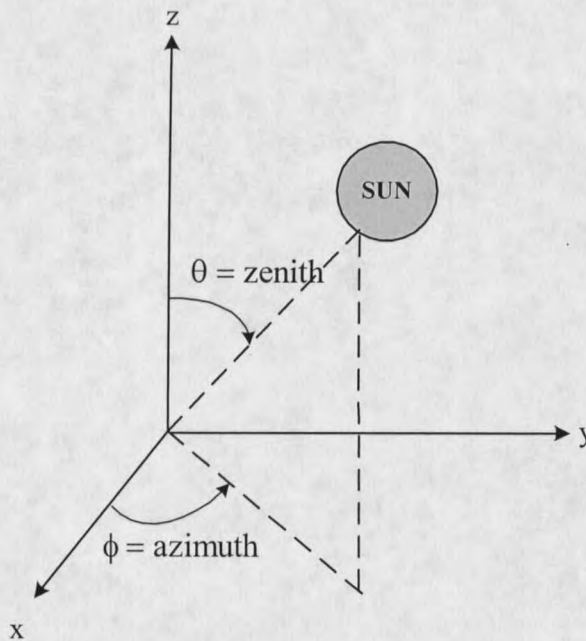


Figure 17. The position of the sun is specified in terms of the zenith and azimuth angles, θ and ϕ , respectively.

A set number of rays, chosen by the user, are cast from each element centroid to determine the view factor to many of the other elements that are visible from the element centroid. The hemisphere is then equally subdivided based on the number of rays to be cast. These subdivisions are a function of the angle θ_i , which is defined so the population of rays cast will be the greatest near the element normal [www.thermoanalytics.com]. The number of rays that intersect another element determines the strength or weight of the view factor. Figure 18b shows multiple rays cast from an element by the use of the ray trace algorithm to determine its view factor relationships to all other elements in the enclosure.

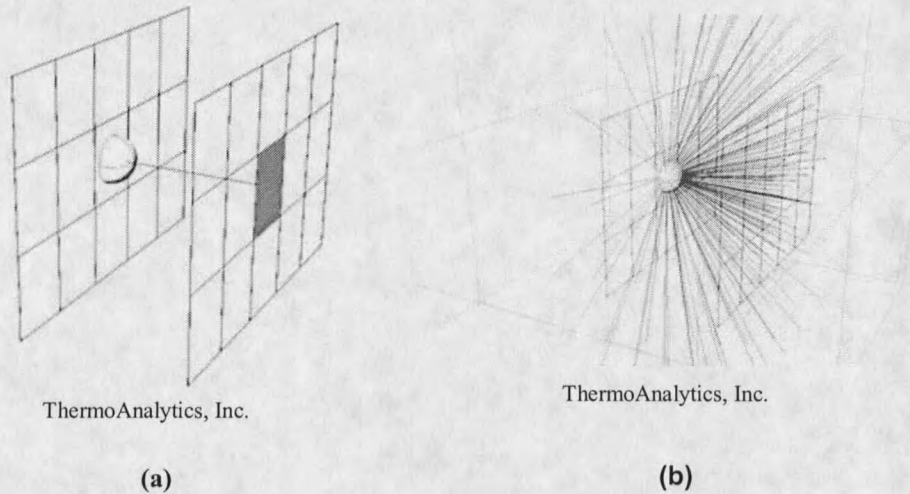


Figure 18. (a) A single ray cast from an element for ThermoAnalytics Ray Trace Algorithm. (b) Multiple rays cast from an element for ThermoAnalytics Ray Trace Algorithm. (Figures from ThermoAnalytics, Inc. website)

View Factor Rays

The number of rays cast from the element centroid is determined by the user-assigned “View Factor Rays”. By adjusting the number of rays cast from each element, the speed and accuracy of the view factor calculation is determined. Five different settings are available in RadTherm. Selecting setting 1 cast 512 rays from each element. This gives the fastest solution time with the least amount of accuracy. Setting 5 cast 4608 rays from each element which gives the most accurate solution but the computation time is much slower. [ThermoAnalytics, 1999]. Table 1 shows the different settings for the view factor rays.

Table 1. View factor ray setting in RadTherm/RT.

Setting	Rays Cast from each Element
1	512 (fastest)
2	1152 (default)
3	2048
4	3200
5	4608 (most accurate)

View Factor Subdivisions

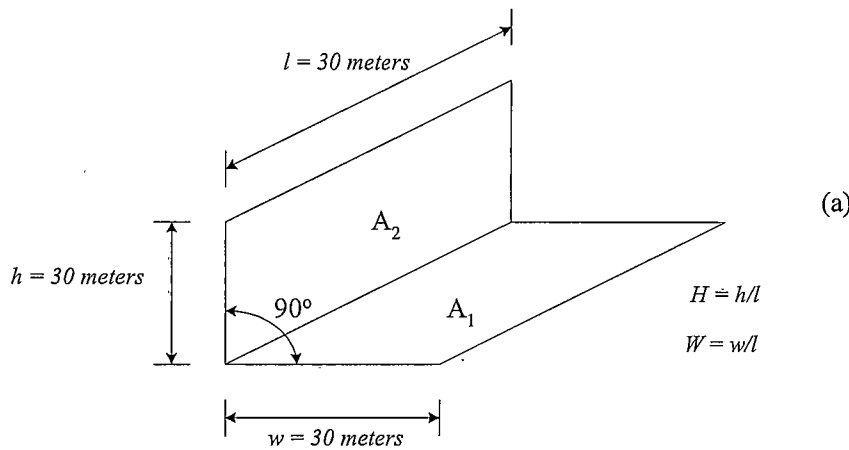
Another user-defined setting is the number of subdivisions per element. Each element has one to five subdivisions and rays are cast from the centroid of each subdivision. The amount of subdivisions increase the number of rays cast from each element from one to five times the view factor ray setting. As the number of subdivisions increase, the thermal run time and view factor calculations can increase significantly [ThermoAnalytics, 1999].

Apparent Area Resolution

Apparent Area Resolution changes the number of rays that are cast from the centroid of each element to the sun used for calculating apparent area. At the lowest setting only a single ray is cast from the centroid of each element to the sun and at the highest setting 64 rays are cast.

Hand Calculation for View Factor

A simple model is used to compare the results of the view factor using the ray trace algorithm and a hand calculation for the view factor. Figure 19a illustrates the simple model that consists of two 30-meter by 30-meter rectangles at an angle of ninety degrees to each other.



$$F_{1-2} = \frac{1}{\pi \cdot W} \left(W \cdot \tan^{-1} \frac{1}{W} + H \cdot \tan^{-1} \frac{1}{H} - \sqrt{H^2 + W^2} \cdot \tan^{-1} \frac{1}{\sqrt{H^2 + W^2}} + \frac{1}{4} \cdot \ln \left\{ \frac{(1+W^2)(1+H^2)}{1+W^2+H^2} \left[\frac{W^2(1+W^2+H^2)}{(1+W^2)(W^2+H^2)} \right]^{W^2} \left[\frac{H^2(1+H^2+W^2)}{(1+H^2)(H^2+W^2)} \right]^{H^2} \right\} \right) = 0.20 \quad (b)$$

Figure 19. (a) A simple model used for the hand calculation of the view factor. The model consists of two finite rectangles of the same length, having one common edge at an angle of 90° to each other. (b) The equation used to calculate the view factor from 1 to 2. (After Siegel and Howell, Figure 16 in the Appendix)

The equation in Figure 19b was used to calculate the view factor from 1 to 2. The result using the equation is $F_{1-2} = 0.20$. The identical model in Figure 19a was created in

RadTherm/RT consisting of two elements, A_1 and A_2 . The view factors calculated for elements 1 and 2 using the ray trace algorithm are shown in Table 2. Element 3 is the default terrain node and element 4 is the sky node.

Table 2. View factor ray setting in RadTherm/RT.

Emitter #	Receiver #	View Factor
1	2	0.1913
1	4 (Sky node)	0.8087
View factor sum:		1.00

Emitter #	Receiver #	View Factor
2	1	0.1913
2	3 (Default terrain node)	0.3089
2	4 (Sky node)	0.4998
View factor sum:		1.00

The hand calculation for the view factor was 0.20 and the value computed using the ray trace algorithm was 0.1913. The ray trace algorithm provides similar results to the hand calculation for the view factor.

The Process to Delete Elements

The motivation for this thesis came from the initial results of the previous section. Basic models were created to determine the importance of the radiation exchange between surfaces. The results prompted the idea that view factors could be used to eliminate elements that have no influence on the temperature of an area of interest. Recall there is a limitation on the size of the thermal map for which RadTherm/RT can compute the thermal solution. If a methodology is created to reduce the number of elements in a thermal map for RadTherm/RT, larger regions of interest can be modeled in

the software. Therefore, the task was to determine a computational approach for reducing the number of elements in the map without losing accuracy or significantly reducing the region of interest. Currently, Bozeman Pass is being used as a test bed for traveler or maintenance weather information. This is the primary region of focus for this thesis and the goal is to increase the length of the highway that can be modeled and/or speed runtime of forecast models.

The current map used for Bozeman Pass was first trimmed down and then remeshed with increasing element size as distance from the interstate increased. Based only on intuition using a topographic map, there is no precise approach for reducing the modeled area. The user must estimate using important ridgelines and create boundaries based on the assumption that the remaining terrain is significant for the thermal solution, and terrain removed was not. A second source of inaccuracy is the process by which the map is remeshed. The procedure for this is to keep the original dimension of the map and combine elements that are further away from the elements of interest, i.e., the highway. This is a misrepresentation of the elements because it involves combining elements of different terrain types, such as water, trees, rock, soil, etc. In addition to these downfalls, it can be a time-consuming and labor-intensive process to generate the coarsened map.

An alternative approach that is more efficient and more accurate is presented here. In order to complete this task, the number of surface elements in the map will be reduced by eliminating elements using the view factor information relating each element to all other elements. The main objective is to ensure that the removed elements do not include

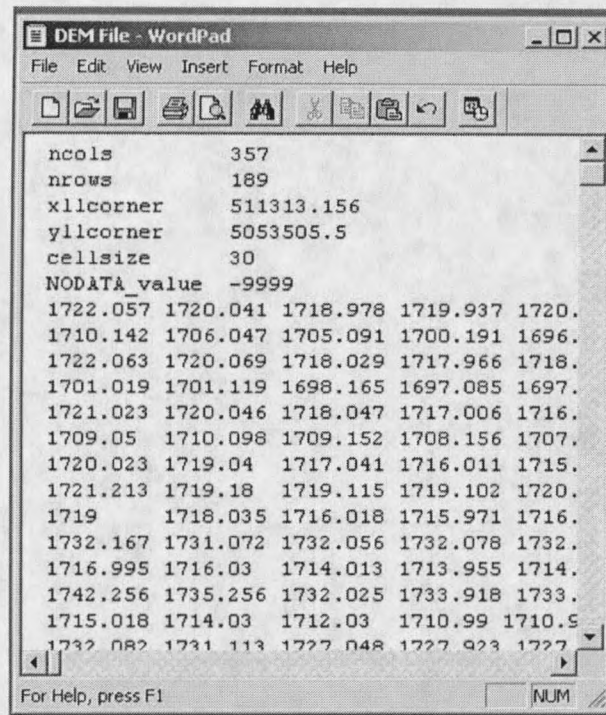
terrain that is pertinent to the temperature of the highway. In the following section, the procedure used to delete the elements is discussed.

Initial Setup

In order to create a thermal model in RadTherm/RT for a particular region of interest, two files are required. The first file contains Digital Elevation Map (DEM) data, and the other provides land cover data used to identify vegetation and road positions in the model. The information is provided in ASCII format and used to create the maps for RadTherm/RT. Montana State University's Geographic Information and Analysis Center (GIAC) produced the ASCII files for elevation, road and vegetation maps using ARC/INFO geographic information system (GIS) software [Adams and Curran, 1999].

Step 1a - Digital Elevation Map Files (.asc)

Elevation data was compiled from a United States Geological Survey (USGS) 30-meter DEM. Digital Elevation Model (DEM) is the terminology adopted by the USGS to describe data sets in a digital raster form. The standard DEM consists of a regular array of elevations cast on a designated coordinate projection system. The orientation of data is by columns and rows. Each column contains a series of elevations ordered from north to south with the order of the columns from west to east. These data were collected as part of the National Mapping Program. The entire set of data was then projected on the Universal Transverse Mercator (UTM) coordinate system. The grids are then exported as ASCII files. An example data file is shown in Figure 20.



```

DEM File - WordPad
File Edit View Insert Format Help

ncols      357
nrows      189
xllcorner  511313.156
yllcorner  5053505.5
cellsize   30
NODATA_value -9999
1722.057 1720.041 1718.978 1719.937 1720.
1710.142 1706.047 1705.091 1700.191 1696.
1722.063 1720.069 1718.029 1717.966 1718.
1701.019 1701.119 1698.165 1697.085 1697.
1721.023 1720.046 1718.047 1717.006 1716.
1709.05  1710.098 1709.152 1708.156 1707.
1720.023 1719.04  1717.041 1716.011 1715.
1721.213 1719.18  1719.115 1719.102 1720.
1719      1718.035 1716.018 1715.971 1716.
1732.167 1731.072 1732.056 1732.078 1732.
1716.995 1716.03  1714.013 1713.955 1714.
1742.256 1735.256 1732.025 1733.918 1733.
1715.018 1714.03  1712.03  1710.99 1710.9
1732.082 1731.113 1727.048 1727.923 1727.

```

For Help, press F1

Figure 20. Example of ASCII input file for elevation data created from USGS 30 meter grid.

Header information is included at the top of the file, containing the number of rows and columns, the x-y origin of the lower left corner of the grid, the cell size, and the value used for “no data” in the grid. The 357 x 189 grid for this map in Figure 20 represents a 10680-meter x 5640-meter terrain map.

Step 1b - Terrain File (.asc)

The vegetation information was a product of the Montana Gap Analysis project (MT-GAP). This project created a classification system for existing vegetation in Montana. Part of this classification system is shown in Table 3.

Table 3. Land cover classification system used by Montana Gap Analysis Project.

Land Cover Class Code	Land Cover Class
2010	Agricultural Lands - Dry
2020	Agricultural Lands - Irrigated
3110	Altered Herbaceous
3130	Very Low Cover Grasslands
3150	Low/Moderate Cover Grasslands
3170	Moderate/High Cover Grasslands
3180	Montana Parklands and Subalpine Meadows
3210	Mixed Mesic Shrubs
3300	Mixed Xeric Shrubs
3310	Salt-Desert Shrub/Dry Salt Flats
3350	Sagebrush
3510	Mesic Shrub-Grassland Associations
3520	Xeric Shrub-Grassland Associations
4000	Low Density Xeric Forest
4140	Mixed Broadleaf Forest
4203	Lodgepole Pine
4205	Limber Pine
4206	Ponderosa Pine
4212	Douglas-fir
4214	Rocky Mountain Juniper
4223	Douglas-fir/Lodgepole Pine
4260	Mixed Whitebark Pine Forest
4270	Mixed Subalpine Forest
4290	Mixed Xeric Forest
4300	Mixed Broadleaf and Conifer Forest
6110	Conifer Riparian
6120	Broadleaf Riparian
6130	Mixed Broadleaf and Conifer Riparian
6200	Graminoid and Forb Riparian
6300	Shrub Riparian
6400	Mixed Riparian
7300	Rock

Montana State University's Geographic Information and Analysis Center (GIAC) produced an ASCII file for vegetation using ARC/INFO software, which is a GIS software program. The data set is an ARC/INFO 90 meter land cover grid (raster file) covering the state of Montana. A raster file contains information that is directly mapped

to the display grid. The specifications of the vegetation grid are set to spatially match the terrain grid: i.e. the cell size (30-meter), the grid origin coordinates, and the number of rows and columns. The roads and mile markers come from a combination of data from the Montana Department of Transportation and the major highway layer from NRIS. The combined vegetation and road data was then converted to a grid and exported as an ASCII file. The ASCII file shown in Figure 21 has been converted from the land cover classification to RadTherm/RT part numbers. For example, part 4 represents a terrain element composed of short grass and part 1 is the interstate.

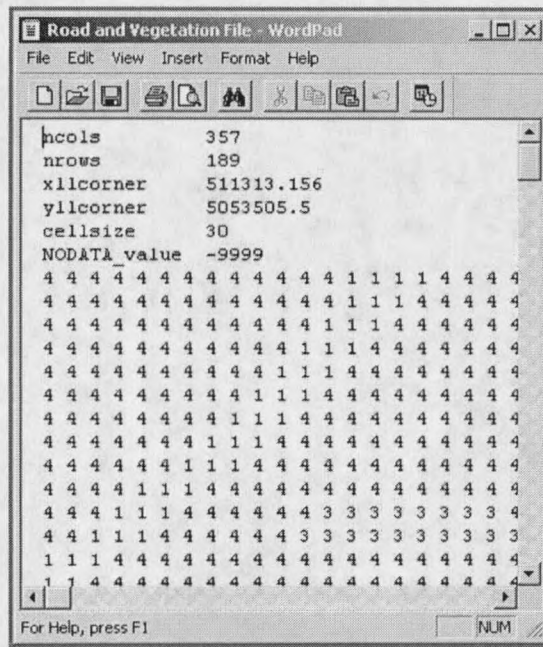


Figure 21. Example of the ASCII input file for the terrain and road information. This file has been converted from the land classification to RadTherm/RT Part ID's. Note that 1 represents a part ID such as the road.

Step 2 - Trimming the Original Map

The main objective is to delete unnecessary elements from a map using the view factor information of each highway element. This information is found in the view factor file (.vfs) initially generated at the start of a run in RadTherm/RT. Once a RadTherm/RT project is created, the model is executed and the first item produced is the view factor file. The software continues to run and the final result is the thermal solution to the terrain map. If a RadTherm/RT project is too large, a limited amount of disk space or computer memory may cause the project to fail. There are two possible outcomes if the RadTherm/RT project contains too many elements. First, the computer can crash and neither the view factor file nor the thermal solution will be completed. Second, the view factor file can complete successfully and then the computer stalls and terminates on the thermal solution. Therefore, RadTherm/RT may produce the view factor file even when the thermal solution cannot be solved. As a result, the largest number of surface elements that can produce the view factor file must be determined. The view factor file is explained in more detail later in this chapter.

A map of Bozeman Pass with dimensions of 16.09 by 27.36 kilometers was used as a starting point to create the view factor file. This map contained 831,174 surface elements and was too large to process even the view factor file. By using a PERL script the map can be cropped by specifying the north, east, south and west UTM coordinates [McKittrick, personal communications]. The command line to execute the script is shown in Figure 22. Note that the script uses both the vegetation and elevation files.

```
$ gis_manip -S[UTM coord] -N[UTM coord] -E[UTM coord] -W[UTM coord] [filename]
```

Note: filename is without _dem.asc or _veg.asc extension

Figure 22. Command line to execute the gis_manip PERL script.

As an example, the first iteration reduces the map to a size of 8.05 by 27.36 kilometers and contains 544,810 surface elements. After over three hours of computation time, the view factor file was only 7% complete. Based on experience, at this point the run was killed, realizing that it would never reach completion. For the next iteration, the map was cut down to 387,340 surface elements. The view factor file after seven hours of run time was only 27% complete and once again the view factor file could not be generated. The final iteration left a map with 327,610 surface elements. A project of this size produced a completed view factor file after almost sixteen hours but crashed on the thermal solution. This was used as the target number of elements to begin the process of deleting surface elements. Table 4 summarizes the findings of this iterative process.

Table 4. Summary of the iterations for trimming a map of Bozeman Pass to a size that produces the view factor file (.vfs).

	# Elements	View Factor File
Original Map	831,174	Failed
Iteration #1	544,810	Failed
Iteration #2	387,340	Failed
Final Iterations	327,610	Complete

There are several items that must be set in RadTherm/RT once a project has been created with less than 330,000 surface elements. First, an appropriate weather file is selected, and then the same start and end times are set on the GUI. This ensures the thermal solution is not calculated, but the view factor file is still generated. Each part in the project is set to type terrain but no other information has to be specified. Figure 23 shows the steps described above.

Finally, the RadTherm/RT project is executed and the view factor file is created. This information is only geometry dependent and therefore does not change at every time step. The file is created once and can be used in successive runs as long as the geometry of the map does not change. Figure 24 highlights the steps of the process to this point.

Step 3 - RadTherm/RT Project Files (.tdf)

Once a project is created in RadTherm, the file is saved as a .tdf file. The .tdf file is the native RadTherm file format, which stores both the geometry for the thermal model and the thermal analysis results. The RadTherm file is a binary file format that is platform independent, so that files written on one machine can be easily read on another machine that uses a different operating system.

In this case, a RadTherm project is created using the ASCII formatted vegetation and elevation file, filename_dem.asc and filename_veg.asc, respectively. First, the "File-open" command on the Graphical User Interface (GUI) prompts the user for the DEM and vegetation file.

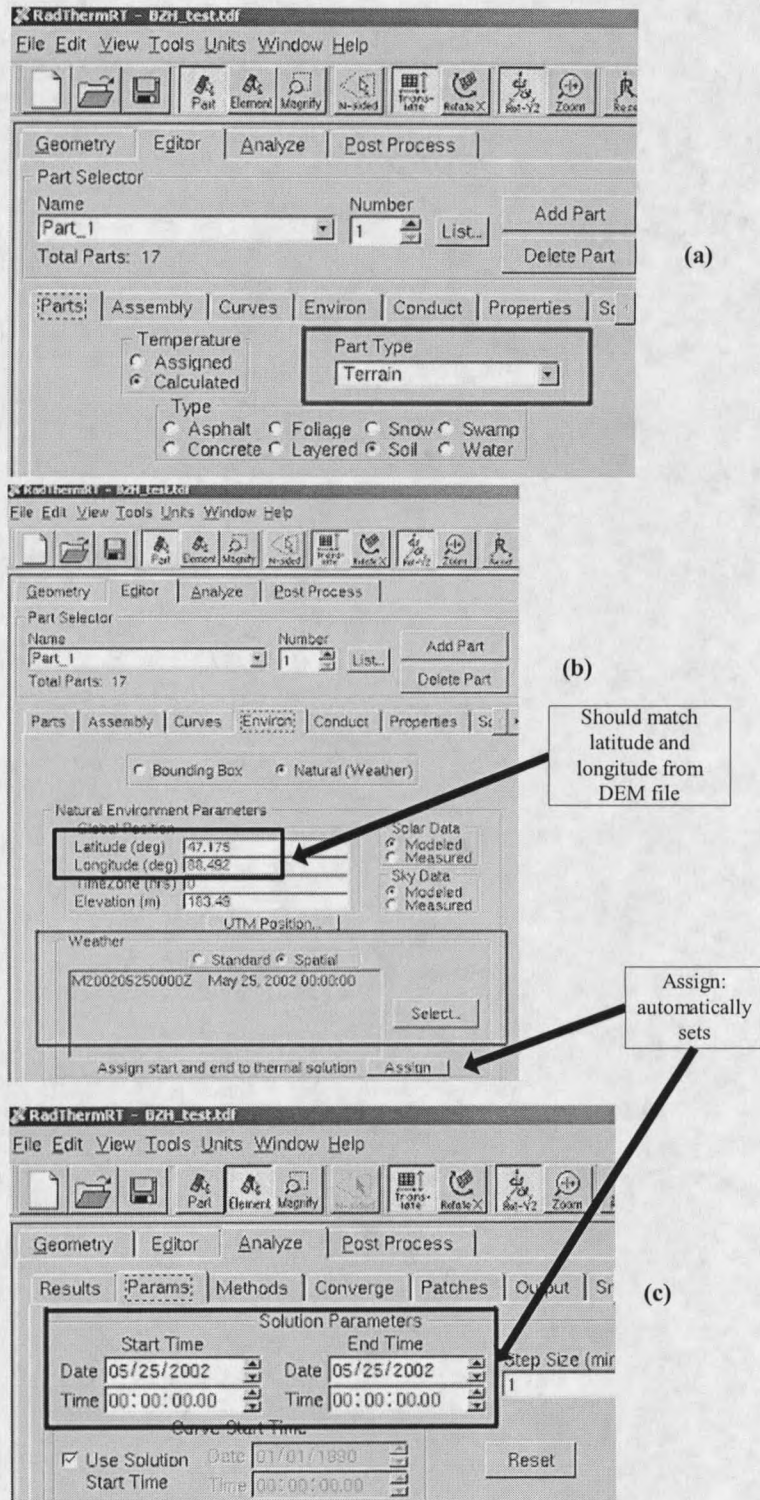


Figure 23. (a) Set each part to type terrain. (b) Specify a weather file. (c) Set the same start and end solution times.

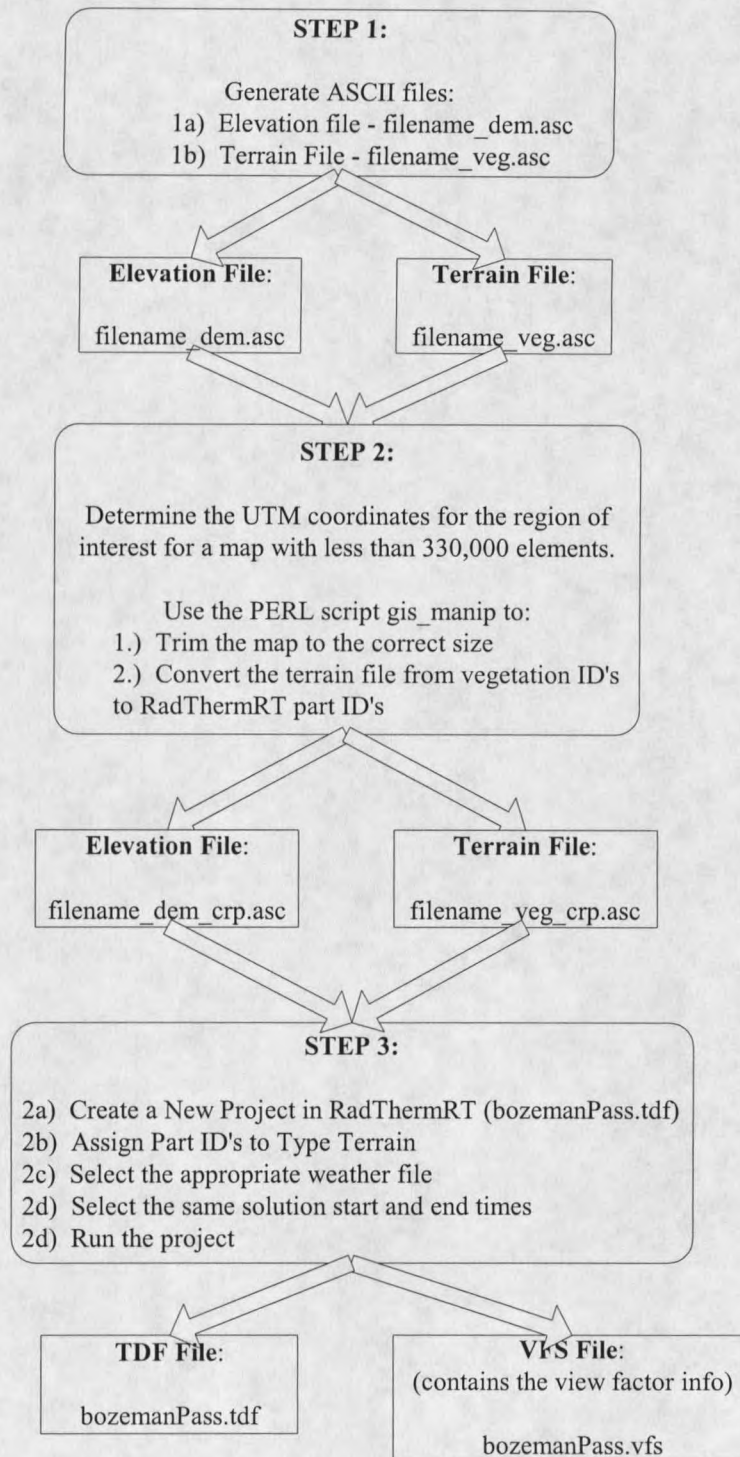


Figure 24. Flow chart of Steps 1 through 3. Step 1 is generating the ASCII files. Step 2 determines the UTM coordinates for a map with less than 330,000 elements. Step 3 creates a new project in RadTherm/RT.

Figure 25 shows a simple terrain model in RadTherm/RT composed of 3,612 elements and two parts.

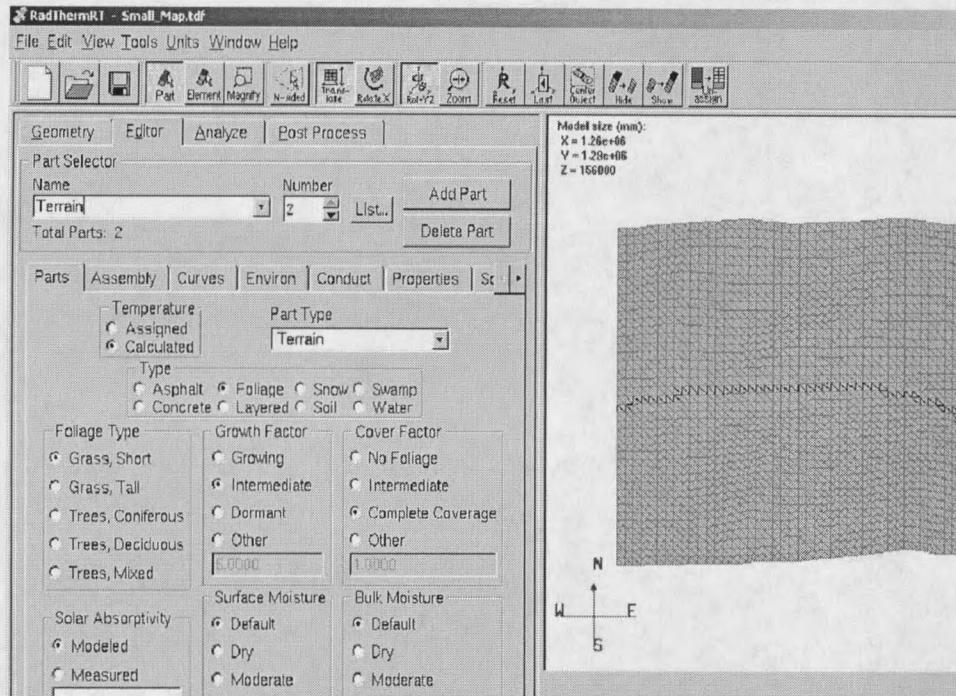


Figure 25. Example of a RadTherm/RT project composed of 2 parts.

Step 4 - Element Attribute File

Figure 26 describes the procedure to retrieve the element attribute information. Certain attributes of every element are pertinent to determine which ones to delete. A C/C++ code was created with the help of several C/C++ functions written by ThermoAnalytics. ThermoAnalytics created a database of C/C++ functions that can be used to modify or obtain information from a RadTherm/RT project (.tdf file). This allows a user with C/C++ background to incorporate these function calls in order to

access necessary information from the project, such as elevation of a particular element or the apparent area at a particular time step.

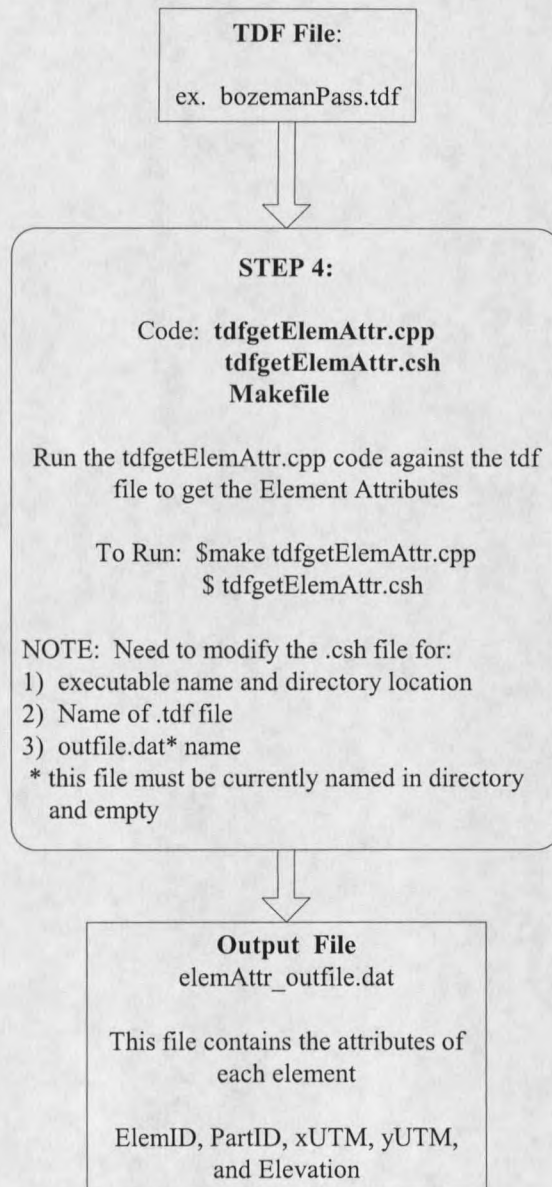
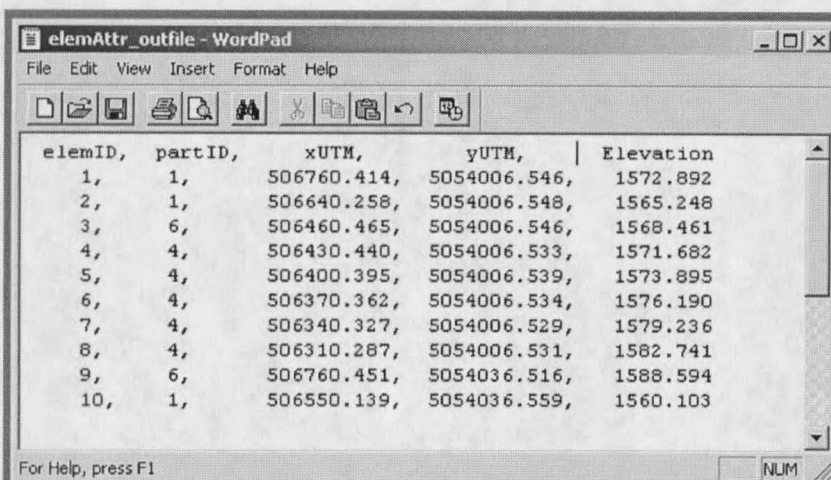


Figure 26. Step 4 of the procedure. A C/C++ code is used to create a text file with the element attributes: Element ID, Part ID, UTM coordinates, and elevation.

This database for the C/C++ functions is called the TDF I/O library. The tdf file is the main database file that is used by ThermoAnalytics' thermal analysis software. A code was created to retrieve the element ID, part ID, UTM coordinates, and elevation for each element. This information is then placed in a text file as shown in Figure 27.



elemID,	partID,	xUTM,	yUTM,	Elevation
1,	1,	506760.414,	5054006.546,	1572.892
2,	1,	506640.258,	5054006.548,	1565.248
3,	6,	506460.465,	5054006.546,	1568.461
4,	4,	506430.440,	5054006.533,	1571.682
5,	4,	506400.395,	5054006.539,	1573.895
6,	4,	506370.362,	5054006.534,	1576.190
7,	4,	506340.327,	5054006.529,	1579.236
8,	4,	506310.287,	5054006.531,	1582.741
9,	6,	506760.451,	5054036.516,	1588.594
10,	1,	506550.139,	5054036.559,	1560.103

Figure 27. Example of the text file that contains the element attributes. This is the output file from Step 4.

Step 5 - View Factor File (.vfs)

Figure 28 summarizes the procedure for Step 5. The view factor file (.vfs) is an hdf data file that contains the view factor information for each element. RadTherm/RT requires the use of the HDF view factor files, but the ASCII files are easier to read and/or modify. A C/C++ code, called vf2asc, was written by ThermoAnalytics to read the view factor data from the specified binary file (HDF or .vfs file) and write the data to an ASCII

file. The computation time for a file that contained 330,00 surface elements was found at most to be five minutes.

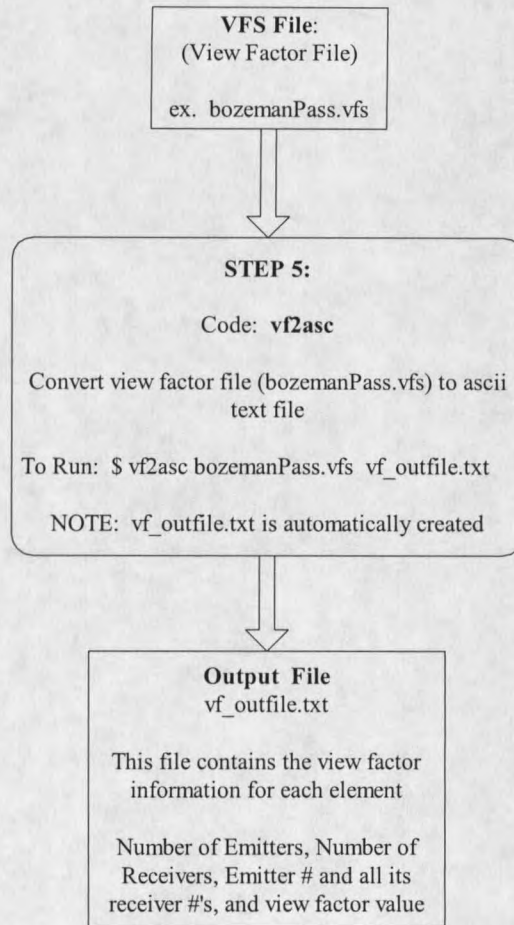


Figure 28. Flow chart for Step 5. This step converts the view factor file to an ASCII text file.

Figure 29 demonstrates the command line to run this executable.

```
$ vf2asc [-p] infile.vfs [outfile.txt]
```

Figure 29. The command line to execute the vf2asc utility.

Referring to Figure 29, vf2asc is the utility executable, -p is optional and when specified, 10 extra digits of precision are used in the output file. The view factor input filename is infile.vfs and outfile.txt is the optional output file name. If not present, the output will be written to a file named infile.vfs.ASC (input file name with ".ASC" appended).

This output file contains the view factor information for each element: the total number of emitters (or elements), the number of receivers, the emitter number and all its receiver numbers, and the view factor value. A sample file is shown in Figure 30.

```

View Factor File-Example - WordPad
File Edit View Insert Format Help

Number of Emitters: 24
Emitter: 1      Number of Receivers: 5
Emitter Receiver View Factor
1      17      0.0010
1      20      0.0010
1      23      0.0005
1      24      0.0005
1      26      0.9971
      VF Sum: 1.0000
Emitter: 2      Number of Receivers: 6
Emitter Receiver View Factor
2      17      0.0010
2      18      0.0010
2      20      0.0015
2      21      0.0010
2      23      0.0010
2      26      0.9946
      VF Sum: 1.0000
Emitter: 3      Number of Receivers: 7
Emitter Receiver View Factor
3      17      0.0020
3      18      0.0029
3      20      0.0015
3      21      0.0015
3      23      0.0010
3      24      0.0010
3      26      0.9941
      VF Sum: 1.0000
  
```

Figure 30. This is the output file from Step 5. Example of the view factor file produced from the vf2asc executable.

Step 6 – Code to Delete Elements

Step 6 takes as input the two files developed in Step 4 and Step 5 and produces a text file containing the number of elements to delete and the associated element ID's. This step is performed with a code written in C/C++ and is shown in the Appendix. Figure 31 shows an overview of Step 6.

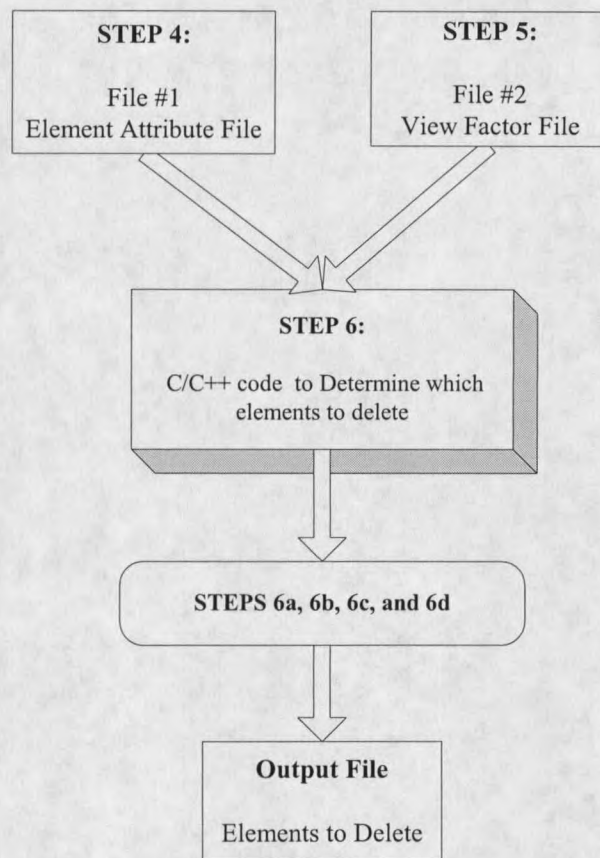


Figure 31. Flow chart for an overview of Step 6. The Element Attribute text file and View factor text file are used as input into Step 6. The output from Step 6 is a text file containing the elements to delete.

Step 6a – Separate Elements. The first stage of this code separates all the elements into two categories, surrounding elements and elements of interest. Figure 32 shows the overview of this step.

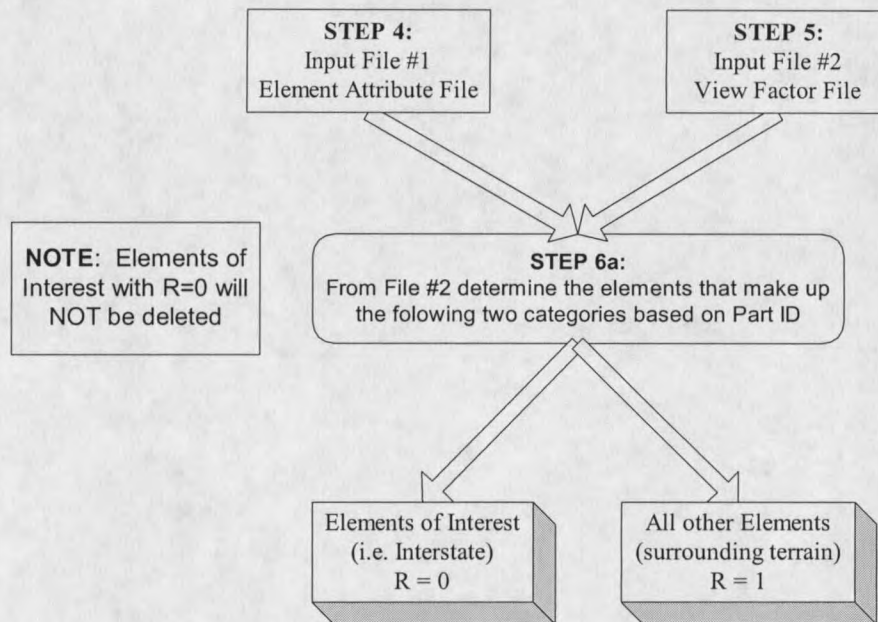


Figure 32. Flow chart for Step 6a. This step separates all the surface elements for the terrain map into two categories. The first category is the elements of interest with a rank of 1 and the second category is the surrounding elements with a rank of 2.

As an example, consider the development of a model for the highway temperature for Bozeman Pass. All the elements making up the highway are considered elements of interest and receive a rank of 0. These are the elements of primary interest for surface temperature forecasts. The remaining elements are called the surrounding elements and are given a rank of 1. Figure 33 shows several elements in a simple demonstration terrain model and their specified rank.

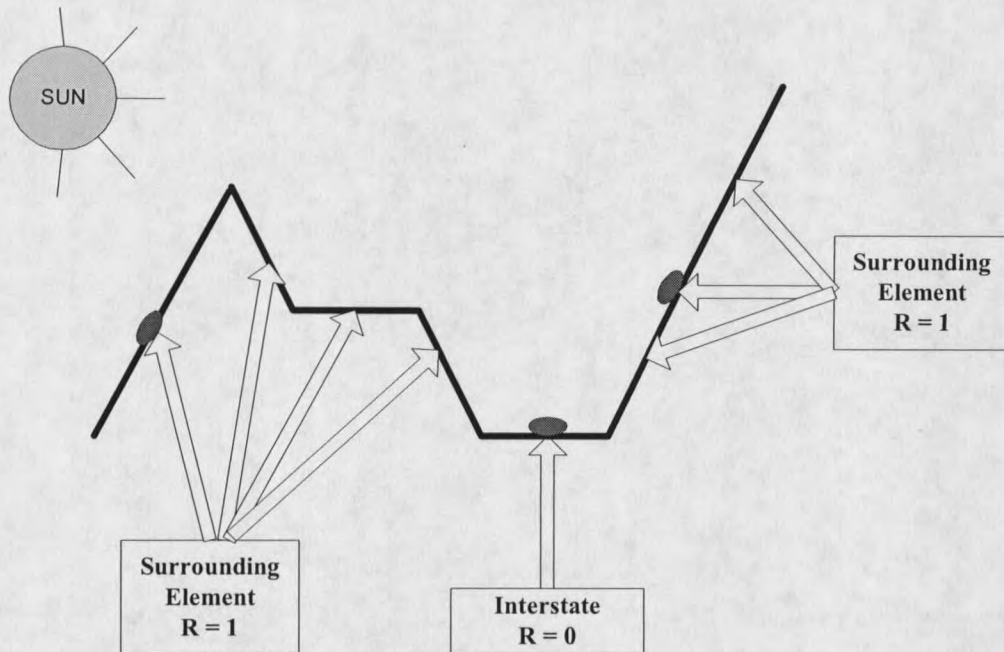


Figure 33. A simple demonstration terrain model to describe the ranking after Step 6a.

The ranking system is a hierarchy and used to determine which elements fall into the different categories described in steps 6a, 6b, 6c, and 6d. A filter is used to best describe this system. The specified elements are filtered and the elements that pass through have failed the criteria and their rank remains the same. The elements that remain in the filter have passed the criteria and their ranks are increased. After steps 6a, 6b, 6c, and 6d the elements that remain with a rank of 1 are the elements to delete. These are the elements that have failed every criterion in each step.

Step 6b – Apply Distance Criteria. The outcome of this entire process is to delete elements from the thermal map. After these elements are deleted, there are holes left in the geometry of the map. In order not to lose the shape of the terrain surrounding the

highway, a distance criterion is set. This states that all elements within a specified distance of the elements of interest (i.e., the interstate) are given an increase in rank. The user can change the specified distance.

Elements with a rank of 1 are placed into the filter. The elements that fall out have failed the criteria and their rank remains at 1. The elements that remain are the ones that have passed the criteria and their ranks are increased to 2. A simple demonstration terrain model is used in Figure 34 to demonstrate the ranking at this point.

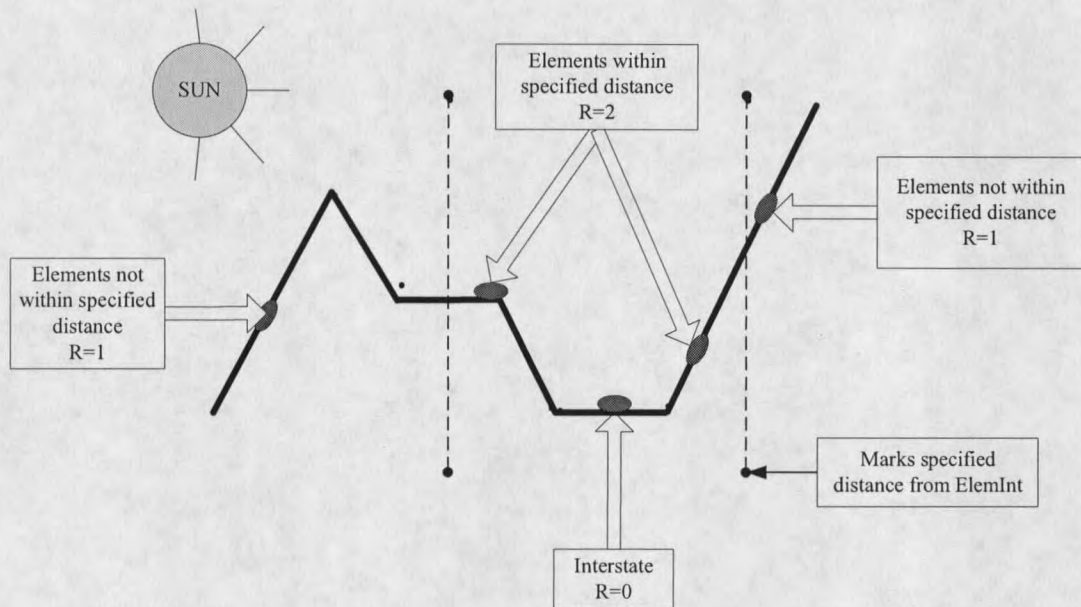


Figure 34. Simple demonstration terrain model to describe the ranking after Step 6b.

Step 6c – View Factors to Elements of Interest. This step applies the view factor theory developed earlier in this chapter. To recall, the view factor file contains each element and each element with which it directly exchanges radiation. This step

determines which elements directly exchange radiation with the elements of interest. To complete this task, the view factor file is read line by line to determine the following: 1) If the emitter is an element of interest, or 2) if the receiver is an element of interest or 3) the emitter or receiver is not an element of interest. Figure 35 illustrates the three options.

Table 1: Emitter: 1, Number of Receivers: 5

Emitter	Receiver	View Factor
1	17	0.0010
1	20	0.0010
1	23	0.0005
1	24	0.0005
1	26	0.9971
VF Sum:		1.0000

Table 2: Emitter: 2, Number of Receivers: 6

Emitter	Receiver	View Factor
2	17	0.0010
2	18	0.0010
2	20	0.0015
2	21	0.0010
2	23	0.0010
2	26	0.9946
VF Sum:		1.0000

Table 3: Emitter: 3, Number of Receivers: 7

Emitter	Receiver	View Factor
3	17	0.0020
3	18	0.0020
3	20	0.0015
3	21	0.0015
3	23	0.0015
3	24	0.0015
3	26	0.9945
VF Sum:		1.0000

For Help, press F1

Figure 35. Illustrates the three options for Step 6c. The emitter is an element of interest, or the receiver is an element of interest, or neither the emitter nor receiver is an element of interest.

If the emitter or receiver is an element of interest, the value of the view factor is compared to the view factor criterion. The value of the view factor criterion is discussed

in Chapter 4. Basically, if an element directly exchanges radiation with an element of interest, the view factor value must be great enough that it strongly influences the element of interest. Otherwise, the element is not important in the energy exchange and should be deleted. An element's rank is increased if its view factor value is greater than the view factor criterion (e.g. 0.0005) to an element of interest. An example of a simple demonstration terrain and the rank of each element after Step 6c are shown in Figure 36.

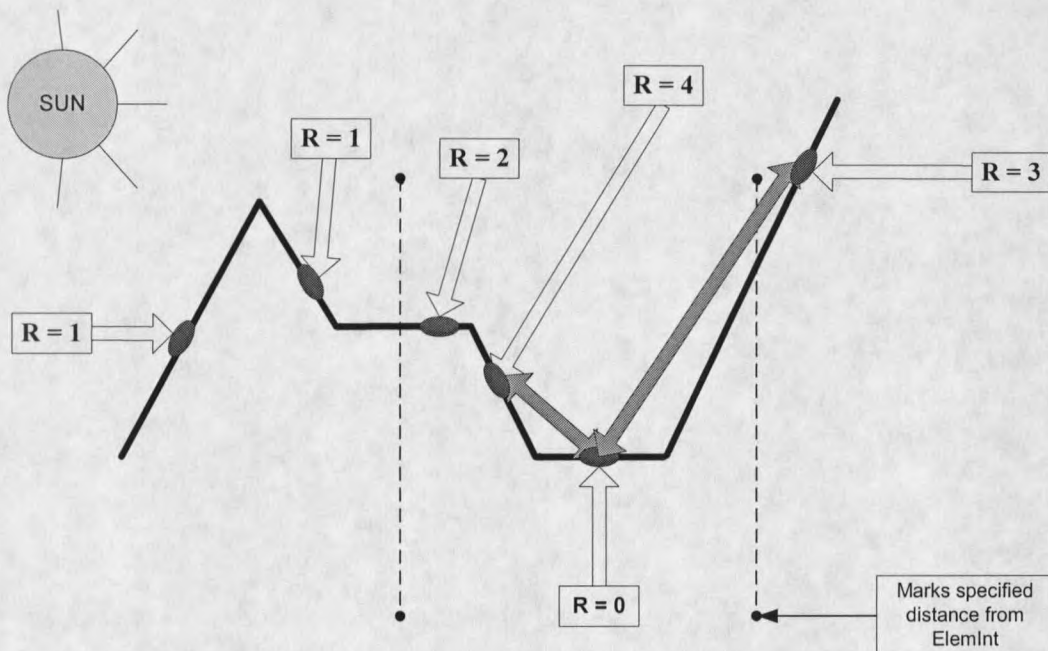


Figure 36. Simple demonstration terrain model to describe the ranking after Step 6c.

Step 6d – View Factors to Elements with View Factors to Elements of Interest. This step applies the same basic view factor principles as step 6c. All elements that still have a rank of 1 are tested against the criteria. At this point, a rank of 1 implies the element is

a surrounding element and has not fallen into any of the categories mentioned above. Although these elements do not emit to or receive from elements of interest it would be inaccurate to delete them at this point. These elements could emit to or receive energy from elements that have a radiation exchange with the elements of interest and therefore, affect the temperature output of the interstate. As in step 6c, an element only passes the criteria if its view factor is greater than the view factor criterion to another element. Figure 37 shows an example of a simple demonstration terrain and the rank of each element after Step 6d. All elements that remain with a rank of 1 are deleted.

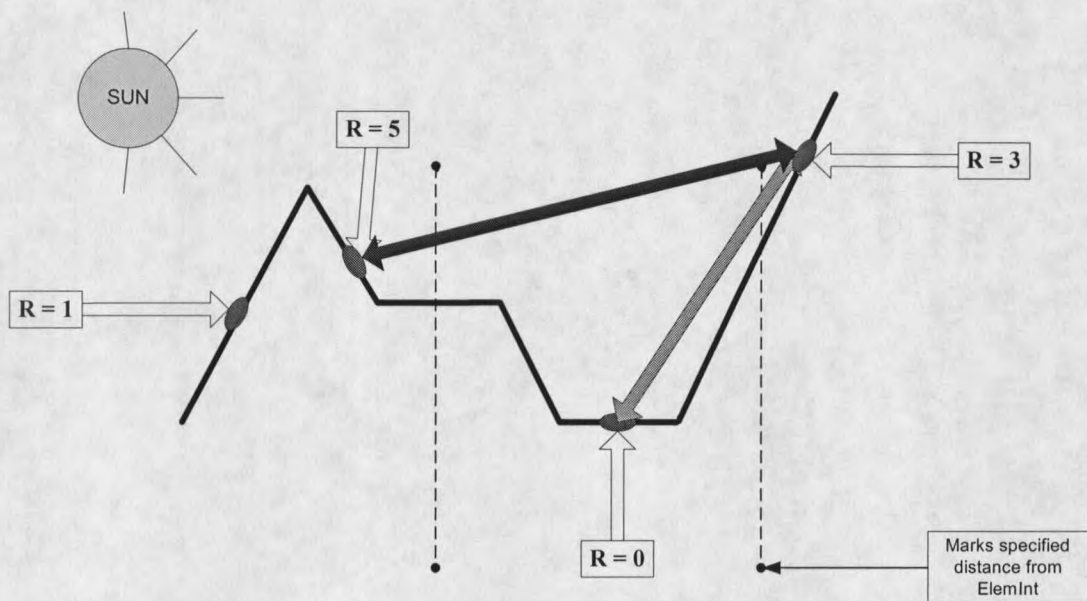


Figure 37. Simple demonstration terrain model to describe the ranking after Step 6d.

Every element has been assigned a rank according to the criteria in the last several steps. If the rank of an element is increased, it is applied in such a way to give a unique rank at every step. For instance, a surrounding element that failed all the criteria remains

with a rank of one. On the other hand, a surrounding element that failed the specified distance criteria but did emit to an element of interest, has a unique rank of three. Therefore, the user is able to analyze how many elements are in each category. A summary of the final ranking is shown in Figure 38.

R = 0:	Element of Interest (ElemInt)
R = 1:	Surrounding Element (SurrElem) NOT within specified distance of ElemInt Does NOT emit to (or receive from) ElemInt Does NOT emit to (or receive from) elements that DO emit to (or receive from) ElemInt
R = 2:	SurrElem within specified distance of ElemInt Does NOT emit to (or receive from) ElemInt
R = 3:	SurrElem NOT within specified distance of ElemInt Does emit to (or receive from) ElemInt
R = 4:	SurrElem within specified distance of ElemInt Does emit to (or receive from) ElemInt
R = 5:	SurrElem NOT within specified distance of ElemInt Does NOT emit to (or receive from) ElemInt Does emit to (or receive from) elements that DO emit to (or receive from) ElemInt

Figure 38. Summary of the ranks applied to the elements in Steps 6a, 6b, 6c, and 6d.

Step 6e – Output File. The final step of this code is to produce a text file with the number of elements deleted and the element ID's to delete. A sample of this file is shown in Figure 39.

Figure 40 summarizes the procedure in Steps 6b through 6e.

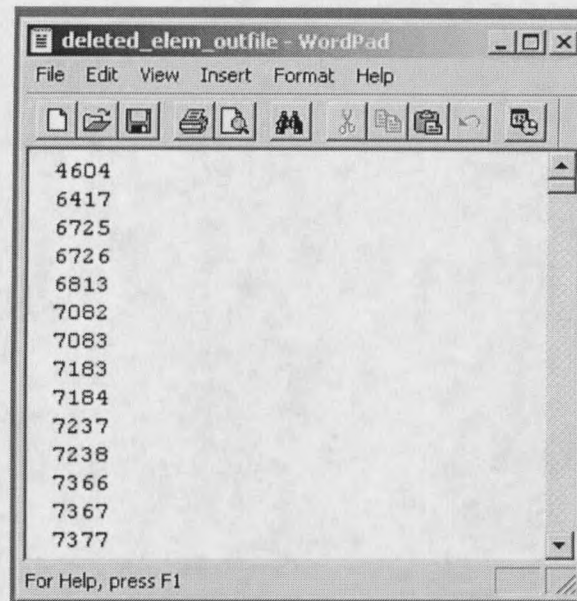


Figure 39. An example of the output text file from Step 6e. The first row of the file contains the number of elements to delete. The subsequent rows are the element numbers to delete ($R = 1$).

Step 7 – Create a new part

The final step in the code takes as input the text file containing the elements to delete. Using the TDF library a new part ID is created, which is one more than the current number of parts, and is labeled as “Elements to Delete”. Every element in the file is assigned to the new part ID. The output of this code is an updated .tdf file called update.tdf.

Creating a RadTherm Project

The new tdf file created in Step 7 is opened in RadTherm/RT. As can be seen in Figure 42 a new part number 18 has been created that contains all the elements to delete.

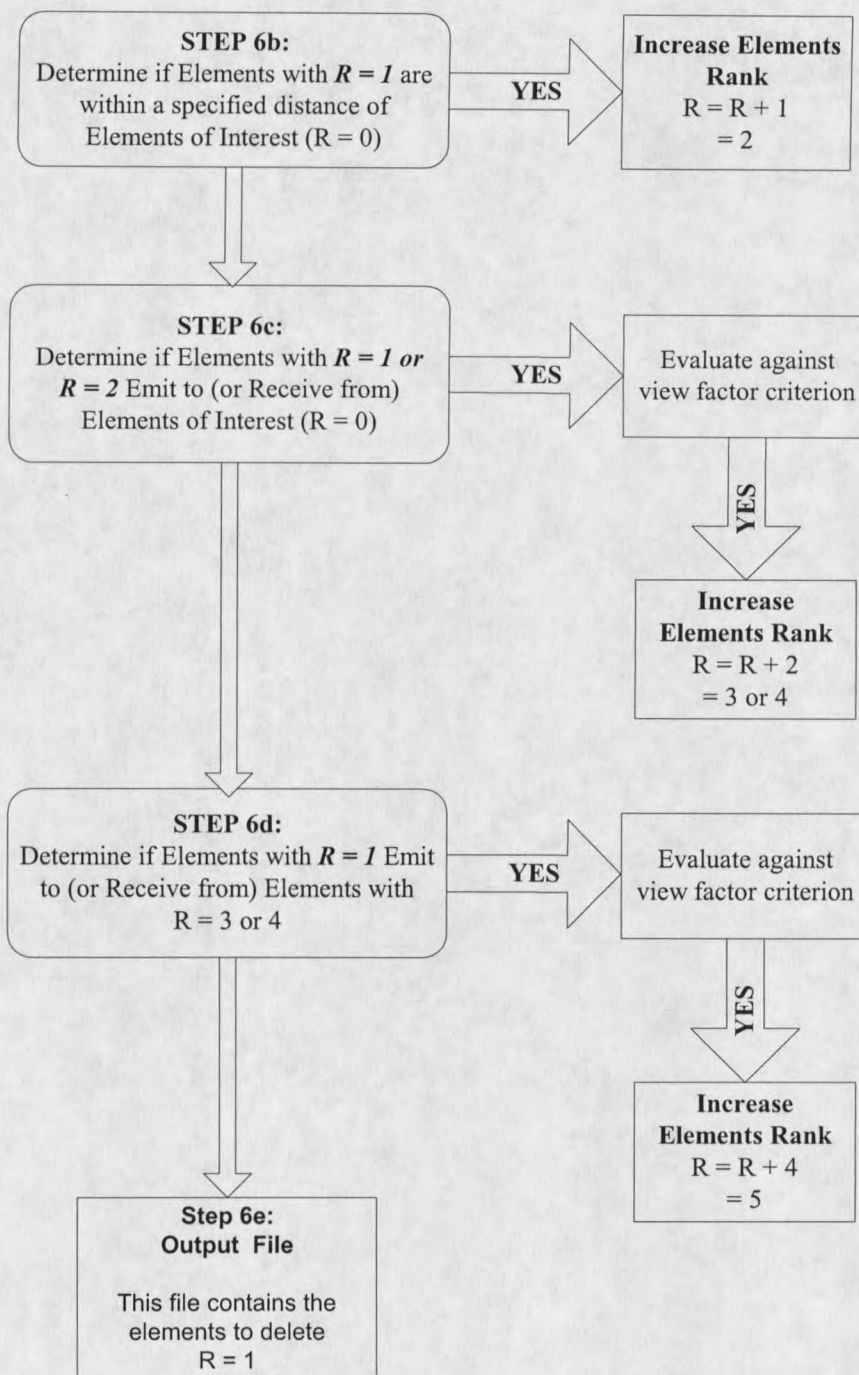


Figure 40. Summary of steps 6b through 6e.

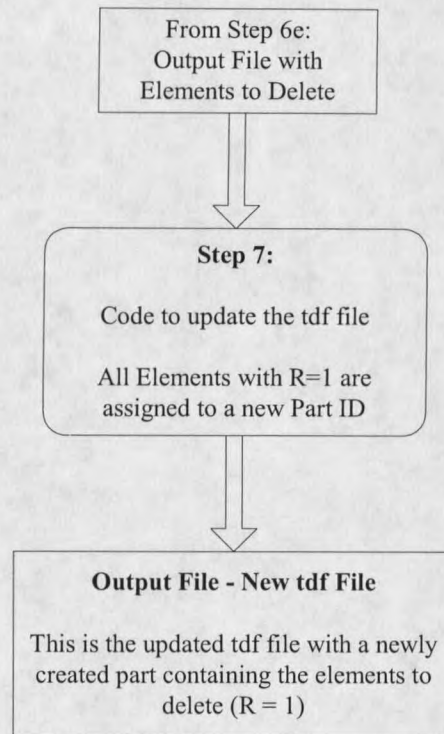


Figure 41. Flow chart for Step 7. The elements to delete, which are contained in the text file, are assigned to a new Part ID.

This step allows a visual check before the elements are permanently deleted. Once the elements are confirmed for deletion the 'delete part' button is used to remove all elements and geometry of the part. An example of the remaining map is illustrated in Figure 43.

Next, each terrain part is assigned the appropriate properties. These thermal properties (e.g. thermal conductivity, albedo, etc.) are applied so that temperatures may be calculated based on the current or forecast environmental conditions [Adams and Curran, 1999]. The thermal model is then executed to calculate the temperatures associated with each cell. Calculated results of the surface temperatures are color coded and displayed as an animated sequence of thermal maps.

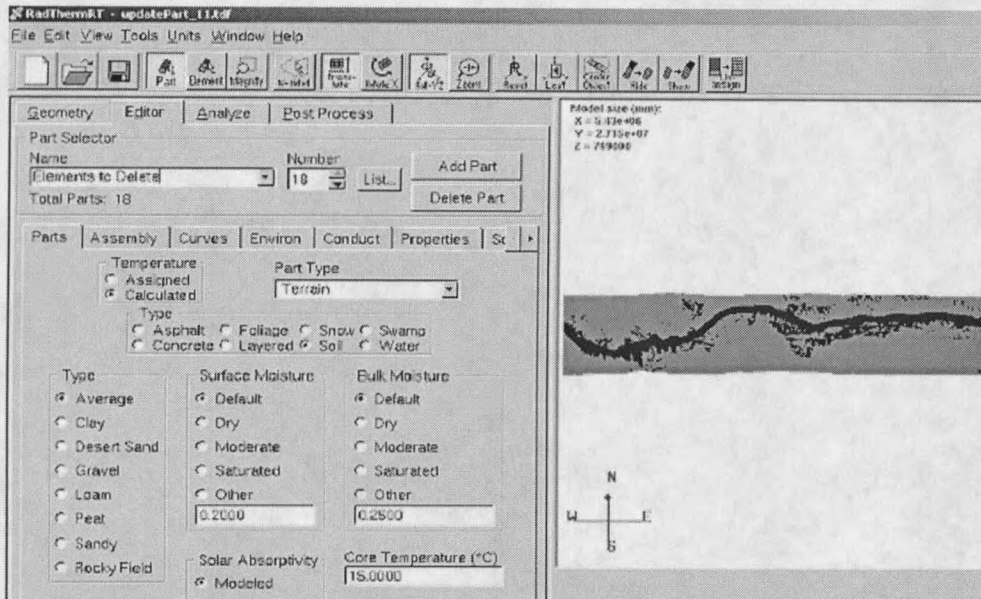


Figure 42. Example of the new update.tdf project created for RadTherm/RT. A new part has been created (Part 18) labeled “Elements to Delete”.

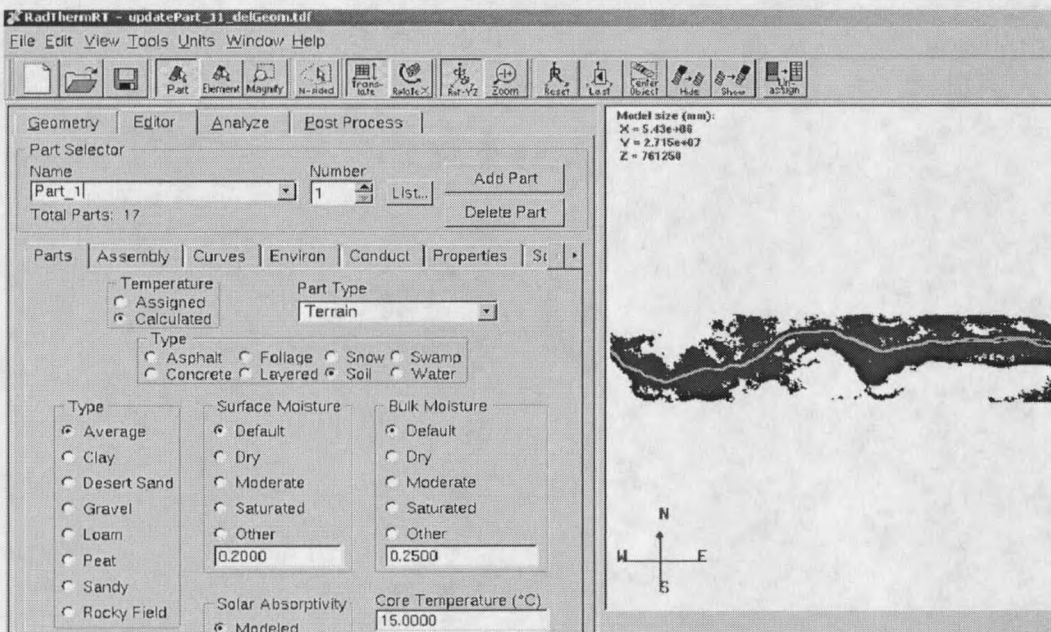


Figure 43. Example of the update.tdf project after Part 18, which contains the elements to delete, has been removed.

CHAPTER 4

FINDINGS/RESULTS

The first section of this chapter provides evidence that the methodology is successful in eliminating elements. The evaluation area is Bozeman Pass from mile marker 313 to mile marker 331 on Interstate 90. For this case, the terrain model was reduced from 327,610 surface elements to 39,302 surface elements that represent the highway or have significant thermal interaction with the highway elements. The code used to eliminate elements based on view factors, is referred to as the view factor trim routine in this chapter.

The next section of this chapter includes a model of Rocky Canyon, which is a smaller segment of Bozeman Pass. This model is used to provide a surface temperature comparison between the original model of Rocky Canyon to the new model of Rocky Canyon with elements deleted. The surface temperatures are compared at increments of $\frac{1}{2}$ mile from mile marker 315 to mile marker 318. A weather file from the winter and summer are used for the results.

Radiation Exchange – Simple Model

In the following section three different scenarios are explored. First, the importance of shading is discussed. Second, the importance of the shortwave energy exchange between surfaces and its effect on the temperature of that region is examined. Next, longwave radiation exchange between surfaces is determined. This section determines

the extent to which the terrain surrounding the interstate affects the temperature output of the highway. In addition, it begins to explain how the use of view factors can be used to eliminate non-participating elements.

Shadowing Effects

Simple models are used to describe the importance of shading from shortwave radiation. Two models are used for a comparison between a shaded and non-shaded region. The first model, as shown in Figure 44, represents a highway (A) with a rock wall (B) at a ninety-degree slope.

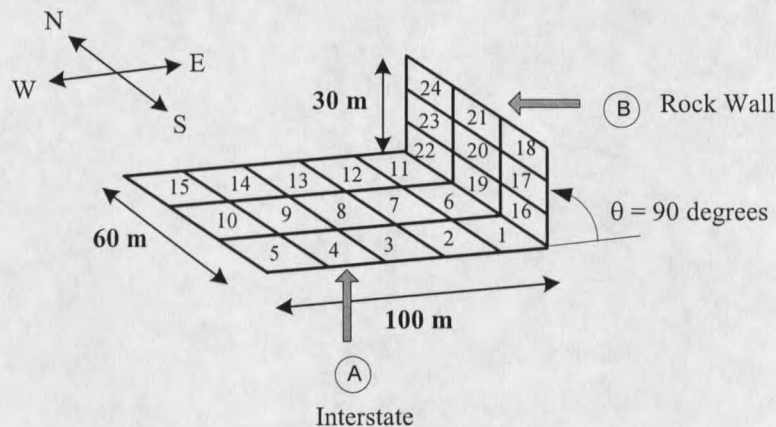


Figure 44. Simple model to demonstrate importance of shading. A represents the interstate and B represents a wall of rock.

This model accounts for the shading effects the rock wall has on the interstate. The interstate has dimensions of 100 meters by sixty meters and a total of fifteen surface nodes. The rock wall has nine surface elements with dimensions of thirty meters by sixty

meters. The second model is a flat section of the interstate composed of fifteen elements (Figure 45). This is simply the first model (Figure 44) without the rock wall.

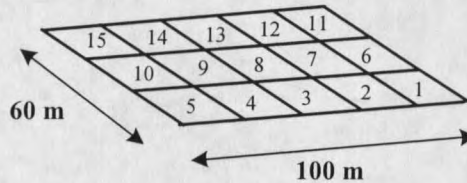


Figure 45. A simple model that represents a flat highway.

The location of the model is 88.5 West longitude and 47.2 North latitude, which is Calumet, Michigan. This location is chosen because the conditions for the day are ideal for the test condition and the weather file is simple to manipulate. An identical weather file with a duration of twenty-four hours on July 22, 1993 is applied to both models. The weather file begins at 12:00 AM and ends on July 23, 1993 at 12:00 AM. The conditions for the day were no clouds, no precipitation, and the temperature varied from 9.5 to twenty-three degrees Celsius. For this particular day, the sunrise was at 6:16 AM and the sunset was at 9:44 PM. To recall, the position of the sun is measured by the azimuth and zenith angle as shown in Figure 17. The angles are shown in two-hour increments from 6:00 AM to 6:00 PM (Table 5). It should be noted the maximum height of the sun on July 22, 1993 was at 2:00 PM at a zenith angle of 27.04° .

Element 6 for both models, which is the center element on the interstate next to the rock wall, is plotted for a full day in Figure 46. As expected, the temperatures diverge at 6:30 AM, which is shortly after sunrise. The maximum temperature difference is 19.1° and occurred at 12:00 PM on July 22, 1993.

Table 5. Solar Azimuth and Zenith angles at two-hour increments on July 22, 1993.

Time of Day	Azimuth Angle (°)	Zenith Angle (°)
6:00 AM	54.44	93.78
8:00 AM	75.88	75.38
10:00 AM	97.6	55.17
12:00 PM	127.5	36.4
* 2:00 PM	179.78	27.04
4:00 PM	232.182	36.32
6:00 PM	262.17	55.07

* Sun is at peak position

Since the rock wall in Figure 44 keeps the element in shade it remains significantly cooler than the element exposed to the direct sun. For this particular day, the maximum zenith angle of the sun was 27.04° directly north at 2:00 PM. As the position of the sun

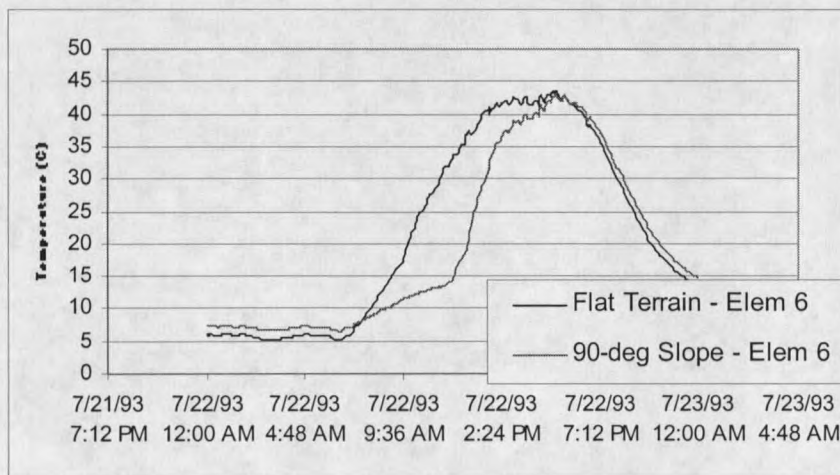


Figure 46. A temperature plot for element 6 of the interstate without a rock wall versus element 6 of the interstate with a rock wall. The plot is from 12:00 AM on July 22, 1993 to 12:00 PM on July 23, 1993.

continues to move west, relative to the model, the temperatures of both models begin to converge after 2:00 PM. The temperatures eventually become uniform at 5:00 PM. This simple example shows the importance of shading on the thermal model. Next, the shortwave radiation exchange between elements is examined.

Shortwave Radiation Exchange

In this section, the shortwave radiation exchange between elements is examined. Recall from Chapter 2, Equation (2) that the sum of the absorptivity and reflectivity must be equal to one for terrain elements. Therefore, if the absorptivity of a material is varied, it affects the amount of reflected radiation. This would imply that for a particular element, the higher the albedo, the greater the amount of radiation reflected, and thus, the more radiation influencing neighboring elements. As described in Chapter 2, view factors play a major role when looking at the energy exchange between surface elements. Several models are used to demonstrate the temperature influence on an element from a neighboring sloped rock wall at a θ value of 180° , 135° , and 90° . The side view of the model is shown in Figure 47.

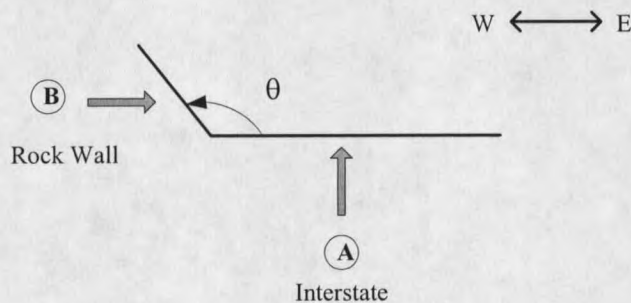


Figure 47. The side view of a model with a sloped rock wall with a varying angle θ .

Part A is modeled as the interstate and Part B is a rock wall with a specified albedo. The temperature difference of selected surface elements is compared.

180° Flat Model. Two models are compared to determine the radiation exchange for a flat terrain. The first model is the interstate with dimensions of 60-meters by 100-meters and composed of fifteen surface elements (Figure 48a). Figure 48a is compared to an identical model with an adjoining terrain part at a theta value of 180° (Figure 48b). The additional terrain part has elements numbering from 16 to 24. This part is modeled as rock and has dimensions of 60-meters by 30-meters as shown in Figure 48b.

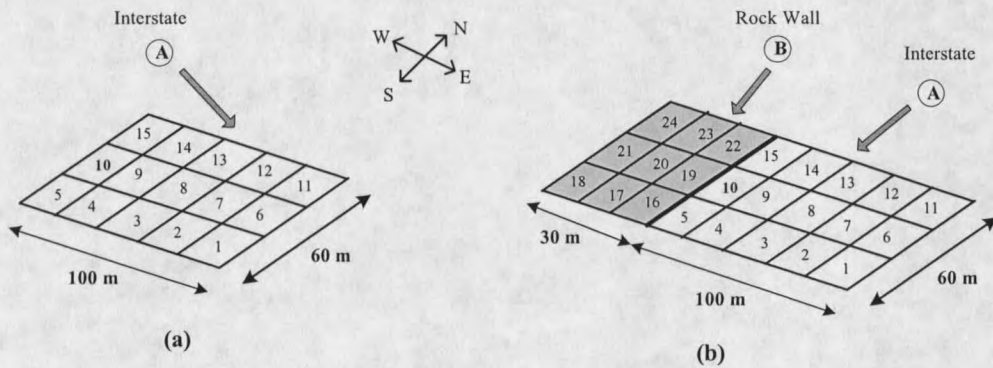


Figure 48. (a) Simple model of the interstate with fifteen surface elements. (b) Simple model of the interstate with an adjoining terrain part composed of nine surface elements (16-24) and modeled as rock.

Each model is subjected to the identical weather file that was used in the previous section. The file consists of a twenty-four hour period from 12:00 AM on July 22, 1993 to 12:00 AM on July 23, 1993. The conditions for the day were no clouds, no precipitation, and the temperature ranged from 9.5°C to 23°C. The temperature for

Element 10, which is at the base of the slope terrain on the interstate, is compared for both models (Figure 48b) as shown in Figure 49.

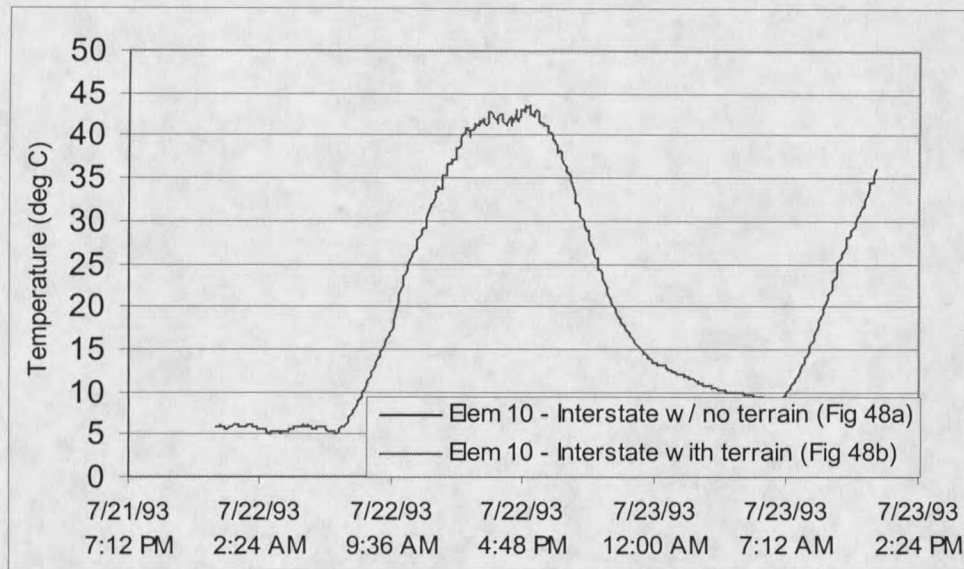


Figure 49. Plot of the 24-hour temperature output for Element 10 for both models described in Figure 48b.

As shown in Figure 49, there is no temperature difference between the two models. This can be explained by looking at the view factor file. The view factor information for each model is shown in Table 6.

Since both models have no geometric variation, every element in each model has a view factor of 1.0 to the sky node (Table 6). Therefore, there is no direct radiation exchange between the surface elements because there are no view factors between the elements. For this reason, the additional horizontal terrain part has no influence on the temperature output of the interstate.

Table 6. (a) View factor values for Figure 48a. Receiver #17 is the sky node. (b) View factor values for Figure 48b. Receiver #26 is the sky node.

Element #	Receiver #	View Factor
		Value
1	17 (sky node)	1.00
2	17	1.00
3	17	1.00
4	17	1.00
5	17	1.00
6	17	1.00
7	17	1.00
8	17	1.00
9	17	1.00
10	17	1.00
11	17	1.00
12	17	1.00
13	17	1.00
14	17	1.00
15	17	1.00

(a)

Element #	Receiver #	View Factor
		Value
1	26 (sky node)	1.00
2	26	1.00
3	26	1.00
4	26	1.00
5	26	1.00
6	26	1.00
7	26	1.00
8	26	1.00
9	26	1.00
10	26	1.00
11	26	1.00
12	26	1.00
13	26	1.00
14	26	1.00
15	26	1.00
16	26	1.00
17	26	1.00
18	26	1.00
19	26	1.00
20	26	1.00
21	26	1.00
22	26	1.00
23	26	1.00
24	26	1.00

(b)

135° Model. This section demonstrates the temperature influence on an element from a neighboring 135° rock wall modeled with two contrasting albedos of 0.05 and 0.90 (Figure 50a). The basic model in this section is similar to the model used for the shading case (Figure 45) but the sloped wall is on the west side of the interstate (Figure 50a). This allows an immediate reading for the shortwave radiation reflecting off the sloped

terrain once the sun begins to rise from the east. For the sun's position in terms of the azimuth and zenith angles, refer to Table 5.

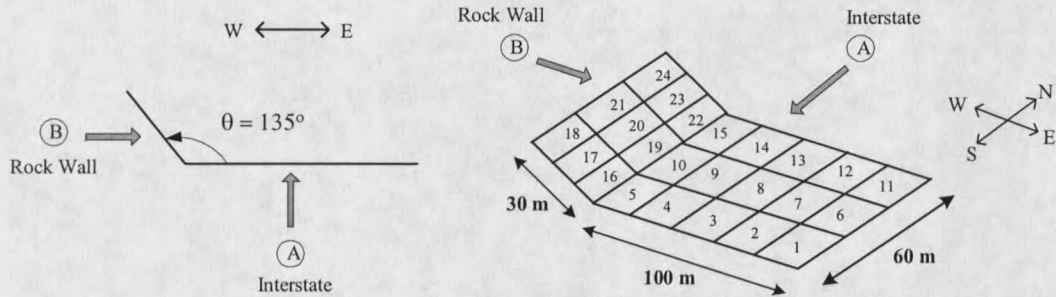


Figure 50. (a) Side view of the model with the sloped wall at an angle of 135° . (b) A model composed of two parts. Part A is the interstate with fifteen elements. Part B is the sloped wall at 135° from the interstate.

The model is composed of two parts. Part A is modeled as the interstate with dimensions of 60-meters by 100-meters and fifteen surface elements. Part B is the sloped wall with dimensions of 60-meters by 30-meters and composed of nine surface elements. Figure 50b illustrates the model with each element labeled.

For the first case, the sloped rock wall in Figure 50b is modeled as highly reflective rock with a high albedo. In the second case, the sloped terrain is assigned a low albedo value. For each case the interstate remains constant. The absorptivity values for each model are shown in Table 7.

For the first case, rock has an absorptivity value of 0.05 which means 95% of the radiation that strikes the surface is reflected. On the other hand, for the second case only

Table 7. The absorptivity values for the sloped terrain for each case.

Case Number	Surface Property	Absorptivity
1 and 2	Interstate	0.85
1	Highly Reflective Rock	0.05
2	Rock with High Absorptivity	0.90

10% of the radiation is reflected from the rock surface. Therefore, elements neighboring the highly reflective rock wall should be influenced by the reflected shortwave radiation. Rock, with a lower albedo is less likely to have an influence on neighboring nodes during direct sunlight. Furthermore, if the albedo value is low then the absorptivity is high and in return, the rock is warming up during direct sunlight. This influences longwave radiation exchange to heat the neighboring nodes if the rock is hotter. The amount of reflected energy is determined by the properties set for each type of terrain. The destination of the reflected energy is determined by view factors. Both models have the same view factors since these are calculated strictly using the geometry. There are 2048 rays cast from the centroid of each element. The view factor relationship for Element 10 is shown in Table 8. Element 10 is at the center of the interstate and its node, which is at the center of the element, is ten meters from the sloped wall.

As shown in Table 8, Element 10 has a view factor to every element on the sloped surface. Furthermore, element 10 receives radiation from every element on the sloped surface. Therefore, Element 10 directly exchanges radiation with each node on the sloped terrain and is influenced by the wall depending on its properties. The view factor relationship is based on two criteria. First, each ray cast using the ray trace algorithm is

Table 8. View factors for Element 10 labeled on Figure 50b with $\theta = 135^\circ$. (a) Shows the receiving elements that element 10 emits to. Note that receiver number 26 is the sky node. (b) The elements that emit to element 10.

Emitter Number	Receiver Number	View Factor
10	16	0.0059
10	17	0.0059
10	18	0.0034
10	19	0.0254
10	20	0.0146
10	21	0.0078
10	22	0.0059
10	23	0.0059
10	24	0.0034
10	26 (Sky Node)	0.9219
View Factor Sum:		1.00

(a)

Emitter Number	Receiver Number	View Factor
16	10	0.0088
17	10	0.0102
18	10	0.0059
19	10	0.0752
20	10	0.0244
21	10	0.0117
22	10	0.0088
23	10	0.0102
24	10	0.0059

(b)

calculated using the view factor equation (Equation (21)). In addition, the number of times a ray strikes an element adds an additional weight to this value. 92.19% of the weighted view factor value for Element 10 is towards the sky node. The remaining 7.81% is towards the 135° rock wall.

Figure 51 shows the temperature plot of Element 10 for the two cases described above. The solid black line represents the model with a high absorptivity and the gray line represents the model with highly reflective rock. There is not much variation between the two lines. The maximum temperature difference is 1.7028°C , which occurs at 11:40 AM.

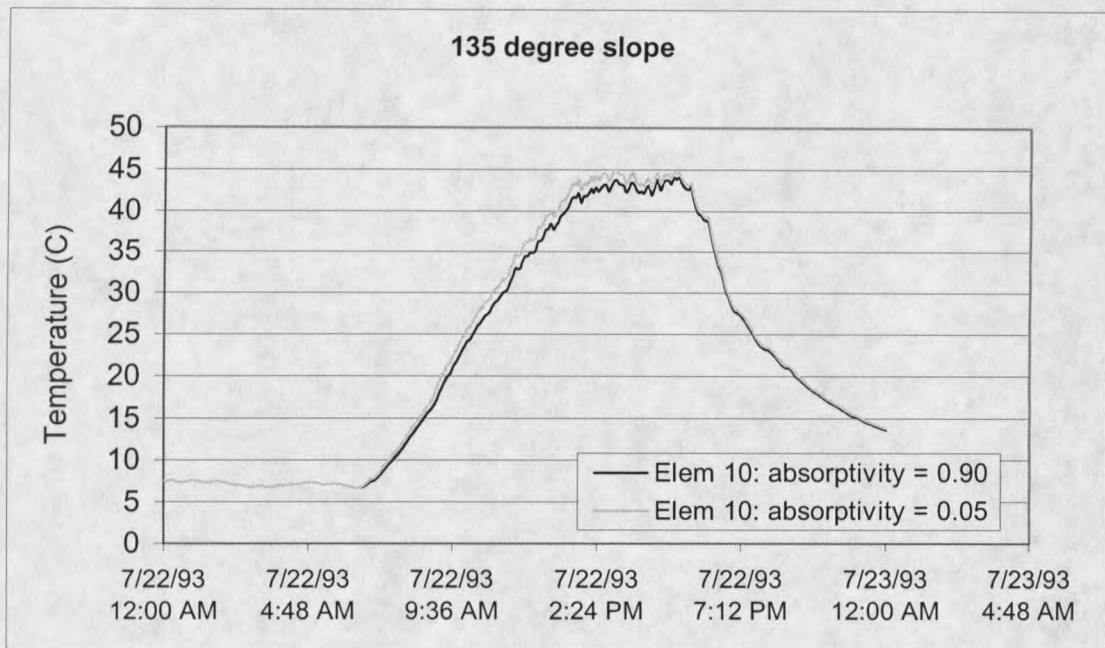


Figure 51. The temperature plot for Element number 10 for two cases. A rock slope with: 1) an absorptivity (α) of 0.05 and 2) $\alpha = 0.90$.

Figure 52 shows the temperature difference for Element 20 for the two cases. Element 20 is the center element on the rock wall. This plot shows a significant temperature difference for the two elements. The temperatures begin to diverge at 6:30 AM. This seems reasonable since the sunrise was at 6:16 AM on July 22, 1993. The maximum temperature difference occurs at 1:30 PM when the highly reflective rock is 17.57°C cooler than the rock with a high absorptivity. The highest temperature of Element 20 for the highly reflective rock wall occurs between 3:00 PM and 4:30 PM and has an average value of 15.3°C. For the rock wall with a high absorptivity, the maximum temperature occurs at 1:30 PM and is 32.2°C.

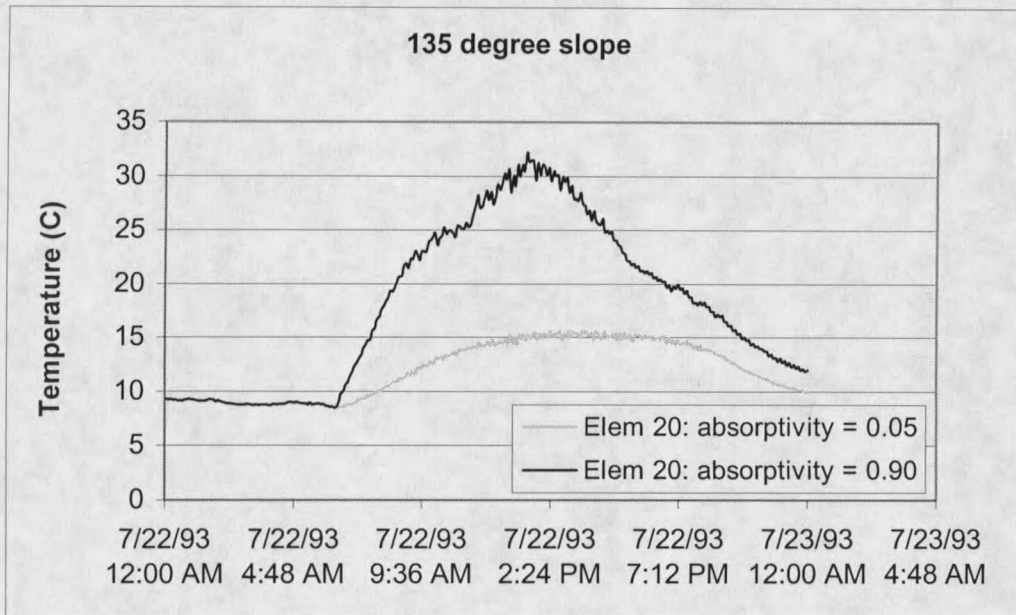


Figure 52. The temperature plot for Element number 20 for both cases. For the first case, the rock slope has an absorptivity (α) of 0.05 and in the second case, $\alpha = 0.90$. Element 20 is the center element on the sloped terrain.

90° Model. For this case, a model similar to Figure 50b is used, except the sloped rock wall is adjusted to an angle of 90° (Figure 53a).

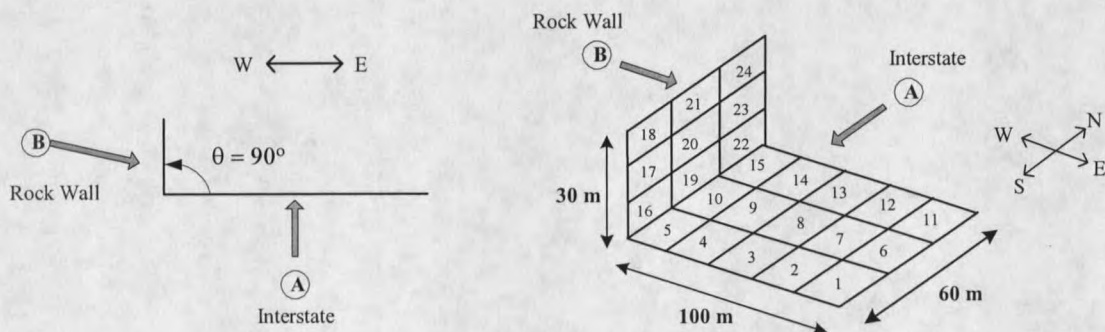


Figure 53. (a) Side view of the simple model consisting of two parts, the interstate and a rock wall at an angle of 90°. (b) A simple RadTherm/RT model consisting of two parts, the interstate and a rock wall at 90°. Fifteen elements are assigned to the interstate and nine elements are assigned to the rock wall at 90°.

The model is created with two parts in RadTherm/RT and assigned element numbers as shown in Figure 53b. This model is used to demonstrate the influence of shortwave radiation reflected off a rock wall at 90° . Again, the view factor file is used to determine the target for the reflected radiation. Since the geometry of the model has changed, new view factors are calculated. The software is set to cast 2048 rays from the each element centroid to compute the view factor relationship. Table 9 compares the view factors of Element 10 for a rock wall at 135° , as in the previous case, and for the rock wall at 90° .

Table 9. View factors for Element 10 labeled for a rock wall at 135° , and for a rock wall at 90° .

Emitter Number	Receiver Number	View Factor 135° Slope	View Factor 90° Slope	Ratio $90^\circ/135^\circ$
10	16	0.0059	0.0161	2.7288
10	17	0.0059	0.0229	3.8814
10	18	0.0034	0.0142	4.1765
10	19	0.0254	0.1122	4.4173
10	20	0.0146	0.0791	5.4178
10	21	0.0078	0.0283	3.6282
10	22	0.0059	0.0161	2.7288
10	23	0.0059	0.0229	3.8814
10	24	0.0034	0.0142	4.1765
10	26	0.9219	0.6740	0.7311
View Factor Sum:		1.00	1.00	

It is apparent from Table 9 that the weight of each view factor is much greater when the rock wall is at an angle of 90° . The value of the view factor for the 90° rock slope is over two to five times that of the 135° rock slope.

As in the previous section, two different scenarios are used. First, the rock wall is highly reflective with an absorptivity value of 0.05 and then assigned a value of 0.90. The same weather file is used as in the previous case for the slope rock wall at 135° . The temperature output for the two scenarios is shown in Figure 54.

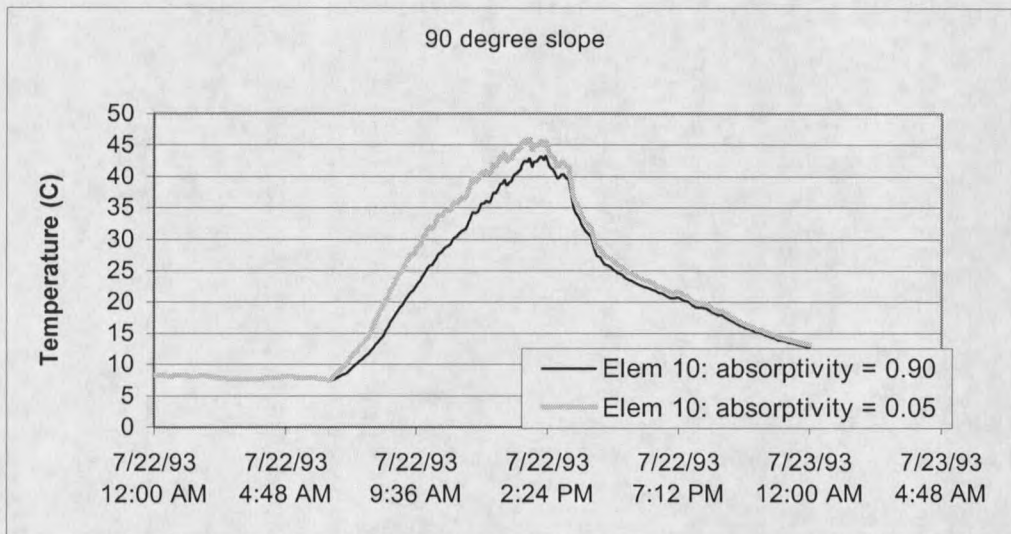


Figure 54. The temperature plot for Element 10 for both cases. For the first case, the rock slope has an absorptivity of 0.05 and in the second case has an absorptivity of 0.90.

For this example, the 90° angle of the rock wall makes a considerable difference on the interstate Element 10. The temperatures begin to differ around 6:30 AM once the sun begins to rise. From 6:30 AM to 2:30 PM the temperature difference ranges from 3°C to 6°C in Figure 54. The maximum temperature difference occurs at 10:05 AM and is 6.18°C . The temperatures finally converge around 3:00 PM. This is the point at which the rock wall begins to shadow Element 10 and the direct shortwave radiation is no

longer influencing either model. Refer to Table 5 for the position of the sun in terms of the azimuth and zenith angles.

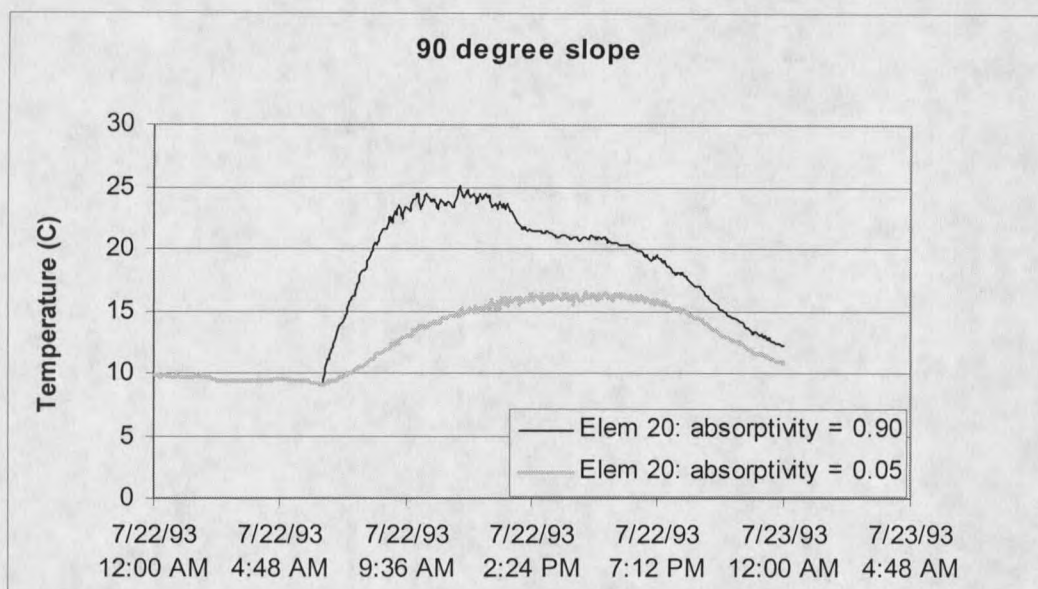


Figure 55. The temperature plot for 90 degree wall, Element 20 for two cases. With a rock slope that has 1) an absorptivity (α) of 0.05 and 2) $\alpha = 0.90$.

The temperature of Element 20 begins to diverge at 6:30 AM. The temperature difference varies from 2°C at 6:40 AM to the maximum temperature difference of 11.2°C, which occurs at 9:20 AM. The greatest temperature of Element 20 for the highly reflective rock wall occurs between 3:00 PM and 5:00 PM and the average temperature is 16.13°C. The rock wall with an absorptivity 0.90 has a maximum temperature of 25.1°C at 11:40 AM. The temperature difference, as well as the maximum temperature, is less than the case with the rock wall at 135°. Since the angle of the rock wall has increased to

90°, there has been a decrease in the apparent area of the rock face. Therefore, the rock wall receives less direct radiation, and the temperatures are lower for the case at 90°.

Figure 56 compares the temperature difference of a highway element that has an adjoining rock wall at two different slopes and the same albedo. The temperature of Element 10 with a rock wall at 90° (Figure 53b) and an absorptivity (α) of 0.90 is shown as the solid black line in Figure 56. The grey line in Figure 56 represents the temperature of Element 10 with a rock wall at 135° (Figure 50b) and an absorptivity (α) of 0.90.

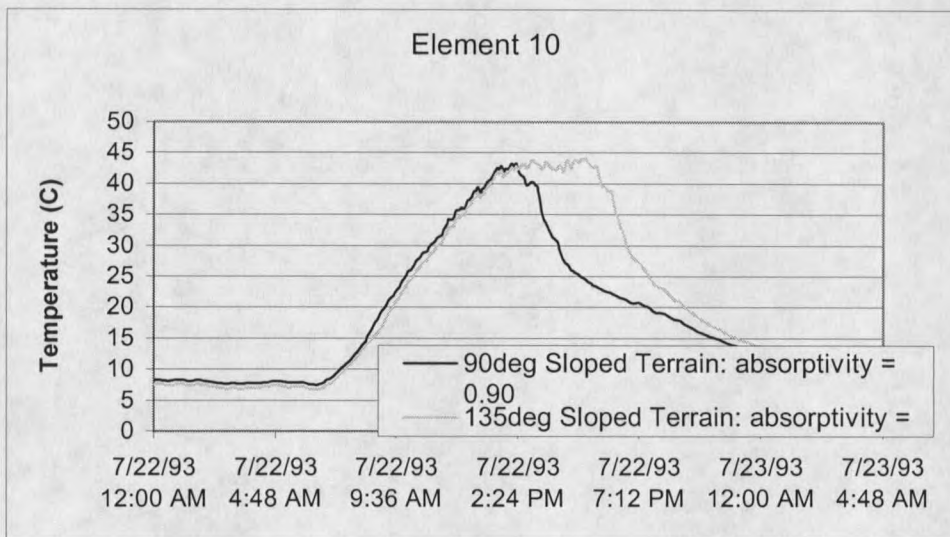


Figure 56. The highway temperature of Element 10 for the model with a sloped terrain at 135° and 90° with an absorptivity value of 0.90.

The temperatures of highway Element 10 remain fairly close after sunrise in Figure 56. Since the rock wall has a low albedo, there is a minimal amount of reflected solar radiation. Therefore, the rock wall has a small influence on neighboring Element 10 from reflected shortwave radiation during sunrise. The temperatures begin to diverge after the

sun reaches its peak at 2:00 pm. The wall at 135° continues to receive direct solar radiation after the sun peaks, as opposed to the rock wall at 90° .

Figure 57 compares the highway temperature difference for an adjoining rock wall at two different slopes and the same albedo. The grey line in Figure 57 represents the temperature of Element 10 with a rock wall at 135° (Figure 50b) and an absorptivity of 0.05. The temperature of Element 10 with a rock wall at 90° (Figure 53b) and an absorptivity of 0.05 is shown as the solid black line in Figure 57.

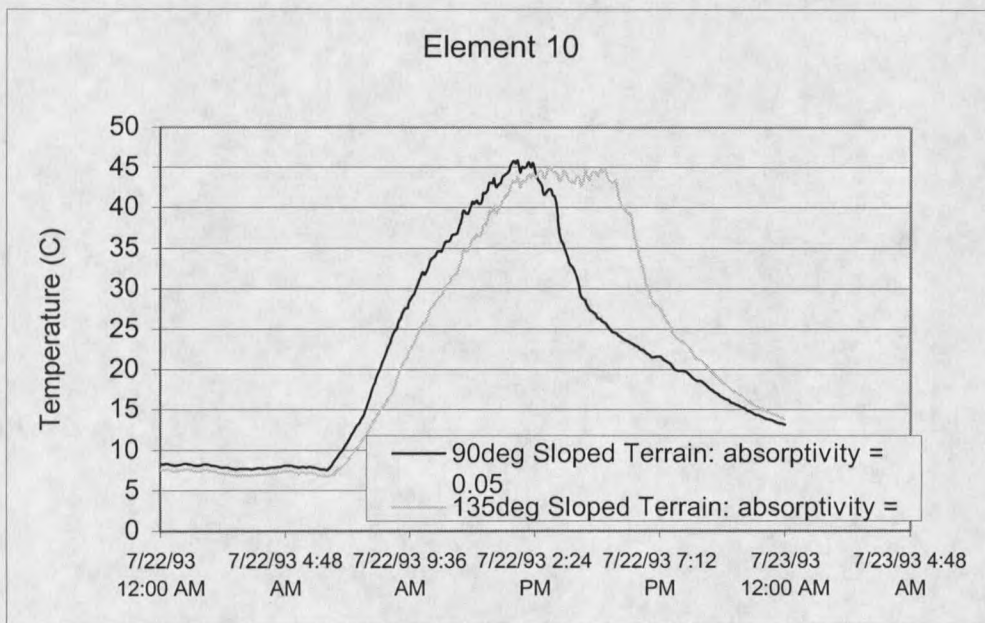


Figure 57. The temperature plot of Element 10 for the model with terrain sloped at 135° and 90° with an absorptivity value of 0.05.

As opposed to Figure 56, the temperatures of Element 10 begin to diverge after 6:30 am. Since each rock wall has a high albedo, the rock wall at 90° has a greater influence on neighboring highway Element 10. After the sun has peaked around 2:00 pm, the rock

wall at 90° begins to receive less direct solar radiation and is much cooler than the rock wall at 135° .

Longwave Radiation Exchange

For this section, the longwave radiation effects are compared using two models.

Figure 58 shows the side view of both models.

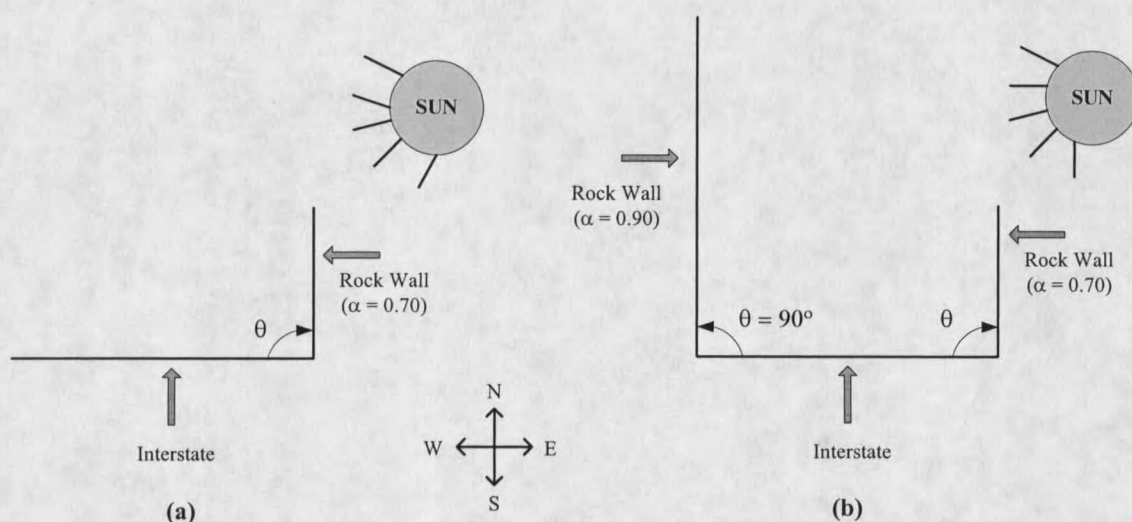


Figure 58. Side view of the models used to compare the longwave radiation effects. (a) Simple model without a left rock wall. (b) Simple model with a rock wall on the left side.

The first model consists of a 90° rock wall that stands 30 meters high and has a length of 300 meters. This wall is composed of rock and has an absorptivity value of 0.70. At the base of the rock wall is a 50-meter by 300-meter flat terrain modeled as the interstate (Figure 58a). The second model is identical to Figure 58a with an additional wall on the left side of the model. This is a rock wall with an absorptivity of 0.90 (Figure 58b).

In both models, the purpose of the right wall is to provide shade to eliminate a majority of the direct shortwave radiation onto the interstate. The intention of the additional left wall in Figure 58b is to provide an object that heats up due to the direct shortwave radiation from the morning sun. Again, refer to Table 5 for the position of the sun throughout the 24-hour period on July 22, 1993.

The weather file for July 22, 1993, used in the previous cases, is again applied to the current scenario with modifications. The natural weather conditions are applied until 9:45 AM on July 22, 1993. Starting at 9:50 AM, the diffuse and direct solar radiations are set to zero. Therefore, the shortwave radiation has disappeared and the longwave radiation effects can be examined. The target time was chosen as 9:50 AM because this is the point at which the sun begins to heat up the interstate elements at the base of the left wall. In order to get accurate results of the longwave radiation effects, the direct solar radiation onto the interstate was avoided. Figure 59 illustrates the model in Figure 58b with the labeled elements. It should be noted that only half of the model is displayed in Figure 59 due to symmetry. Figure 58a represents a model that is identical to the model shown in Figure 59 without the left rock wall.

The view factors for element number 40 are calculated for the model shown in Figure 59 with and without the left rock wall. For the model without a left rock wall, 91.80% of the view factors encountered the sky. On the other hand, the model with the left rock wall had only 45.14% reached the sky. The results from the view factor file are shown in Table 10.

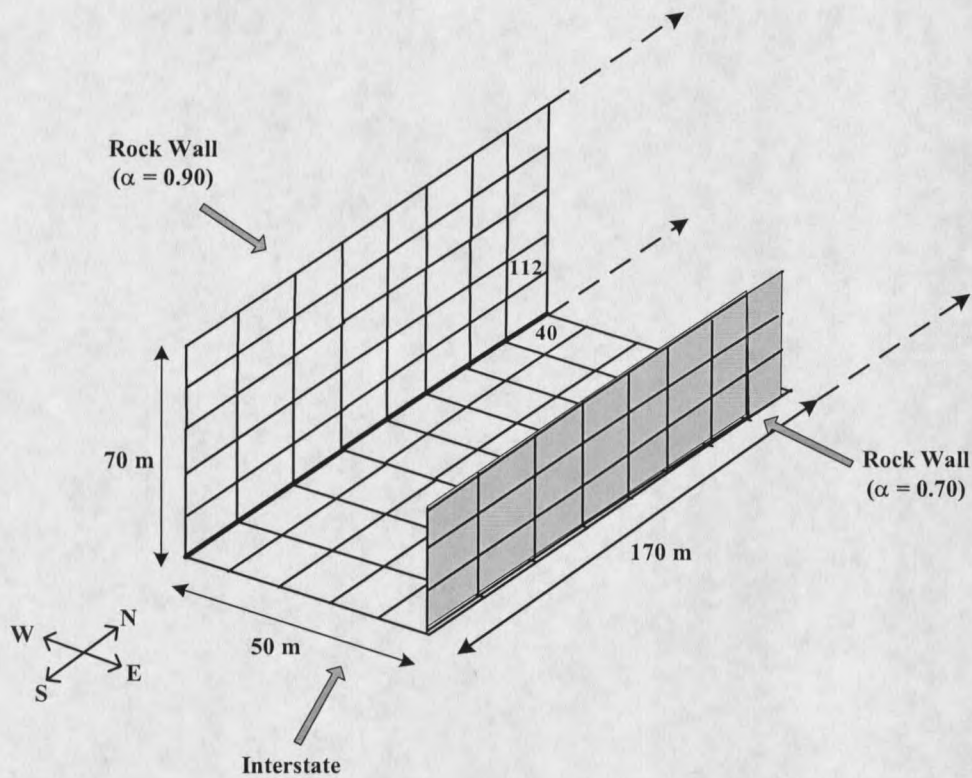


Figure 59. The model used to compare the longwave radiation effects with the pertinent elements labeled.

Table 10. The percentage of the view factor values for element 40 in each model in Figure 60.

	% Left Wall	% Right Wall	% Sky
No Left Rock Wall	0.00%	8.20%	91.80%
Left Rock Wall	46.70%	8.23%	45.14%

The weather file for July 22, 1993 is applied to both models with the solar radiation set to zero at 9:50 AM. The temperature of the interstate element 40 for both models is compared in Figure 60.

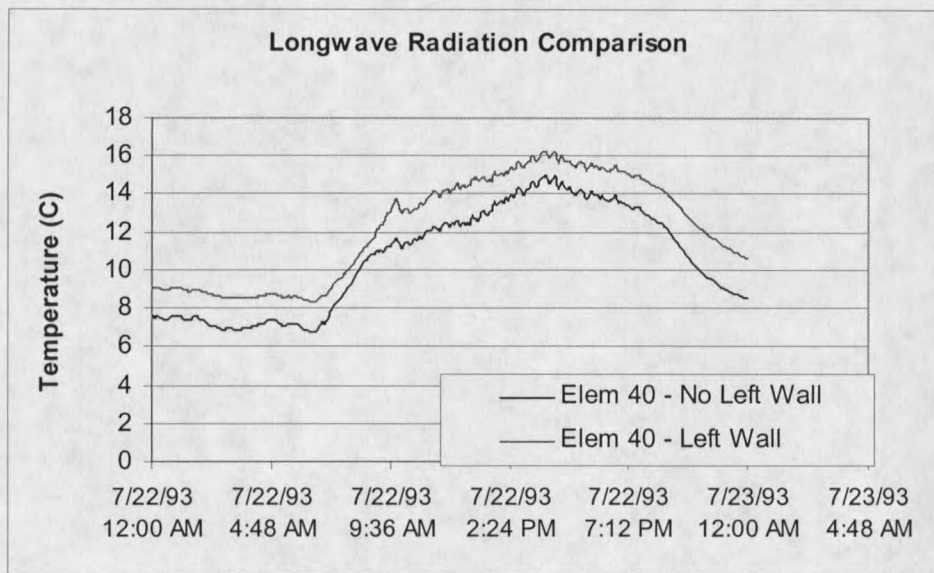


Figure 60. The temperature plot of interstate element 40 for both models. Note that the solar radiation is set to zero at 9:50 AM.

The first temperature difference to note is at 12:00 AM on July 22, 1993. RadTherm/RT initializes the temperatures of Element 40 in each model to be a degree and a half apart. RadTherm/RT applies the core temperature as the initial temperature of the element and then solves the equations at steady state for the surface temperature. The software calculates the initial temperature of Element 40 (without the left rock wall) to be cooler because there is no longwave radiation emitting to the element from the wall, forcing element 40 to carry out additional interactions with the cooler sky. By creating a

variation of the model in Figure 59, the initial temperature difference of Element 40 can be proven. For this model, the left rock wall in Figure 59 is incremented each run by 10 meters. As shown in Figure 61, the temperatures converge as the height of the rock wall approaches the original height of 70 meters.

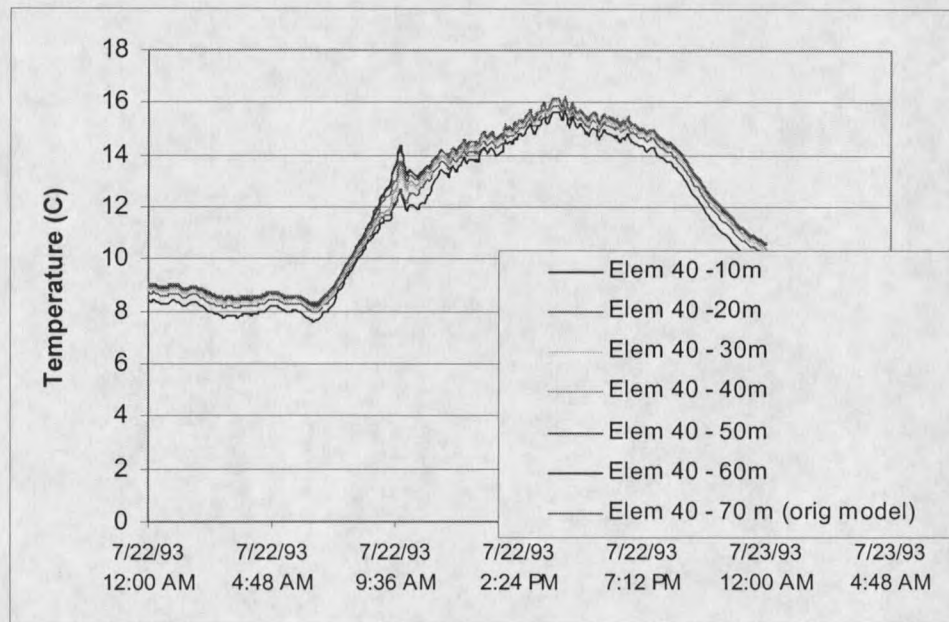


Figure 61. The temperature plot of Element 40 where the left rock wall increases in 10-meter segments.

As the sun begins to rise at 6:16 AM, Element 40 for both models increase in temperature (Figure 60). At 9:50 AM, the solar radiation is set to zero and a slight dip can be seen in Figure 60. Both elements continue to heat up and remain an average of 2°C apart. The greatest temperature difference is 2.27°C and occurs at 11:00 PM.

Figure 60 shows a slight significance due to longwave radiation, but not as influential as shortwave radiation.

The next figure compares the interstate Element 40 to Element 112 on the left rock wall.

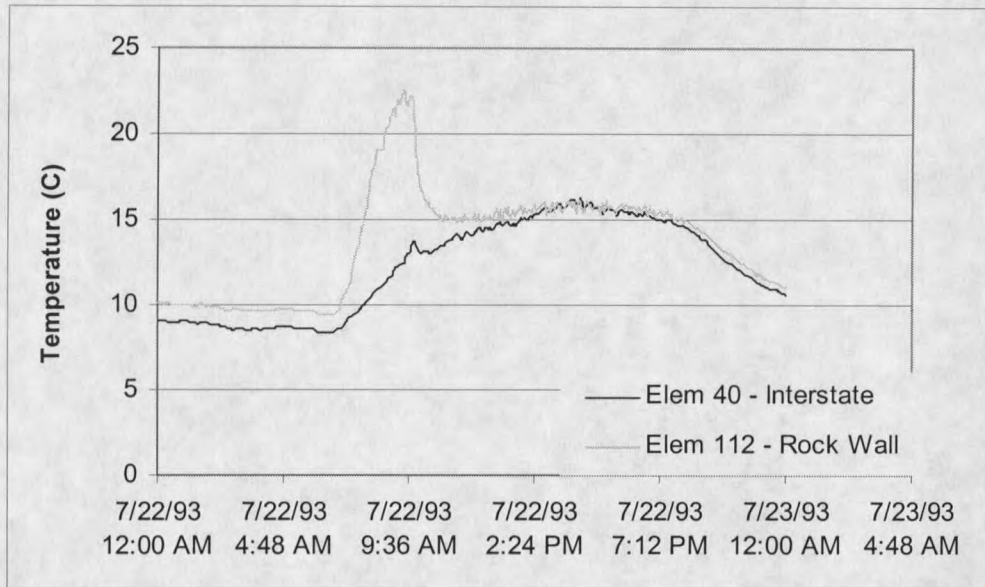


Figure 62. A plot of the temperature of interstate Element 40 to Element 112 on the left rock wall.

The rock wall heats up quickly as the shortwave radiation strikes the surface from 6:16 AM to 9:45 AM. Once the solar radiation is set to zero, the curve drops and by 3:00 PM, Element 112 on the rock wall and Element 40 on the interstate have converged to a uniform temperature.

Figure 63 compares the interstate element 40 (for the model with no left wall) to the sky temperature. Both curves observe the same trend throughout the 24-hour weather file. Element 40 does not seem to lose a significant amount of heat to the sky. One reason is element 40 gains heat by conduction from the interior layers or surface convection may also provide heat to the surface element.

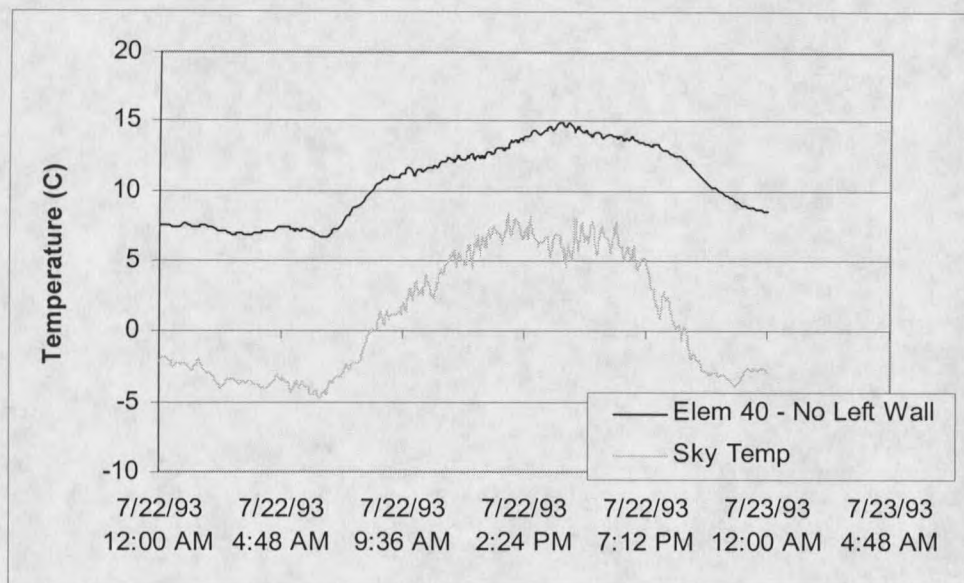


Figure 63. A plot of the temperature of interstate Element 40 (with no left wall) to the sky temperature for the model.

Model of Bozeman Pass

This section provides validation that the methodology is successful in eliminating elements. Bozeman Pass from mile marker 313 to mile marker 331 is used as the region of interest. The UTM coordinates for the boundary are listed in Figure 64. These are the coordinates used to trim the map to the appropriate size (Step 2 of Chapter 3).

West UTM: 504000
South UTM: 5052600
North UTM: 5058000
East UTM: 531000

Figure 64. The UTM coordinates used to create the model for Bozeman Pass.

From the UTM coordinates in Figure 64, two ASCII files are created containing the vegetation/road and elevation information. Using the ASCII files, a RadTherm/RT project is created that contains 327, 610 surface elements and 17 parts. Figure 65 shows the model of Bozeman Pass in RadTherm/RT.

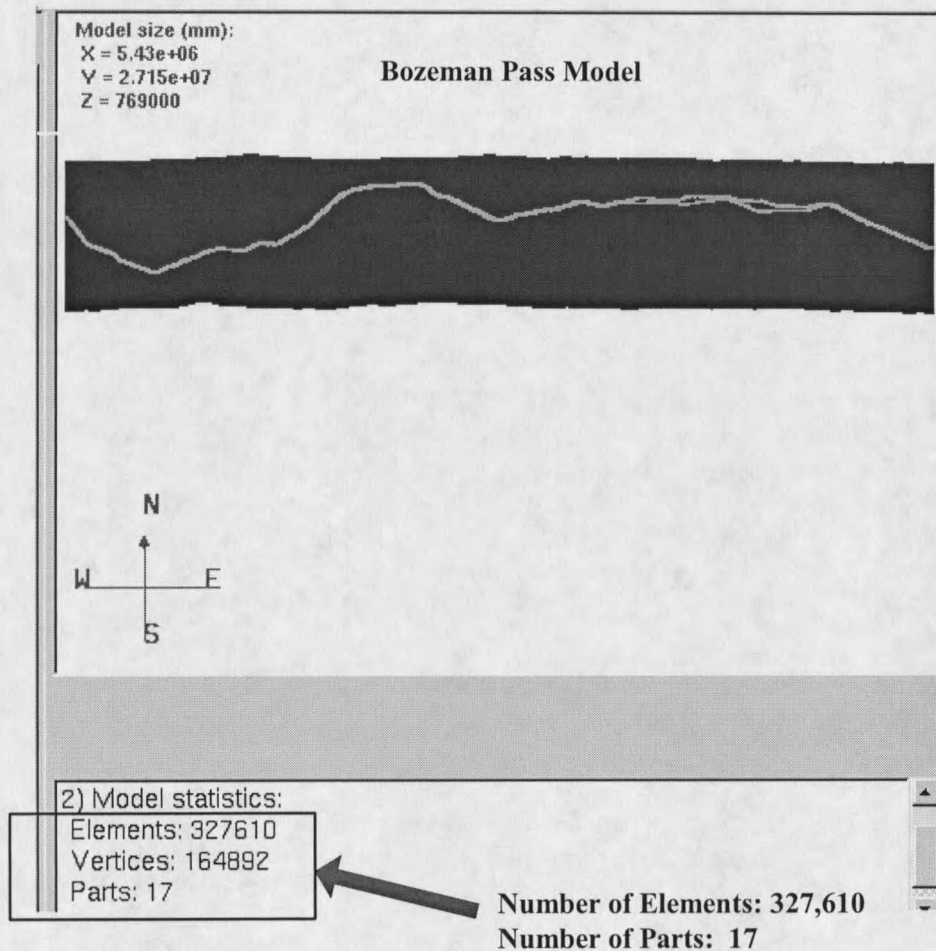


Figure 65. RadTherm/RT model for Bozeman Pass consisting of 327,610 surface elements and 17 parts.

As described in Step 3 of Chapter 3, each part is assigned to type "terrain", a weather file is chosen, and the same start and end solution times are selected. The RadTherm/RT project is now executed and the file that contains the view factor information (.vfs file) is created. Since the same start and end solution times were chosen, the thermal solution is not calculated. At this point, there is an adequate amount of information to begin the process of deleting particular elements.

Since the two files that contain the RadTherm/RT project (.tdf) and the view factor information (.vfs) are not in a readable format, Step 4 and Step 5 in Chapter 3, convert these to ASCII files. The resulting two files contain the element attributes and the view factor information for each element.

The output files from Step 4 and Step 5 are used as inputs into Step 6 as described in Chapter 3. The output file from Step 6 is a text file that contains the elements to delete.

Before Step 6 can be implemented, there are three variables that must be set in the code. First, for Step 6a in Chapter 3, the Part ID(s) that contains the elements of interest must be specified. For this case, the elements of interest for Bozeman Pass are the elements that make up Interstate 90, which are Parts 1 and 2, the east and westbound highway lanes. Figure 66 shows the line of code that contains Part 1 and Part 2 as the elements of interest. The code in its entirety is shown in the Appendix. Two lines of code must be set and are labeled as A and B (Figure 66).

```

//*****
// STEP 6a:
// Separate the elements object into two objects based on partID
//   surrElem and elemInt
// It will also assign an initial rank
//   elemInt (R = 0)
//   surrElem (R = 1)
//*****

// Determine the number of elements EOI (elements of interest)
int nElemInt = 0;
int nSurrElem;

for (int i=1; i<=linecount; i++)
{
    if((elements[i].partID == 1) || (elements[i].partID == 2))
    {
        nElemInt = nElemInt + 1;
    }
}

nSurrElem = linecount - nElemInt;
cout << "Number of Elements of Interest: " << nElemInt << endl;
cout << "Number of Surrounding Elements: " << nSurrElem << endl << endl;

const int SIZE_1 = nElemInt;

Element *elemInt = new Element[SIZE_1](0, 0);

// fill the 2 objects with the correct elements
int count_1 = 1;
for (int i=1; i<=linecount; i++)
{
    if((elements[i].partID == 1) || (elements[i].partID == 2))
    {
        elements[i].rank = 0; // also create rank for elements object
        elemInt[count_1].x = elements[i].x;
        elemInt[count_1].y = elements[i].y;
        count_1++;
    }
    else
    {
        elements[i].rank = 1; // also create rank for elements object
    }
}

```

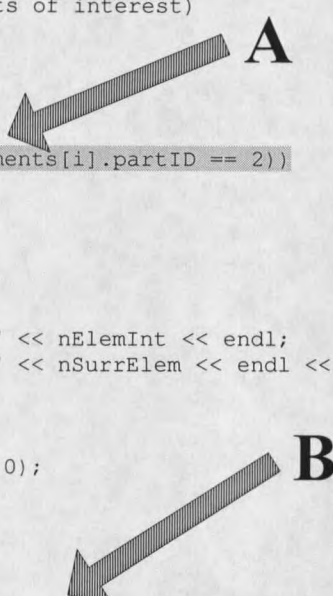


Figure 66. The Part ID that contains the elements of interest must be specified. The two lines of code that must be set for Step 6 are labeled A and B.

The second variable that must be set is the distance criteria in Step 6b of Chapter 3.

This is the distance from the elements of interest in which all elements are kept. To

better illustrate this, Figure 67 shows a simple model with a specified distance criterion of one hundred meters. The line of code that must be set is identified as C in Figure 68.

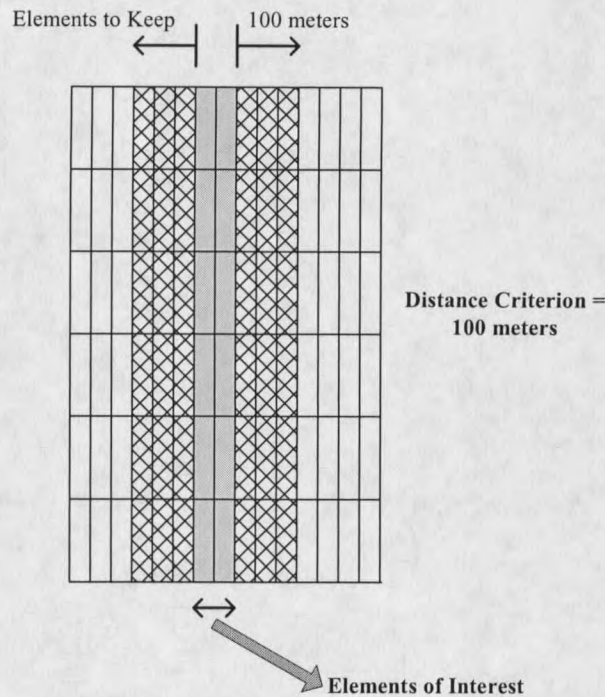


Figure 67. A simple model to illustrate the distance criterion set at 100 meters.

The final variables that must be assigned are for Step 6c and 6d of Chapter 3, for which the view factor value criteria are set. If an element has a view factor to another element, the view factor value indicates the intensity at which it sees the element. An element's rank is not increased if its view factor to another element is less than the view factor criteria. Figure 69 labels the line of code (D and E) for which the value of the view factor criteria is set. Note that the lines of code between D and E have been eliminated for simplicity.

```

//*****
// STEP 6b:
//   Determine if elements with R = 1 are within a specified distance
//   of elements EOI
//   At this point all surrElem should have a rank R = 1
//*****

float tempXdist, tempYdist, tempDist;
float minDist = 999.9; // set initial min distance
int tempElemID; // dummy variable

// Set minimum distance criteria - Within this distance we keep the elements
float distCriteria = 250.0; // 0.25 km

cout << "The distance criteria for which to keep Surrounding Elements is: ";
cout << distCriteria << " meters" << endl << endl;

```

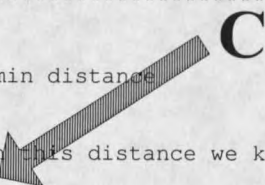


Figure 68. The distance criterion in Step 6b is assigned in the code, which is labeled as C. In this case, the distance criterion is 250 meters.

```

//*****
// STEP 6c:
//   Determine if elements with R = 1 emit to (or receive from)
//   elements with R = 3 or R = 4
//
//*****

float vf_crit_1 = 0.0005;

... (missing code) ...

//*****
// STEP 6d:
//   Determine if elements with R = 1 emit to (or receive from)
//   elements with R = 3 or R = 4
//
//*****

float vf_crit_2 = .0005;

```

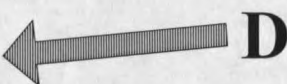
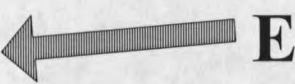



Figure 69. The view factor criteria must be set for lines D and E. In this case, the view factor criteria are 0.0005.

The output from Step 6 is a file that contains the element numbers to delete. This file is used as input into the next Step. In Step 7 every element contained in the file is reassigned to a new part number labeled "elements to delete". The output from this code is a new TDF file that contains an additional part with all the elements to delete. Figure 70 shows an identical model of Bozeman Pass as Figure 65, but contains an additional part. Part number 18 is the additional part that contains the elements to delete.

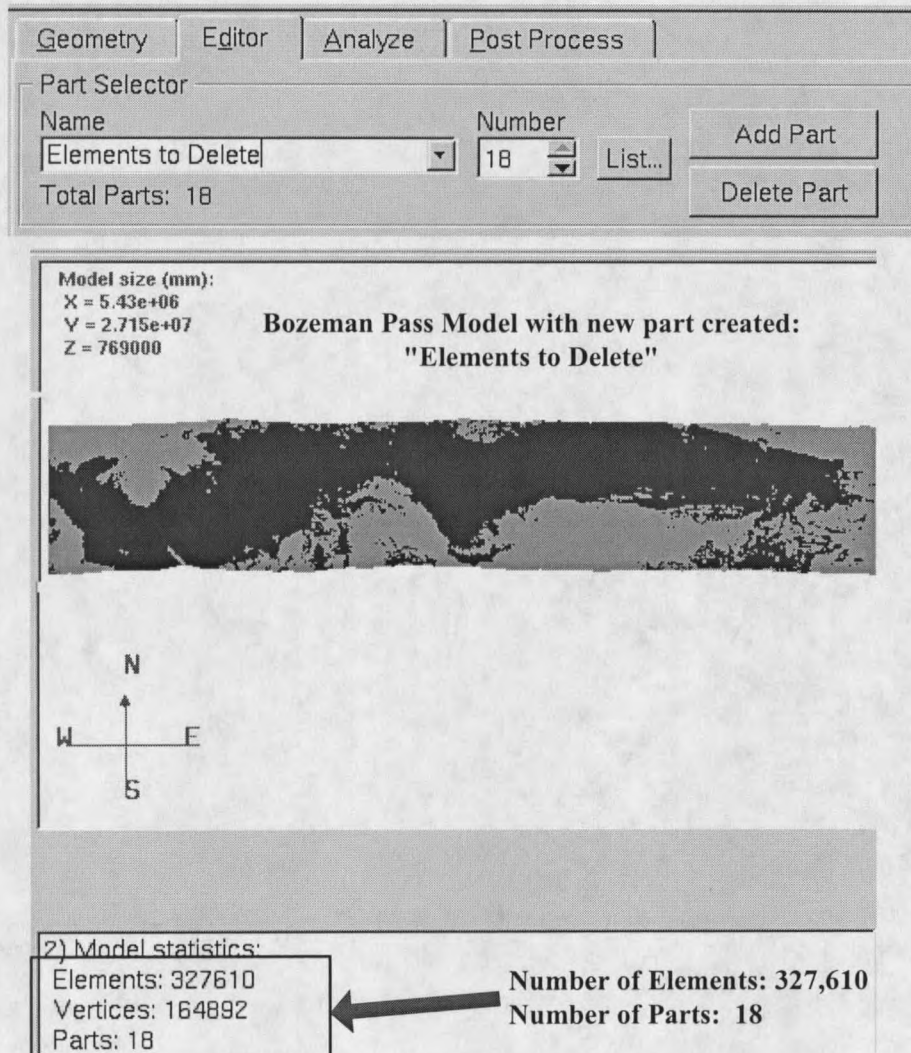


Figure 70. This is the identical model as in Figure 65, but contains an additional part. This part (Part 18) contains the elements to delete.

In the following sections, different values are applied to the view factor criteria and distance criteria and the results of the model are summarized for each scenario. Variation in the value of the view factor criteria and distance criterion affect the number of elements that are deleted. The goal of this procedure is to create a new model with 30,000 to 60,000 elements.

No View Factor or Distance Criteria

For this case, the following properties are set for the model of Bozeman Pass. First, the distance criterion is set to zero. Therefore, no elements are kept because they are within a certain distance from the elements of interest. Second, the minimum view factor criteria are set to zero and hence, there is no minimum requirement of the view factor value.

The resulting rank of each element after the view factor trim routine is applied to the Bozeman Pass model is displayed in Table 11. Refer to Figure 38 in Chapter 2 for a summary of the ranking system. Table 11 shows there are 4354 elements of interest, which is the number of elements that make up Interstate 90 for this particular model.

Since the distance criterion is set to zero, holes appear beside the interstate after elements are deleted. This does not pose a problem for the calculation but makes it difficult to distinguish between the interstate and terrain. Figure 71 demonstrates a magnified picture of Bozeman Pass and the empty spaces near the interstate where elements have been deleted.

Table 11. Results for the model of Bozeman Pass with the distance criterion and view factor criteria set to zero. For this case, 82,075 elements were deleted.

Distance Criterion:	0 meters
View Factor Criterion 1:	0.00
View Factor Criterion 2:	0.00
# Elements of Interest	4354
# Surrounding Elements	323256
# Elements within Dist Criterion	0
# Elements (R = 2)	0
# Elements (R = 3)	67828
# Elements (R = 4)	0
# Elements (R = 5)	173353
# Elements to be Deleted (R = 1)	82075

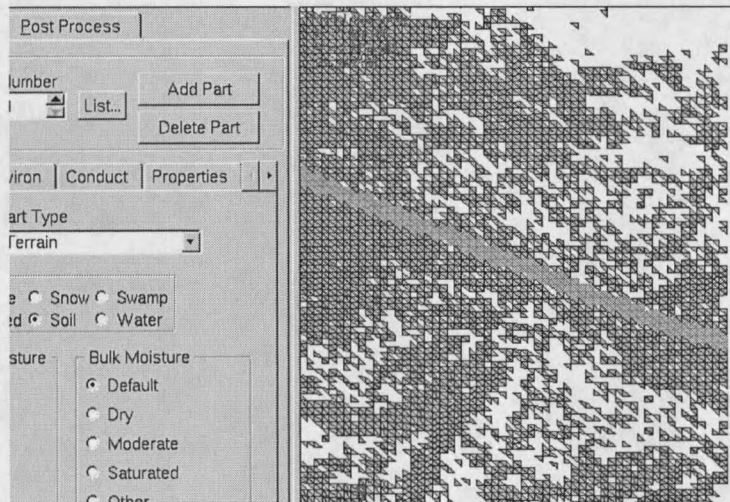


Figure 71. A magnified picture of the interstate through Bozeman Pass. The empty spaces are a result of the deleted elements.

In order to eliminate the holes near the interstate, the next case applies a distance criterion to the model.

No View Factor and Distance Criteria Set

For this case, the distance criterion is set to 250 meters from the elements of interest. This ensures all elements within 250 meters from the interstate are kept. The view factor criteria in this case remain at zero. Table 12 shows the results from this example.

Table 12. Results for the model of Bozeman Pass with the distance criterion set to 250-meters and the view factor criteria set to zero. For this case, 81,448 elements are deleted.

Distance Criterion:	250 meters
View Factor Criterion 1:	0.00
View Factor Criterion 2:	0.00
# Elements of Interest	4354
# Surrounding Elements	323256
# Elements within Dist Criterion	34306
# Elements (R = 2)	21260
# Elements (R = 3)	54782
# Elements (R = 4)	13046
# Elements (R = 5)	152720
# Elements to be Deleted (R = 1)	81448

As expected, the number of surrounding elements and elements of interest remain the same as the previous example. There are 34,306 elements that fall within the distance criterion. Comparing Table 11 and Table 12, it shows applying the distance criterion did not drastically change the number of elements to delete. Only 627 more elements were deleted when the distance criterion was set to zero. This shows that nearly all the surrounding elements within 250 meters of the interstate receives view factors from the interstate elements, or from elements that see the interstate. The resulting model is illustrated in Figure 72.

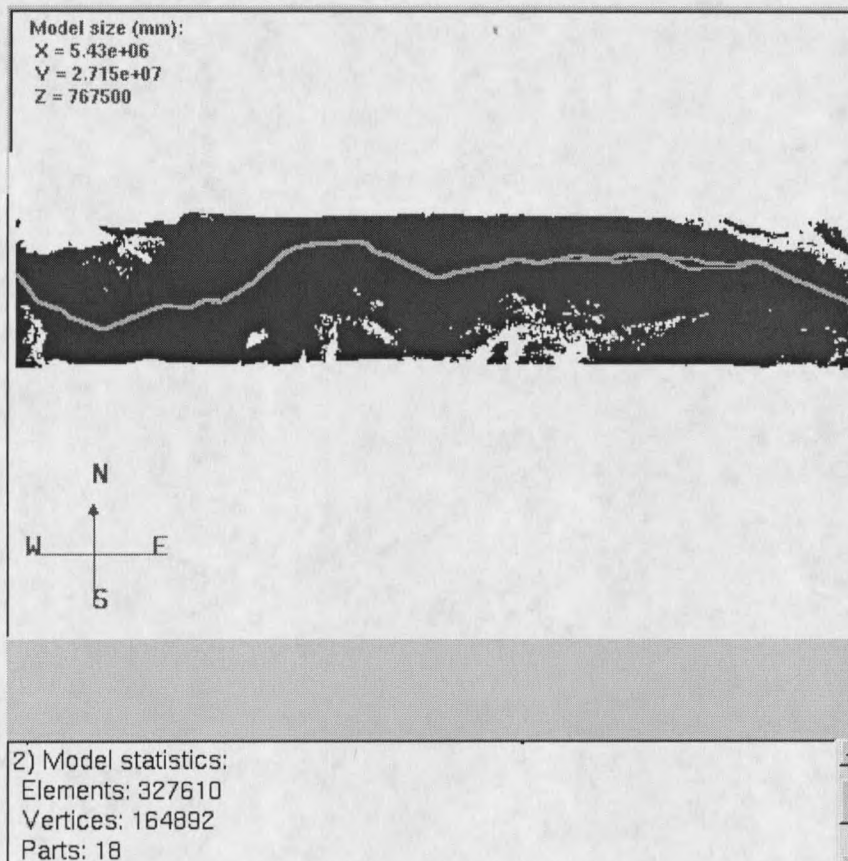


Figure 72. The resulting model of Bozeman Pass with the distance criterion set to 250 meters and the view factor criteria set to zero. Only 82,000 surface elements have been deleted and 245,000 surface elements remain in the model.

At this point, the goal is not achieved of scaling the map to roughly 30,000 to 60,000 surface elements. Only 82,000 surface elements have been deleted, and the model remains with approximately 245,000 surface elements.

View Factor and Distance Criteria Set

In order to eliminate additional elements, the view factor criterion is determined. To recall, the view factor file contains each element and each element for which it directly exchanges radiation. In addition, it has a view factor value associated with each element. As opposed to keeping all elements that exchange radiation with the interstate, a view factor criterion is set. If the emitter or receiver is an element of interest, the value of the view factor is compared to the view factor criteria. The algorithm discussed in the next section determines the value for the view factor criteria.

Evaluate View Factor Criterion. The view factor criterion is determined by a sensitivity study on the radiation term. The net radiation equation (Equation (72)) is duplicated for convenience as Equation (80)

$$q_k = \frac{\varepsilon_k \cdot A_k}{1 - F_{k-k} \cdot (1 - \varepsilon_k)} \left[\sum_{j=1}^N \sigma \cdot F_{k-j} \cdot (T_k^4 - T_j^4) + \sum_{j=1}^N \left((1 - \delta_{kj}) \cdot F_{k-j} \cdot \frac{1 - \varepsilon_j}{\varepsilon_j} \right) \cdot \frac{q_j}{A_j} \right], \quad (80)$$

where N is the number of elements, F_{k-j} is the view factor and q_k is the net radiative loss from the k^{th} surface. Recall, an element is deleted if its view factor value does not exceed the view factor criterion. This section determines the view factor criterion and estimates the amount of energy lost if an element is eliminated. It is assumed for this case the terrain is convex and element k does not see itself and therefore, F_{k-k} is set to zero in

Equation (80). Equation (80) is rewritten to account for the energy exchange between surface element k and a single element j . This is accomplished by removing the summation notation, so that for a single interaction Equation (80) becomes

$$q_k = \varepsilon_k \cdot A_k \cdot \left[\sigma \cdot F_{k-j} \cdot (T_k^4 - T_j^4) + \left(F_{k-j} \cdot \frac{1 - \varepsilon_j}{\varepsilon_j} \right) \cdot \frac{q_j}{A_j} \right] \quad (81)$$

Values are chosen for the terms in Equation (81) to estimate the amount of energy lost to surface element j if surface element k is deleted. A value of 0.0005 was set as the view factor criteria, F_{k-j} . For this view factor value, 0.05% of the view factors from element k see element j . The emissivity values, ε_k and ε_j , were assigned a value of 0.95 for this demonstration. To reiterate, emissivity is a characteristic of a surface, which determines its ability to emit heat by radiation. Table 13 list the emissivity values for terrain types in RadTherm/RT, which range from 0.94 to 1.0

Table 13. The emissivity values for the terrain types in RadTherm/RT.

Terrain	Emissivity
Asphalt	0.94
Concrete	0.94
Foliage	0.95
Snow	1.00
Soil	0.95

The area of the element, A_k and A_j , were given an average value of 450-meters². This is assuming a 30-m x 30-m original grid size for each element and the grid is divided in half to create triangular elements giving an area of 450-m². The temperature of surface element k , T_k , was set to 290K or 16.85°C. The temperature of surface element j was assigned four different temperatures, T_{j1} , T_{j2} , T_{j3} , and T_{j4} that are 325K (51.85°C), 300K (26.85°C), 270K (-3.15°C), and 240K (-33.0°C), respectively. The y-axis in Figure 73 shows the amount of radiation leaving surface k and intercepted by j for four different surface temperatures of element j .

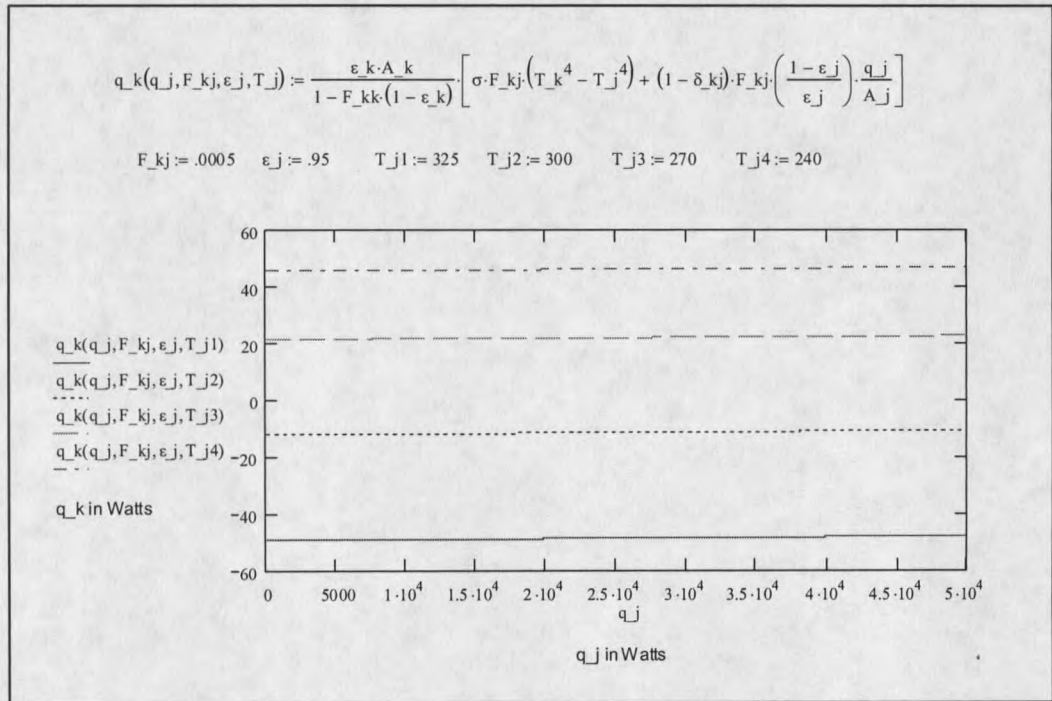


Figure 73. Plot of Equation (81), which is the energy lost or gained from surface element k to j . The x-axis is the radiation from surface element j and varies from 0 to 50,000 Watts. The graph plots the radiation term of surface k for four different temperature values of j .

The x-axis in Figure 73 varies from 0 Watts to the highest observed value of 50,000 Watts. As shown in Figure 73, the largest amount of energy lost is 46 Watts from surface k to j , when the temperature of surface j is 240K. On the other hand, 48 Watts are radiated from surface j to surface k , when the temperature of surface j is 325K. This is a relatively small value compared to the thousands of Watts exchanged between elements. Therefore, the view factor value criteria are set to 0.0005.

Results. For this case, a view factor criterion is set at 0.0005 and the distance criterion remains at 250 meters. Table 14 shows the results after the view factor trim routine is applied.

Table 14. Results for the model of Bozeman Pass with the distance criterion set to 250-meters and the view factor criteria set to 0.0005. For this case, 288,308 elements are deleted.

Distance Criterion:	250 meters
View Factor Criterion 1:	0.0005
View Factor Criterion 2:	0.0005
# Elements of Interest	4354
# Surrounding Elements	323256
# Elements within Dist Criterion	34306
# Elements (R = 2)	29706
# Elements (R = 3)	77
# Elements (R = 4)	4600
# Elements (R = 5)	565
# Elements to be Deleted (R = 1)	288308

The number of elements deleted is significantly changed from the previous run. The total number of elements eliminated in this run is 288,308 and therefore, the remaining model has only 39,302 elements. Figure 74 shows the resulting model with the above criteria in place.

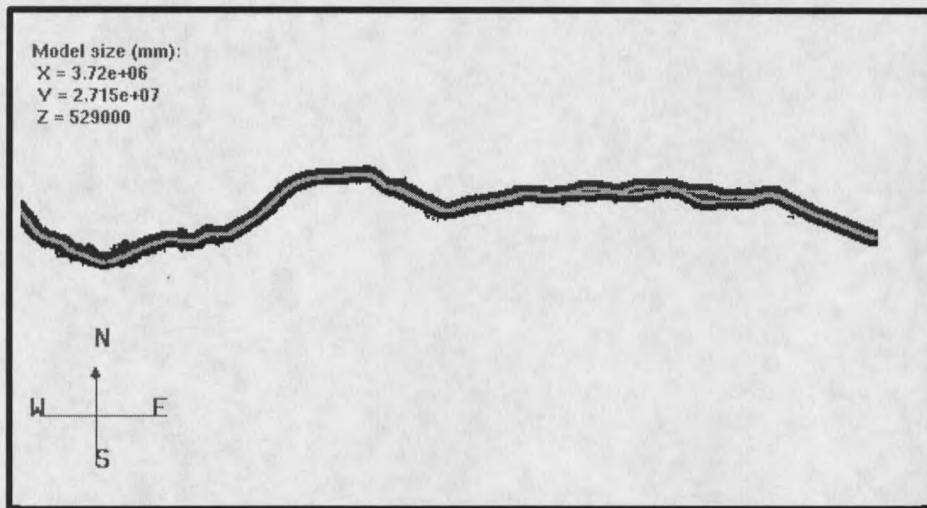


Figure 74. The resulting model of Bozeman Pass with the distance criterion set to 250 meters and the view factor criteria set to 0.0005. 288,308 surface elements have been deleted and 39,302 surface elements remain in the model.

At this point, the map has been scaled to a size RadTherm/RT can handle and the next step is to determine the accuracy of the new model with considerably less elements. To accomplish this, the temperature output of the interstate for the original model will be compared to the temperature output of the same elements in the new model with fewer elements. Due to the limitation of the computer's memory capacity used to run RadTherm/RT, the thermal solution for Bozeman Pass with 327,610 elements cannot be completed. Therefore, a smaller section of Bozeman Pass is used for the validation. This

section is known as Rocky Canyon and is the most topographically complex terrain of Bozeman Pass. For this reason, it is an excellent section to validate the accuracy of the methodology.

Model of Rocky Canyon

Validation is a very important part of the modeling process. Without validation the accuracy of the methodology is unknown. A smaller section of Bozeman Pass is modeled in order to produce a map with fewer elements. Therefore, the thermal solution can be calculated for the original map and then compared to the newly created model with deleted elements.

Coordinates for Rocky Canyon Model

The section of Bozeman Pass used is Rocky Canyon for which the Interstate spans from mile marker 314½ to mile marker 318. Figure 75 shows the area used including the mile markers on Interstate 90. To better illustrate the canyon in Figure 75, a 3-dimensional image is used. Figure 76 shows a view of Interstate 90 through Rocky Canyon from mile marker 314 heading east.

The UTM coordinates for this section of Rocky canyon are North: 5055500, East: 0511500, South: 5053000, and West: 0505600. Again, the ASCII files from GIAC for Bozeman Pass are trimmed to the UTM coordinates listed above using the gis_manip PERL script. Using the ASCII files a new project is created in RadTherm/RT, which has 33,096 surface elements as shown in Figure 77.

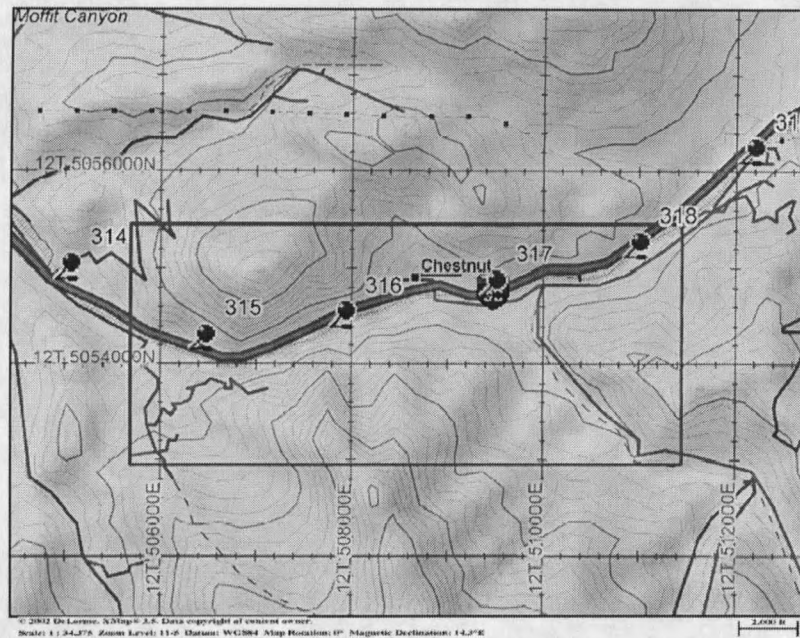


Figure 75. The region selected for the model of Rocky Canyon. The interstate spans from mile marker 314½ to mile marker 318.



Figure 76. A 3-dimensional map of Interstate 90 through Rocky Canyon from mile marker 314 heading east.

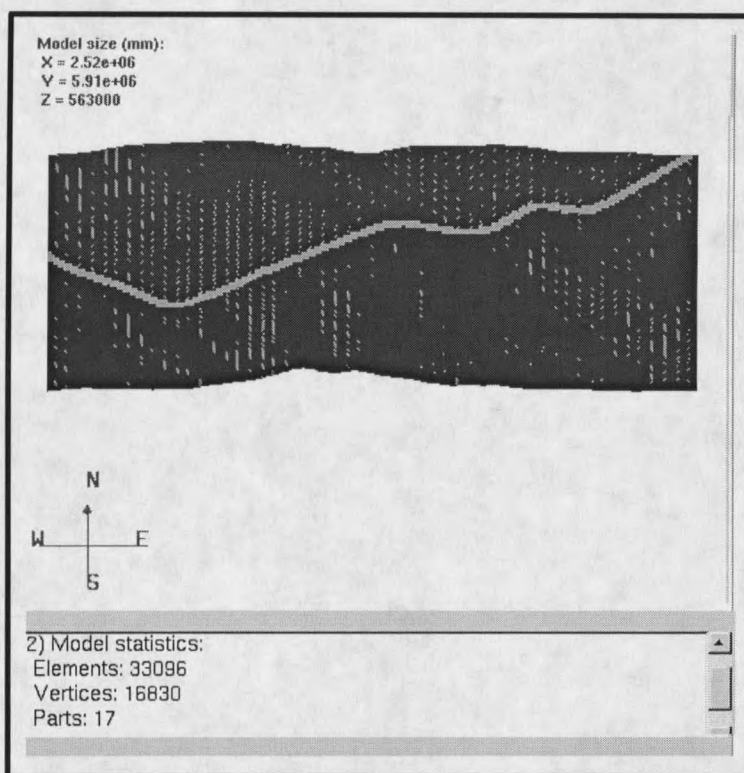


Figure 77. RadTherm/RT model for Rocky Canyon consisting of 33,096 surface elements.

The identical procedure is performed on Rocky Canyon as applied to Bozeman Pass. The distance criterion is set at 0 meters and the view factor criteria were set at 0.0005. The results for Rocky Canyon after applying the view factor trim routine are listed in Table 15. As shown in Table 15, the elements of interest ($R=0$) that make up Interstate-90 are composed of 728 surface elements. The resulting output file from Step 6 contains 27,319 surface elements to delete. The model of Rocky Canyon remains with 5,777 surface elements as illustrated in Figure 78.

Table 15. Results for Rocky Canyon using the view factor trim routine with the distance criteria set to 0 and the view factor criteria set to 0.0005.

Distance Criterion:	0
View Factor Criterion 1:	0.0005
View Factor Criterion 2:	0.0005
# Elements of Interest	728
# Surrounding Elements	32368
# Elements within Dist Criterion	0
# Elements (R = 2)	0
# Elements (R = 3)	2675
# Elements (R = 4)	0
# Elements (R = 5)	2374
# Elements to be Deleted (R = 1)	27319

Assign RadTherm/RT Part ID.

Since the two comparison models have been created, the next step is to assign properties for each part in RadTherm/RT. As mentioned in Chapter 3 the land cover class assigned to each element in the vegetation map has to be assigned to a RadTherm/RT part ID. Since RadTherm/RT is only capable of a maximum of twenty parts, the land cover classification system has to be reduced. Also note, a part must be reserved for the "elements to delete" after the procedure in Chapter 3 is complete. Because RadTherm allows a maximum of 20 parts, the terrain classification must be limited to at most nineteen parts.

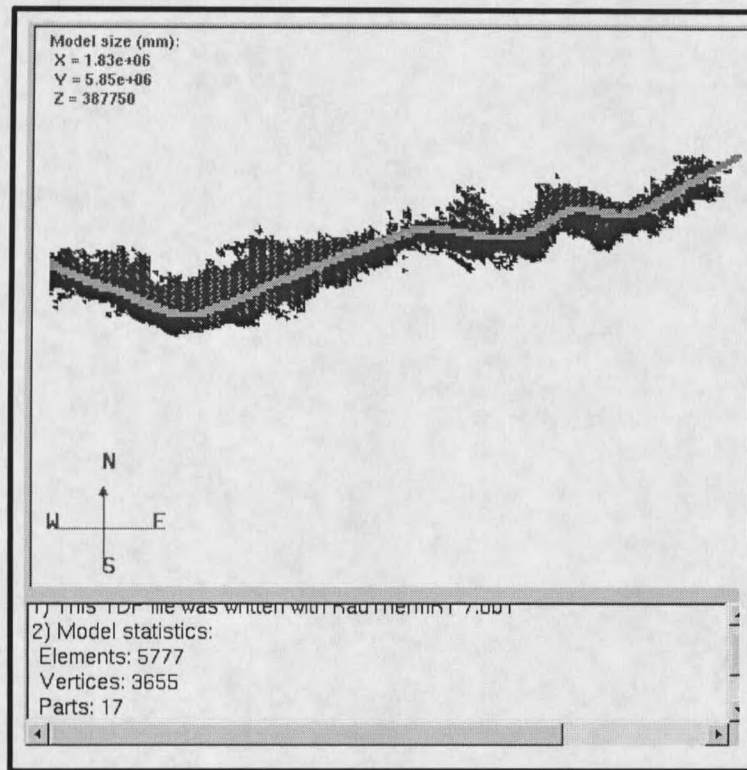


Figure 78. RadTherm/RT model for Rocky Canyon after the view factor trim routine is applied. The distance criterion was set at zero and the view factor criteria were set at 0.0005. This model consists of 5,777 surface elements.

Originally, the land cover class was simply truncated to create the new part ID. For instance, a Ponderosa Pine with a land cover classification of 4206 was assigned to part number 4 and Sagebrush with a land cover classification of 3350 became part number 3. In order to more effectively create RadTherm/RT part ID's the land cover class was grouped based on description of the part as well as its published albedo value [Green et al, 1999]. Table 16 shows the land classification and its assigned albedo value. Then, each land cover class is assigned to a RadTherm/RT part number. The land cover types in Rocky Canyon generated a total of seventeen parts.

Table 16. Land cover classifications mapped to RadTherm/RT part ID's.

Land Cover Class Code	Land Cover Class	Albedo Values	RadThermRT Part ID's	Modeled as in RadThermRT
1000	Highway East	default	1	Asphalt, Interstate
1001	Highway West	default	2	Asphalt, Interstate
2010	Agricultural Lands - Dry	0.20	4	Foliage, Short Grass
2020	Agricultural Lands - Irrigated	0.18	4	Foliage, Short Grass
3101	Foothills Grassland	0.26	5	Foliage, Short Grass
3110	Altered Herbaceous	0.26	5	Foliage, Short Grass
3130	Very Low Cover Grasslands	0.25	5	Foliage, Short Grass
3150	Low/Moderate Cover Grasslands	0.26	5	Foliage, Short Grass
3170	Moderate/High Cover Grasslands	0.16	6	Foliage, Tall Grass
3180	Montana Parklands and Subalpine Meadows	0.16	6	Foliage, Tall Grass
3210	Mixed Mesic Shrubs	0.15	7	Foliage, Tall Grass, Moderate Surface Moisture
3300	Mixed Xeric Shrubs	0.10	8	Foliage, Tall Grass, Dry
3310	Salt-Desert Shrub/Dry Salt Flats	N/A	8	Foliage, Tall Grass, Dry
3350	Sagebrush	N/A	8	Foliage, Tall Grass, Dry
3510	Mesic Shrub-Grassland Associations	0.15	7	Foliage, Tall Grass, Moderate Surface Moisture
3520	Xeric Shrub-Grassland Associations	0.10	8	Foliage, Tall Grass, Dry
4000	Low Density Xeric Forest	0.18	10	Foliage, Tall Grass
4140	Mixed Broadleaf Forest	0.20	11	Foliage, Deciduous Trees
4203	Lodgepole Pine	0.10	12	Foliage, Coniferous Trees
4205	Limber Pine	0.18	10	Foliage, Tall Grass
4206	Ponderosa Pine	0.10	12	Foliage, Coniferous Trees
4212	Douglas-fir	0.10	12	Foliage, Coniferous Trees
4214	Rocky Mountain Juniper	0.18	10	Foliage, Tall Grass
4223	Douglas-fir/Lodgepole Pine	0.10	12	Foliage, Coniferous Trees
4260	Mixed Whitebark Pine Forest	0.10	12	Foliage, Coniferous Trees
4270	Mixed Subalpine Forest	0.10	12	Foliage, Coniferous Trees
4290	Mixed Xeric Forest	0.10	12	Foliage, Coniferous Trees
4300	Mixed Broadleaf and Conifer Forest	0.15	14	Foliage, Mixed Trees
5000	Water	0.14	15	Water, First Principles
6110	Conifer Riparian	0.10	12	Foliage, Coniferous Trees
6120	Broadleaf Riparian	0.20	11	Foliage, Deciduous Trees
6130	Mixed Broadleaf and Conifer Riparian	0.15	14	Foliage, Mixed Trees
6200	Graminoid and Forb Riparian	0.16	6	Foliage, Tall Grass
6300	Shrub Riparian	0.18	6	Foliage, Tall Grass
6301	Deep Marsh	0.12	16	Swamp
6400	Mixed Riparian	0.18	9	Foliage, Tall Grass
7300	Rock	default	17	Soil, Rocky Field

Once the parts are created, the properties for each part are determined and assigned in the GUI of RadTherm/RT. Figure 79 shows a sample of the GUI for a foliage terrain part.

Part Selector

Name: Part_1 Number: 1 List... Add Part Delete Part

Total Parts: 17

Parts Assembly Curves Environ Conduct Properties

Temperature: ☐ Assigned ☒ Calculated

Part Type: Terrain

Type: ☐ Asphalt ☒ Foliage ☐ Snow ☐ Swamp ☐ Concrete ☐ Layered ☐ Soil ☐ Water

Foliage Type: ☒ Grass, Short ☐ Grass, Tall ☐ Trees, Coniferous ☐ Trees, Deciduous ☐ Trees, Mixed

Growth Factor: ☒ Growing ☐ Intermediate ☐ Dormant ☐ Other 0.0000

Cover Factor: ☐ No Foliage ☒ Intermediate ☐ Complete Coverage ☐ Other 0.7500

Solar Absorptivity: ☒ Modeled ☐ Measured

Surface Moisture: ☒ Default ☐ Dry ☐ Moderate ☐ Saturated ☐ Other 0.2000

Bulk Moisture: ☒ Default ☐ Dry ☐ Moderate ☐ Saturated ☐ Other 0.2500

Core Temperature (°C): 15.0000

Figure 79. An example of the RadTherm/RT GUI for a foliage terrain part.

A complete list of each part and their associated properties is shown in Table 17.

Table 17. Each RadTherm/RT part and the associated properties.

Part #	1,2
Part Type	Terrain
Type	Asphalt
Type	Interstate
Surface Cond	Aged
Wetness	Exposed
Part #	3,4,5
Type	Foliage
Foliage Type	Short Grass
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Surface Moisture	Default
Bulk Moisture	Default
Part #	6,9,10
Type	Foliage
Foliage Type	Tall Grass
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Surface Moisture	Default
Bulk Moisture	Default
Part #	7
Type	Foliage
Foliage Type	Tall Grass
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Surface Moisture	Moderate
Bulk Moisture	Default
Part #	8
Type	Foliage
Foliage Type	Tall Grass
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Surface Moisture	Dry
Bulk Moisture	Default
Part #	11
Type	Foliage
Foliage Type	Deciduous Trees
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Surface Moisture	Default
Bulk Moisture	Default

Part #	12
Type	Foliage
Foliage Type	Coniferous Trees
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Surface Moisture	Default
Bulk Moisture	Default
Part #	13
Type	Foliage
Foliage Type	Mixed Trees
Growth Factor	Dormant
Cover Factor	No Foliage
Solar Absorptivity	Modeled
Surface Moisture	Default
Bulk Moisture	Default
Part #	14
Type	Foliage
Foliage Type	Mixed Trees
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Surface Moisture	Default
Bulk Moisture	Default
Part #	15
Type	Water
Type	First Principles
Surface Atten.	Default
Solar Absorptivity	Modeled
Depth	1.0 meter
Part #	16
Type	Swamp
Dew Mass	Moderate
Growth Factor	Growing
Cover Factor	Intermediate
Solar Absorptivity	Modeled
Depth	1.0 meter
Part #	17
Type	Soil
Type	Rocky Field
Surface Moisture	Default
Bulk Moisture	Default
Solar Absorptivity	Modeled
Fraction Quartz Content	0.0

*note all Part Type's are Terrain

Location for Temperature Comparison

The temperature difference between the original model of Rocky Canyon and the newly created model of Rocky Canyon with 5,777 elements is compared. Seven different locations are used which include every ½ mile increment from mile marker 315 to mile marker 318 on Interstate 90. The easting and northing UTM coordinates for each location is listed in Table 18. In addition, the element number for the original map and trimmed map are shown.

Table 18. The temperatures are compared at each ½ mile between mile marker 315 and 318. The UTM coordinates and element numbers at listed for each location.

Mile Marker		UTM Coordinates	Original Map Element Number	Trimmed Map Element Number
315	Easting Northing	506274.121261 5054175.911120	17775	4545
315 1/2	Easting Northing	507054.121261 5054085.911120	19009	4964
316	Easting Northing	507774.121261 5054445.911120	14329	3310
316 1/2	Easting Northing	508504.121261 5054725.911120	10832	1903
317	Easting Northing	509344.121261 5054755.911120	10494	1813
317 1/2	Easting Northing	510034.121261 5054995.911120	7388	810
318	Easting Northing	510814.121261 5055145.911120	5470	379

Forecasted meteorological data for one day in the summer and winter are used in this section. First, the temperature difference between the original and trimmed map are compared using a forecasted weather file from July 2nd, 2003. Next, a forecasted weather file from February 2nd, 2003 is applied. This tests the accuracy of the methodology for two contrasting months.

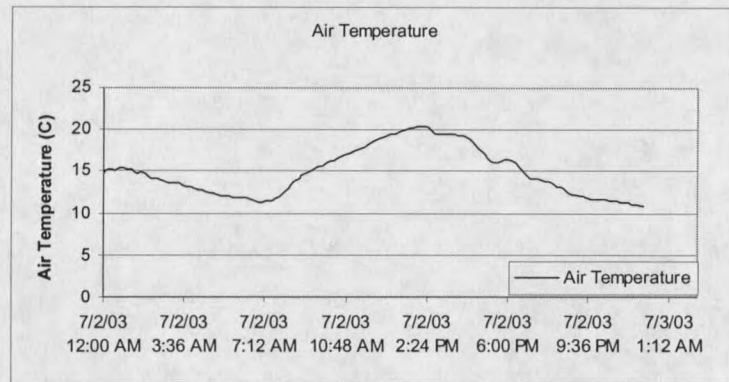
Summer Test Case – July 2nd, 2003

The first case utilizes a weather file for July 2nd, 2003. The conditions for the day are discussed in the following section. Next, the results are shown for the interstate temperature between the original Rocky Canyon model and the Rocky Canyon model with deleted elements. Seven locations are compared, which are at ½ mile increments from mile marker 315 to mile marker 318.

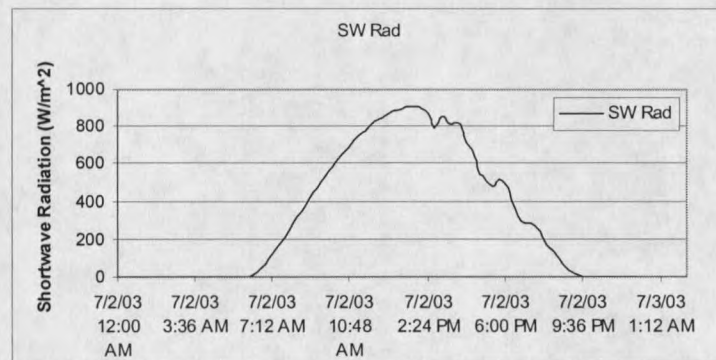
Weather File - Summer. The morning of July 2nd, 2003 is fairly warm, with the average air temperature, from 12:00 AM to 6:00 am, of 14.35°C. The sunrise is at 6:21 am for this particular morning and the sunset at 10:34 pm. At sunrise, the temperatures begin to steadily increase and the average temperature between 6:00 am and 2:00 pm is 19.10°C. As shown in Figure 80a, the temperature remained fairly constant from 2:00 pm until 7:30 pm, with an average temperature of 24.88°C.

Figure 80b is a plot of the measured solar radiation (W/m^2) versus time of day. The solar radiation peaked at 2:30 pm at 982.5 W/m^2 . The decrease in solar radiation between 2:00 pm and 4:00 pm and again at 6:30 pm to 10:00 pm was probably due to the

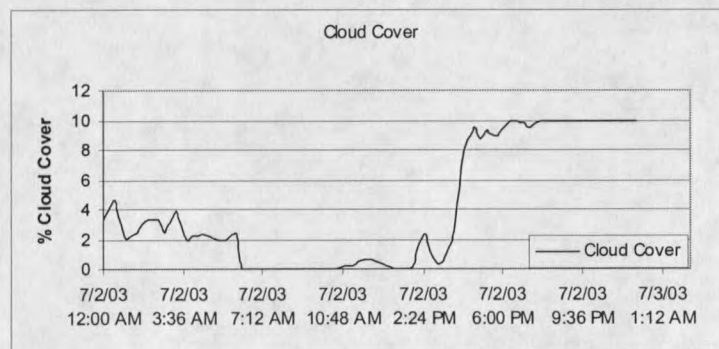
increase of cloud cover between these periods. Figure 80c shows the cloud cover throughout the day.



(a)



(b)



(c)

Figure 80. Meteorological Plots from 12:00 am on July 2nd, 2003 for (a) air temperature vs. time (b) solar radiation vs. time (c) cloud cover verse time of day

Temperature Comparison – Mile Marker 315. The surface temperature of particular elements for the original Rocky Canyon map and the map of Rocky Canyon with deleted elements is compared. Table 18 lists the elements location at each mile marker. The results for element 17775-verse element 4545 are shown in Figure 81.

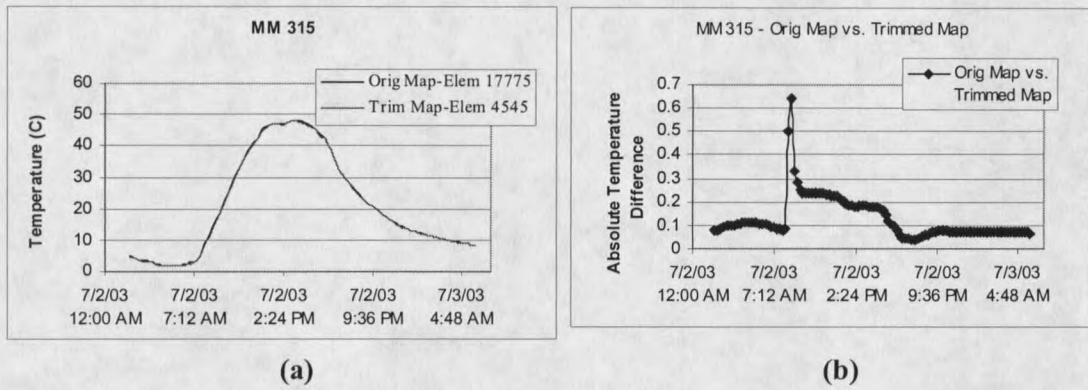


Figure 81. (a) Temperature comparison for element 17775 (original map) versus element 4545 (map with deleted elements) for July 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

As shown in Figure 81b, there is not a major temperature variation between the models. The maximum temperature during the 24-hour period is 48°C, which occurs at 3:15 PM. A difference of 0.64°C is the maximum temperature difference between the models and occurs at 8:45 AM. By taking the absolute value of the temperature difference between the models, the average difference is 0.127°C. Therefore, there is not a significant difference in temperature throughout the 24-hour period.

Temperature Comparison – Mile Marker 315½. The next comparison is at mile marker 315½ on Interstate 90. The temperature comparison for both models is shown in Figure 82.

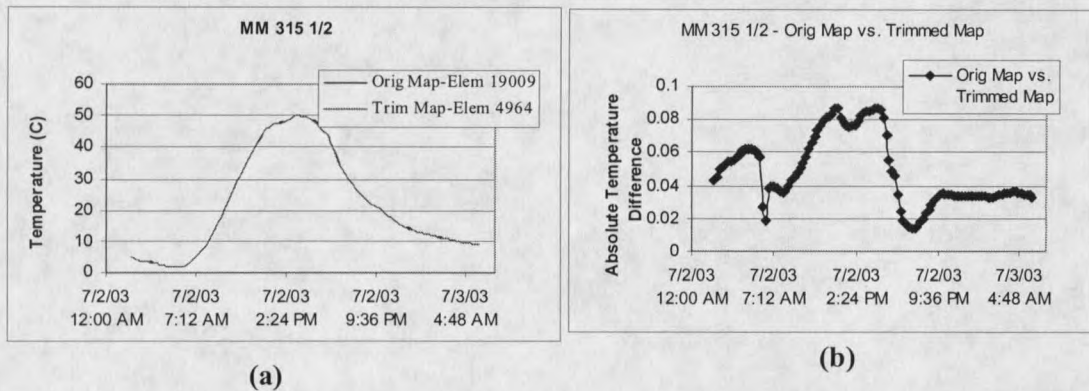


Figure 82. (a) Temperature comparison for element 19009 (original map) versus element 4964 (map with deleted elements) for July 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

Once again, there is very slight temperature difference between the two models. Figure 82a follows the same trend as Figure 81 and the maximum temperature occurs at 3:15 PM, which is 49.9°C. For this case, the maximum temperature difference is very low at 0.087°C (Figure 82b). The average absolute temperature difference throughout the day is 0.048°C.

Temperature Comparison – Mile Marker 316. The temperatures between the two models at mile marker 316 are shown in Figure 83.

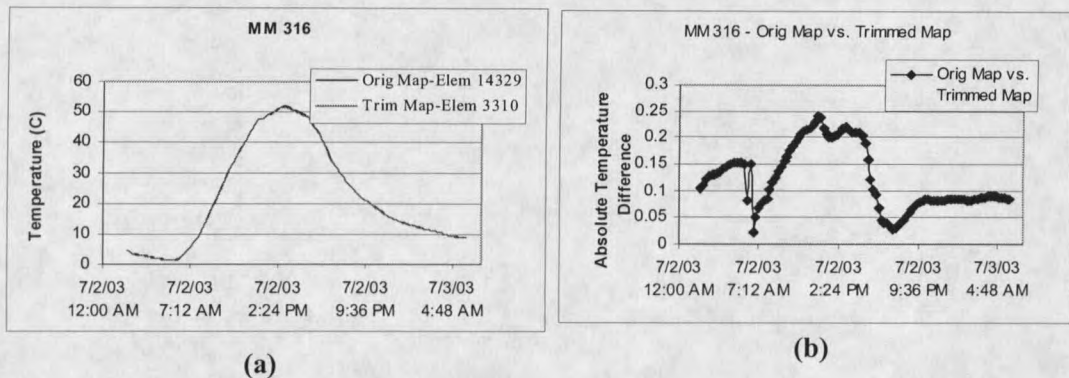


Figure 83. (a) Temperature comparison for element 14329 (original map) versus element 3310 (map with deleted elements) for July 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

At mile marker 316 the maximum temperature occurs at 2:45 PM and is 51.7°C (Figure 83a). The greatest temperature difference between the models is 0.24°C as shown in Figure 83b. For the 24-hour period, the average absolute temperature difference is 0.12°C.

Temperature Comparison – Mile Marker 316½. Figure 84b shows the temperature difference between element 10832 and element 1903 for mile marker 316½. The maximum temperature for the 24-hour period occurs at 3:00 PM and is 51.2°C (Figure 84a). In Figure 84b, the maximum temperature difference between the two elements is 0.326°C. Again, the average absolute temperature difference is very small at 0.14°C.

Temperature Comparison – Mile Marker 317. At mile marker 317 the temperature for element 10494 for the original Rocky Canyon model is compared to element 1813 for the model with deleted elements (Figure 85a).

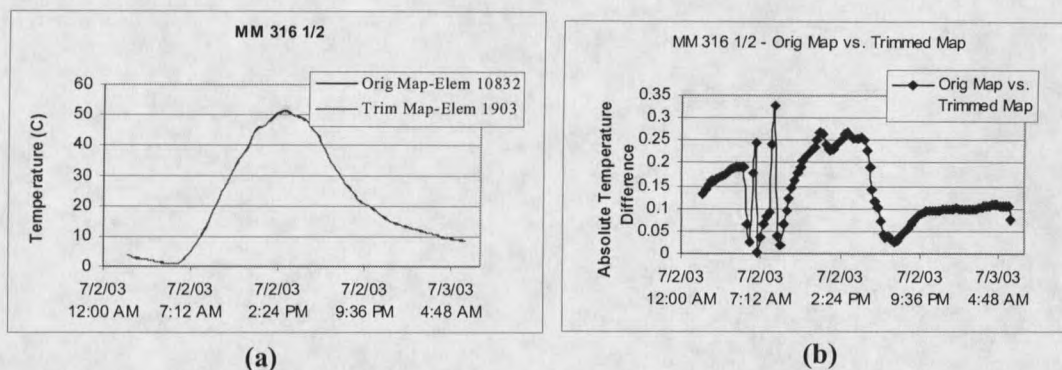


Figure 84. (a) Temperature comparison for element 10832 (original map) versus element 1903 (map with deleted elements) for July 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

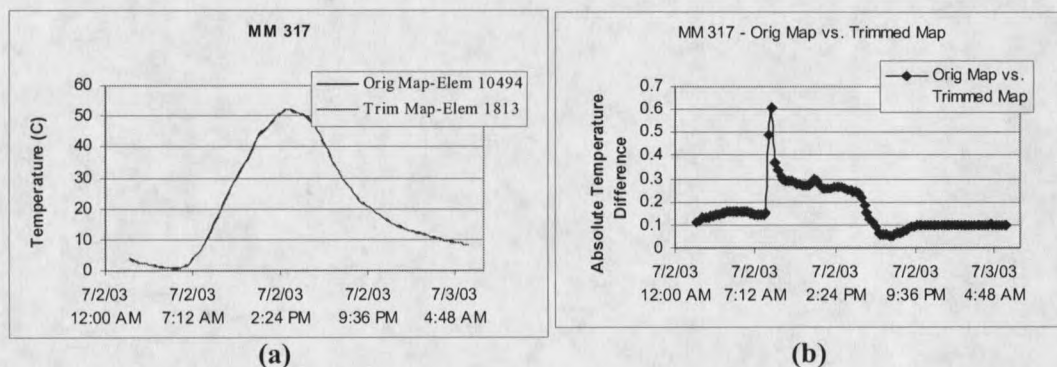


Figure 85. (a) Temperature comparison for element 10494 (original map) versus element 1813 (map with deleted elements) for July 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

The maximum temperature is 52.0°C and occurs at 3:00 PM as shown in Figure 85a. Throughout the entire day, the maximum temperature difference is 0.61°C (Figure 85b). This is slightly higher than for the previous cases, but the average absolute temperature difference remains low at 0.16°C.

Temperature Comparison – Mile Marker 317½. The temperature comparison between element 7388 (original model) and element 810 (model with deleted elements) is shown in Figure 86a.

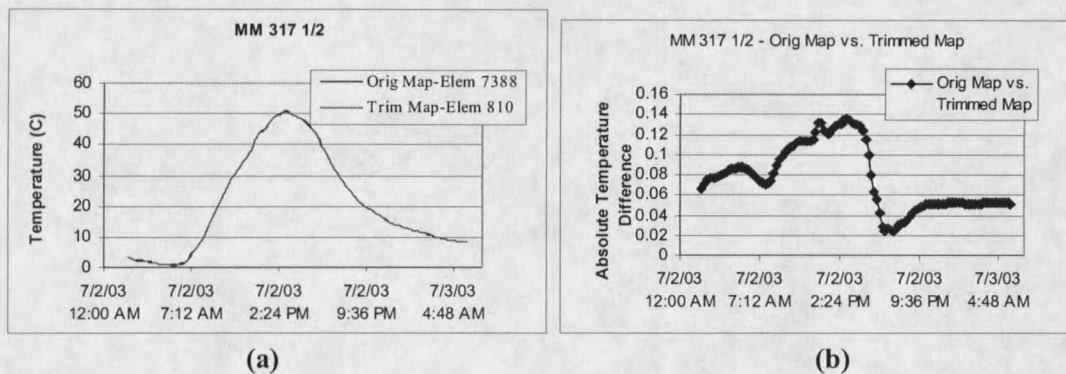


Figure 86. (a) Temperature comparison for element 7388 (original map) versus element 810 (map with deleted elements) for July 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

For the 24-hour period, the maximum temperature is 50.74°C, which occurs at 3:00 PM. In this case, the maximum temperature difference is very low at 0.136°C and the average absolute temperature difference is 0.076°C (Figure 86b).

Temperature Comparison – Mile Marker 318. Figure 87a shows the temperature difference between the two models at mile marker 318. A maximum temperature of 50.0°C occurs at 2:45 PM. During the 24-hour period, the maximum temperature difference is very small at 0.167°C (Figure 87b). The average absolute temperature difference for Figure 87b is 0.095°C. The difference between the temperatures indicates the topographic influence in the canyon.

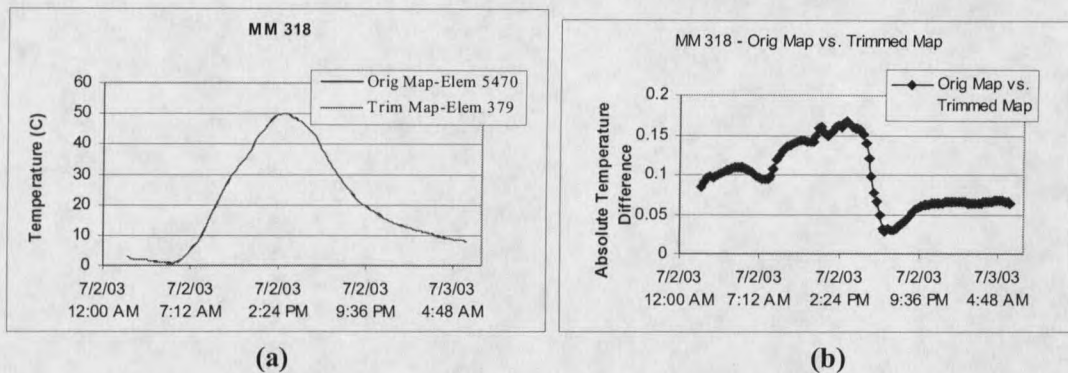


Figure 87. (a) Temperature comparison for element 5470 (original map) versus element 379 (map with deleted elements) for July 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

Winter Test Case – February 2nd, 2003

For the second validation case, a weather file is used from February 2nd, 2003. The meteorological conditions for the day are discussed in the next section. The results between the original Rocky Canyon model and the Rocky Canyon model with fewer elements are discussed in the following section. As in the previous case, the interstate nodes are compared for each ½ mile increments between mile marker 315 and mile marker 318.

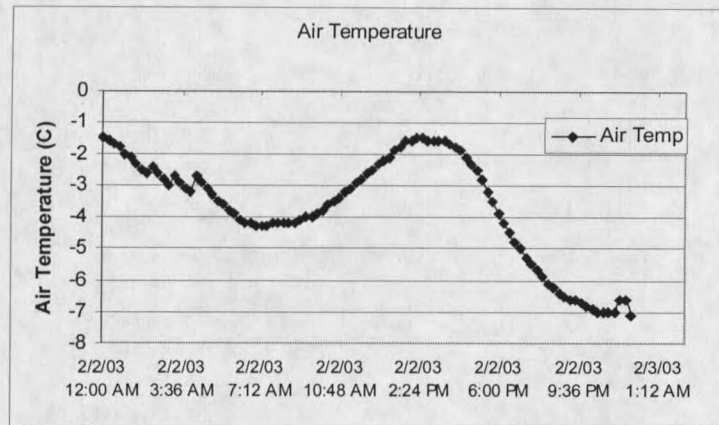
Weather File - Winter. The air temperature for February 2nd, 2003 was fairly cool with a minimum temperature of -7.7°C and a maximum temperature of -1.6°C throughout the entire day (Figure 88a). The sunrise for this day is at 6:32 am and the sunset is at 6:43 pm. The temperature did not fluctuate significantly from 12:00 am to 11:00 am with an average temperature of -3.07°C . Between 11:00 am and 4:30 pm the

temperatures varied due to the change in cloud cover causing an increase and decrease in solar radiation.

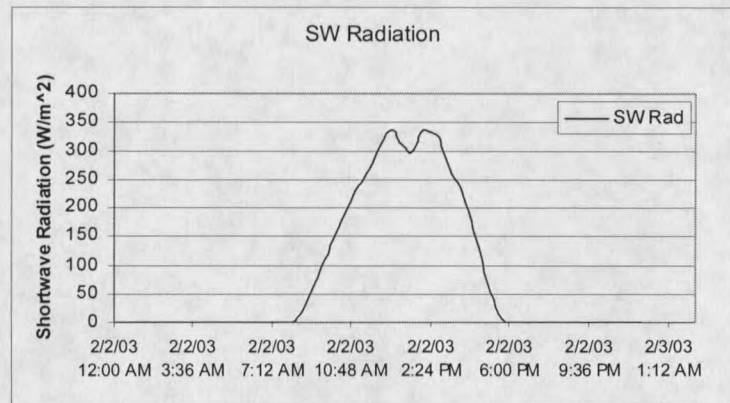
Figure 88b is a plot of the measured solar radiation (W/m^2) versus time of day. The solar radiation peaked at 12:15 pm at 720.3 W/m^2 . The sudden spike in solar radiation from 11:00 am to around 1:00 pm was probably due to the sudden decrease in cloud cover. Figure 88c shows the cloud cover throughout the day.

Temperature Comparison – Mile Marker 315. The surface temperature of an element at mile marker 315 for the original Rocky Canyon model is compared to an element in the identical location on the new model for Rocky Canyon with fewer elements. The UTM coordinates for the elements location at each mile marker as well as the element number is shown in Table 18. The results for element 17775 compared to element 4545 at mile marker 315 are shown in Figure 89. As shown in Figure 89b, there is not a major temperature variation between the models. The maximum temperature during the 24-hour period is 4.36°C , which occurs at 2:15 PM (Figure 89a). At 1:30 PM there is a maximum temperature difference of 0.048°C between the models (Figure 89b). By taking the absolute value of the temperature difference between the models, the average difference is 0.030°C . This shows there is not a significant difference in surface temperature after eliminating over 25,000 elements.

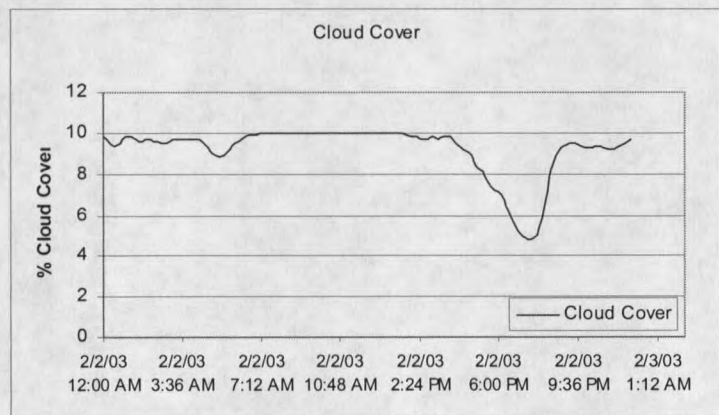
Temperature Comparison – Mile Marker 315½. The results for mile marker 315½ are very similar to the results for mile marker 315. The temperature comparison between interstate element 19009 and 4964 are shown in Figure 90a.



(a)



(b)



(c)

Figure 88. Meteorological Plots from 12:00 am on February 2nd, 2003 for (a) air temperature vs. time (b) solar radiation vs. time (c) cloud cover verse time of day.

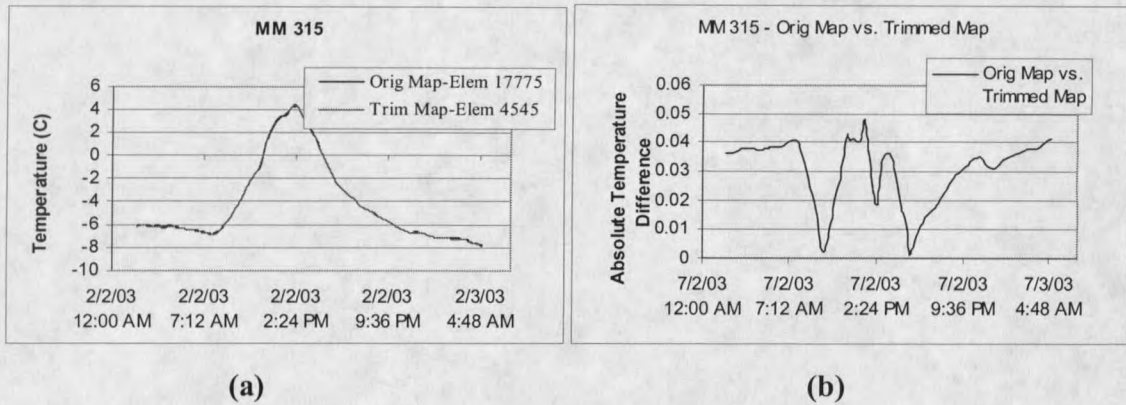


Figure 89. (a) Temperature comparison for element 5470 (original map) verse element 379 (map with deleted elements) for February 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

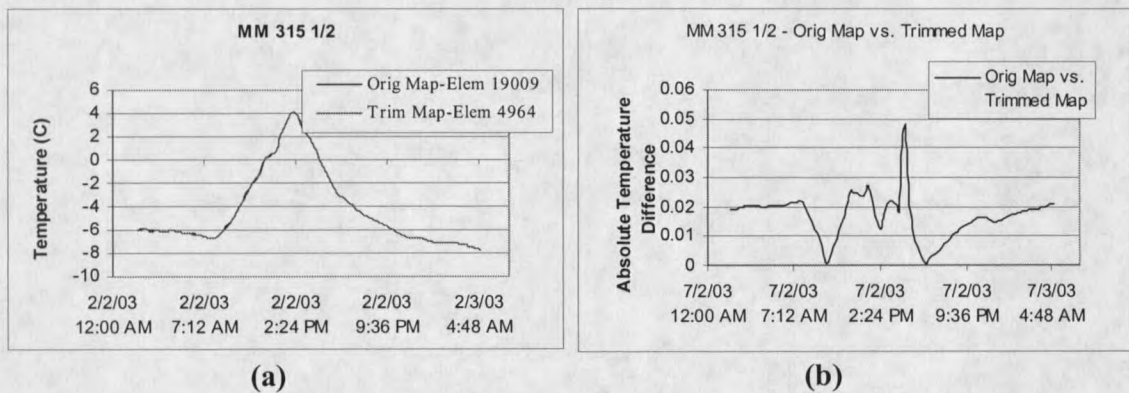


Figure 90. (a) Temperature comparison between element 19009 (original map) verse element 4964 (map with deleted element) for February 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

The maximum temperature occurs at 2:15 PM and is 4.18°C. The maximum temperature difference during the 24-hour period is 0.047°C (Figure 90b). Once again, the average absolute temperature difference is very low at 0.016°C.

Temperature Comparison – Mile Marker 316. The temperatures between the elements at mile marker 316 are shown in Figure 91a.

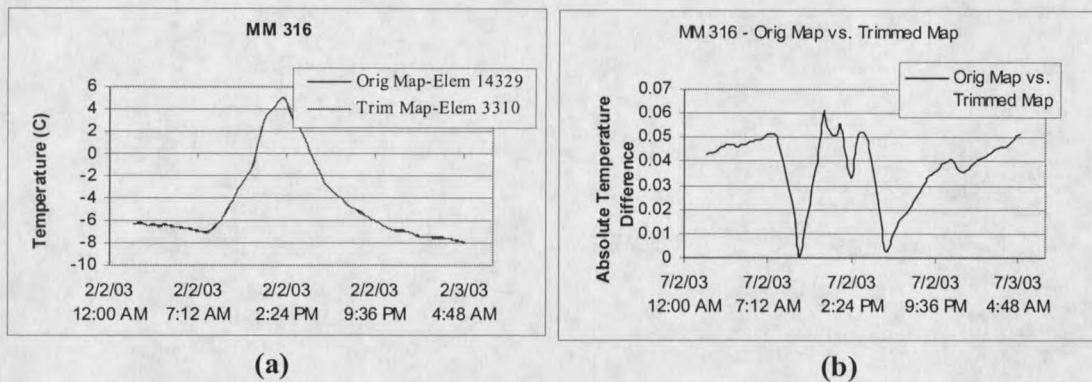


Figure 91. (a) Temperature comparison between element 14329 (original map) verse element 3310 (map with deleted element) for February 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

At mile marker 316 the maximum temperature is 5.05°C, which is slightly higher than at mile marker 315½, and occurs at 2:00 PM. The greatest temperature difference between the models is 0.06°C (Figure 91b). For the 24-hour period, the average absolute temperature difference is 0.038.

Temperature Comparison – Mile Marker 316½. At mile marker 316½ the temperature difference between elements 10832 and 1903 are shown in Figure 92a. The maximum temperature occurs at 2:00 PM and is 4.71°C. In Figure 92b, the maximum temperature difference between the two elements is 0.076°C. Again, the average absolute temperature difference is insignificant at 0.046°C.

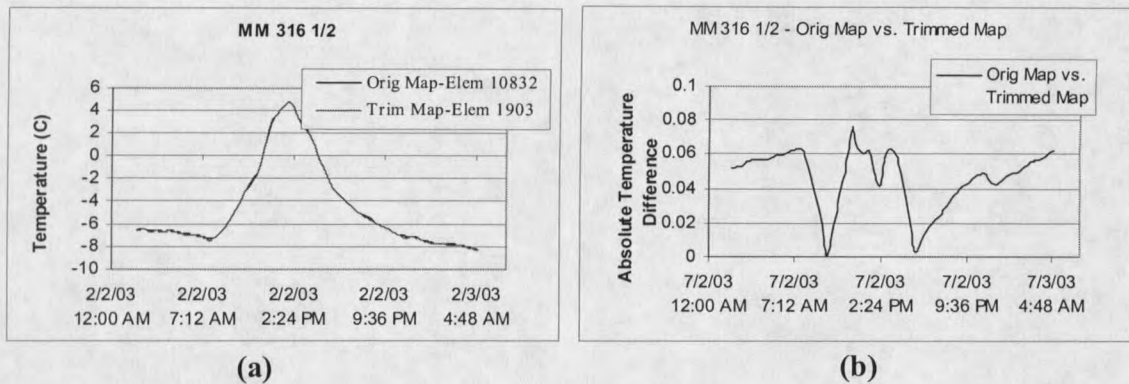


Figure 92. (a) Temperature comparison between element 10832 (original map) verse element 1903 (map with deleted element) for February 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

Temperature Comparison – Mile Marker 317. At mile marker 317 the temperature for element 10494 for the original Rocky Canyon model is compared to element 1813 for the model with deleted elements (Figure 93a).

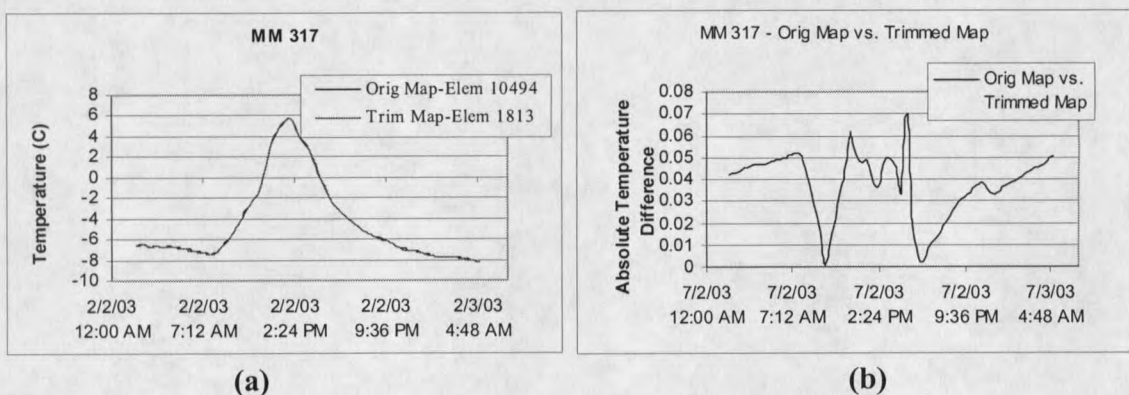


Figure 93. (a) Temperature comparison between element 10494 (original map) verse element 1813 (map with deleted element) for February 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

The maximum temperature is 5.78°C and occurs at 2:00 PM. Throughout the entire day, the maximum temperature difference is 0.069°C as shown in Figure 93b. The average absolute temperature difference remains small at 0.037°C .

Temperature Comparison – Mile Marker 317½. The temperature comparison between element 7388 (original model) and element 810 (model with deleted elements) is shown in Figure 94a.

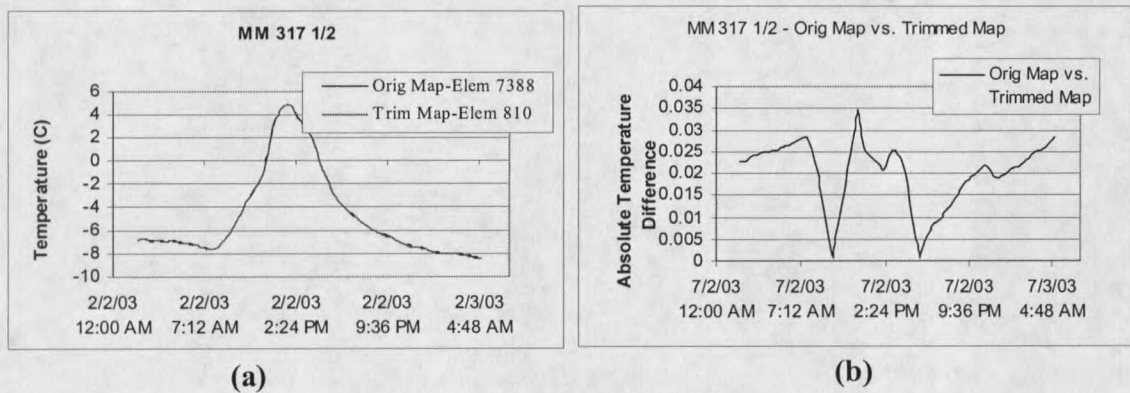


Figure 94. (a) Temperature comparison between element 7388 (original map) verse element 810 (map with deleted element) for February 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

Throughout the day, the maximum temperature is 4.85°C , which occurs at 1:45 PM. The maximum temperature difference remains low at 0.034°C (Figure 94b). The average absolute temperature difference during the 24-hour period is very low at 0.020°C .

Temperature Comparison – Mile Marker 318. Figure 95a shows the temperature difference between the two models at mile marker 318. A maximum temperature of 4.71°C occurs at 1:30 PM.

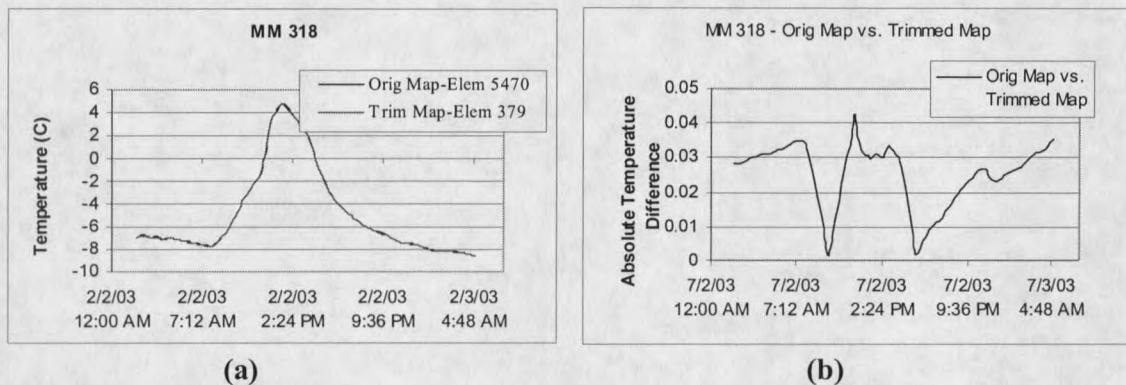


Figure 95. (a) Temperature comparison between element 5470 (original map) verse element 379 (map with deleted element) for February 2nd, 2003. (b) Difference in Temperature between the original map and the trimmed map.

During the 24-hour period, the maximum temperature difference is 0.04°C (Figure 95b). The average absolute temperature difference throughout the day is 0.025°C.

The results indicate that eliminating about 80% of the elements in the original model did not significantly change the resulting temperatures of the interstate. The computer computation time to create the view factor file and calculate the thermal solution for the original Rocky Canyon model was 4 hours and 5 minutes. To generate the view factor file and calculate the thermal solution for the Rocky Canyon model with deleted elements requires 27 minutes of computation time. This is a decrease in 3 hours and 38 minutes.

CHAPTER 5

SUMMARY/CONCLUSIONS

In this thesis an analytical approach was used to optimize the number of elements required to produce accurate thermal maps utilizing RadTherm/RT. RadTherm/RT is a thermal model that uses all modes of heat transfer to predict road temperatures. The goal is to maximize the length of highway modeled and minimize the computation time. A methodology is implemented using a 4-part C/C++ algorithm to select particular elements to delete. The section of code that specifically determines the elements to delete is referred to as the view factor trim routine.

Initial steps are performed in order to obtain the information needed to determine the elements to delete. First, two files are necessary to create a new project in RadTherm/RT. Montana State University's Geographic Information and Analysis Center produced the ASCII files for elevation, road and vegetation maps using ARC/INFO GIS software. These files are loaded into RadTherm/RT to create a new project (.tdf). The project is executed to produce the view factor file (.vfs). Geometric view factors are used as the basis to eliminate elements. Radiation view factors are used in ThermoAnalytics' software to determine the intensity of radiation of each element to every other element in the model and to the environment. This allows the terrain model to include full effects of solar shadowing, multiple reflections, and reradiation of geometric objects. The geometric view factors of each element are calculated using ThermoAnalytics Ray Trace Algorithm and the results are stored in the view factor file (.vfs).

Two text files are required as input into the view factor trim routine. The first file contains the view factor information. This file is obtained by applying an algorithm written by ThermoAnalytics. The routine converts the view factor file (.vfs), which is in HDF format, to a text file. A second file is obtained by applying a routine that retrieves each element and its attributes, which includes the element number, part number, UTM coordinates, and elevation.

The view factor trim routine uses the element attributes and view factor information text files to determine which elements to delete from the model. The routine was developed to have the capability of designating any part(s) as the elements of interest, which are the elements of primary interest for temperature. Furthermore, the values used for the criteria to determine which elements to delete can be modified. The first criterion is the distance from the elements of interest, where all elements that meet this criterion are kept. The next criterion determines the minimum view factor value to keep an element that directly exchanges radiation with an element of interest. Similarly, a minimum view factor criterion is set for elements that emit to or receive energy from elements that have a direct radiation exchange to the elements of interest. The result from the view factor trim routine is a text file that contains the elements to delete.

The final part of the code uses the text file that contains the elements to delete and creates a new project in RadTherm/RT. The primary difference in the new project is that it contains a new part composed of the elements to delete. This allows a user to view the elements, before deleting them.

Simple models were developed to describe the energy exchange between elements due to shortwave and longwave radiation. In addition, the view factors of particular elements are shown to describe the effects on surrounding elements. It was found that shadowing and reflection from solar radiation had a significant influence on the temperature distribution of the surrounding terrain. As expected, the amount of reflected energy increased as the surface absorptivity value decreased. The temperature influence on surrounding elements due to longwave radiation had a slight effect on the surrounding terrain, but not as influential as shortwave radiation.

A model of Bozeman Pass from mile marker 313 to mile marker 331 on Interstate 90 was created to show that the methodology was successful in eliminating elements. The project consisted of 327,610 surface elements and the goal was to reduce the map to between 30,000 to 60,000 surface elements. Different values were applied to the distance criteria and view factor criteria and the results were summarized. The optimal values chosen were 250 meters for the distance criteria and 0.0005 for the view factor criteria. These values produced a model of Bozeman Pass that contained 39,302 elements.

Since the original model of Bozeman Pass required a large amount of computer memory and disk space, it is often difficult for a computer to calculate the thermal solution. A small enough model was required in order to compare the trimmed model with the original model. Therefore, the thermal solution was calculated for the original model and compared to the new model with selected elements deleted. The region modeled from mile marker 314½ to mile marker 318 is known as Rocky Canyon, which contains topographically complex terrain. For this reason, it is a good section for testing

the accuracy of the methodology. The model of Rocky Canyon was originally composed of 33,096 surface elements. After the view factor trim routine was applied, 5,777 surface elements remained in the model. The view factor trim routine had the distance criteria set at zero and a view factor criteria set to 0.0005. The temperatures of each model were compared at ½ mile increments from mile marker 315 to mile marker 318 on interstate 90. The difference between the temperatures showed the effects of eliminating elements from the thermal map.

Forecast meteorological data for one day in the summer and one day in the winter were used to compare results. First, the difference between the original and trimmed map were compared using the weather file for July 2nd, 2003. Next, a weather file from February 2nd, 2003 was applied. This examines the accuracy of the methodology for two contrasting months.

The temperature plots at each ½ mile increment differed slightly for the summer model. This indicated a topographic influence in the canyon. The greatest temperature difference between models for all comparison locations occurred at mile marker 317 and was 0.61°C. This is almost double the second greatest temperature difference of 0.326°C at mile marker 316½. The greatest average absolute temperature difference for the 24-hour period was 0.16°C and occurred at mile marker 317. Mile marker 315½ had the smallest maximum temperature difference of 0.087°C and the lowest absolute average temperature difference for the entire day of 0.048°C.

The winter model had less temperature variation at each mile marker than the summer model. The greatest temperature difference was 0.076°C at mile marker 316½. The

largest absolute temperature difference for the entire day occurred at mile marker 316½ and was 0.046°C. The best results were at mile marker 317½ with a maximum temperature difference of 0.034°C. Mile marker 315½ had the smallest absolute temperature difference for the entire 24-hour period of 0.016°C.

These results show that after eliminating about 80% of the elements in the original model, the resulting temperatures of the interstate did not significantly change. The computer computation time to create the view factor file and calculate the thermal solution for the original Rocky Canyon model was 4 hours and 5 minutes. To generate the view factor file and calculate the thermal solution for the Rocky Canyon model with deleted elements required 27 minutes of computation time. This is a decrease of 3 hours and 38 minutes.

It was shown using simple models that surface temperatures were greatly influenced by shortwave radiation due to shadowing and reflected radiation from nearby elements. In addition, temperatures varied from emitted longwave radiation from surrounding terrain. Therefore, it was shown that surrounding terrain has a strong influence on the highway and would greatly affect the temperature if influential elements were removed.

The temperature difference between the original Rocky Canyon model and the Rocky Canyon model with deleted elements was very small considering Rocky Canyon is the most topographically complex terrain through Bozeman Pass. The greatest absolute average temperature difference between the two models was 0.64°C at mile marker 315 utilizing the July weather file. A difference in temperature between the two models indicates that there was a topographic influence through the Canyon. The overall results

conclude that the methodology was successful in using the concept of view factors to eliminate elements in a thermal map. The goals for this thesis to reduce the number of elements in the model, reduce the computer computation time, reduce user time, while maintaining accuracy of the thermal model were achieved.

REFERENCES CITED

1. Adams, E. E. *Proof of concept for prediction of pavement temperature: a tactical decision aid for highway safety*. Prepared for MDT with US DOT FHA, Montana State University, MT. 1999.
2. Adams, E. E., S. McDowell. *Thermal model for snow on three dimensional terrain*. Proc. Japan-U.S. Workshop on Snow Avalanche, Landslide, Debris Flow Prediction and Control. 1991.
3. Ballard, L. *Greater Yellowstone regional traveler and weather information system*. Scope of Work, version 1.0, June 1, 2001.
4. Beckwith, Thomas, Marangoni, Roy, Lienhard, John H. *Mechanical Measurements 5th edition*. Addison-Wesley Publishing Company. 1993.
5. Cengel, Y.A., M.A. Boles. *Thermodynamics: An engineering approach, 2nd edition*. McGraw-Hill, Inc. 1994.
6. Incropera, F.P., D.P. DeWitt. *Introduction to Heat Transfer, 3rd edition*. John Wiley & Sons. New York.
7. Marttila, Eric A. *A multi-mode heat transfer code for thermal analysis of automobile exhaust systems*. Partial fulfillment of Master's Thesis, Michigan Technological University. 1999.
8. Sparrow, E.M., R.D. Cess. *Radiation Heat Transfer*. Brooks/Cole Publishing Company. 1970.
9. Ozisik, Necati M. *Basic Heat Transfer*. McGraw-Hill Book Company. 1977.
10. Ozisik, Necati M. *Heat Conduction: Second Edition*. John Wiley & Sons, Inc. 1993.

REFERENCES CITED - CONTINUED

11. Pluss, C. *The energy balance over an alpine snowcover*. Zurcher Geographische Schriften, Heft 65. 1997.
12. Siegel, Robert, John R. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Publishing Corporation. 1981.
13. Tannehill, John C., Dale A. Anderson, Pletcher, Richard H. *Computational Fluid and Mechanics and Heat Transfer: Second Edition*. Taylor & Francis. 1997.
14. ThermoAnalytics, Inc. *Terrain Model Technical Manual*. Calumet, MI. 1999.
15. ThermoAnalytics, Inc. *WinTherm 6.0 Manual*. Calumet, MI. 2000.
16. ThermoAnalytics, Inc. *RadTherm Software*. Calumet, MI. 2002.

APPENDICES

APPENDIX A

C/C++ CODE – RETRIEVE ELEMENT ATTRIBUTES

MAKEFILE

```

# $Id: Makefile,v 1.1.1.1 2001/10/11 01:34:35 ladeanm Exp ladeanm $
# $Source: /apps/wxneve/AWMC_UTIL/TDF_SRC/Makefile,v $
# $Description: Makefile to update tdf.routines $
# $Log: Makefile,v $
# Revision 1.1.1.1 2001/10/11 01:34:35 ladeanm
# Expanded and updated to work with winthermrt-6.X
#

# List of Commands that can be compiled
SRC = tdfgetElemAttr tdfDeleteElem tdfupdatePart
# Each command can be compiled by typing, for example: make tdfupdatem

CC      = cc
CXX     = CC
CFLAGS  = -n32 -mips4 -LANG:bool=ON -woff
1021,1209,1233,1314,1355,1375,1506 -fullwarn -O2
CXXFLAGS = -n32 -mips4 -LANG:bool=ON -woff
1021,1174,1209,1233,1314,1355,1375,1506 -ptused -fullwarn -O2
LINK     = CC
LFLAGS  = -n32 -mips4

TDFIO = /home/mercury/apps/coe/RTWx/TAI/tdfio
INCLUDE = -I ${TDFIO}/include
LIBES = -L ${TDFIO}/lib/ -ltdfio -lm

${SRC}:  $$@.cpp
        . ${CXX} -o $@ $? ${CXXFLAGS} ${INCLUDE} ${LIBES}

# For discussion of makefile syntax, see \cite[p.21]{Oram-Talbot:1991}

# End of $RCSfile:$

```

C/C++ CODE

```

/!!
*  \file  tdfgetElemAttr.cpp
*  \brief Reads element data from a TDF file and writes
*         the patch num, elem number, part ID, UTM coords
*         and elevation to a text file.

*  To compile this program, use the command
*  $make
*  $tdfgetElemAttr.csh
*/

/*-----
HISTORY
DATE          07/23/2003 (last updated)
-----*/

/*-----
standard includes
-----*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/*-----
non-standard includes
-----*/
#include "tdfio.h"
#include "tdfpart.h"
#include "tdfelement.h"

/*-----
public function definitions
-----*/

TDF::TdfStatus GetElemInfo(char *datei, TdfIO &tdfio);

/*-----
*  main
*/
/!!
*  \brief  Main entry point for this program
*
*  \param argc  Number of command line arguments
*  \param argv  Array of command line arguments
*
*  \return  Returns 0 on successful completion, 1 on unsuccessful
*          completion
*/

int main(int argc, char **argv)
{
    // Check to make sure the command line looks like expected

```

```

if (argc != 3) {
    fprintf( stderr, "Usage: %s infile.tdf  outfile.dat \n",
        argv[0] );
    return 1;
}

// Read the TDF file
TdfIO tdfio;
TDF::TdfStatus status = tdfio.readFile( argv[1] );
if (status != TDF::Success) {
    fprintf( stderr, "Error reading TDF file %s\n", argv[1] );
}else {

    GetElemInfo(argv[2], tdfio);

}

return 0;

} /* End main() */

/*-----
 *  GetElemInfo
 */
/*!
 *  \brief Reads data from a TDF file and writes the
 *         data to a text file.
 *
 *  \param fp      Pointer to the opened ascii text file to write to
 *  \param tdfio   TdfIO class to retrieve the data from
 *
 *  \return Returns TDF::Success upon successful completion.
 *         Other return values indicate a failure occurred.
 */
TDF::TdfStatus GetElemInfo(char *datei, TdfIO &tdfio)
{
    TDF::TdfStatus status = TDF:: Success;

    // open the file
    FILE *fp = fopen(datei, "w");

    int nElem = tdfio.numberOfElements();
    TdfElement element;
    TdfPart part;

    int partID;
    int elemID;
    int patch;
    double x,y;
    float xx, yy, zz;

    int *partID_o = new int [nElem];
    int *elemID_o = new int [nElem];

```

```

    int *patch_o = new int [nElem];
    double *x_o = new double [nElem];
    double *y_o = new double [nElem];
    double *zz_o = new double [nElem];

    // Print Header Line
    fprintf(fp, ".Patch #, elemID, partID, xUTM, yUTM,
Elevation \n ");

    for (int i = 0; i<nElem; i++)
    {
    // Returns the element ID for the specified element index
        elemID = tdfio.elementID(i);
        elemID_o[i] = elemID;

    // retrieve the indexed element
        tdfio.element( elemID, element);

    // retrieve the part that the element belongs to
        partID = element.part();
        partID_o[i] = partID;

    // commented this out because I did not think it was necessary
    // tdfio.part(partID, part);

    // retrieve the view factor patch that the element belongs to
        patch = element.patch();
        patch_o[i] = patch;

    // Retrieve the UTM coordinates of each element
        status = tdfio.elementUTMposition(elemID, x, y);
        x_o[i] = x;
        y_o[i] = y;

    // Retrieve the Centroid of the Element
    // Use zz as elevation
        status = element.centroid(xx, yy, zz);
        zz_o[i] = zz;
    }

    for (i = 0; i<nElem; i++)
    {
        fprintf(fp, "%8i, %8i, %5i, %10.3f, %10.3f, %10.3f \n",
patch_o[i], elemID_o[i], partID_o[i], x_o[i], y_o[i], zz_o[i]);
    }

    fclose(fp);

    delete [] partID_o;
    delete [] elemID_o;
    delete [] patch_o;
    delete [] x_o;
    delete [] y_o;

```



```
delete [] zz_o;
```

```
    status = TDF:: Success;
```

```
return status;
```

```
}
```

.CSH FILE

```
#!/bin/csh
# Run-Time Load Library Path
setenv LD_LIBRARY_PATH /home/mercury/apps/coe/RTWx/TAI/tdfio
setenv LD_LIBRARY64_PATH /home/mercury/apps/coe/RTWx/TAI/tdfio
setenv LD_LIBRARYN32_PATH /home/mercury/apps/coe/RTWx/TAI/tdfio
setenv PATH /home/mercury/apps/coe/RTWx/bin:sbin:/apps/freeware/bin

env

### Extract data from TDF file ###
#~/TDF_SRC/tdfgettsfcm \ # Program to extract data
#BZH_2002081500Z.tdf \ # tdf filename
#BZH_2002081500Z.dat # Data Extracted

~/Test_case_bzpass_vf/Step3/tdfgetElemAttr \
BZH_test.tdf \
elemAttr_outfile.dat

# End of csh script
```

APPENDIX B

C/C++ CODE – VIEW FACTOR TRIM ROUTINE

MAKEFILE

```
driver:      fileio_demo.o element.o
            g++ -o driver element.o fileio_demo.o

fileio_demo.o:  element.cpp element.h fileio_demo.cpp
            g++ -c fileio_demo.cpp element.cpp

element.o:  element.cpp element.h
            g++ -c element.cpp
```

ELEMENT.H - HEADER FILE

```
// element.h
#ifndef ELEMENT_H_
#define ELEMENT_H_

//#include <cstring>

class Element
{
private:

public:
    int npatch;
    int elemID;
    int partID;
    // char prtname[256];
    // char prtname[20];
    float x;
    float y;
    float z;
    int rank;

    Element();

    Element(float xUTM, float yUTM);

    ~Element(); //destructor
};

#endif
```

ELEMENT.CPP - SOURCE FILE

```
// element.cpp

#include <cstring>
#include <iostream>
#include "element.h"

using namespace std;

Element::Element()
{
    npatch=-1;
    elemID=-1;
    partID=-1;
    x=0.0;
    y=0.0;
    z=0.0;
    rank = -1;
}

Element::Element(float xUTM, float yUTM)
{
    x = xUTM;
    y = yUTM;
}

Element::~Element() {}
```

FILEIO_CEMO.CPP - SOURCE FILE

```

// fileio_demo.cpp : Defines the entry point for the application.
//
// To Run: $make
//          $fileio_demo.csh

//stream libs and assert lib

#include <iostream>
#include <fstream>
//#include <cstdlib>
#include <stdlib.h>
#include <stdio.h>
#include <string>
#include <assert.h>

#include <math.h>

#include "element.h"

using namespace std;

// Element class - is in a seperate .h and .cpp file

int main(int argc, char *argv[])
{
    char str[256];
    char ch;

    //Check to make sure command line looks like expected
    if(argc!=4)
    {
        fprintf(stderr, "Usage:  %s elemAttr.dat vf_info.txt
deleted_elem_outfile.txt \n", argv[0]);
        return 1;
    }

    cout << endl << endl;

    // NOTE NOTE NOTE!!!!
    // Be sure that there is no empty space in the elemAttrFile
    // The first line should be the header, then the info

    char *elemAttrFile = argv[1];

    char *vfFile = argv[2];

    ifstream infile;

    //*****
    // Retrieve the linecount data of each file
    // compare the number of elements to be sure they match

```



```

//*****
// Open Element Attribute file
    infile.open(elemAttrFile);

// Start count at 1 so index and element # match
    int count=1;
    int linecount = 0;

    //make sure we can open the file.
    if(!infile)
    {
        cout << "Can not open element attribute infile" << endl;
    }

    //get the number of lines in the file
    while(infile.getline(str,255))
        linecount++;
    linecount--; //decrement by one to get rid of the EOF line
    cout << "Line count of Element Attribute Infile: " << linecount <<
endl;

    infile.close();
    infile.clear();    // necessary for some compilers

// Open second file and retrieve linecount2
    int linecount2=0;

    infile.open(vfFile);

    //make sure we can open the file.
    if(!infile)
    {
        cout << "Can not open view factor infile" << endl;
    }

    //get the number of lines in the file
    // while(infile.getline(str,255))
    //     linecount2++;
    // linecount2--; //decrement by one to get rid of the EOF line

    cout << "Line count of View Factor Infile: " << linecount2 <<
endl << endl;

    infile.close();
    infile.clear();    // necessary for some compilers

//*****
// Open View Factor File and determine the Number of Emitters
// This number should match the linecount
//*****
    int nEmitters;
    infile.open(vfFile);

    // find the Number of Emittersemmitter number

```

```

do{
    infile>>ch;
    }while(ch!=':');
    infile>>nEmitters;
    cout << "Number of Emitters in View Factor Infile: " <<
nEmitters << endl;
    cout << "Number of Emitters in Elem Attribute Infile: " <<
linecount << endl << endl;
    infile.close();
    infile.clear();    // necessary for some compilers

//*****
// Check to be sure linecount = nEmitters
// If not -- display error message
//*****
    if(nEmitters != linecount)
        cout << "ERROR: The number of elements in each file DO NOT
MATCH" << endl;

//*****
// Create elements object and load with each elements attributes
//
//*****
    //allocate memory for the elements
    // it is [linecount+1] so index matches element number (starts at
zero index)
    const int SIZE = linecount+1;
    Element* elements = new Element[SIZE];

    // reopen the file for reading
    // - not sure how to reset to the beginning of the file without
reopening
    infile.open(elemAttrFile);

    //parse out the first header line
    infile.getline(str,255);

    //loop over all elements in the file
    while(infile.peek() != EOF)
    {
        //make sure we don't write beyond the allocated memory
        assert(count<=linecount+1);

        infile>>elements[count].npatch;
        infile>>ch;//get rid of the comma

        infile>>elements[count].elemID;
        infile>>ch;//get rid of the comma
        infile>>elements[count].partID;
        infile>>ch;//get rid of the comma

    }

// This code is to read in the part name
// do{
//     infile>>ch;

```

```

//          if(ch!=' ',')
//          {
//              elements[count].prtname[i]=ch;
//              i++;
//          }
//      }while(ch!=' ',');

infile>>elements[count].x;
infile>>ch; //get rid of the comma
infile>>elements[count].y;
infile>>ch; //get rid of the comma
infile>>elements[count].z;
count++; //increment array index
//i=0;
}
infile.close(); //always close the file
infile.clear(); // reset infile

/*****
*
// STEP 3:
// Separate the elements object into two objects based on partID
//   surreElem and elemInt
// It will also assign an initial rank
//   elemInt (R = 0)
//   surreElem (R = 1)
//
/*****

// How many parts of interest are there????

// Determine the number of elements EOI (elements of interest)
int nElemInt = 0;
int nSurreElem;

/*
for (int i=1; i<=linecount; i++)
{
    if(elements[i].partID == 1)
    {
        nElemInt = nElemInt + 1;
    }
}

*/
for (int i=1; i<=linecount; i++)
{
    if((elements[i].partID == 1) || (elements[i].partID == 2))
    {
        nElemInt = nElemInt + 1;
    }
}

nSurreElem = linecount - nElemInt;

```

```

    cout << "Number of Elements of Interest: " << nElemInt << endl;
    cout << "Number of Surrounding Elements: " << nSurrElem << endl <<
endl;

    const int SIZE_1 = nElemInt;

    cout << "Create a new array of only Elements of Interest" << endl
<< endl;
    Element *elemInt = new Element[SIZE_1](0, 0);

// fill the 2 objects with the correct elements
    int count_1 = 1;
    for (int i=1; i<=linecount; i++)
    {
        if((elements[i].partID == 1) || (elements[i].partID == 2))
        {
            elements[i].rank = 0; // also create rank for elements
object
            elemInt[count_1].x = elements[i].x;
            elemInt[count_1].y = elements[i].y;
            count_1++;
        }
        else
        {
            elements[i].rank = 1; // also create rank for elements
object
        }
    }
    cout << "Elem Num: " << elements[i].elemID << " Elem Rank: " <<
elements[i].rank << endl;
}

//*****
*
// STEP 4:
// Determine if elements with R = 1 are within a specified distance
// of elements EOI
// At this point all surrElem should have a rank R = 1
//*****
*

float tempXdist, tempYdist, tempDist;
float minDist = 999.9; // set initial min distance
int tempElemID; // dummy variable

// Set minimum distance criteria - Within this distance we keep the
elements
float distCriteria = 250.0; // 0.25 km
// float distCriteria = 0.0;

cout << "The distance criteria for which to keep Surrounding Elements
is: ";
cout << distCriteria << " meters" << endl << endl;

```

```

int tally = 0;
for (int i=1; i<=linecount; i++)
{
    if(elements[i].rank == 1)
    {
        for(int j=1; j<=nElemInt; j++)
        {
            tempXdist = (elements[i].x - elemInt[j].x);
            tempYdist = (elements[i].y - elemInt[j].y);
            tempDist = sqrt(pow(tempXdist,2) + pow(tempYdist,2));

            if(tempDist < minDist)
                minDist = tempDist;
        } //end for

        // Increase rank by 1 if meets distance criteria
        if(minDist < distCriteria)
        {
            elements[i].rank = elements[i].rank + 1; // increase rank of
elements object
            tally++;
        } //end if

    } // end if
    minDist = 999.9; //reset minDist.
} //end for

cout << "The Number of Elements that are within Distance Criteria: " <<
tally << endl << endl;
cout << "About to delete elemInt and surrElem" << endl;

//*****
*
// Delete the elemInt object
// This object was only needed to calculate the minimum distance (Step
4)
//*****
*

//delete elemInt from memory
delete [] elemInt;

//*****
*
// Create a 2D array containing the emit to and receive from
information
// as well as the view factor percentage
//*****
*

int emitterNum, numRec;
int tempEmitter, tempReceiver;
float tempVF;

```

```

int count_rank_1 = 0;
int count_rank_2 = 0;

//*****
*.
// STEP 6c:
// Determine if elements with R = 1 emit to (or receive from)
// elements with R = 3 or R = 4
//
//*****

float vf_crit_1 = 0.0005;

// initialize max and min values
int numRecMin = 999999;
int numRecMax = 0;

int max_vfDel_emit5 = 0;
int max_vfDel_rec5 = 0;

//*****
// Open the view factor file and determine the
// emit, receive, and view factor % information
//
//*****

cout << "about to open vf file" << endl;
// reopen the view factor file for reading
infile.open(vfFile);

cout << "successful in opening vf file" << endl;

//make sure we can open the file.
if(!infile)
{
    cout << "Can not open the view factor file" << endl;
}

//parse out the first line (Number of emitters: nElems)
infile.getline(str,255);

int count2 = 0;
//loop over all elements in the file
while(infile.peek() != EOF)
{
    //make sure we don't write beyond the allocated memory
    assert(count2<=nEmitters+1);
    // find the emitter number
    do{
        infile>>ch;
    }while(ch!=':');
    infile>>emitterNum;

    // find the number of receivers

```

```

do{
    infile>>ch;
    }while(ch!=':');
    infile>>numRec;

//-----
// Determine the max and min value for the number of receivers (numRec)
// This will give an idea the range of nRec
// Then you can determine the max Q loss you will get
// if eliminate a view factor value of vf_crit_1 and vf_crit_2
//-----
    if(numRec > numRecMax)
        numRecMax = numRec;

    if(numRec < numRecMin)
        numRecMin = numRec;
//-----

// grab dummy line
    infile.getline(str,255);
//      cout << "this is dummy line:  " << str[0] << str[1] << str[2]
<< endl;

// get rid of header line (Emitter Receiver View Factor)
    infile.getline(str,255);

// retrieve all information for emitter
    int nVf_emitEOI_del = 0;
    int nVf_recEOI_del = 0;

    for(int i=0; i<numRec; i++)
    {
        infile>>tempEmitter;
        infile>>tempReceiver;
        infile>>tempVF;

        infile.getline(str,255);

        // check if emitter is an elemInt or surrElem
        // check if emitter is surrElem
        if((elements[tempEmitter].rank == 1) ||
(elements[tempEmitter].rank == 2))
        {
            // if surrElem --> see if it emits to an elemInt
            if(elements[tempReceiver].rank == 0)
            {
                // if the view factor is > vf_crit_1
                if(tempVF > vf_crit_1)
                {
                    elements[tempEmitter].rank = elements[tempEmitter].rank +
2;
                    count_rank_1++;
                }
            }
        }
    }

```



```

        else // Determine number of view factors deleted from each
element
        {
            nVf_emitEOI_del = nVf_emitEOI_del + 1;
        }
    }

    // check if emitter is elemInt
    else if(elements[tempEmitter].rank == 0)
    {
        // if the receiver is a surrElem --> increase its rank
        if((elements[tempReceiver].rank == 1) ||
(elements[tempReceiver].rank == 2))
        {
            // if the view factor is > vf_crit_1
            if(tempVF > vf_crit_1)
            {
                elements[tempReceiver].rank =
elements[tempReceiver].rank + 2;
                count_rank_2++;
            }
            else
            {
                nVf_recEOI_del = nVf_recEOI_del + 1;
            }
        }
    }

} //end for

if(nVf_emitEOI_del > max_vfDel_emit5)
    max_vfDel_emit5 = nVf_emitEOI_del;

if(nVf_recEOI_del > max_vfDel_rec5)
    max_vfDel_rec5 = nVf_recEOI_del;

// get rid of VF Sum line
infile.getline(str,255);

// increment count so do not have infinite loop
count2++;

} //end while

infile.close(); //always close the file
infile.clear(); // reset infile

//Results
cout << endl << "Results from Step 5" << endl;

```

```

    cout << " The Number of Elements(R=1,2) that Emit to Elements(R=0): "
    << count_rank_1 << endl;
    cout << " The Number of Elements(R=1,2) that Receive from
    Elements(R=0): " << count_rank_2 << endl << endl;

```

```

//*****
*
// STEP 6d:
//   Determine if elements with R = 1 emit to (or receive from)
//   elements with R = 3 or R = 4
//
//*****
*

```

```

    float vf_crit_2 = .0005;
    //float vf_crit_2 = 0.00;

```

```

int max_vfDel_emit6 = 0;
int max_vfDel_rec6 = 0;

```

```

cout << "Ready to open the View Factor File for Step 6" << endl;
    // reopen the view factor file for reading
    infile.open(vfFile);

```

```

cout << "successful in opening vf file" << endl;

```

```

//make sure we can open the file.

```

```

    if(!infile)
    {
        cout << "Can not open the view factor file" << endl;
    }

```

```

    //parse out the first line (Number of emitters: nElems)
    infile.getline(str,255);

```

```

    count2 = 0;
    // reset counters
    count_rank_1 = 0;
    count_rank_2 = 0;

```

```

//loop over all elements in the file

```

```

while(infile.peek() != EOF)

```

```

{
    //make sure we don't write beyond the allocated memory
    assert(count2<=nEmitters+1);
    // find the emitter number
    do{
        infile>>ch;
    }while(ch!=':');
    infile>>emitterNum;

```

```

// find the number of receivers
do{

```

```

        infile>>ch;
    }while(ch!=':');
    infile>>numRec;

// grab dummy line
    infile.getline(str,255);

// get rid of header line (Emitter Receiver View Factor)
    infile.getline(str,255);

// retrieve all information for emitter
    int nVf_rec_del = 0;
    int nVf_emit_del = 0;

    for(int i=0; i<numRec; i++)
    {
        infile>>tempEmitter;
        infile>>tempReceiver;
        infile>>tempVF;

        infile.getline(str,255);

// check if emitter is a surrElem with R=3,4 or R=1
        // check if emitter is surrElem with rank=1
        if(elements[tempEmitter].rank == 1)
        {
            // if surrElem with R=1 --> see if it emits to a surrElem with
R=3,4
            if((elements[tempReceiver].rank == 3) ||
(elements[tempReceiver].rank == 4))
            {
                // if the view factor is > vf_crit_2
                if(tempVF > vf_crit_2)
                {
                    elements[tempEmitter].rank = elements[tempEmitter].rank
+ 4;
                    count_rank_1++;
                }
                else
                {
                    nVf_emit_del = nVf_emit_del + 1;
                }
            }

            // check if emitter is surrElem with R=3,4
            else if((elements[tempEmitter].rank == 3) ||
(elements[tempEmitter].rank == 4))
            {
                // if surrElem with R=3,4 --> see if it receives from a
surrElem with R=1
                if(elements[tempReceiver].rank == 1)
                {
                    // if the view factor is > vf_crit_2

```

```

        if(tempVF > vf_crit_2 )
        {
            // if so, increase rank of element
            elements[tempReceiver].rank = elements[tempReceiver].rank
+ 4;
            count_rank_2++;
        }
        else
        {
            nVf_rec_del = nVf_rec_del + 1;
        }
    }
}

} //end for

if(nVf_emit_del > max_vfDel_emit6)
    max_vfDel_emit6 = nVf_emit_del;

if(nVf_rec_del > max_vfDel_rec6)
    max_vfDel_rec6 = nVf_rec_del;

// get rid of VF Sum line
infile.getline(str,255);

// increment count so do not have infinite loop
count2++;

} //end while

infile.close(); //always close the file

cout << endl << "Results from Step 6" << endl;
cout << "The Number of Elements(R=1) that Emit to Elements(R=3,4): " <<
count_rank_1 << endl;
cout << "The Number of Elements(R=1) that Receive from Elements(R=3,4):
" << count_rank_2 << endl;

//*****
*
// Elements to be deleted R=1
//*****
*

int nElem_delete = 0;
int nElem_rank0 = 0;
int nElem_rank2 = 0;
int nElem_rank3 = 0;
int nElem_rank4 = 0;
int nElem_rank5 = 0;

for (int i=1; i<=nEmitters; i++)
{

```

```

        if(elements[i].rank == 1)
        {
            nElem_delete++;
        }
        else if(elements[i].rank == 0)
        {
            nElem_rank0++;
        }
        else if(elements[i].rank == 2)
        {
            nElem_rank2++;
        }
        else if(elements[i].rank == 3)
        {
            nElem_rank3++;
        }
        else if(elements[i].rank == 4)
        {
            nElem_rank4++;
        }
        else if(elements[i].rank == 5)
        {
            nElem_rank5++;
        }
    }
}
//*****
*
// Results from Ranking
//
//*****
*
    cout << endl << "Minimum Number of Receivers for an Element: " <<
numRecMin << endl;
    cout << "Maximum Number of Receivers for an Element: " << numRecMax <<
endl;

    cout << endl << "The Max Number of Deleted View Factors (emit-Step5): "
<< max_vfDel_emit5 << endl;
    cout << "The Max Number of Deleted View Factors (receive-Step5): " <<
max_vfDel_rec5 << endl;
    cout << "The Max Number of Deleted View Factors (emit-Step6): " <<
max_vfDel_emit6 << endl;
    cout << "The Max Number of Deleted View Factors (receive-Step6): " <<
max_vfDel_rec6 << endl;

    cout << endl << "The Critical Value for Step 5c: " << vf_crit_1 <<
endl;
    cout << "The Critical Value for Step 5d: " << vf_crit_2 << endl;
    cout << "NOTE: The critical value is for > rather than >= " << endl;

    cout << endl << "The Total Number of Elements to be Deleted (R=1): " <<
nElem_delete << endl;

```

```

    cout << "The Total Number of Elements of Interest (R=0): " <<
nElem_rank0 << endl;
    cout << "The Total Number of Elements (R=2): " << nElem_rank2 << endl;
    cout << "The Total Number of Elements (R=3): " << nElem_rank3 << endl;
    cout << "The Total Number of Elements (R=4): " << nElem_rank4 << endl;
    cout << "The Total Number of Elements (R=5): " << nElem_rank5 << endl
<< endl;

//*****
*
//  Output Deleted Elements to a Test File
//
//*****
*
    char *output_file = argv[3];

    // one line:  ofstream fout("deleted_elem.txt");
ofstream fout;
fout.open(output_file);

    // Make sure output file will open
if(!fout.is_open())
{
    cerr << "Can't open " << output_file << " file for output.\n";
    exit(1);
}

    // The top line of the output file is the number of elem to delete
fout << nElem_delete << "\n" ;

    // write to file the element ID of the elements to be deleted
for(int i=1; i<=nEmitters; i++)
{
    if(elements[i].rank == 1)
    {
        fout << elements[i].elemID << "\n" ;
    }
}

    fout.close(); //close output connection to file

cout << "The Output File " << output_file << " is Complete. \n\n\n";

//*****
*
// Delete the elements object
//*****
*

    //delete elements from memory
delete[] elements;

    return 0;
}

```

FILEIO_DEMO.CSH FILE

```
### Extract data from TDF file ###
#~/TDF_SRC/tdfgettsfcm \ # Program to extract data
#BZH_2002081500Z.tdf \ # tdf filename
#BZH_2002081500Z.dat # Data Extracted

~/Test_case_bzpass_vf/Step5/driver \
~/Test_case_bzpass_vf/Step3/elemAttr_outfile.dat \
~/Test_case_bzpass_vf/Step4/vf_outfile.txt \.
~/Test_case_bzpass_vf/Step5/deleted_elem_outfile.txt

# End of csh script
```


APPENDIX C

C/C++ CODE – UPDATE RADTHERM PROJECT FILE

MAKEFILE

```

# $Id: Makefile,v 1.1.1.1 2001/10/11 01:34:35 ladeanm Exp ladeanm $
# $Source: /apps/wxneve/AWMC_UTIL/TDF_SRC/Makefile,v $
# $Description: Makefile to update tdf routines.$
# $Log: Makefile,v $
# Revision 1.1.1.1 2001/10/11 01:34:35 ladeanm
# Expanded and updated to work with winthermrt-6.X
#

# List of Commands that can be compiled
SRC = tdfupdatePart
# Each command can be compiled by typing, for example: make tdfupdatem

CC      = cc
CXX     = CC
CFLAGS  = -n32 -mips4 -LANG:bool=ON -woff
1021,1209,1233,1314,1355,1375,1506 -fullwarn -O2
CXXFLAGS = -n32 -mips4 -LANG:bool=ON -woff
1021,1174,1209,1233,1314,1355,1375,1506 -ptused -fullwarn -O2
LINK    = CC
LFLAGS  = -n32 -mips4

TDFIO = /home/mercury/apps/coe/RTWx/TAI/tdfio
INCLUDE = -I ${TDFIO}/include
LIBES = -L ${TDFIO}/lib/ -ltdfio -lm

${SRC}:  $$@.cpp
        ${CXX} -o $@ $? ${CXXFLAGS} ${INCLUDE} ${LIBES}

# For discussion of makefile syntax, see \cite[p.21]{Oram-Talbot:1991}

# End of $RCSfile:$

```

C/C++ CODE - TDFUPDATEPART.CPP

```

/*
* \file tdfupdatePart.cpp
*
* Date: 04/02/03
*
* This code takes the output file of the deleted elements
* and changes the part ID's of these elements
*
* To run:
* $ make tdfupdatePart
* $ tdfupdatePart.csh
*/

//-----
// Standard Includes
//-----
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

//-----
// Non-Standard Includes
//-----
#include "tdfio.h"
#include "tdfelement.h"
#include "tdfpart.h"

//-----
// Private Function Prototypes
//-----

TDF::TdfStatus changePart(FILE *fp_rd, TdfIO &tdfio);

//-----
// Main
// \param argc Number of Command line arguments
// \param argv Array of command line arguments
//
// \return Returns 0 on successful completion, 1 on
// unsuccessful completion
//-----

int main(int argc, char **argv)
{
    //Check to make sure command line looks like expected
    if(argc !=4)
    {
        fprintf(stderr, "Usage: %s infile.tdf delElemFile.txt outfile.tdf\n", argv[0]);
        return 1;
    }
}

```

```

}

FILE *fp_rd = fopen(argv[2], "r"); //open for reading only
if (fp_rd == NULL)
{
    fprintf( stderr, "Error opening %s for reading!\n", argv[2] );
    return 1;
}

// Read the TDF file
TdfIO tdfio;
TDF::TdfStatus status = tdfio.readFile(argv[1]);

if(status != TDF:: Success)
{
    fprintf(stderr, "Error reading TDF file %s \n", argv[1]);
    return 1;
}
else
{
    changePart(fp_rd, tdfio);

    fprintf( stderr, "In main after changePart call \n" );

    TDF::TdfStatus status = tdfio.writeFile( argv[3] );
    fprintf( stderr, "after tdfio.writeFile call" );

    if (status != TDF::Success)
    {
        fprintf( stderr, "Error writing TDF file %s\n", argv[3] );
    }
    else
    {
        fprintf( stderr,
            "Wrote TDF file %s with modified environment
parameters.\n",
            argv[3] );
    }
}

fclose(fp_rd);

return 0;

} // end main

/*-----
--
* changePart
*
*/
TDF::TdfStatus changePart(FILE *fp_rd, TdfIO &tdfio)
{

```

```

TDF::TdfStatus status = TDF:: Success;

//*****
// Read in the text file
// First line: Number of Deleted Elements
// Next lines: Element ID of each element to delete
//*****
int nElems;
fscanf(fp_rd, "%i \n", &nElems);

int *elemID_toDelete = new int [nElems+1];
fprintf(stderr, "nElems: %10i \n", nElems);

int counter;
for(counter=1; counter<=nElems; counter++)
{
    fscanf(fp_rd, "%i \n", &elemID_toDelete[counter]);
//    fprintf(stderr, "elemID_toDelete: %5i \n",
elemID_toDelete[counter]);
}

//*****
// Add a new part to contain the elements to delete
//
//*****
TdfElement element;
TdfPart newPart;

int nParts = tdfio.numberOfParts();
fprintf(stderr, "nParts: %5i \n", nParts);

int partID = tdfio.partID( nParts-1 );
fprintf(stderr, "partID: %5i \n", partID);

status = newPart.setName( "Elements to Delete" );
if (status != TDF::Success)
{
    fprintf( stderr, "Error setting part name!\n" );
}

status = tdfio.addPart( ++partID, newPart );
if (status != TDF::Success)
{
    fprintf( stderr, "Error adding a new Part");
}

// Check to see if the new number of parts is as expected
// Should be the new part ID
int new_partID = tdfio.numberOfParts();
fprintf(stderr, "New Part ID: %5i \n", new_partID);

// set the element and part parameters using the modified copy

```

```

// status = tdfio.changePart(new_partID, newPart); //Not sure if need
this

//*****
// Change the partID of the elements to delete
//
//*****

for(int i=1; i<=nElems; i++)
{
    int elemID = elemID_toDelete[i];

    // retrieve the indexed element
    tdfio.element( elemID, element);

    // set the elements new part ID
    element.setPart( new_partID );

    // retrieve the part that the element belongs to -- check
    // fprintf(stderr, "Elem ID: %7i Part ID: %5i \n", elemID,
    element.part() );

    // set the element and part parameters using the modified copy
    status = tdfio.changeElement(elemID, element);
}

status = TDF:: Success;

delete [] elemID_toDelete;

return status;
}

```

TDFUPDATEPART.CSH FILE

```
#!/bin/csh
# Run-Time Load Library Path
setenv LD_LIBRARY_PATH /home/mercury/apps/coe/RTWx/TAI/tdfio
setenv LD_LIBRARY64_PATH /home/mercury/apps/coe/RTWx/TAI/tdfio
setenv LD_LIBRARYN32_PATH /home/mercury/apps/coe/RTWx/TAI/tdfio
setenv PATH /home/mercury/apps/coe/RTWx/bin:sbin:/apps/freeware/bin

### Extract data from TDF file ###
#~/TDF_SRC/tdfgettsfcm \ # Program to extract data
#BZH_2002081500Z.tdf \ # tdf filename
#gettsfc_id BZH.dat \ # Elements to Extract data for
#BZH_2002081500Z.dat # Data Extracted

~/Test_case_bzpass_vf/Step6/tdfupdatePart \
BZH_test.tdf \
~/Test_case_bzpass_vf/Step5/deleted_elem_outfile.txt \
~/Test_case_bzpass_vf/Step6/updatePart.tdf

# End of csh script
```


MONTANA STATE UNIVERSITY - BOZEMAN



3 1762 10388380 5