University of Nevada, Reno

Mine Ventilation and User’s Interface with MULTIFLUX software

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Mining Engineering

by

Rajeev Gunda

Dr. George Danko/Thesis Advisor

August, 2011
We recommend that the thesis prepared under our supervision by

RAJEEV GUNDA

entitled

Mine Ventilation And User’s Interface With MULTIFLUX Software

be accepted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Dr. George Danko, Advisor

Dr. Jaak J. K. Daemen, Committee Member

Dr. Dhanesh Chandra, Graduate School Representative

Marsha H. Read, Ph. D., Dean, Graduate School

August, 2011
Abstract

MULTIFLUX is an underground ventilation simulation software developed at the University of Nevada, Reno by Professor George Danko. The objectives of the current work are to test the software and to develop a graphical user interface that enables the user to draw any ventilation network, input the associated data, and see the simulation results after processing the data and executing MULTIFLUX.

As a first step, a manual user interface is developed in MATLAB software with which the user can work on small sized ventilation networks. For the development of the graphical user interface, commercial software such as VnetPC, Ventsim, CLIMSIM, ICAMPS and VUMA are studied in detail and a basic layout of the user interface for MULTIFLUX software is designed.

As the second step, a template-based user interface for MULTIFLUX is developed. Various graphical templates and data tables are designed for the input of data by the user for various types of branches. The Google SketchUp software is used as the 3-dimensional graphical platform for the MULTIFLUX’s user interface. Mine ventilation problems are solved to test the graphical user interface. Time dimension is added to the data input, which facilitates the user to study the dynamic behavior of heat sources and sinks, and their effect on the mine airways regarding time.

As a final step to improve the user’s interaction with the software, a scenario based concept is developed, with the help of which, one can analyze the mine behavior in various scenarios such as blasting, mine fire, etc. Barrick’s Meikle mine is taken as an example to demonstrate the pre-processing, data input, processing tasks, and the display of results in the MULTIFLUX software.
Dedicated to my amma and nanna
Acknowledgements

I owe my deepest gratitude to my advisor Dr. George Danko, professor and research leader, for his invaluable guidance, support and encouragement throughout my thesis and research work. The research fund from NIOSH, USA under the grant number, 200-2009-30157 is gratefully acknowledged.

I am heartily thankful to Dr. Davood Bahrami, research professor and team member, for his immense help in many ways, and for being available whenever I needed him.

I thank my committee members Dr. Jaak Daemen and Dr. Dhanesh Chandra for their valuable time, suggestions and recommendations. I am grateful to Dr. Pierre Mousset-Jones who introduced me to Mine Ventilation. It is my pleasure to thank my research teammates Pedram Rostami, Jon Fox and Raymond Grymko for their support and assistance.

I also take this opportunity to appreciate my brother Chipsy, my grandparents, and friends in Reno and India, for giving me moral support to complete this work.
Table of contents

Abstract..................................................................................................................................................i

List of Tables.......................................................................................................................................vi

List of Figures......................................................................................................................................ix

1 Problem description and approach .................................................................................................1

2 Background study.............................................................................................................................5

   2.1 Summary report on VnetPC - User interface..............................................................................8

   2.2 Summary report on Ventsim - User interface ...........................................................................10

   2.3 Summary report on CLIMSIM - User interface........................................................................12

   2.4 Summary report on ICAMPS - User interface.........................................................................15

   2.5 Summary report on VUMA - User interface.........................................................................17

3 MULTIFLUX - List of input parameters............................................................................................21

4 Manual – mode user interface development...................................................................................25

5 Conceptualizing the interface – Pre-Processor and Post-Processor................................................33

6 Designing templates and data tables...............................................................................................36

   6.1 Template 1: Airflow Connections...............................................................................................36

   6.2 Template 2: Fan Branch ............................................................................................................44

   6.3 Template 3: Air – Thermal – Moisture Branch .........................................................................48

   6.4 Template 4: Air – Thermal – Moisture – Contaminant Branch ................................................62

   6.5 Template 5: Branch with duct: Air – Thermal – Moisture – Contaminant connections ...............65
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6</td>
<td>Template 6: Branch with duct including fan and cooler</td>
<td>67</td>
</tr>
<tr>
<td>6.7</td>
<td>Template 7: Branch with duct and conveyor: Air – Thermal – Moisture – connections</td>
<td>72</td>
</tr>
<tr>
<td>6.8</td>
<td>Template 8: Development-end scenario in hard rock and soft rock mines</td>
<td>79</td>
</tr>
<tr>
<td>6.9</td>
<td>Special templates</td>
<td>81</td>
</tr>
<tr>
<td>6.10</td>
<td>The three types of interactions</td>
<td>81</td>
</tr>
<tr>
<td>7</td>
<td>Developing HTML forms for interfacing with Google Sketchup software</td>
<td>83</td>
</tr>
<tr>
<td>8</td>
<td>Time dependency in data input</td>
<td>90</td>
</tr>
<tr>
<td>9</td>
<td>Scenario based simulations</td>
<td>93</td>
</tr>
<tr>
<td>10</td>
<td>Illustration of the Pre-processor and Post-processor</td>
<td>94</td>
</tr>
<tr>
<td>11</td>
<td>Conclusions</td>
<td>99</td>
</tr>
<tr>
<td>12</td>
<td>Future work</td>
<td>101</td>
</tr>
<tr>
<td>13</td>
<td>References</td>
<td>102</td>
</tr>
<tr>
<td>14</td>
<td>Appendices</td>
<td>106</td>
</tr>
<tr>
<td>14.1</td>
<td>Appendix 1</td>
<td>106</td>
</tr>
<tr>
<td>14.2</td>
<td>Appendix 2</td>
<td>148</td>
</tr>
</tbody>
</table>
List of Tables:

1. Branch and Junction input data – VnetPC.........................................................9
2. Fan - Fixed Quantity- Contaminant input data – VnetPC...............................10
3. Airway input data – Ventsim........................................................................11
4. Fan / Fixed flow input data – Ventsim...............................................................11
5. Heat data – Ventsim.......................................................................................11
6. Contaminant data – Ventsim........................................................................12
7. Physical description of airway – CLMSIM.......................................................13
8. Ventilation or airflow parameters – CLMSIM..................................................14
9. Thermal Parameters and Heat Sources – CLMSIM........................................14
10. Branch input – ICAMPS................................................................................16
11. Fan input data – ICAMPS.............................................................................16
12. Airflow (Aero) input parameters for Development Heading branch type – VUMA software..............................................................................................18
13. Thermal input parameters for Development Heading branch type – VUMA software..............................................................................................18
14. Contaminant input parameters for Development Heading branch type – VUMA software......................................................................................19
15. List of input parameters for MULTIFLUX........................................................23
16. Types of connections for all the network types.............................................27
17. Airway Geometry Input form for Template 1 shown in Figure 10..................42
18. Input form for Parametric tables, Template 1................................................43
19. Airway Geometry Input form - Template 2....................................................46
20. Input form for Fan - Template 2……………………………………………………….47
21. Airway Geometry Input form – Template 3……………………………………….51
22. Input form for the Rock Strata (Heat and Moisture) – Template 3…………………52
23. MULTIFLUX rock property library associated with the input form for the Rock Strata (Heat and Moisture) – Template 3…………………………………………………………..52
24. Input form for the Heat Source / Sink data – Template 3………………………….53
25. MULTIFLUX explosive energy library, associated with heat source / sink form for Template – 3 ………………………………………………………………………..55
26. Data entry forms for all types of heat sources in the Heat Source / Sink form – Template 3…………………………………………………………………………..56
27. The MULTIFLUX metabolic heat library ………………………………………….57
28. Data entry forms for all types of cooling sources in the Heat Source / Sink form – Template 3…………………………………………………………………………..61
29. Input form for the contaminant window – Template 4……………………………..64
30. Data entry table for the Radon term in the contaminant form – Template 4……….64
31. Duct Geometry Input form – Template 5……………………………………………67
32. Input form for the duct window – Template 6………………………………………70
33. Input form for the heat sink / source window – Template 6………………………..71
34. Duct / Conveyor Belt Geometry Input form – Template 7………………………75
35. Duct Fan and Cooler Input form – Template 7……………………………………..76
36. Heat Source / Sink Data Input form – Template 7…………………………………..78
37. Road Header / Shuttle input form…………………………………………………..80
38. The utilization trend (as per shift) of a diesel machine in a mine during a normal day of operation………………………………………………………………………….91

39. Input form for time dependent input variables………………………………………………91

A1. Meikle and Roadeo mines annual fuel consumption………………………………………148

A2. Duetz 912 family – Construction engine specifications……………………………..149
List of Figures:

1. Outline of the CLIMSIM structure.................................................................13
2. Thermal-Hydrological-Airflow-Contaminant modeling with MULTIFLUX........22
3. Layout of the GUI for Manual mode of input for MULTIFLUX in GUIDE module of MATLAB........................................................................................................26
4. GUI window showing the data entry tables for one of the connection types in the „Air“ network...........................................................................................................29
5. GUI window showing the data entry tables for one of the connection types in the „Heat“ network, and illustrating “Generator” connection.................................31
6. Display of the error message for an inappropriate combination of selections.....32
7. Process flowchart between CAD, MULTIFLUX and MATLAB environments................................................................................................................35
8. The schematic of a sample ventilation network that is drawn in the GUI window of MULTIFLUX.................................................................................................36
9. Basic view of the network shown in Figure 8....................................................37
10. Template Selection Window............................................................................38
11. Simple airway template that has only airflow connections...............................40
12. Special case I: Fan branch, airflow line..........................................................44
13. A branch with air, thermal and moisture connections.......................................48
14. Side view of Figure 13....................................................................................48
15. Front view of Figure 13..................................................................................49
16. Branch with Air – Thermal – Moisture – Contaminant connections..................62
17. Side view of Figure 16....................................................................................63
18. A drift with one duct; thermal (including radiation) and moisture connections…62
19. Special Case II: Branch with duct – duct fan and cooler………………...68
20. Front view of Figure 19…………………………………………………………69
21. Branch with duct and conveyor belt; thermal (including radiation) and moisture connections……………………………………………………72
22. Front view of Figure 21…………………………………………………………73
23. Side view of Figure 21…………………………………………………………73
24. Development end scenario in a metal mine ……………………………...79
25. Development end scenario in a soft rock mine, consisting of auxiliary ventilation, road header and material conveying system…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………
35. The temperature distribution in the network. The post-processor part of the GUI.
1. Problem description and approach

Due to the rapidly increasing production rates and deepening of working areas, there is a drastic rise in the number of thermal issues to be addressed for maintaining a safe mine environment. As high temperatures and humidity in underground mines could affect the health and performance of workers, and ultimately the production rate, it is necessary to provide a properly controlled environment (Gunda et al., 2011).

An engineer must predict the overall heat loads within mine, to effectively design and implement integrated mine ventilation and climatic control scheme. For the engineer to assess, quantify and cost the effectiveness of alternative climatic strategies, including ventilation and refrigeration systems, there should exist an effective mine network climate prediction and planning tool. Though there are several mine ventilation simulation software applications on the market, the MULTIFLUX software has an edge over others regarding the capability of modeling all possible and necessary transport connections between elements of mine to simulate the mine ventilation environment accurately.

MULTIFLUX software (Danko, 2008) was originally developed for analyzing the thermal-hydrologic processes in a hot and humid environment involving the storage of high-level nuclear waste and spent fuel. The MULTIFLUX model and software is well documented since it became a qualified code for studying the above mentioned application, and has also been used in mining applications. The software provides the necessary simulation model elements to comply with the need for the fast-changing, transient conduction in the rock wall, complex heat transport interactions between the airflows in the auxiliary air duct and the drift, viscous dissipation, and compression-expansion work in the variable-pressure air flow.

The objective of my research work is to add Graphical User-Interface (GUI) capacity to MULTIFLUX, for ease-of-use support for typical mine ventilation tasks with which the user conveniently enters input geometry and user-selectable parameters and properties for typical mining tasks.
Before the project commenced, the only way to setup MULTIFLUX model and execute it was by manually defining convective, conductive and other transport network connections. To setup such a model, one has to use MATLAB or any other text editor, and create MULTIFLUX input files that define node-to-node connections for heat, airflow, contaminant, moisture and other parameters. For software like MULTIFLUX to be available for an end user such as a mining engineer or a student, it has to be equipped with an easy to use interface. The interface should be able to assist the user to:

1. Setup the complex transport connections in the network easily, by taking in to account for various kinds of heat and mass transfers including diffusion, radiation, and moisture and contaminant interactions, etc. In addition, the events like auto-compression, heat exchange between drain and air, presence of fissure water, etc. should be included whenever appropriate.
2. It will also enable the planning engineer to check design variations and ventilation and cooling configurations, to produce the desired climatic conditions, and to modify and extend the existing systems.

The task of development of graphical user interface for MULTIFLUX can be organized into 3 parts:

1. Studying user interfaces of existing mine ventilation software such as VnetPC (McPherson, 2003), Ventsim (Ventsim, 2007), CLIMSIM (CLIMSIM, 2006), MINEVENT (ICAMPS, 2007), and VUMA (Bluhm et al, 2001) to incorporate common useful features into the new MULTIFLUX interface for the sake of compatibility, thus reducing the learning time of a new user.
2. Designing user-interface to conveniently enter input geometry and user-selectable parameters and properties for typical mining tasks.
3. Developing simple database connections between a Computer Aided Design software, a mine ventilation design software such as VnetPC, and MULTIFLUX for ventilation network definition and graphical documentation.
For compatibility with other CAD and mine ventilation software, the user interface should be equipped with following features:

- One should be able to import the mine schematic files which are drawn not only in a CAD environment, but also in other software such as VnetPC or Ventsim.

- The additional attributes might be entered manually or it will have a program which automatically appends the CAD „.dxf” file with additional parameters such as k-factor, etc.

Section 2 is a critical review of the literature studies made on several previously and currently existing mine ventilation software, and their user interface. List of input parameters are made from a selected ventilation modeling software that are currently available in the market. Section 3 gives a basic understanding on the internal functioning of the MULTIFLUX software and the list of input parameters developed, based on studying the software mentioned in Section 2.

A manual mode of input to the MULTIFLUX is developed as an initial step towards achieving the goal of designing a full-fledged 3-dimensional graphical user interface. This is briefed in Section 4. Then, the pre-processor and post-processor of the interface are conceptualized, which are covered in Section 5. In the process of design of the mode of input parameters data, a template-based concept is introduced. Section 6 covers various templates and the data input forms developed for those templates.

After the templates and corresponding data tables are developed, the process of integrating these with the Google SketchUp software’s 3-dimensional platform is achieved through the development of HTML forms, which is explained in Section 7. The “time” dimension is introduced in the data input and the significance of the time dependency in the data input is explained in Section 8. Scenario based simulations are developed which aid the user to study the mine response behaviors in a specific scenario. This is given in Section 9.
The 3-dimensional platform, the pre-processor and post-processor of MULTIFLUX are illustrated with screen-shots of the working version of MULTIFLUX’s user interface in Section 10. The methods of communication of the user with the code to view the results and various kinds of results that can be viewed by the user are explained in this section.

One has to notice that this thesis is part of a 5 year research project in its second year, and the task of developing the graphical user interface (GUI) will be fully completed by the end of the project. The work presented here details the current form of the user-interface, and is yet to be worked on for the next three years to reach its final form.
2. **Background study**

Several computer programs were developed as early as in the 1960s that focused on ventilation modeling of underground mines. These early computer programs were intended for predicting heat pick-up, and temperature and humidity changes in the air in mine tunnels. Written in programming languages like BASIC, FORTRAN and PASCAL, they lacked any user interface owing to the text based computer technology at those times. The mode of input was through either punched cards or command line entry, and the output used to be in the form of printouts of large tables. A literature study is conducted in which some of such early software are studied. An outline of the input and output functionalities of those software are summarized in the following paragraphs.

Gooch (1971) has designed a computer model written in FORTRAN which combines the laws of heat conduction, heat and mass transfer, autocompression and psychrometry in order to determine the increase in temperature and humidity of air flowing down the mine shafts. The program required standard punch cards for the data input by the user categorically, and the data output, i.e., dry and wet bulb temperatures of air, is given in the form of printouts.

Starfield’s computer program (Lawrence & Whiller, 1975) is one of the earlier ventilation simulation programs. The user inputs the input parameters such as virgin rock temperature, rock conductivity, age and perimeter of the airway, the dry and wet bulb temperatures, volumetric flow rate and barometric pressure of the intake air. And the program computes the change in dry and wet bulb temperatures, moisture and energy content and overall heat gain of the air at specified intervals along the length of the airway. All the input data was fed to the computer in the form of punched cards, and the data is output in the form of printed tables.

Michigan Tech. Ventilation Network Program which is developed during 1970s takes natural ventilation phenomenon into account (Gaines, 1978). This program has similar input and output functionalities.
CANMET is another thermodynamic mine ventilation program (Hall et al., 1981) for use by deep Canadian mines in planning ventilation flows, considering autocompression of air, moisture liftwork and natural ventilation effects. This software also has a plotting tool to simplify the data interpretation, which represents the air flow circuits through pressure and airflow quantity information. The output of the program includes branch data, fan results and annual ventilation costs.

ENVIRON (Von Glehn et al., 1987) is a Pascal based program that simulates the cooling and ventilating systems in gold mines. Large air density variations of the ventilation air in deep South African mines are taken into account. This consists of two modules HEATFLOW and VENTFLOW which may be run separately as stand-alone programs to perform a complete thermodynamic analysis of a mine. HEATFLOW, using an air flow distribution previously calculated by the VENTFLOW module, determines the total heat load. VENTFLOW, on the other hand, uses temperatures calculated by HEATFLOW and determines the air flow distribution throughout the mine. Thus these two modules are iteratively solved until the solution converges. Various heat sources and sinks are termed as boxes and are linked together in a network representing the mine air path layout. These boxes are positioned on the screen relative to the rest of the network such that the airflow path progresses down the screen. This program is menu-driven and makes use of „windows”. Default values are displayed in the fields which can be modified by the user. The results can be saved in ASCII files.

VENTCLIM (Mizuta & Vutukuri, 1990) is a computer program which models a mine ventilation network and calculates the airflow rate, temperature and humidity for the entire network of airways. The program is written in BASIC, which can accommodate networks up to 500 airways, 500 junctions and 10 fans. The user provides the data through another program DATAPREP (Mizuta & Vutukuri, 1990), which is stored in files for use by the VENTCLIM program. The input data is organized into four different groups, namely branch data, data about surface, inlet and outlet junctions, junction data, and fan data. The secondary parameters, that are determined from initial parameters, are calculated by another program, PARDET (Mizuta & Vutukuri, 1990). There is a
provision to modify the mine model and correct any mistakes entered, but this process is tedious. The output data is stored in a file. This program is developed to run on single or double floppy disk systems.

AUXVECL (Mizuta & Vutukuri, 1990) is a program that models drivages i.e., development end drifts and calculates the airflow rate, temperature and humidity in the drivages, which assists the engineers in the design of auxiliary ventilation and air-cooling systems for underground mines and tunnels. This program takes into account the auto-compression, presence of inflow ground water, heat produced by machinery, spot heat sources, heat exchange between drain and air, etc in the drivage, and leakage of the duct. The method of data input by the user and the data output by the program is similar to VENTCLIM. AUXVECL1 is the program through which the user inputs the data, and AUXVECL2 is the main program that processes the data and gives the results in tabulated form. This program is developed to run on single or double floppy disk systems.

TRANCLIM (Mizuta & Vutukuri, 1990) is a mine ventilation program which employs a numerical method to predict the temperature and humidity variations of airflow along tunnels. This program has similar user interface to that of VENTCLIM and AUXVECL (Mizuta & Vutukuri, 1990).

Similar to the above mentioned software, there were several other programs that were developed to model underground mine ventilation networks. However, there is no inclusion of graphics input/output routines that will enable data to be entered directly from mine plans to the software until late 1990s. The quality of mine ventilation modeling software became more error free as the graphical user interface was introduced. GUIs provided the user capacity to review the input parameters and analyze complex network solutions more easily, thereby generating valid results. In early 2000s, came a new generation of software with graphical tools. Some of them are detailed in the following sections.
GRUBE-V (Heim et al., 2001) is another mine ventilation modeling software with a fully developed graphical user interface. The interface of this software utilizes AutoCAD as its basic platform for interaction with the user. Thus, along with the AutoCAD tools, the user can manipulate the model with specific functions for network analysis in addition. The user inputs data through the data input forms that pop up when the user clicks on a branch or a node. The results can be seen in the same manner. The input data for all the airways is presented in tabular form, which can be edited by the user.

Several other software such as MFIRE (MFIRE, 1995), FLOWNEX (FLOWNEX, 2009), etc. are available in the market currently with advanced graphical tools and easy-to-use interface. Each program has its own kind of data inputting method.

To reduce the learning time for a new user, the new user interface design should be designed similar to the currently existing mine ventilation and climate simulation software. In this process, software such as VnetPC (McPherson, 2003), Ventsim (Ventsim, 2007), CLIMSIM (CLIMSIM, 2006), MINEVENT (ICAMPS, 2007), and VUMA (Bluhm et al, 2001) were studied. The following section is a conglomeration of all the summary reports of the detailed studies made on the user interfaces of above mentioned ventilation simulation software.

2.1 Summary report on VnetPC – User interface

VnetPC software (McPherson, 2003) has a simple to use interface. There are several windows through which the user can interact with the program, by giving inputs, seeking results, editing the branch information, viewing the network, etc. Those windows are:

- Model Information
- Branch Input
- Branch Results
- Schematic
- Fixed Quantities
- Fan Input
- Fan Results
•  Junction Data

The data input and output is done either through the Schematic graphically, or through the rest of windows in the form of tables. The user can switch his view in between these windows through a drop down menu. The user can:

1. Color branches based on the airway type.
2. Export ‘.dxf’ files to CAD or any planning software with a custom attribute export option.
3. Introduce fans, contaminants, resistances, etc, easily with the use of icons provided in the tool bar. One can click and drag these icons into the desired branch of the mine network.
4. Input branch resistance in four different ways.
5. Determine the operating cost for the system.

The input parameters given in the data input tables of VnetPC are categorized into two tables, and are presented in Tables 1 and 2.

Table 1. Branch and Junction input data - VnetPC

<table>
<thead>
<tr>
<th>Branch Number</th>
<th>Node from</th>
<th>Node to</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch Type:</th>
<th>R: Resistance, Parallel Factor.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P/Q: Pressure Drop, Quantity, Parallel Factor.</td>
</tr>
<tr>
<td></td>
<td>k Factor: Friction Factor, Length, Equivalent Length, Area, Perimeter, Parallel Factor.</td>
</tr>
<tr>
<td></td>
<td>R/L: Resistance per length, Length, Equivalent Length, Parallel Factor.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch Code</th>
<th>Surface State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junction</td>
<td>Junction Number</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Fan - Fixed Quantity - Contaminant input data - VnetPC

<table>
<thead>
<tr>
<th>X Coordinate</th>
<th>Y Coordinate</th>
<th>Z Coordinate</th>
<th>Relative Pressure</th>
</tr>
</thead>
</table>

Fan Data
- No. of fans in parallel
- No. of fans in series
- Operating pressure
- Fan curve with pressure and quantity data

Fixed Quantities
- Fixed Quantity
- Booster Pressure
- Branch resistance
- Regulator resistance
- Total Resistance
- Orifice Area

Contaminants
- Quantity
- Emission Rate
- Emission Concentration
- Contaminant Flow
- Contaminant Concentration

2.2 Summary report on Ventsim – User interface

Ventsim software (Ventsim, 2007) has a sophisticated 3-dimensional graphic window to draw ventilation networks. The input data tables pop up when the user double clicks on each airway. All the simulation tools are presented as icons in the tool bar. Animation is the striking feature of this software. Lists of all the input parameters are given in the tables 3 through 6.
Table 3. Airway input data - Ventsim

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td></td>
</tr>
<tr>
<td>Shape</td>
<td></td>
</tr>
<tr>
<td>Square</td>
<td></td>
</tr>
<tr>
<td>Round</td>
<td></td>
</tr>
<tr>
<td>Arched</td>
<td></td>
</tr>
<tr>
<td>Shanty</td>
<td></td>
</tr>
<tr>
<td>Irregular</td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>m</td>
</tr>
<tr>
<td>Height</td>
<td>m</td>
</tr>
<tr>
<td>Obstruction</td>
<td>m²</td>
</tr>
<tr>
<td>Length</td>
<td>m</td>
</tr>
<tr>
<td>Air type</td>
<td></td>
</tr>
<tr>
<td>Evasee</td>
<td>m²</td>
</tr>
<tr>
<td>Orifice</td>
<td>m²</td>
</tr>
<tr>
<td>Resistance</td>
<td>Ns²/m⁸</td>
</tr>
<tr>
<td>Friction factor</td>
<td>kg/m³</td>
</tr>
<tr>
<td>Shock – equivalent length</td>
<td>m</td>
</tr>
</tbody>
</table>

Table 4. Fan / Fixed flow input data - Ventsim

<table>
<thead>
<tr>
<th>Fans</th>
<th>Number of fans, Configuration (Parallel or Series)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed flow</td>
<td>Quantity Restriction Fixed pressure</td>
</tr>
</tbody>
</table>

Table 5. Heat data - Ventsim

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameters</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot sources</td>
<td>Sensible heat</td>
<td>kW</td>
</tr>
<tr>
<td>Parameter</td>
<td>Units</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>Latent heat</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Refrigeration</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Diesel engine</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Moisture</td>
<td>mlitres/s</td>
<td></td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Diesel engine</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Moisture</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Oxidation</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Conductivity</td>
<td>W/m-K</td>
<td></td>
</tr>
<tr>
<td>Diffusivity</td>
<td>m²/s</td>
<td></td>
</tr>
<tr>
<td>Specific heat</td>
<td>J/kg-K</td>
<td></td>
</tr>
<tr>
<td>Rock density</td>
<td>kg/m³</td>
<td></td>
</tr>
<tr>
<td>Wetness fraction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age at entry and exit</td>
<td>Days</td>
<td></td>
</tr>
<tr>
<td>temperatures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rock temperature</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>Air dry bulb and wet bulb</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>temperatures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diesel emission</td>
<td>kg</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Contaminant data - Ventsim

2.3 Summary report on CLIMSIM – User interface

This gives a description of the detailed study made on CLIMSIM”s user interface (CLIMSIM, 2006) and mode of input and output of data in setting up a model for climatic simulation study of an airway or series of branches. Basic structure of the user
interface of CLIMSIM comprises 3 kinds of input modules and 2 modes of output results, as shown in Figure 1. Various blocks of the software are:

1. Branch Input Table.
2. Air Source Table
3. Heat Source Table
4. Branch Results Table
5. Graphs

The various kinds of input required to be input by the user in the process of setting up a model are presented in the form of tables. They are categorized into 3 groups, presented in Table 7 through Table 9.

Table 7. Physical description of airway- CLIMSIM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>m</td>
</tr>
<tr>
<td>Cross-sectional area</td>
<td>m²</td>
</tr>
<tr>
<td>Perimeter</td>
<td>m</td>
</tr>
<tr>
<td>Parameter</td>
<td>Unit</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Depth in</td>
<td>m</td>
</tr>
<tr>
<td>Depth out</td>
<td>m</td>
</tr>
<tr>
<td>Friction Coefficient</td>
<td>kg/m$^3$</td>
</tr>
<tr>
<td>Wetness Factor</td>
<td>-</td>
</tr>
<tr>
<td>Age at inlet</td>
<td>days</td>
</tr>
<tr>
<td>Age at outlet</td>
<td>days</td>
</tr>
</tbody>
</table>

Table 8. Ventilation or airflow parameters - CLIMSIM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>m$^3$/s</td>
</tr>
<tr>
<td>Pressure</td>
<td>kPa</td>
</tr>
<tr>
<td>Dry bulb temperature</td>
<td>°C</td>
</tr>
<tr>
<td>Wet bulb temperature</td>
<td>°C</td>
</tr>
</tbody>
</table>

Table 9. Thermal Parameters & Heat Sources - CLIMSIM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virgin Rock Temperature at inlet</td>
<td>°C</td>
</tr>
<tr>
<td>Geothermal step</td>
<td>m/°C</td>
</tr>
<tr>
<td>Conductivity</td>
<td>W/m²/°C</td>
</tr>
<tr>
<td>Diffusivity</td>
<td>m$^2$/sec</td>
</tr>
<tr>
<td>Heat Transfer Coefficient</td>
<td>W/m²/°C</td>
</tr>
<tr>
<td>Node – Node interval</td>
<td>m</td>
</tr>
</tbody>
</table>

Machine spot heat source:
| Distance from intake end         | m             |
| Full load power output           | kW            |
| % utilization at equivalent full load | %          |
| Type                             | Diesel, etc.  |

Non – machine spot heat source:
| Distance from intake end         | m             |
| Sensible Heat Load               | kW            |
The output of CLIMSIM is provided both in the form of graphs with values given at various nodes, and as discrete data in tabular form. Various types of temperatures and other parameters are reported by the software, some of them are:

1. Dry bulb temperature
2. Wet bulb temperature
3. Relative humidity
4. Effective temperature
5. Dry wall temperature

2.4 Summary report on MINEVENT (ICAMPS) – User interface

The MINEVENT is a mine ventilation software module of ICAMPS software (Integrated Computer Aided Mine Planning Software) (ICAMPS, 2007), which is based on Penn State University’s PENVENT ventilation engine, developed by Ohio Automation company (Ohio Automation, 2003).

This software is built on AutoCAD, and so works from within AutoCAD software, utilizing all its graphical features. Thermal parameters are not used in the program. Gravity is not included in the calculations, thus the software assumes air as an incompressible fluid. Table 10 and Table 11 give the list of input parameters used for model configuration in ICAMPS software.

Table 10. Branch input - ICAMPS

<table>
<thead>
<tr>
<th>Different modes of inputting branches:</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Leakage</td>
</tr>
<tr>
<td></td>
<td>Fixed Q</td>
</tr>
<tr>
<td>Regulator</td>
<td>Dummy Regulator</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------</td>
</tr>
</tbody>
</table>

Q
NVP (° H₂O)
Resistance
Gas contaminant %
C Factor \((P = R \times Q^C)\) for leakage branches
Height & width of regulator
Fan curve

| Resistance Formula | Branch Length, Air way Height, Width, k-factor
|                   | Resistance per 1000 units of length
|                   | Shape Factor Area
|                   | Perimeter – Area
|                   | Stoppings Resistance,
|                   | Pressure – Quantity
|                   | Entry Resistance
|                   | NE B-P: Non-equal branches in parallel
|                   | NE B-S: Non-equal branches in series

Table 11. Fan input data - ICAMPS

<table>
<thead>
<tr>
<th>Blade setting</th>
<th>Pressure – Quantity data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Axial speed</td>
</tr>
<tr>
<td></td>
<td>Air density</td>
</tr>
</tbody>
</table>

This software is designed to work along with as-mined and projected timing maps. These planning maps serve as the basis for drawing the ventilation schematics to sale. As the input file is created by the user, the MineVent solver runs from within
AutoCAD. Once the model is solved, the output is displayed in the form of color-coordinated drawings which illustrate parameters such as airflows in the branches, pressures at nodes, etc.

2.5 Summary report on VUMA – User Interface

VUMA (Marx et al., 2001) (Bluhm et al., 2001) was one of the earliest software developed with graphical display functionalities for an easy user interface. It incorporates user-friendly interface that allows users to construct simulation networks and view them graphically.

A 2-dimensional graphical editor is provided in which the user can construct the ventilation network level-by-level, and can interconnect the levels with shafts or declines. A 3-dimensional graphical viewer is provided for the user to view the network. The results can also be viewed in tabular format.

In VUMA software, there are different types of branch elements provided for the user to choose from. They are:

1) Tunnel
2) Production zone
3) Development heading
4) Shaft
5) Fan
6) Shaft station
7) Control element

Unique user input tables pop up for each selection of the branch element type. A list of parameters is made on the basis of the data tables pop up for the “development end” selection. They have been categorized into three different sections; airflow, thermal and contaminant parameters. These are shown in Tables 12, 13 and 14 respectively.
Table 12. Airflow (Aero) input parameters for Development Heading branch type – VUMA software.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sub-selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Inclination</td>
<td>Up</td>
</tr>
<tr>
<td></td>
<td>Down</td>
</tr>
<tr>
<td>Inclination category</td>
<td>Category 1: 1°</td>
</tr>
<tr>
<td></td>
<td>Category 2: 8°</td>
</tr>
<tr>
<td></td>
<td>Category 3: 15°</td>
</tr>
<tr>
<td></td>
<td>Category 4: 30°</td>
</tr>
<tr>
<td></td>
<td>Category 5: 90°</td>
</tr>
<tr>
<td>Mass flow (kg/s)</td>
<td></td>
</tr>
<tr>
<td>In-heading leakage</td>
<td></td>
</tr>
<tr>
<td>Vent option</td>
<td>Forced</td>
</tr>
<tr>
<td></td>
<td>Exhaust</td>
</tr>
<tr>
<td>Shape</td>
<td>Rectangular</td>
</tr>
<tr>
<td></td>
<td>Circular</td>
</tr>
<tr>
<td></td>
<td>Irregular</td>
</tr>
</tbody>
</table>

Table 13. Thermal input parameters for Development Heading branch – VUMA software.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sub-selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face advance rate (m/day)</td>
<td></td>
</tr>
<tr>
<td>Rock type</td>
<td>Quartzite</td>
</tr>
<tr>
<td></td>
<td>Norite</td>
</tr>
<tr>
<td></td>
<td>Shale</td>
</tr>
<tr>
<td></td>
<td>Etc</td>
</tr>
<tr>
<td>Clothing factor</td>
<td>Category 1: Unclothed</td>
</tr>
</tbody>
</table>
| Other heat/cool loads | Category 2: Light  
Category 3: Moderate  
Category 4: Heavy  
Category 5: Very heavy | Motor Rating (kW)  
Motor rating (kW)  
Machine Work Cycle (Rating)  
Point / Linear (kW or kW/m)  
Heat – Moisture (Rating) |
|----------------------|-------------|-----------------|-----------------|------------------|-----------------|
| Auxiliary fans       | Continuous miners  
Road Headers  
Tunnel Boring Machinery  
Winches          | User defined  
Vehicles        | Service water | Compressed air  
Fissure water     |
|                      |              | Heat – Moisture (Rating)  
Type of Vehicle (Diesel / Electrical)  
Heat Load (kW)  
Pollutants          | Inlet temperature (°C)  
Pipe Diameter (mm)  
Pipe Insulation (Rating) |

Table 14. Contaminant input parameters for Development Heading branch type – VUMA software.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sub-selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1 Gas</td>
<td>Gas Load (l/min)</td>
</tr>
<tr>
<td>Type 1 Gas decay</td>
<td>Radon Source Term (Atoms/m³/s)</td>
</tr>
<tr>
<td>Type 1 Dust</td>
<td>Value (mg/s; delta -mg/m³)</td>
</tr>
</tbody>
</table>
User can change element type on a right click, or in the data input box. He can also reverse the airflow. There are rating categories for several parameters, which make the user input process of critical parameters easier. Some of them are:

1. Duct leakage ratings
2. Development heading inclination ratings
3. Rock cover ratings
4. Pipe insulation ratings
5. Heat-moisture ratings
6. Machine work cycle ratings, etc.

The production zone type has several choices to choose from. Each selection has its own set of parameters to be input by the user. The choices provided for the user are:

1. Narrow reef stoping
2. Drift (and fill)
3. Bending / cutting
4. Open stoping
5. Stop shrinkage
6. Vertical crater retreat
7. Room and pillar
8. Coal longwall

The processes are displayed through a schematic in the data forms, which facilitate the user to understand the input parameters and their role in calculations.
3. **MULTIFLUX - List of input parameters**

The process of the design of user interface for MULTIFLUX requires a basic understanding of the internal functioning of MULTIFLUX. The software uses an innovative solution method for coupling in-rock and in-drift processes. The main idea of the coupling is to separate the two problems, analyze each of them numerically, describe the general behavior of each problem mathematically with surrogate models fitted to data, and re-couple the general, mathematical, surrogate models for the solution of the composite problem. MULTIFLUX has three different modules (Danko, 2008):

1. **NTCF (Numerical Transport Code Functionalization)**
   
   The NTCF module (Danko, 2006) provides a surrogate model for time-dependent heat and moisture flow in the rockmass around the drift. A typical, user-selected NTCF operator equation in the mine ventilation applications is a first-order, linear matrix equation.

2. **CFD (Computational Fluid Dynamics)**
   
   The CFD module solves for heat, moisture, contaminant, and airflow in the airway. This is a lumped-parameter CFD with coupled, CFD sub-models for heat, moisture/vapor, contaminant and airflow.

3. **DISAC (Direct Iteration and Successive Approximation Coupler)**
   
   The results of NTCF and CFD modules are balanced by iteration and successive approximation on the interfacing drift wall during each time division by DISAC module.

   All the three modules are vectorized by design, capable of performing simulations for more than one time divisions consecutively and for the entire 3-dimensional space grid of the in-drift model domain. The NTCF, CFD and DISAC modules are interconnected through shared data files specifying parameters like Temperature, Pressure, heat and mass fluxes, etc, on the shared boundary, that is the drift wall. This is described in Figure 2.
Figure 2. Thermal-Hydrological-Airflow-Contaminant modeling with MULTIFLUX.

From studying the software detailed in Sections 2.1 through 2.5, the input data required for carrying out a simulation in MULTIFLUX are categorized into various subcategories and given below. All the input parameters are categorically presented in Table 15.

1. Airways geometry and attributes
2. Rock strata properties
3. Fan characteristics
4. Power input and heat sources
5. Contaminant sources
6. Cooling power and moisture sources
Table 15. List of input parameters for MULTIFLUX

<table>
<thead>
<tr>
<th><strong>Airway – Geometry and Attributes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Location</strong>: Coordinates, Elevation (Depth).</td>
</tr>
<tr>
<td>• <strong>Size</strong>: Cross sectional area, Perimeter</td>
</tr>
<tr>
<td>• Surface roughness</td>
</tr>
<tr>
<td>• Air doors, Regulators</td>
</tr>
<tr>
<td>• Surface visible wetness factor</td>
</tr>
<tr>
<td>• Air ducts, Obstructions.</td>
</tr>
<tr>
<td>• Ground control, support system</td>
</tr>
<tr>
<td>• Materials handling in the airways</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Rock strata properties</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Rock type</td>
</tr>
<tr>
<td>• Virgin rock temperature (VRT)</td>
</tr>
<tr>
<td>• Bulk density</td>
</tr>
<tr>
<td>• Specific heat</td>
</tr>
<tr>
<td>• Conductivity</td>
</tr>
<tr>
<td>• Age of opening</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Airflow input</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Fan data</strong>: Fan curve, Nominal power, Efficiency</td>
</tr>
<tr>
<td>• <strong>Configuration</strong>: Number of fans, Series / Parallel arrangement</td>
</tr>
<tr>
<td>• Air density during calibration</td>
</tr>
<tr>
<td>• Fixed quantity</td>
</tr>
<tr>
<td>• <strong>Air duct</strong>: Evase, Type, Length, Conductivity, Cross-section, Leakage, Number of segments, Location (within drift cross section).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Power input and Heat sources</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Source type</strong>: Point source, Line source, Diesel, Electrical.</td>
</tr>
<tr>
<td>• <strong>Power input</strong>: Sensible heat, Latent heat, Explosive power, Oxidation power, Geothermal inflow, Machine power, Mine crew size (Metabolic heat), Compressed air usage, Electrical power usage, Diesel fuel usage, Process</td>
</tr>
</tbody>
</table>
Contaminant sources:

- **Type:** Diesel, Methane, Dust, CO₂, etc.
- **Generation rate:** Volumetric flow rate (m³/s) and density, or mass flow rate (kg/s).
- Percentage in air

Cooling power input:

- Spray water
- Spot coolers
- Spray chambers
- Chilling power
4. **Manual – mode user interface development**

A manual mode of input to the MULTIFLUX software is developed as an initial step towards achieving the goal of designing a full-fledged 3-dimensional graphical user interface. A simple layout of the interface for MULTIFLUX is made in MATLAB. This interface program (written using the GUIDE module of MATLAB) takes the input information from the user, regarding the nodal parameters and connection parameters, and writes the parameters into respective files for further processing of data both for CFD module (Danko, 2008) and for the NTCF module (Danko, 2006). The NTCF module is used to model and represent the rock-mass surrounding the mine airway. Using the manual input mode, one can set up a small network with a few nodes, and define various heat, moisture and airflow connections between them. The picture shown in Figure 2 is a snap shot of that interface. The user can select the network type, assign the coordinates of nodes, and define various transport connections. The user is there by able to study simple networks in MULTIFLUX.

The window shown in Figure 3 depicts various drop-down menus that are provided for the selection of the type of network, the direction vector of nodes (type of „from” and „to” nodes) and the type of connection between those nodes. Once the user gives his selection and presses the „Ok” button, various corresponding tables pop up, which the user has to fill the data into. The first drop down menu provides the user four different networks to choose from:

a) Air  
b) Thermal  
c) Moisture  
d) Contaminant

The second and fourth drop down menus give the user a list of different types of nodes, based on the network selected:

a) Dry Rock Surface  
b) Wet Rock Surface
Figure 3. Layout of the GUI for Manual mode of input for MULTIFLUX in GUIDE, MATLAB.
e) Conveyor  
f) Air in Drift  
g) Air in Duct  
h) Heat Source / Sink  

The third drop down menu lists the types of connections one can setup in between the nodes „i” and „j”, depending up on the type of network chosen from the first drop down menu. They are given in Table 16.

After the user defines the type of connection, 5 tables pop up. They are as follows:

1. Nodal parametric table for node – i  
2. Nodal parametric table for node – j  
3. Connection parametric table  
4. Error limits table  
5. Control parametric table  

Table 16. Types of connections for all the network types.

<table>
<thead>
<tr>
<th>Network type</th>
<th>Connection type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air</td>
<td>1. With energy conversion</td>
</tr>
<tr>
<td></td>
<td>2. Without energy conversion</td>
</tr>
<tr>
<td></td>
<td>3. Momentum</td>
</tr>
<tr>
<td></td>
<td>4. Mass Flux</td>
</tr>
<tr>
<td></td>
<td>5. User Defined</td>
</tr>
<tr>
<td>Heat</td>
<td>1. Conduction</td>
</tr>
<tr>
<td></td>
<td>a) Parallel Plates</td>
</tr>
<tr>
<td></td>
<td>b) Cylindrical Surfaces</td>
</tr>
<tr>
<td></td>
<td>2. Convection</td>
</tr>
<tr>
<td></td>
<td>a) Free Convection</td>
</tr>
<tr>
<td></td>
<td>(i) External</td>
</tr>
<tr>
<td></td>
<td>(ii) Enclosed Vertical Plates</td>
</tr>
<tr>
<td></td>
<td>(iii) Circular Annulus</td>
</tr>
<tr>
<td></td>
<td>(iv) Horizontal Enclosure</td>
</tr>
<tr>
<td></td>
<td>(v) Dispersion</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td>a) Evaporation Condensation</td>
</tr>
<tr>
<td></td>
<td>b) Tripole Evaporation Condensation</td>
</tr>
<tr>
<td></td>
<td>c) User Defined</td>
</tr>
<tr>
<td></td>
<td>d) In front of Vertical Plate</td>
</tr>
<tr>
<td></td>
<td>e) Circular Annulus</td>
</tr>
<tr>
<td></td>
<td>f) Horizontal Enclosure</td>
</tr>
<tr>
<td></td>
<td>g) Switched Connection</td>
</tr>
<tr>
<td></td>
<td>h) Darcy Flow</td>
</tr>
</tbody>
</table>

Figure 4 shows the data entry tables for the Air network. The “i” node is “Air in Drift”, and the “j” node is “Air in Drift”, the connection type is “Laminar or turbulent flow with energy conversion”. If the nodes “i” and “j” are of same type, it means that the connection is axial. All the tables display default values. The user can enter the values of his choice. If the “Generator” check box is checked, it is a self-ended connection, where both the nodes are one and the same. The “Generator” function is illustrated in Figure 5. Figure 5 also shows the data tables when “Heat” network is chosen.
Figure 4. GUI window showing the data entry tables for one of the connection types in the „Air” network.
The nodal parametric tables (both for i and j nodes) have the following data:

- Node number
- X Co-ordinate
- Y Co-ordinate
- Z Co-ordinate
- Source / Sink

According to the selection of the types, the Connection parametric table varies. The Error Limits table and Control Parametric table have a long list of essential parameters that need to be set for the execution of CFD (Computational Fluid Dynamics). If any inappropriate selection is done, error messages pop up which let the user know that a change in the selection should be made. This can be seen in Figure 6.

After the data input to the tables is done, the user can click the button “Next Connection” to enter the other connections and data. After the input of all the data is completed, the user can click the button “Save MF files” to load various data into respective files, for further use by CFD.
Figure 5. GUI window showing the data entry tables for one of the connection types in the "Heat" network, and illustrating "Generator" connection.
Figure 6. Display of the error message for an inappropriate combination of selections. There can never be conduction heat transfer in between dry rock surface and a force duct.
5. Conceptualizing the user interface – Pre-Processor and Post-Processor

Model definition involves setting up of all possible transport processes between various elements such as pressure, heat, and moisture sources, flow lines, tunnel/drift walls, air ducts. The model provides a coupled solution between rockmass and airway domains, obeying all relevant thermodynamic and heat/mass/momentum transport principles and conservation laws and compressible fluid properties.

In order to create input for MULTIFLUX, a 3-dimensional node network needs to be generated. Google’s SketchUp software’s GUI is used as the CAD platform for graphical user interface for the software. The CAD interface provides planners access to complex heat and moisture transport functions that would otherwise be unavailable to the end user.

Secondly, connection to MATLAB should be provided, an environment in which the user may create data post-processing scripts as well as result visualization. This will be a separate, stand-alone interface for the engineer to conduct numerical analysis and to allow data manipulation, once the geospatial node network has been established.

As part of developing graphical user interface for the MULTIFLUX, we need to build a bridge in between CAD, MULTIFLUX and MATLAB codes. This bridge would be a code which essentially translates CAD data into the format that is understandable by MATLAB.

The user first defines a wireframe of the mine ventilation network for the CAD platform by drawing it in the CAD environment or importing the wireframe from software such as VULCAN. For each branch in the wireframe network, the user may select a detailed model-element. This could be a simple task, such as a ventilation problem without heat and moisture. Or, it could be a branch with multiple, simultaneous processes with air, heat and moisture flow. The templates will allow for further discretization of each branch, by defining the number of sub-divisions. Further nodes will be added to those shown in the simple wireframe network. These internal nodes will
be processed in the MULTIFLUX environment together with model-elements processing. These will be internal nodes, not necessary for the users to worry about.

The user will be given choices in the form of templates. Templates will be offered for choosing from a pop-up window in the CAD platform. The concept in a nutshell is:

1. The user defines the wire diagram of a ventilation network in the CAD platform.
2. He/she enhances the wire diagram by selecting a task for the network (is it an air flow problem, or a mine climate problem, or a development-end problem?). He/she picks a template for each (i,j) branch in CAD platform, then, fills in data tables that pop up for that template. CAD software supports this activity with the template graphics. Table pop-up follows the needs for a specific template;
3. CAD then creates an ASCII data vector from the user's input for each (i, j) branch and creates a readable file for MULTIFLUX.
4. MULTIFLUX takes on from here, and solves the model. MULTIFLUX reads and re-configures the wireframe network by squeezing in the internal nodes. The network MULTIFLUX solves is much bigger than the simple wireframe the user sees. MULTIFLUX can be started outside CAD.
5. Once MULTIFLUX creates the full solution, it gives the solution back in the form of ASCII files to CAD for simple display.
6. MULTIFLUX will make data files for animation of the results.

This process is presented as a flow chart in Figure 7.
Figure 7. Process flowchart between CAD, MULTIFLUX and MATLAB environments.
6. Developing templates and data tables

Because of the complexity involved in the calculations and establishment of various transport connections, the design of data input is a tedious task in the MULTIFLUX software. So, a template-based concept is developed according to which the input data can be assigned easily by the user for selected branches in a ventilation network.

Figure 8. The schematic of a sample ventilation network that is drawn in the GUI window of MULTIFLUX.

Figure 8 shows the schematic of the ventilation network of the California mine located at El Rico, Chile, as an example. The schematic is drawn on the graphical user interface developed for MULTIFLUX software. 3-Dimensional drawing space and 2-dimensional grids are provided to the user for easy drawing of ventilation networks.
together with cross-sectional views of networks in XY, YZ and XZ planes. Figure 9 shows a simpler (skeleton) view of the network shown in Figure 8.

![Figure 9. Basic view of the network shown in Figure 8.](image)

With the introduction of the template based concept, the data input and model configuration in the MULTIFLUX software becomes much easier. According to this concept, the input data can be assigned easily by the user for selected branches (lines in the network schematic – Figures 8 and 9) for a given ventilation network. Various templates are developed, considering all the possible scenarios that an underground mine ventilation network could have. Figures 11 to 25 show some templates that are already developed.

Template assignment can be done in two ways:
1. The user can select one or a group of branches in the network, which is shown in Figure 9. Then a window called “Template Selection Window” will pop up. By selecting a template from the array of templates provided in the template selection widow, the user can assign that template to the branch.

2. Or the user can select a template beforehand, and start drawing the ventilation network. He can switch to another template and then draw another set of branches, and so on.

Once the template assignment is done, the next window pops up, which is called “MULTIFLUX Parameter Input Form”. In this form, the necessary data to be input by the user is displayed in various tabs.

![Select a Template Table](image)

Figure 10. Template Selection Window. This window shows a few templates. It is being populated with all the possible templates.
The user gives necessary input data for calculations in those forms of the respective input data window. Following are different tabs that will be displayed depending on the template selected.

1. Branch window
2. Airflow / Fan window
3. Rock Strata window
4. Heat Source / Sink window
5. Contaminant window
6. Duct Geometry window
7. Duct Fan and Cooler window
8. Conveyor window
9. Development - End window
10. Parametric Table window

Based on the template selected, the data input window could comprise either single window from the above list or various combinations of them. User input forms for various templates are given below, under respective templates, in Tables 17 through 36.

6.1 Template 1 - Airflow Connections:

Figure 11 shows the basic template, which depicts a branch that has only airflow connections and does not include any heat, moisture or contaminant connections. Table 17 is the input form for this template, which is the “Airway Geometry Input form” described above. It has three sections, namely branch identifier, branch geometry and branch resistance sections:

1. **Branch Identifier Section:** This section asks the user to input the branch name, and surface connection type. It displays the node numbers and the x, y, z coordinates of both ends of the branch. The user can change the default node numbers given for the branch ends.

   The x, y, and z coordinates data of the branch are extracted from the 3-dimensional space of the SketchUp Graphic window. The length of the airway can either be given manually by the user or left unchecked for the software to calculate
from the coordinate information. The user is provided to choose the discretization length of the airway. The surface state can be left unchecked if the branch is not connected to surface. This section displays the figure of the respective template.

2. **Branch Geometry Section:** This section is for choosing the shape of the airway and to give the respective size data. Five options are provided for the user to choose from. They are rectangular, circular, horse-shoe (tunnel), trapezoidal, and irregular. Depending on the shape selected, respective size parameters will be highlighted in this section. Area and perimeter are calculated and displayed instantaneously, when the user gives the size data like height, width, etc. All the fields have lower and upper limits. If the value entered is beyond the acceptable range, validation messages pop up. The obstruction area accounts for any objects that are present in the branch such as transformers, cooling pipe lines, etc.

3. **Branch Resistance Section:** The user can choose the resistance type he would like to input, and give the respective data. Similar to the Branch Geometry, the relevant parameters get highlighted in the Branch resistance after selecting the Resistance type. Except for the “Resistance” type, for all the other types, the resistance value is calculated based on the user input, or based on the simulation results, and displayed.
The user can give the shock loss factor value or leave it blank. Following are the
equations involved to calculate resistance from the input data (Hartman et al., 1997).
Equations 1 and 2 are for calculating the resistance from pressure and quantity input,
and from friction factor and dimensions of the branch respectively.

\[ R = \frac{P}{Q^2} \]  \hspace{1cm} \text{Eq (1)}

where,
\[ R = \text{Resistance (N-s}^2/\text{m}^8) \]
\[ P = \text{Pressure (Pa)} \]
\[ Q = \text{Quantity (m}^3/\text{s)} \]

\[ R = \frac{k(L + L_{eq})P}{A^3} \]  \hspace{1cm} \text{Eq (2)}

where,
\[ R = \text{Resistance (N-s}^2/\text{m}^8) \]
\[ k = \text{Friction factor (kg/m}^3) \]
\[ L = \text{Length of the branch} \]
\[ L_{eq} = \text{Equivalent length for shock loss} \]
\[ P = \text{Perimeter of the branch (m)} \]
\[ A = \text{area of the branch (m}^2) \]

The user clicks the Submit button, and the input data is returned to the SketchUp main interface, and stored in the MULTIFLUX data libraries. The Reset button is for clearing the input data, and the Cancel button is to cancel the input and get out of (close) the window.

Table 18 shows the parametric table input form for Template 1. This window is included in the input forms of all the templates, as this is necessary for every template. This form comprises of two different lists of parameters. One is the list of error limits, and the other is list of control parameters which are required for internal calculations in the DISAC module of MULTIFLUX. Default values are displayed in this form to reduce the user input time. The user can modify these values.
Table 17. Airway Geometry Input form for Template 1 shown in Figure 10. The user provides information about the branch such as branch name, surface connection, shape, geometry and resistance.
Table 18. Input form for Parametric tables, Template 1.

<table>
<thead>
<tr>
<th>Error Limits</th>
<th>Control Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errlim_heat</td>
<td>Latentf</td>
</tr>
<tr>
<td>Errlim_moist</td>
<td>Preqf</td>
</tr>
<tr>
<td>Looplim</td>
<td>Task2f</td>
</tr>
<tr>
<td>wP</td>
<td>Tc_f</td>
</tr>
<tr>
<td>Errlim_latent</td>
<td>Ala_h</td>
</tr>
<tr>
<td>Looplim_latent</td>
<td>Ala_m</td>
</tr>
<tr>
<td>wT_latent</td>
<td>Initf</td>
</tr>
<tr>
<td>Errlim_cond</td>
<td>NTCF_f</td>
</tr>
<tr>
<td>Looplim_cond</td>
<td>CFD_iteration_f</td>
</tr>
<tr>
<td>wP_cond</td>
<td>Vd_f</td>
</tr>
<tr>
<td>Errlim_air</td>
<td>Ce_f</td>
</tr>
<tr>
<td>Looplim_air</td>
<td>Air_model</td>
</tr>
<tr>
<td>w_air</td>
<td>Heat_model</td>
</tr>
<tr>
<td>w_airD</td>
<td>Moisture_model</td>
</tr>
<tr>
<td>W_admita</td>
<td>Gravity (m/s²)</td>
</tr>
<tr>
<td>Errlim_T</td>
<td>1.000e-03</td>
</tr>
<tr>
<td>DwC</td>
<td>1.000e-09</td>
</tr>
<tr>
<td>Dwsd</td>
<td>1.000e-12</td>
</tr>
<tr>
<td>Dwvar</td>
<td>1.000e-12</td>
</tr>
<tr>
<td>Errlim_airD</td>
<td>1.000e-03</td>
</tr>
<tr>
<td>wq_latent</td>
<td>0.950</td>
</tr>
<tr>
<td>Looplim_airD</td>
<td>3</td>
</tr>
</tbody>
</table>
6.2 Template 2 - Fan Branch

In mines, sometimes fans are installed such that they take up the whole cross-sectional area of branch. This template is for such special case, and the user input window for this template comprises of three tabs. One tab is the Branch window, shown in Table 19, which is the same as the one shown in Table 17. The other one is the “Airflow / Fan window”, shown in Table 20. This window has two different sections to be chosen by the user. The user can select one among these two sections:

1. **Fixed Flow Section**: If fixed values for the pressure and flow rate are desired, associated values should be entered in the respective fields. The user will give either fixed quantity (m$^3$/s) or fixed pressure (Pa).

2. **Fan Section**: In this section, there are three sub-sections, namely fan configuration, fan curve, and fan data sections.
   a) **Fan Configuration**: The user can give the number of fans, and their arrangement in the airway. The airflow direction can be reversed by checking the “Reverse” check box.
b) **Fan Curve:** The user can select a fan curve from the MULTIFLUX fan library, or can build a user defined fan curve by filling the fan curve input table, such as pressure and airflow. Using the ,,Add” button, the entered fan curve data can be stored in the MULTIFLUX fan library with a unique name.

If there are multiple fans, then there would be different tabs named as “Fan 1”, “Fan 2” and so on, depending on the number of fans in the branch. In those tabs, the user can give the respective fan curve data. The results of this input are plotted dynamically in the graph area beside the table. The fan curve has various parameters represented by the plot of airflow quantity vs. fan total pressure. These curves are fit using either cubic or linear interpolation based on the user’s preference. The fan curve for the combination of fans is also displayed under another tab.

c) **Fan Data:** Data such as fan speed, fan geometry including evase and bell mouth etc, are input by the user in this section. A figure depicting a fan with evase and bell mouth is displayed. The user could give the bellmouth and evase diameters or leave them blank, if there is no bellmouth or evase in the fan system.

The third window in the input form for Template 2 is the Parametric Table Input form, which is similar to the one shown in Table 18.
Table 19. Airway Geometry Input Form - Template 2.

<table>
<thead>
<tr>
<th>Branch Identifier</th>
<th>Node i:</th>
<th>Node j:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X - Coordinate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y - Coordinate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z - Coordinate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Discretization</td>
<td>m</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Surface Connection</th>
<th>Intake</th>
<th>Exhaust</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Branch Geometry</th>
<th>Shape</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rectangular</td>
<td>Width m</td>
</tr>
<tr>
<td></td>
<td>Circular</td>
<td>Height m</td>
</tr>
<tr>
<td></td>
<td>Tunnel</td>
<td>Diameter m</td>
</tr>
<tr>
<td></td>
<td>Trapezoidal</td>
<td>Area m²</td>
</tr>
<tr>
<td></td>
<td>Irregular</td>
<td>Perimeter m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Top Width m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bottom Width m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Obstruction Area m²</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch Resistance</th>
<th>Resistance Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Resistance</td>
<td>N·s²/m⁴</td>
</tr>
<tr>
<td></td>
<td>P/Q</td>
<td>Pa</td>
</tr>
<tr>
<td></td>
<td>K-Factor (Atkinson’s)</td>
<td>m³/s</td>
</tr>
<tr>
<td></td>
<td>Resistance/Length</td>
<td>kg/m³</td>
</tr>
<tr>
<td></td>
<td>KOD (Relative Roughness)</td>
<td>kg/m⁴</td>
</tr>
<tr>
<td></td>
<td>Shock Loss Factor</td>
<td>Parallel Factor</td>
</tr>
</tbody>
</table>
Table 20. Input form for Fan - Template 2.

<table>
<thead>
<tr>
<th>Fan Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Fans: 2</td>
</tr>
<tr>
<td>Reverse: ✔</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fan Curve</th>
<th>Select from Library ▼</th>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quantity (m³/s)</th>
<th>Fan Static Pressure (Pa)</th>
<th>Fan Total Pressure (Pa)</th>
<th>Fan Total Efficiency (%)</th>
<th>Absorbed Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.0</td>
<td>2500.0</td>
<td>2557.6</td>
<td>50.0</td>
<td>61.4</td>
</tr>
<tr>
<td>18.0</td>
<td>2250.0</td>
<td>2479.8</td>
<td>55.0</td>
<td>81.1</td>
</tr>
<tr>
<td>24.0</td>
<td>2000.0</td>
<td>2361.2</td>
<td>60.0</td>
<td>90.3</td>
</tr>
<tr>
<td>30.0</td>
<td>1800.0</td>
<td>2159.6</td>
<td>65.0</td>
<td>95.2</td>
</tr>
<tr>
<td>36.0</td>
<td>1600.0</td>
<td>2008.4</td>
<td>70.0</td>
<td>92.5</td>
</tr>
<tr>
<td>42.0</td>
<td>1400.0</td>
<td>1816.4</td>
<td>75.0</td>
<td>89.2</td>
</tr>
<tr>
<td>48.0</td>
<td>1200.0</td>
<td>1675.2</td>
<td>80.0</td>
<td>84.0</td>
</tr>
<tr>
<td>54.0</td>
<td>1000.0</td>
<td>1516.4</td>
<td>85.0</td>
<td>78.4</td>
</tr>
<tr>
<td>60.0</td>
<td>800.0</td>
<td>1377.2</td>
<td>90.0</td>
<td>68.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fan Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic</td>
</tr>
<tr>
<td>Linear</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fan Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan Speed RPM</td>
</tr>
<tr>
<td>Reverse Pressure Pa</td>
</tr>
<tr>
<td>Reverse Quantity m³/s</td>
</tr>
<tr>
<td>Air Density during Calibration kg/m³</td>
</tr>
<tr>
<td>Fan Diameter (a) m</td>
</tr>
<tr>
<td>Bell Mouth Diameter (b) m</td>
</tr>
<tr>
<td>Evase Diameter (c) m</td>
</tr>
</tbody>
</table>

Submit | Reset | Cancel
6.3 Template 3 - Air – Thermal – Moisture Branch

Figure 13. A branch with air, thermal and moisture connections. Wet floor is shown in light blue.

Figure 14. Side view of Figure 13.
Template-3 shows a branch with airflow, moisture and thermal connections. It could have a heat source, a cooling source, and a water source. Figure 13 depicts this template, and Figures 14 and 15 are the side and front views of Figure 13, respectively.

The user input window for this template has four tabs. They are:

1. Branch Geometry Window
2. Rock Strata Window
3. Heat Source / Sink Window
4. Parametric table window

First tab is the “Airway Geometry Input Form” (Table 18) which is similar to the one shown in Table 17. And the fourth tab, the “Parametric Table Input Form” is similar to the input form shown in Table 18.

The second tab is “Rock Strata Input Form” which has three different sections, namely thermophysical data, age data, and temperature data. Table 22 shows the input form for rock strata input.
a. **Thermophysical data:** The user can select the rock type from a given list of rock types or choose “others” and define a user-defined type. The data such as rock density, thermal conductivity, specific heat, and thermal diffusivity are displayed automatically from the MULTIFLUX rock property library (Table 23) once the user selects a rock type that is already defined. Or else, the user will give custom values for the user-defined type. Any three of the four data is required to be input by the user, and the fourth one will be automatically calculated and displayed dynamically, using the following relation, given in Equation 3:

\[
\alpha = \frac{k}{\rho C_p} \quad \text{............... Eq (3)}
\]

where,

- \(\alpha\) = Thermal diffusivity (m\(^2\)/s)
- \(k\) = Thermal conductivity (W/m-K)
- \(\rho\) = Bulk density (kg/m\(^3\))
- \(C_p\) = Specific heat (J/kg-K)

The user can select the rock type surrounding the drift from either homogeneous or heterogeneous. There is a figure showing the shape of the drift, with check boxes provided on the four walls of the branch. If homogeneous is selected, all the boxes are checked, implying all the walls are selected. The data given by the user is applied to all the walls. If heterogeneous type is selected, the user can select the roof, floor, and side walls independently and assign unique rock properties to each.

b. **Age data:** The age of the airway at inlet and outlet can be given in this section.

c. **Temperature:** There are two modes of input for virgin rock temperature:

1) By directly giving the virgin rock temperature at inlet and outlet of the airway.
2) By defining the geothermal step (°C/m) and surface ground temperature. The software gets the depth information of depth by extracting the coordinate data from the SketchUp 3-dimensional platform. By multiplying the depth of the
end and geothermal step, one gets the temperature. By choosing the respective radio button provided, the user can give the data.


<table>
<thead>
<tr>
<th>Branch Identifier</th>
<th>Surface Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Number</td>
<td></td>
</tr>
<tr>
<td>Node i:</td>
<td></td>
</tr>
<tr>
<td>Node j:</td>
<td></td>
</tr>
<tr>
<td>X - Coordinate</td>
<td></td>
</tr>
<tr>
<td>Y - Coordinate</td>
<td></td>
</tr>
<tr>
<td>Z - Coordinate</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Discretization</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch Geometry</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>Width</td>
</tr>
<tr>
<td>Circular</td>
<td>Height</td>
</tr>
<tr>
<td>Tunnel</td>
<td>Diameter</td>
</tr>
<tr>
<td>Trapezoidal</td>
<td>Area</td>
</tr>
<tr>
<td>Irregular</td>
<td>Perimeter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch Resistance</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistance Type</td>
<td>Resistance</td>
</tr>
<tr>
<td>P/Q</td>
<td>Pressure (P)</td>
</tr>
<tr>
<td>K-Factor (Atkinson’s)</td>
<td>Flow Rate (Q)</td>
</tr>
<tr>
<td>Resistance / Length</td>
<td>K-Factor</td>
</tr>
<tr>
<td>KOD (Relative Roughness)</td>
<td>Resistance / 1000 m</td>
</tr>
<tr>
<td>Shock Loss Factor</td>
<td>Parallel Factor</td>
</tr>
</tbody>
</table>
Table 22. Input form for the Rock Strata (Heat and Moisture) – Template 3.

Table 23. MULTIFLUX rock property library associated with the input form for the Rock Strata (Heat and Moisture) – Template 3. (Clark, S. P., Jr., 1966) (Kim, J. et al., 2007) (The Engineering ToolBox, 2011)

<table>
<thead>
<tr>
<th>Rock type</th>
<th>Density „ρ” (kg/m³)</th>
<th>Thermal conductivity „λ” (W/m-K)</th>
<th>Specific heat „C_p” (kJ/kg-K)</th>
<th>Diffusivity „α” (m²/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argillite</td>
<td>2620</td>
<td>2.09</td>
<td>0.870</td>
<td>9.169e-04</td>
</tr>
<tr>
<td>Sandstone</td>
<td>2470</td>
<td>1.97</td>
<td>0.837</td>
<td>9.529 e-04</td>
</tr>
<tr>
<td>Mudstone</td>
<td>1923</td>
<td>3.36</td>
<td>0.816</td>
<td>2.141e-03</td>
</tr>
<tr>
<td>Limestone</td>
<td>2610</td>
<td>2.1</td>
<td>0.840</td>
<td>9.579e-04</td>
</tr>
<tr>
<td>Granite</td>
<td>2670</td>
<td>3.3</td>
<td>0.816</td>
<td>1.515e-03</td>
</tr>
<tr>
<td>Shale</td>
<td>2620</td>
<td>3.1</td>
<td>0.921</td>
<td>1.285e-03</td>
</tr>
</tbody>
</table>
The third tab is the “Heat Source / Sink data” window, shown in Table 24. This allows the user to give various kinds of heat sources such as diesel, electrical, metabolic heat, etc. and cooling sources such as spray water, spot coolers, etc. This input form has two sections, heating source and cooling source:

1. **Heating source**: The user can choose the source as either point or linear. Once selected, the user is prompted to define the source type from a drop down menu which has diesel, electrical, oxidation, explosive, ground water and metabolism as
types. After the selection is made, the appropriate input form is displayed below, which is the dynamic part of the window. The data entry form for each type of heat source is given in Table 26.

a. Diesel source:

There are two sets of input data that may be given by the user for a diesel machine. If the diesel consumption rate is given, the corresponding calculations involved to get the values of heat output, moisture output, and contaminant output are presented in Appendix 3. (Engineering ToolBox, 2011b) (Deutz, 2011) (Shell, 2010). The governing equation for the combustion of a diesel molecule is given in Equation 4.

$$1 \text{ kg of Diesel} = 3.116 \text{ kg (CO}_2\text{)} + 2.352 \text{ kg of (H}_2\text{O)} (\uparrow) + 28,800 \text{ kJ (heat)}.$$  

............. Eq (4)

If nominal power is input by the user, the following equation (Equation 5) is used to calculate the sensible heat output. The moisture output is calculated from the equations provided in Appendix 3.

$$Sensible\ heat = Nominal\ power \times (1 - efficiency) \quad \ldots \quad \text{Eq (5)}$$

b. Electrical source:

When the user inputs nominal power of the electrical power source such as an electrical transformer or substation, etc., and the efficiency of the power source, the sensible heat output is calculated and used as heat source in MULTIFLUX. It is given Equation 6.

$$Sensible\ heat = Nominal\ power \times (1 - efficiency) \quad \ldots \quad \text{Eq (6)}$$

c. Explosive energy:

The user is provided with a variety of explosive types, bulk explosive, dynamite, emulsions, ANFO and seismic. The energy released per kilogram of each of these explosives is given in Table 25, the MULTIFLUX explosive energy library. Once the user gives the data for explosive amount used for blasting and efficiency of the energy released to blast, the sensible heat is calculated and
displayed as explosive power in the window. Generally 50% to 90% of the explosive energy is wasted (Hartman et al., 1997). It is assumed that almost all the wasted energy converts into heat which is absorbed by the rock immediately, and eventually is gradually released into the air stream (Hemp and Deglon, 1979) (Fenton, 1980).

\[ \text{Explosive energy} = S \times M \times (1 - \eta) \quad \text{.... Eq (7)} \]

where,

- \( S \) = Specific energy of the explosive
- \( M \) = Amount of the explosive used
- \( \eta \) = efficiency

The energy released from the explosives is calculated from the above equation, Equation 7. This heat source is then averaged over the simulation time division.

Table 25. MULTIFLUX explosive energy library, associated with heat source / sink form for Template – 3. (Dyno, 2009)

<table>
<thead>
<tr>
<th>Explosive type</th>
<th>Specific energy (kCal/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk explosive</td>
<td>780</td>
</tr>
<tr>
<td>Dynamite</td>
<td>1000</td>
</tr>
<tr>
<td>Emulsion (Packaged)</td>
<td>860</td>
</tr>
<tr>
<td>ANFO</td>
<td>1100</td>
</tr>
<tr>
<td>Seismic explosive</td>
<td>2000</td>
</tr>
</tbody>
</table>

d. Metabolic heat:

When the user selects the level of manual activity from one of the four choices, light, moderate, heavy, and very heavy, and gives the number of people working in the given airway, the metabolic heat is given from Table 27, the MULTIFLUX metabolic heat library.
Table 26. Data entry forms for all types of heat sources in the Heat Source / Sink form – Template 3.

<table>
<thead>
<tr>
<th>Source type</th>
<th>Diesel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal power</td>
<td>kW</td>
</tr>
<tr>
<td>Efficiency</td>
<td>%</td>
</tr>
<tr>
<td>Diesel consumption rate</td>
<td>kg/s</td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
</tr>
<tr>
<td>Latent heat</td>
<td>kW</td>
</tr>
<tr>
<td>Moisture input</td>
<td>kg/s</td>
</tr>
<tr>
<td>CO₂ release</td>
<td>kg/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source type</th>
<th>Metabolism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity type</td>
<td>Light Heavy</td>
</tr>
<tr>
<td>Mine crew size</td>
<td></td>
</tr>
<tr>
<td>Metabolic heat</td>
<td>kW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source type</th>
<th>Groundwater</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water temperature at emission</td>
<td>°C</td>
</tr>
<tr>
<td>Water flow rate</td>
<td>m³/s</td>
</tr>
<tr>
<td>Drainage channel</td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>m</td>
</tr>
<tr>
<td>Length</td>
<td>m</td>
</tr>
<tr>
<td>Depth</td>
<td>m</td>
</tr>
<tr>
<td>Location from entry</td>
<td>m</td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
</tr>
<tr>
<td>Latent heat</td>
<td>kW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source type</th>
<th>Oxidation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ore type</td>
<td>Coal Sulfide</td>
</tr>
<tr>
<td>O₂ content by volume at inlet</td>
<td>%</td>
</tr>
<tr>
<td>O₂ content by volume at outlet</td>
<td>%</td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
</tr>
</tbody>
</table>
Table 27. The MULTIFLUX metabolic heat library. (McPherson, 1993) (Hartman & Mutmansky, 2002)

<table>
<thead>
<tr>
<th>Level of manual activity</th>
<th>Power per unit skin area (W/m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>95</td>
</tr>
<tr>
<td>Moderate</td>
<td>145</td>
</tr>
<tr>
<td>Heavy</td>
<td>245</td>
</tr>
<tr>
<td>Very Heavy</td>
<td>340</td>
</tr>
</tbody>
</table>

The skin area is calculated from the empirical formula given in Equation 8:

\[
Skin \ area = 0.202 \times (m_b)^{0.425} \times (h_b)^{0.725} \quad \ldots \ldots \ Eq \ (8)
\]

where,

\[ m_b = \text{Mass of the worker’s body} \]
\[ = 70 \text{ kg for an average person} \]
\[ h_b = \text{Height of the worker’s body} \]
\[ = 1.73 \text{ m for an average person} \]

So, we get 1.83 m² as the skin area for an average person.

e. **Ground water heat:**

In case of any water oozing out of a rock fissure at high temperatures, that ground water can be a significant source of heat for the branch. For calculating the heat generated from ground water, the user will give the temperature of the groundwater at the point of emission, the flow rate, and the surface area exposed to the mine airflow, in terms of the width and length of the drainage channel. The airflow channel and the drainage channel behave as a heat exchanger and exchange heat. This amount of heat (both sensible and latent) is calculated from MULTIFLUX’s internal defined functions (Danko, 2008).
f. Oxidation:

Oxidation becomes a significant heat source in ores where sulfur and carbon are exposed to the air and become oxidized. This heat can be calculated from the oxygen content by percent volume at the inlet and outlet of branch. Thus, with the amount of oxygen being consumed, we can calculate the sensible heat from the following equations, Equation 9 through 11 (McPherson, 1993):

\[ C + O_2 = CO_2 + 12,675 \text{ kJ of heat (per 1 kg of } O_2) \]  

\[ S + O_2 = SO_2 + 9,304 \text{ kJ of heat (per 1 kg of } O_2) \]  

\[ \text{Amount of } O_2 \text{ consumed} = (O_2 \text{ at inlet} - O_2 \text{ at outlet}) \times \text{Density of } O_2 \]

Assuming density of \( O_2 \) = 1.3 kg/m\(^3\) (McPherson, 1993).

2. Cooling Source (Heat sink):

Along with the heat sources, the user can also input heat sink data. The user can select the type of cooling source from a drop down menu that consists of spray chamber, spray water, spot cooler and ice system as types. He can specify the location of it in the branch. After the selection is done, the appropriate input form comes up below it, which is the dynamic part of the window. The data entry form for each type of heat sink is given in Table 28.

a. Spot cooler:

Spot cooler is the generalization of any closed loop heat exchanger in which the working fluid (water or any other coolant) does not have any physical contact with the air stream in the airway. The only interaction that takes place between the coolant and the air is through heat exchange. Chilled coolant at low temperature enters the system, absorbs heat from the air, and leaves at a higher temperature. The sensible heat is calculated from Equation 12 (McPherson, 1993):

\[ Q_s = m_w \times C_{pw} \times (T_2 - T_1) \]  

\[ \text{Eq (12)} \]
where,

\[ Q_s = \text{Sensible heat lost by the air} \]
\[ m_w = \text{Mass flow rate of water} \]
\[ C_{pw} = \text{Specific heat capacity of water at inlet temperature} \]
\[ T_1 = \text{Water inlet temperature} \]
\[ T_2 = \text{Water outlet temperature} \]

b. Ice system:

Ice system is similar to the spot cooler heat exchanger, except that ice is fed into the system instead of feeding chilled water. The sensible heat lost by the air consists of three components:

1. The sensible heat gained by ice from inlet temperature to 0°C.
2. The latent heat of fusion of ice to convert to water at 0°C.
3. The sensible heat gained by water from 0°C to outlet temperature.

These three components are calculated from Equations 13 through 16 (McPherson, 1993):

\[ Q_{si} = m \times C_{pi} \times T_1 \quad \text{......... Eq (13)} \]
\[ Q_i = m \times L_i \quad \text{......... Eq (14)} \]
\[ Q_{sw} = m \times C_{pw} \times T_2 \quad \text{......... Eq (15)} \]
\[ Q = Q_{si} + Q_{sw} + Q_i \quad \text{......... Eq (16)} \]

where,

\[ Q = \text{Sensible heat lost by air} \]
\[ Q_{si} = \text{Sensible heat gained by the ice} \]
\[ Q_{sw} = \text{Sensible heat gained by the water} \]
\[ m = \text{Mass flow rate of ice/water} \]
\[ C_{pi} = \text{Specific heat capacity of ice at inlet temperature} \]
\( C_{pw} \) = Specific heat capacity of water at outlet temperature

\( L_i \) = Latent heat of fusion of ice

\( T_1 \) = Water inlet temperature

\( T_2 \) = Water outlet temperature

c. **Spray water:**

This cooling system is fully open loop system. The water that is sprayed does not enter the system again. Systems such as dust suppression units come under this category. After sprayed on the face, some portion of the water evaporates into the air stream. The water that is left flows into the drainage channel at a higher temperature. The sensible heat and latent heat are calculated as shown in Equations 17 through 19 (McPherson, 1993):

\[
Q_l = (m_1 - m_2) \times L_i \quad \text{......... Eq (17)}
\]

\[
Q_s = m_2 \times C_p \times (T_2 - T_1) \quad \text{......... Eq (18)}
\]

\[
Q = Q_s + Q_l \quad \text{......... Eq (19)}
\]

where,

\( Q \) = Sensible heat loss from air

\( m_1 \) = Water spray rate

\( m_2 \) = Drainage water rate

\( T_1 \) = Spray water temperature

\( T_2 \) = Drainage water temperature

\( C_p \) = Specific heat capacity of water at inlet temperature

\( Q_i \) = Latent heat gain by the portion of water evaporated

\( Q_s \) = Sensible heat gained by the water
Table 28. Data entry tables for all the types of cooling sources in the Heat Source / Sink form – Template 3.

<table>
<thead>
<tr>
<th>Cooling sources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sink type</strong></td>
</tr>
<tr>
<td>Distance from drift entrance</td>
</tr>
<tr>
<td>Water initial temperature</td>
</tr>
<tr>
<td>Water final temperature</td>
</tr>
<tr>
<td>Water flow rate</td>
</tr>
<tr>
<td>Refrigeration</td>
</tr>
</tbody>
</table>

| **Sink type** | **Spot cooler** |
|------------------|
| Ice system |
| Distance from drift entrance | m |
| Ice supply temperature | °C |
| Water final temperature | °C |
| Ice supply rate | kg/s |
| Refrigeration | kW |

| **Sink type** | **Spray water** |
|------------------|
| Distance from drift entrance | m |
| Spray rate | kg/s-m² |
| Surface area | m² |
| Drainage flow rate | kg/s |
| Input temperature | °C |
| Drainage water temperature | °C |
| Refrigeration | kW |
| Moisture input | kg/s |

| **Sink type** | **Spray chamber** |
|------------------|
| Distance from drift entrance | m |
| Water flow rate at inlet | kg/s |
| Water flow rate at outlet | kg/s |
| Water initial temperature | °C |
| Water final temperature | °C |
| Refrigeration | kW |
| Moisture input | kg/s |

d. **Spray chamber:**

Both mass and heat exchange take place between the working fluid and air in this system. Sensible and latent heat are calculated from the following equations, Equations 20 through 22 (McPherson, 1993):

\[ Q_l = (m_1 - m_2) * L_1 \] \hspace{1cm} ........... Eq (20)

\[ Q_s = m_2 * C_p * (T_2 - T_1) \] \hspace{1cm} ........... Eq (21)

\[ ........... Eq (22) \]
\[ Q = Q_s + Q_l \]

where,

- \( Q \) = Sensible heat loss from air
- \( m_1 \) = Water spray rate
- \( m_2 \) = Drainage water rate
- \( T_1 \) = Spray water temperature
- \( T_2 \) = Drainage water temperature
- \( C_p \) = Specific heat capacity of water at inlet temperature
- \( Q_l \) = Latent heat gain by the portion of water evaporated
- \( Q_s \) = Sensible heat gained by the water

6.4 Template 4 - Air – Thermal – Moisture – Contaminant Branch

Figure 16. Branch with Air – Thermal – Moisture – Contaminant connections.
Template 4 has four networks to be modeled simultaneously, as shown in Figure 16. This template consists of one additional axial node connection in the branch to that of the previous template (Air – Thermal – Moisture Branch Template), which is the contaminant connection. The side view of Figure 16 is shown in Figure 17. The data input form consists of five tabs:

1. Airway Geometry Input form
2. Rock Strata Input form
3. Heat Source / Sink Input form
4. Contaminant Window Input form
5. Parametric Table Input form

The first, second, third and fifth input forms are the same as the required forms for the previous templates (Tables 17, 22, 24 and 18 respectively). The fourth one, the contaminant input form, is shown in Table 29. This window consists of a drop down menu containing a list of contaminants from which the user can select one. Once the user selects the type of contaminant, the next step is the mode of input. It is either contaminant percentage concentration in air, or generation. If it is generation, it is either a fixed generation for some time, or a pulse release of a certain amount. After selecting the type of generation, the user will give the type of decay, either linear (with time) or logarithmic. Depending on the type of decay, the contaminant simulation is animated, which is a post-processor feature of the interface.

Figure 17. Side view of Figure 16.
Table 29. Input form for the contaminant window – Template 4.


Gases such as CO₂, Methane, etc. have the same input data form. But for Radon as a contaminant, it has a separate data form, which is shown in Table 30. The user enters the half-life and initial concentration of the radon. This will be further developed such that the software will assist the user more in terms of studying contaminant distributions as it has been suggested in the literature (Rawlins, 1997).
6.5 Template 5: Branch with duct: Air – Thermal – Moisture – Contaminant connections

Figure 18. A branch with one duct; thermal (including radiation) and moisture connections.

This template consists of a duct thus introducing internal (air – thermal – moisture) connections in the duct and connections (thermal - moisture) between the duct and walls
of the branch, in addition to the connections required for the previous template, Template 4. This template is depicted in Figure 18. The internal connections of the duct are shown as a blow-up in the same figure. The data input form consists of six tabs:

1. Airway Geometry Input form
2. Rock Strata Input form
3. Duct Geometry Input form
4. Heat Source / Sink Input form
5. Contaminant Input form
6. Parametric Table Input form

The first, second, fourth, fifth and sixth input forms are the same as the required forms in previous templates (Tables 17, 22, 24, 29 and 18, respectively). The third input form consists of the user input form for the geometry of the ducts present in the branch. This form is shown in Table 31.

There are two different drop down menus provided in this input form, through which the user specifies the type of duct. One is for the type of duct (force or exhaust), and the other for the material type. After selecting the type of duct, the user gives the thickness, diameter, resistance, airflow leakage, shock loss factor of the duct, and the evase diameter. From the user’s selection of material, the thermal conductivity and specific heat are automatically populated. If the user selects the “others” option, custom values for conductivity and specific heat must be provided by the user. Below this, there are two figures, with graphical depiction of various lengths and distances for which the user will input values:

1) Distance between the start node of duct and the inlet of drift.
2) Distance between the end node of duct and the face or outlet of the drift.
3) Length of the duct.
4) Distance between the center of the duct and center of the drift.
5) Distance between the center of the duct and floor of the drift.
There are tabs provided for the user to add more ducts, if there are multiple ducts in the branch. The user can click on the “Add” tab, to add a new duct and give the necessary input in the same way.

Table 31. Duct Geometry Input form – Template 5.

<table>
<thead>
<tr>
<th>Duct 1</th>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duct Type</td>
<td>Force Return</td>
</tr>
<tr>
<td>Material</td>
<td>Steel Plastic (PVC) Others</td>
</tr>
<tr>
<td>Conductivity</td>
<td>W/m K</td>
</tr>
<tr>
<td>Specific heat</td>
<td>J/kg K</td>
</tr>
<tr>
<td>Thickness</td>
<td>m</td>
</tr>
<tr>
<td>Diameter</td>
<td>m</td>
</tr>
<tr>
<td>Elbow Diameter</td>
<td>m</td>
</tr>
<tr>
<td>Leakage percentage</td>
<td>%</td>
</tr>
<tr>
<td>No. of segments</td>
<td></td>
</tr>
<tr>
<td>Duct resistance</td>
<td>N.s/m²</td>
</tr>
<tr>
<td>Relative roughness</td>
<td>(KOD)</td>
</tr>
<tr>
<td>Shock Loss Factor</td>
<td></td>
</tr>
</tbody>
</table>

6.6 Template 6: Branch with duct including fan and cooler

This template is for the scenario in which the duct consists of a fan and any other fan related cooling equipment. This template has four different networks, air, thermal, moisture and contaminant, in the airway and three networks air-thermal-moisture in the duct, to model. All the related connections, both axial and radial are modeled for airway and duct in this template. The inter-connections between airway and duct are also modeled, such as radiation and convection heat transfer between duct and airway, etc.
Figure 19 depicts Template 6, and the front view of the template is shown in Figure 20. All the possible connections in this template are not shown in the figures.

Because of the inclusion of fan and cooling sources in the duct, the “Duct Geometry Input form” and “heat Source / Sink Input form”, which are the third and fourth tabs for Template 5, are slightly altered in Template 6.

The Duct Geometry Input form consists of two sections, which is shown in Table 32:

- **a. Duct Geometry Input section:**
  The geometry parameters required to be input by the user, as shown in Table 28 are in this section.

- **b. Duct Fan Input Section:**
  For any fan installed in the duct, the user will input the fan performance, or choose from MULTIFLUX Fan library. For the input fan curve data, the fan curve will be displayed graphically, depicting airflow quantity vs. fan static pressure. The user will also input other necessary characteristic data for the fan(s), in the fields
provided below the fan curve. If there are no fans in the duct, the user will leave this section blank.

The fourth tab is the heat source / sink input form. This is similar to the heat source / sink input form of previous templates shown in Tables 24, 26 and 28. But it has an additional section for duct heat sources and sinks. This window is shown in Table 33. The heat source in the duct section consists of a duct fan. This section has necessary data fields to be input by the user. The cooling source in the duct section consists of a spot cooler heat exchanger. The user inputs necessary data for the spot cooler, in case of a duct having any spot coolers in it. Following are the calculations that are involved in solving the heat source and sink power in the ducts:

When the user inputs nominal power and the efficiency of the fan (power source), the sensible heat output is calculated and used as heat source in MULTIFLUX.

\[
Sensible \ heat = \ Nominal \ power \times (1 - efficiency) \quad \ldots \ldots \text{Eq (23)}
\]
Similar to the heat source, if there are any heat sinks such as spot coolers inside the duct, the refrigeration load is calculated as:

\[ \text{Refrigeration} = \text{Nominal power} \times (1 - \text{efficiency}) \quad \ldots \quad \text{Eq (24)} \]

Table 32. Input form for the duct window – Template 6
Table 33. Input form for the heat sink / source window – Template 6.

<table>
<thead>
<tr>
<th>Heat sink / source data - Branch</th>
<th>Heating source</th>
<th>Cooling source</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source 1</strong></td>
<td><strong>Source 2</strong></td>
<td><strong>Add</strong></td>
</tr>
<tr>
<td><strong>Sink 1</strong></td>
<td><strong>Sink 2</strong></td>
<td><strong>Add</strong></td>
</tr>
<tr>
<td>Point source</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Linear source</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td><strong>Source type</strong></td>
<td>Diesel</td>
<td>Electrical</td>
</tr>
<tr>
<td></td>
<td>Oxidation</td>
<td>Ground water</td>
</tr>
<tr>
<td></td>
<td>Explosive</td>
<td>Metabolism</td>
</tr>
<tr>
<td><strong>Distance from drift entrance</strong></td>
<td>m</td>
<td></td>
</tr>
<tr>
<td><strong>Nominal power</strong></td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>Diesel consumption rate</td>
<td>kg/s</td>
<td></td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Latent heat</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Moisture input</td>
<td>kg/s</td>
<td></td>
</tr>
<tr>
<td>CO₂ release</td>
<td>kg/s</td>
<td></td>
</tr>
<tr>
<td><strong>Sink type</strong></td>
<td>Spray water</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spot cooler</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spray chamber</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chiller</td>
<td></td>
</tr>
<tr>
<td><strong>Distance from drift entrance</strong></td>
<td>m</td>
<td></td>
</tr>
<tr>
<td><strong>Spray rate</strong></td>
<td>kg/s-m²</td>
<td></td>
</tr>
<tr>
<td><strong>Surface area</strong></td>
<td>m²</td>
<td></td>
</tr>
<tr>
<td><strong>Input temperature</strong></td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td><strong>Refrigeration</strong></td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td><strong>Moisture input</strong></td>
<td>kg/s</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heat sink / source data - Duct</th>
<th>Heating source</th>
<th>Cooling source</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duct 1</strong></td>
<td><strong>Duct 2</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Distance from duct entrance</strong></td>
<td>m</td>
<td></td>
</tr>
<tr>
<td><strong>Fan nominal power</strong></td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td><strong>Distance from duct entrance</strong></td>
<td>m</td>
<td></td>
</tr>
<tr>
<td><strong>Spot cooling power</strong></td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td><strong>Refrigeration</strong></td>
<td>kW</td>
<td></td>
</tr>
</tbody>
</table>
6.7 Template 7: Branch with duct and conveyor: Air – Thermal – Moisture – Contaminant connections

Figure 21. Branch with duct and conveyor belt; thermal (including radiation) and moisture connections.
Figure 22. Front view of Figure 21.

Figure 23. Side view of Figure 21.
This template consists of a conveyor belt in addition to the previous template (Template 6), and is shown in Figure 21. The front and side views of this figure are shown in Figures 22 and 23 respectively. One can observe various air, thermal and moisture connections between all the elements of the branch in radial and axial directions. The input data forms for this template are very similar to the previous template except that the data for conveyor belt is covered in addition to the previous one. The input data form for this template consists of 7 tabs:

1. Airway Geometry Input form
2. Rock Strata Input Form
3. Duct / Conveyor Geometry Input form
4. Duct Fan and Cooler Input form
5. Heat Source / Sink input form
6. Contaminant Input form
7. Parametric Table Input form

**Duct / Conveyor Geometry Input form:**

This input form is similar to Table 31 of Template 6. But in addition to the data forms present in Table 31, this incorporates the data for conveyor belt geometry. This is shown in Table 34. This has two sections. The first section is for the geometry parameters of the duct. Below this is the second section, which is the graphical depiction of various dimensions of duct and conveyor belt:

1) Distance between the start node of duct and the inlet of drift.
2) Distance between the end node of duct and the face or outlet of the drift.
3) Length of the duct.
4) Distance between the center of the duct and center of the drift.
5) Distance between the center of the duct and floor of the drift.
6) Distance between the start node of conveyor and the inlet of drift.
7) Distance between the end node of conveyor and the face or outlet of the drift.
8) Length of the conveyor belt.
9) Width of the conveyor belt.
Table 34. Duct / Conveyor Belt Geometry Input Form – Template 7.

Tabs 1, 2, 6 and 7 are similar to that of the previous template (Tables 17, 22, 29 and 18, respectively). The rest of the tabs are detailed below.

Duct Fan and Cooler Input form:

This input form is shown in Table 35. This form comprises of all the parameters required to model duct fans and coolers present in the ducts. This has two sections. First section shows the fan data, which is the same as the Section 2 of Table 32 – Template 6.

The second section has parameters to calculate the heat generated from duct fan and refrigeration power of the spot cooler present in duct.
Table 35. Duct Fan and Cooler Input form – Template 7.

**Duct Fan and Cooler Data**

<table>
<thead>
<tr>
<th>Quantity</th>
<th>m³/s</th>
<th>Fan Static Pressure Pa</th>
<th>Fan Total Pressure Pa</th>
<th>Fan Total Efficiency %</th>
<th>Absorbed Power kW</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.0</td>
<td>2100.0</td>
<td>2577.0</td>
<td>50.0</td>
<td>61.4</td>
<td></td>
</tr>
<tr>
<td>18.0</td>
<td>2500.0</td>
<td>2479.4</td>
<td>55.8</td>
<td>81.1</td>
<td></td>
</tr>
<tr>
<td>23.0</td>
<td>2500.0</td>
<td>2361.0</td>
<td>60.0</td>
<td>90.5</td>
<td></td>
</tr>
<tr>
<td>27.0</td>
<td>2000.0</td>
<td>2290.0</td>
<td>65.0</td>
<td>95.1</td>
<td></td>
</tr>
<tr>
<td>30.0</td>
<td>1800.0</td>
<td>2159.0</td>
<td>70.0</td>
<td>92.3</td>
<td></td>
</tr>
<tr>
<td>32.0</td>
<td>1600.0</td>
<td>2008.0</td>
<td>72.0</td>
<td>89.2</td>
<td></td>
</tr>
<tr>
<td>34.0</td>
<td>1500.0</td>
<td>1861.0</td>
<td>70.0</td>
<td>90.6</td>
<td></td>
</tr>
<tr>
<td>36.0</td>
<td>1400.0</td>
<td>1716.0</td>
<td>65.0</td>
<td>84.1</td>
<td></td>
</tr>
<tr>
<td>38.0</td>
<td>1300.0</td>
<td>1575.0</td>
<td>60.0</td>
<td>83.0</td>
<td></td>
</tr>
<tr>
<td>40.0</td>
<td>1200.0</td>
<td>1437.0</td>
<td>45.0</td>
<td>56.4</td>
<td></td>
</tr>
</tbody>
</table>

**Curve Estimation**

**Fan Speed**

- RPM
- Air Density during Calibration: kg/m³
- Reverse Pressure: Pa
- Reverse Quantity: m³/s

**Heat Source**

- Distance from duct entrance: m
- Fan nominal power: kW
- Efficiency: %
- Sensible heat: kW

**Cooling Source**

- Distance from duct entrance: m
- Fan nominal power: kW
- Efficiency: %
- Refrigeration: kW

**Heat Source / Sink Input form:**

This input form is similar to Table 24, and is shown in Table 36. This input form has two sections. The first section is identical to Table 24 – Template 3. The second section comprises of heat data related to conveyor. In this section, the user could give either production rate or the advance rate of the working. Based on the selection, following formulae are used to calculate the speed of the conveyor belt, which are given in Equations 25 through 27.
\[ M = \frac{AR \cdot A \cdot \rho}{24 \cdot 3600} \quad \text{........ Eq (25)} \]

\[ M = \frac{PR \cdot 907.19}{24 \cdot 3600} \quad \text{........ Eq (26)} \]

where,

\( M \) = mass flow rate (kg/s)

\( AR \) = Advance rate of the working (m/day)

\( A \) = Cross-sectional area of the drift (m\(^2\))

\( \rho \) = Bulk density of the rock (kg/m\(^3\))

\( PR \) = Production rate of the drivage (Ton/day)

1 Ton = 907.19 kg

\[ v = \frac{M}{A_c \cdot \rho} \quad \text{........ Eq (27)} \]

where,

\( v \) = Velocity of the belt drive

\( A_c \) = Cross-sectional area of the conveyor belt

\( = \frac{W^2}{18} \)

\( W \) = width of the conveyor belt

These two parameters are used to calculate the radiative and convective heat transfer between the conveyor belt and other elements in the branch, through internally defined functions of MULTIFLUX code.
Table 36. Heat Source / Sink Input form – Template 7.

<table>
<thead>
<tr>
<th>Heat sink / source data - Branch</th>
<th>Heating source</th>
<th>Cooling source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source 1</td>
<td>Source 2</td>
<td>Add</td>
</tr>
<tr>
<td>Point source</td>
<td>Linear source</td>
<td></td>
</tr>
<tr>
<td>Source type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diesel</td>
<td>Electrical</td>
<td>Oxidation</td>
</tr>
<tr>
<td>Distance from drift entrance</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Nominal power</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>Diesel consumption rate</td>
<td>kg/s</td>
<td></td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Latent heat</td>
<td>kW</td>
<td></td>
</tr>
<tr>
<td>Moisture input</td>
<td>kg/s</td>
<td></td>
</tr>
<tr>
<td>CO₂ release</td>
<td>kg/s</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conveyer Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working advance rate</td>
</tr>
<tr>
<td>Production rate</td>
</tr>
<tr>
<td>Drive position from the entrance</td>
</tr>
<tr>
<td>Wetness factor</td>
</tr>
<tr>
<td>Mineral age</td>
</tr>
<tr>
<td>Motor power</td>
</tr>
<tr>
<td>Efficiency</td>
</tr>
<tr>
<td>Heat generated</td>
</tr>
</tbody>
</table>

Submit | Reset | Cancel
6.8 Template 8: Development-end template for hard rock and soft rock mines

There are two types of development scenarios, i.e., of metal mines and coal mines, which are shown in the Figures 24 and 25 respectively. The mode of advancement makes the difference. In the development end template for coal mines – Template 8 (Figure 24), there are six different elements in between which there are multiple air, heat, moisture, and contaminant interactions. They are:

1. Airway walls
2. Force Duct – Fan and Cooler
3. Return / Exhaust Duct – Fan and Cooler
4. Road Header
5. Shuttle Vehicle (Conveyor Loader)
6. Conveyor Belt

The input data for airway walls, force duct, exhaust duct, duct fans and coolers, and conveyor belt are taken care of by Tables 17, 22, 33, 34 and 35. The input data required to model the road header and shuttle vehicle are given in Road header and Shuttle Input form, which is given in Table 37.

Table 37. Road Header / Shuttle input form

<table>
<thead>
<tr>
<th></th>
<th>Conveyerloader</th>
<th>Road header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle</td>
<td>Add</td>
<td>Add</td>
</tr>
<tr>
<td>Nominal power</td>
<td>kW</td>
<td>kW</td>
</tr>
<tr>
<td>Efficiency</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>Sensible heat</td>
<td>kW</td>
<td>kW</td>
</tr>
</tbody>
</table>

Table 37 consists of two sections, one for shuttle vehicle / conveyor loader and one for road header. Each section consists of a drop down menu to select the type of equipment from a library of vehicles. From the selection, the nominal power and the efficiency are populated automatically from the library. Or the user could enter the nominal power and efficiency nominally.
6.9 Special templates

Apart from the above discussed templates, there are a few other templates that are already developed. The input data tables for these templates are yet to be developed. One such case is shown in Figure 26.

![Figure 26. Specia case III: Fan branch, airflow line.](image)

6.10 The three types of user interactions:

The user interface, in the form of the template based data entry, provides three different types of user interactions for the user with the software. Following are the tasks integrally involved in describing the ventilation model:

1. **Definition of a model/system with given processes:**
   Mining methods, the regulations, safety considerations, the mine ore, strata conditions, etc. come into this category.

2. **Definition of boundary conditions of the system and processes involved within the system:**
VRT, the presence of heat sources, moisture sources and heat sinks, nature of the freshly cut ore, ground water problems, type of mining equipment, ground control, type of development, thermophysical properties of rock, etc.

3. **Defining the values of boundary conditions of the system:**
   
   RPM of fan, location of the heat sources and sinks, resistance of the airways, water inflow rate, power rating of the equipment, mine crew size, etc.

The three model interactions then systematically describe the model through the sequential data input by the user through input forms. This is explained in detail by analyzing the method of data input designed for Template 3, the branch with Air-Thermal-Moisture connections:

a. The user defines the system to be homogeneous or heterogeneous in the thermophysical data section of Table 22.

b. The processes such as the convection, conduction heat transfers between the heat sources and sinks, and mass transport are defined in Table 24.

c. The user defines the boundary conditions by giving virgin rock temperature, choosing the types of heat sources and sinks, their locations in the drift, etc.

d. The boundary condition values are input by the user through inputting data such as the age of drift at entry and exit, the power ratings and efficiencies of the heat sources and sinks, and all the relevant information on the heat inputs and moisture inputs. This is done in Tables 26 and 28.
7. **Developing HTML forms for interfacing with Google Sketchup software**

The development of the MULTIFLUX Graphical User Interface input forms involves a concerted effort by three team members. The team consists of the software designer who extends the functionality of the MULTIFLUX engine by defining the templates and all the input parameters that must be gathered from the user. A form programmer fills the role of coding input forms to guide the user and gather characteristics for graphically selected airways. A ruby programmer provides the Google SketchUp interface programming, defining the look and feel of the user interface. Google SketchUp has many native features that are available for navigation around the 3-D airflow network, and graphical airway creation or modification.

The user’s input forms have been designed to help gather the required data, based on the assigned template. Multiple levels of data validation ensure that the MULTIFLUX preprocessor will generate error-free modeling scenarios. Much of this validation requires input of integers, radio button and drop-down menu selections, or input of real numbers within a given range. Graphical prompts that assist the user in correcting the input data appear adjacent to an input element when the validation of that element fails. Additionally, alert windows appear when invalid scenarios are selected.

The input forms are coded in HTML and JavaScript. Much of the programming required to provide functionality for passing data to the MULTIFLUX processor involves a combination of Java Scripting and manipulation of HTML document object model (DOM) element characteristics. JavaScript is required to enact many attributes of the DOM document methods. The disparity of JavaScript is that different browsers gain access to and modify elements in the DOM quite differently. This current issue of cross-browser compliance may well be solved in future versions of HTML.

The data gathering and referencing functionality of the input forms relies heavily upon a simple database type, known as XML. The beauty of XML is the cross-platform compatibility and open structure that allows interaction with resources outside the
MULTIFLUX application. By using native XML data sets to process model elements means that the modeling data, both pre and post-processing can be shared with other commercially available ventilation simulation software applications.

In order to bridge the current technology gap in browser technologies, the Dojo JavaScript toolkit has been employed. The Dojo toolkit provides many data-driven graphical solutions that are directly applicable to the input forms specified by the software designer, in addition to the inherent cross-browser functionality that will maintain the application’s forward compatibility. The Spry Ajax framework is also employed in menu driven data selection and entry validation. Both the Dojo and Spry JavaScript libraries are abstracts on pure JavaScript. This implication means that three dialects of JavaScript are utilized in many input forms, further adding to the complexity of the programming task.

The input forms, and all their dependent resources, are included as webdialog objects for the SketchUp interface. These forms are then interpreted in SketchUp using a rudimentary, freely distributed MS Internet Explorer browser for display of the webdialog objects. The application flow for the progression of input forms for each airway relies upon the underlying template definition. Input form display will follow dependencies mandated by airway template menu selection.

Following is the workflow for generating the MULTIFLUX Graphical User Interface input forms. The same is presented in the form of a flow chart in Figure 27:

1. Form content has been authored and conceptual layout of graphical elements approved. Mock-up of input form in MS Word.
2. Forms authored in HTML format and checked against MS Word copy.
3. Form validation JavaScript added to form elements to ensure proper data gathering.
4. Form submission JavaScript added and subsequent testing and debugging on the SketchUp platform.
Figure 27. Flow chart showing the process of integration with SketchUp software.
5. Graphical data handling capability added programmatically to process user selected airway branch(s), then subsequent testing and debugging on the SketchUp platform.

6. Forms given to SketchUp programmer for evaluation.

7. Input forms returned to implement programmatic changes for additional SketchUp platform functionality.

8. Subsequent revision cycles yield final form dialogs for software inclusion.

Figure 28. The HTML form of the input data table for Branch form – Table 17 from Template 1.
HTML forms that are duplicated from the MS word tables of a few templates are shown in Figures 28, 29 and 30. The forms for the data input forms for the rest of templates will be developed soon. The HTML form demonstrates the inputting method by the user. In Figure 28, the branch is connected to the surface and shape is given as “horse-shoe”. Subsequent data fields are filled in for the shape and resistance selections.

![HTML form](image)

Figure 29. The HTML form of the input data table for Fan form – Table 19 from Template 2.

Similarly, in Figure 29, the fan input window in MS word (Table 20 of Template 3) is duplicated as HTML form. The „Fan” option is selected to give the airflow attributes in the airway. The arrangement of the fans is chosen to be parallel. The number of fans is
given as 2, and depending on the number of fans given by the user, the appropriate number of tabs will be displayed dynamically.

From the MULTIFLUX fan curve library that is being populated currently, one of the fan curves is chosen, and the respective fan curve data is loaded automatically, and the fan curve is under display, adjacent to the data.

Figure 30 shows the HTML form developed for the Rock Strata Input form which is shown in Table 21, for Template 3. Heterogeneous rock type is selected, and Shale Rock is chosen as the rock type for the roof and both the side walls. One can see that the properties of Shale rock are automatically populated upon the selection. The Age data and the Virgin rock temperature data are also shown in the respective fields.
Figure 30. The HTML form of the input data table for Rock strata form – Table 21 from Template 3.
8. **Time dependency in data input**

All the existing mine ventilation software allow the user to simulate mine environments for a given set of input parameters and conditions at given time frames. Although there are a few instances such as with the software developed by Dziurzynski et al (1997) and Ventsim (Ventsim, 2007) simulating the contaminant flow through the mine network, most of the available modeling software do not give the results for dynamic conditions, i.e., time-variable input parameters such as conductive heat flux due to aging of the mine workings. Time is an important factor in two scenarios:

1. Heat – strata “age” variation, where the information on the airway’s strata history is important.
2. Concentration distributions, where mixing and convection phenomena occur together, and volume buffers concentration variation. There is a time delay in the reporting, especially in scenarios like mine fires where the airflow distribution is rapidly affected by the varying intensity of the fires that interact with the mine ventilation system.

MULTIFLUX software has the capability to accept time dependent input data conditions, a data vector, and give results. The user interface for MULTIFLUX will be able to provide the user with this choice of entering time dependent input variables in the form of a span and discretization interval vector, such that one can model mine ventilation networks over a desired time span.

Usually, the data provided by the user in a typical mine ventilation software is for a time instance. For example, a heat source input is given in terms of its power, and there is no further data provided to the software in terms of the span of the heat source equipment usage, the time span of its maximum utilization and any break time, etc. So, if the user has the data, such as the operating time of the heat source such as a diesel machine, and if the user is able to input this time dependent data to the simulation software, then with the help of simulation software, he will be able to assess the changing dependencies of
ventilation and cooling requirements for that specific airway, and in turn the whole mine.

Difference between time independent and time dependent input data:

1. Time-independent data of a machine heat source (diesel equipment):
   Sensible heat – 300kW; Latent heat – 100kW

2. Time-dependent input on the power utilization of another diesel machine, given in Table 38. Nominal power (100% utilization) being 500kW.

Table 38. The utilization trend (as per shift) of a diesel machine in a mine during a normal day of operation.

<table>
<thead>
<tr>
<th>Time (hours)</th>
<th>Input data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0600 – 1400</td>
<td>80% utilization</td>
</tr>
<tr>
<td>1400 – 1500</td>
<td>0% utilization</td>
</tr>
<tr>
<td>1500 - 2100</td>
<td>80% utilization</td>
</tr>
<tr>
<td>2100 - 2200</td>
<td>0% utilization</td>
</tr>
<tr>
<td>2200-0600</td>
<td>80% utilization</td>
</tr>
</tbody>
</table>

The time discretization could be seconds, minutes, hours, shifts, days, weeks, months and years. The user will be provided the option to upload a simple comma delimited file for the vector data from the user input window for the respective template. The current version of the user input form for time dependent variables is given in Table 39.

Table 39. Input form for time dependent input variables.
Also, in the actual mines, several branches are shut down and several airways are newly added to the network, and this is a regular process. The frequently changing ventilation network for the mine needs to be updated regularly on the software, for modeling purposes. It could only get better if the user is provided with a tool to study the effects of newly added airways and the deleted old airways, with respect to time.

For example, if the user wants to see the changes in the ventilation network over a time period of a few months or a year, there is no such tool in any of the currently existing simulation software. This could be achieved if the user can add the new branches to the existing mine ventilation network in the form of a new layer.
9. **Scenario based simulations**

This feature is intended to improve the user’s interaction with the software. The concept is to have some tools in the software which help the user study the behavior of a mine ventilation network, effects on the ventilation in special scenarios, such as:

1. Equipment shut down
2. Evacuation
3. Blasting
4. Mine fires
5. Fan break down
6. Break time in between shifts

The user will be able to select a group of branches in the network, and apply one of those scenarios to the selected group, simulate and analyze the scenario. For example, if the user wants to see the behavior of the ventilation model from the time blasting starts through the time when the fumes are cleared from the mine, during which the following conditions are applied:

a) A few fans are shut down.

b) Few others are run in reverse direction to drive the fumes out.

c) No metabolic heat load due to the absence of mine workers in the mine.

d) The mine equipment is shut down, etc.

The user would select all the branches that are related to the blasting scenario, including the exhaust shaft, and click on the icon for “blasting”. Then the software would simulate the whole scenario for the user which serves the educational (training) purposes, helps in mine maintenance and also future planning purposes.
10. **Illustration of Pre-processor and Post-processor**

This section illustrates the pre-processor and post-processor of the MULTIFLUX software’s interface that has been developed so far. The ventilation network of Barrick Gold Strike’s Meikle mine is successfully loaded in the Google SketchUp software’s 3-dimensional CAD platform via MULTIFLUX software. The schematic of ventilation network is shown in Figure 31. Figure 32 shows close-up of a few branches of the network from Figure 31. It is also shown that a few branches are selected, which are highlighted in blue color.

![Figure 31. Schematic of Meikle underground mine’s ventilation network in SketchUp 3-dimensional CAD platform.](image)

The rendered network is shown in Figure 33. This figure demonstrates the working model of the interface where, an airway is selected and a template is assigned to it, and the data is fed into the relevant data input form. It also features a query tool that is below the data input form in the figure.
The close up of a few branches in rendered form are shown in Figure 34. Figure 35 shows the temperature distribution of the mine, with increasing temperature from blue colored to red colored branches. This is the post-processor part of the GUI.

Figure 32. Close-up of a few branches of the network from Figure 31. A few branches are selected which are highlighted in blue color.

The user will be able to see the results for each branch individually, that are displayed adjacent to the branches in the 3-dimensional network. The user can choose the parameters he wishes to see with the branches on the display. A detailed view of the results for an airway can be seen up on clicking on that specific airway. Also, the results of all the branches along with the input data will be displayed in the form of tables.

Various parameters can be plotted as graphs by the user, including velocity distributions, temperature distributions, volumetric flowrate, and others.

The user will be able to animate the results such as contaminant distribution, and the airflows in the mine network over the time.
Figure 33. The rendered network shown in Figure 31. Query tool and branch input form are also shown.

The user will also be provided tools to do several sensitivity analyses. He can decide on the optimum location of heat sources, cooling sources and fans, or vary the RPM of a fan in an airway, or change the size or configuration of fans, etc. For these studies, the code is not required to run on the entire system, but with the relevant parts of the model.

The NTCF part of the processor is made independent with respect to the CFD part of the processor, and vice versa. Thus, for any change in the model configuration of CFD such as changes in fan configuration, the NTCF part (Age of the rock, dimensions of the airway, etc.) need not be executed over and over. This saves a lot of execution time during sensitivity analyses.
Figure 34. Close up of a few branches in rendered form.
Figure 35. The temperature distribution in the network. The post-processor part of the GUI.
11. Conclusions

A graphical user interface for MULTIFLUX software is partially developed and will take its final form in the course of the completion of the research project. From the work explained above in previous sections, it can be concluded that:

1. A background study on the previously and currently existing mine ventilation modeling software has supplied invaluable input to develop a graphical user interface. In particular, a few mine ventilation modeling software that are currently available on the market are studied in detail. Lists of the input parameters for those software are being used as the basis for the new model.

   MULTIFLUX software’s basic functionalities were studied in order to determine the additional input parameters for the more complicated transport model. Based on the studies, the new input parameters required for the user interface have been introduced accordingly.

2. A manual mode of data entry user interface is developed in MATLAB as a first step towards designing the graphical user interface. The user can establish user-defined connections in between chosen elements in an airway or a small sized ventilation network, and can study the effects of various transport connections in between them.

   This will be incorporated as a manual tool in the main GUI of MULTIFLUX in the future, which will serve educational purposes.

3. Google SketchUp software is chosen as the 3-dimensional CAD platform for MULTIFLUX GUI. Then, the user interface is conceptualized, and the tasks of pre-processor and post-processor are laid out.

4. A template based concept is adopted for an easier input of the data by user. Templates are developed for several scenarios that prevail in the underground mine workings. The user can click on a branch or select a group of branches and
append a template to them. The data input forms are also developed which are unique for each template. These are window based, and they appear in various tabs. All the possible templates will be developed and the corresponding data input forms will be designed.

Three different types of interactions the user will have with the software while inputting the data are explained. They are:

a) Definition of model configuration
b) Definition of boundary conditions
c) Definition of boundary values

5. The data input forms which are designed in MS word are converted into HTML forms by which the forms are interfaced with Google SketchUp software. The interfacing involves JAVA coding to incorporate various functionalities in the input forms.

6. The pre-processor and post-processor of the GUI for MULTIFLUX are illustrated by modeling Barrick Gold Strike’s Meikle mine as an example.

7. Future work is proposed regarding time dependency in the input data in modeling. Scenario based simulations are conceptualized, which aid the user to study the mine ventilation network’s response behaviors in various scenarios in an efficient way.
12. **Future work**

Future works and recommendations are listed as following:

1. All the possible templates will be populated.
2. The input forms for the new templates have to be developed and duplicated into HTML.
3. The post-processor tasks are to be developed in terms of the display of the results in the form of tables, graphics and animation.
4. The time dependent data input should be fully developed so that a user can analyze the behavior of a mine in course of time.
5. Scenario based simulations are to be developed for typical scenarios to serve training purposes and study purposes.
13. References


14. **Appendices**

Appendix 1 includes the MATLAB code for the two programs related to the manual interface of MULTIFLUX, which is explained in Section 4. The first program is a macro script and to create the files to be input to the DISAC module. The second program contains the code in which all the tables are created.

14.1 Appendix 1:

**MACRO – MATLAB CODE**

```matlab
% To avoid reset of global variables, this is a macro-
script
clear all
MatInit

% initial values
global T_tm P_tm TG_tm PG_tm qc_tm qab_tm v_tm Mtmi_tm time
qh_tm qm_tm;
global wf_tm Pb_tm Pbg_tm qabg_tm vg_tm qa_tm;

% empty matrices to which tables are written
global FOCa FOBa DUCa VFFa FOCag FOBag DUCag VFFag FRCh
FOCh COCh CODh RADh;
global DUCh FRChg FOChg COChg CODhg RADhg DUCHg FRChm FOCm
COChm COChmg DUCm;
global FRChmg FOCmg COCmg CODmg DUCmg;

% creation of disac_d folder
if ~exist('disac_d')
    mkdir disac_d
end
```
%creation of result_d folder
if ~exist('result_d')
    mkdir result_d
end

%required tables for MF
%**** initialize MULTIFLUX code table variables to empty matrices ********
FOCa=[];    % FOCa table: forced convection
FOBa=[];    % FOBa table: booster fan, forced convection
DUCa=[];    % DUCa table: air flow model
VFFa=[];    % vffa table: air flow model
%definition of generators
FOCag=[];   % FOCag table: forced convection
FOBag=[];   % FOBag table: booster fan, forced convection
DUCag=[];   % DUCag table: air flow model
VFFag=[];   % vffag table: air flow model
%initialize MULTIFLUX code table variables with empty matrices for heat
%flow model
FRCh=[];    % FRCh table: free convection
FOCh=[];    % FOCh table: forced convection
COCh=[];    % COCh table: controlled convection, capacitive
CODh=[];    % CODh table: conduction
RADh=[];    % RADh table: radiation
DUCh=[];    % direct user connection
% Definition of generators
FRChg=[];   % FRChg table: free convection
FOChg=[]; % FOChg table: forced convection
COChg=[]; % COChg table: controlled convection, capacitive
CODhg=[]; % CODhg table: conduction
RADhg=[]; % RADhg table: radiation
DUCHg=[]; % direct user connection

%initialize MULTIFLUX code table variables with empty matrices for moisture flow model
FRCm=[]; % FRCm table: free convection
FOCm=[]; % FOCm table: forced convection
COCm=[]; % COCm table: controlled convection, capacitive
CODm=[]; % CODm table: conduction
DUCm=[]; % direct user connection

% Definition of generators
FRCmg=[]; % FRCmg table: free convection
FOCmg=[]; % FOCmg table: forced convection
COCmg=[]; % COCmg table: controlled convection, capacitive
CODmg=[]; % CODmg table: conduction
DUCmg=[]; % direct user connection

ManualEntry_Electrical

MANUAL USER INTERFACE – MATLAB CODE

function varargout = ManualEntry_MULTIFLUX(varargin)

MANUALENTRY_MULTIFLUX('CALLBACK',hObject,eventData,handles,...) calls the local function named CALLBACK in
MANUALENTRY_MULTIFLUX.M with the given input arguments. 
MANUALENTRY_MULTIFLUX('Property','Value',...) creates a new 
MANUALENTRY_MULTIFLUX or raises the existing singleton*. 
Starting from the left, property value pairs are applied to 
the GUI before ManualEntry_MULTIFLUX_OpeningFcn gets 
called. An 
unrecognized property name or invalid value makes property 
application stop. All inputs are passed to 
ManualEntry_MULTIFLUX_OpeningFcn via varargin. 
% 
gui_Singleton = 1; 
gui_State = struct('gui_Name', mfilename, ... 
     'gui_Singleton', gui_Singleton, ... 
     'gui_OpeningFcn', 
@ManualEntry_MULTIFLUX_OpeningFcn, ... 
     'gui_OutputFcn', 
@ManualEntry_MULTIFLUX_OutputFcn, ... 
     'gui_LayoutFcn', [], ... 
     'gui_Callback', []); 
if nargin && ischar(varargin{1}) 
    gui_State.gui_Callback = str2func(varargin{1}); 
end 

if nargout 
    [varargout{1:nargout}] = gui_mainfcn(gui_State, 
varargin{:}); 
else 
    gui_mainfcn(gui_State, varargin{:}); 
end 
% End initialization code - DO NOT EDIT
% --- Executes just before ManualEntry_MULTIFLUX is made visible.
function ManualEntry_MULTIFLUX_OpeningFcn(hObject, eventdata, handles, varargin)

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

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = ManualEntry_MULTIFLUX_OutputFcn(hObject, eventdata, handles)

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

% --- Executes on selection change in popupmenu_connectionheat.
function popupmenu_connectionheat_Callback(hObject, eventdata, handles)

contents = get(hObject,'String');
data2 = contents{get(hObject,'Value')};
disp(data2);
get(handles.popupmenu_network,'Value');
a=get(handles.popupmenu_network,'Value');
get(handles.popupmenu_inode,'Value');
b=get(handles.popupmenu_inode,'Value');
get(handles.popupmenu_connectionheat,'Value');
c=get(handles.popupmenu_connectionheat,'Value');

%H Heat
\[
c_1 = (a==2) \land (c==7); \quad \text{%User-defined}
\]
\[
c_2 = (a==2) \land (c==1); \quad \text{%Conduction - parallel plates}
\]
\[
c_3 = (a==2) \land (c==2); \quad \text{%Conduction - cylindrical surfaces}
\]
\[
c_4 = (a==2) \land (c==3); \quad \text{%Convection - free}
\]
\[
c_5 = (a==2) \land (c==4); \quad \text{%Convection - forced}
\]
\[
c_6 = (a==2) \land (c==5); \quad \text{%Convection - controlled}
\]
\[
c_7 = (a==2) \land (c==6); \quad \text{%Radiation}
\]

%Moisture
\[
c_8 = (a==3) \land (c==7); \quad \text{%User-defined}
\]
\[
c_9 = (a==3) \land (c==1); \quad \text{%Conduction - parallel plates}
\]
\[
c_{10} = (a==3) \land (c==2); \quad \text{%Conduction - cylindrical surfaces}
\]
\[
c_{11} = (a==3) \land (c==3); \quad \text{%Convection - free}
\]
\[
c_{12} = (a==3) \land (c==4); \quad \text{%Convection - forced}
\]
\[
c_{13} = (a==3) \land (c==5); \quad \text{%Convection - controlled}
\]

%HEAT

if c1
    set(handles.uitable_inodal,'Visible','on')
set(handles.uitable_conduction,'Visible','off')
set(handles.uitable_radiation,'Visible','off')
set(handles.uitable_controlledconvection,'Visible','off')
set(handles.uitable_forcedconvection,'Visible','off')
set(handles.uitable_userdefined,'Visible','on')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_mforcedconvection,'Visible','off')
set(handles.uitable_muserdefined,'Visible','off')
set(handles.uitable_aforcedconvection,'Visible','off')
set(handles.uitable_velocityflowfield,'Visible','off')
set(handles.uitable_jnodal,'Visible','on')
set(handles.uitable_errlim,'Visible','on')
set(handles.uitable_controlparameter,'Visible','on')
set(handles.text5,'Visible','on')
set(handles.text6,'Visible','on')
set(handles.text7,'Visible','on')
set(handles.text8,'Visible','on')
set(handles.text9,'Visible','on')
set(handles.errormsg,'Visible','off')

elseif c2
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','on')
    set(handles.uitable_radiation,'Visible','off')

    set(handles.uitable_controlledconvection,'Visible','off')
set(handles.uitable_forcedconvection,'Visible','off')
set(handles.uitable_userdefined,'Visible','off')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_mforcedconvection,'Visible','off')
set(handles.uitable_muserdefined,'Visible','off')
set(handles.uitable_aforcedconvection,'Visible','off')
set(handles.uitable_velocityflowfield,'Visible','off')
set(handles.uitable_jnodal,'Visible','on')
set(handles.uitable_errlim,'Visible','on')
set(handles.uitable_controlparameter,'Visible','on')
set(handles.text5,'Visible','on')
set(handles.text6,'Visible','on')
set(handles.text7,'Visible','on')
set(handles.text8,'Visible','on')
set(handles.text9,'Visible','on')
set(handles.errormsg,'Visible','off')

%c3 (conduction - cylindrical surfaces) to be added

%c4 (convection - free) to be added

elseif c5
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','off')
    set(handles.uitable_radiation,'Visible','off')

    set(handles.uitable_controlledconvection,'Visible','off')
    set(handles.uitable_forcedconvection,'Visible','on')
set(handles.uitable_userdefined,'Visible','off')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_mforcedconvection,'Visible','off')
set(handles.uitable_aforcedconvection,'Visible','off')
set(handles.uitable_velocityflowfield,'Visible','off')
set(handles.uitable_jnodal,'Visible','on')
set(handles.uitable_errlim,'Visible','on')
set(handles.uitable_controlparameter,'Visible','on')
set(handles.text5,'Visible','on')
set(handles.text6,'Visible','on')
set(handles.text7,'Visible','on')
set(handles.text8,'Visible','on')
set(handles.text9,'Visible','on')
set(handles.errormsg,'Visible','off')

elseif c6
set(handles.uitable_inodal,'Visible','on')
set(handles.uitable_conduction,'Visible','off')
set(handles.uitable_radiation,'Visible','off')
set(handles.uitable_controlledconvection,'Visible','on')
set(handles.uitable_forcedconvection,'Visible','off')
set(handles.uitable_userdefined,'Visible','off')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_mforcedconvection,'Visible','off')
set (handles.uitable_muserdefined, 'Visible', 'off')
set (handles.uitable_aforcedconvection, 'Visible', 'off')
set (handles.uitable_velocityflowfield, 'Visible', 'off')
set (handles.uitable_jnodal, 'Visible', 'on')
set (handles.uitable_errlim, 'Visible', 'on')
set (handles.uitable_controlparameter, 'Visible', 'on')
set (handles.text5, 'Visible', 'on')
set (handles.text6, 'Visible', 'on')
set (handles.text7, 'Visible', 'on')
set (handles.text8, 'Visible', 'on')
set (handles.text9, 'Visible', 'on')
set (handles.errormsg, 'Visible', 'off')

elseif c7
    set (handles.uitable_inodal, 'Visible', 'on')
    set (handles.uitable_conduction, 'Visible', 'off')
    set (handles.uitable_radiation, 'Visible', 'on')

    set (handles.uitable_controlledconvection, 'Visible', 'off')
    set (handles.uitable_forcedconvection, 'Visible', 'off')
    set (handles.uitable_userdefined, 'Visible', 'off')
    set (handles.uitable_mconduction, 'Visible', 'off')
    set (handles.uitable_mfreeconvection, 'Visible', 'off')

    set (handles.uitable_mcontrolledconvection, 'Visible', 'off')
    set (handles.uitable_mforcedconvection, 'Visible', 'off')
    set (handles.uitable_muserdefined, 'Visible', 'off')
    set (handles.uitable_aforcedconvection, 'Visible', 'off')
    set (handles.uitable_velocityflowfield, 'Visible', 'off')
    set (handles.uitable_jnodal, 'Visible', 'on')
    set (handles.uitable_errlim, 'Visible', 'on')
set(handles.uitable_controlparameter,'Visible','on')
set(handles.text5,'Visible','on')
set(handles.text6,'Visible','on')
set(handles.text7,'Visible','on')
set(handles.text8,'Visible','on')
set(handles.text9,'Visible','on')
set(handles.errormsg,'Visible','off')

%MOISTURE

elseif c8
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','off')
    set(handles.uitable_radiation,'Visible','off')

    set(handles.uitable_controlledconvection,'Visible','off')
    set(handles.uitable_forcedconvection,'Visible','off')
    set(handles.uitable_userdefined,'Visible','off')
    set(handles.uitable_mconduction,'Visible','off')
    set(handles.uitable_mfreeconvection,'Visible','off')

    set(handles.uitable_mcontrolledconvection,'Visible','off')
    set(handles.uitable_mforcedconvection,'Visible','off')
    set(handles.uitable_muserdefined,'Visible','on')
    set(handles.uitable_aforcedconvection,'Visible','off')
    set(handles.uitable_velocityflowfield,'Visible','off')
    set(handles.uitable_jnodal,'Visible','on')
    set(handles.uitable_errlim,'Visible','on')
    set(handles.uitable_controlparameter,'Visible','on')
    set(handles.text5,'Visible','on')
    set(handles.text6,'Visible','on')
set(handles.text7,'Visible','on')
set(handles.text8,'Visible','on')
set(handles.text9,'Visible','on')
set(handles.errormsg,'Visible','off')

elseif c9
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','off')
    set(handles.uitable_radiation,'Visible','off')

    set(handles.uitable_controlledconvection,'Visible','off')
    set(handles.uitable_forcedconvection,'Visible','off')
    set(handles.uitable_userdefined,'Visible','off')
    set(handles.uitable_mconduction,'Visible','on')
    set(handles.uitable_mfreeconvection,'Visible','off')

    set(handles.uitable_mcontrolledconvection,'Visible','off')
    set(handles.uitable_mforcedconvection,'Visible','off')
    set(handles.uitable_muserdefined,'Visible','off')
    set(handles.uitable_aforcedconvection,'Visible','off')
    set(handles.uitable_velocityflowfield,'Visible','off')
    set(handles.uitable_jnodal,'Visible','on')
    set(handles.uitable_errlim,'Visible','on')
    set(handles.uitable_controlparameter,'Visible','on')
    set(handles.text5,'Visible','on')
    set(handles.text6,'Visible','on')
    set(handles.text7,'Visible','on')
    set(handles.text8,'Visible','on')
    set(handles.text9,'Visible','on')
    set(handles.errormsg,'Visible','off')
elseif c11
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','off')
    set(handles.uitable_radiation,'Visible','off')
    set(handles.uitable_controlledconvection,'Visible','off')
    set(handles.uitable_forcedconvection,'Visible','off')
    set(handles.uitable_userdefined,'Visible','off')
    set(handles.uitable_mconduction,'Visible','off')
    set(handles.uitable_mfreeconvection,'Visible','on')
    set(handles.uitable_mcontrolledconvection,'Visible','off')
    set(handles.uitable_mforcedconvection,'Visible','off')
    set(handles.uitable_muserdefined,'Visible','off')
    set(handles.uitable_velocityflowfield,'Visible','off')
    set(handles.uitable_jnodal,'Visible','on')
    set(handles.uitable_errlim,'Visible','on')
    set(handles.uitable_controlparameter,'Visible','on')
    set(handles.text5,'Visible','on')
    set(handles.text6,'Visible','on')
    set(handles.text7,'Visible','on')
    set(handles.text8,'Visible','on')
    set(handles.text9,'Visible','on')
    set(handles.errormsg,'Visible','off')

elseif c12
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','off')
set(handles.uitable_radiation,'Visible','off')
set(handles.uitable_controlledconvection,'Visible','off')
set(handles.uitable_forcedconvection,'Visible','off')
set(handles.uitable_userdefined,'Visible','off')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_mforcedconvection,'Visible','on')
set(handles.uitable_muserdefined,'Visible','off')
set(handles.uitable_aforcedconvection,'Visible','off')
set(handles.uitable_velocityflowfield,'Visible','off')
set(handles.uitable_jnodal,'Visible','on')
set(handles.uitable_errlim,'Visible','on')
set(handles.uitable_controlparameter,'Visible','on')
set(handles.text5,'Visible','on')
set(handles.text6,'Visible','on')
set(handles.text7,'Visible','on')
set(handles.text8,'Visible','on')
set(handles.text9,'Visible','on')
set(handles.errormsg,'Visible','off')

elseif c13
set(handles.uitable_inodal,'Visible','on')
set(handles.uitable_conduction,'Visible','off')
set(handles.uitable_radiation,'Visible','off')

set(handles.uitable_controlledconvection,'Visible','off')
set(handles.uitable_forcedconvection,'Visible','off')
set(handles.uitable_userdefined,'Visible','off')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')

set(handles.uitable_mcontrolledconvection,'Visible','on')
set(handles.uitable_mforcedconvection,'Visible','off')
set(handles.uitable_muserdefined,'Visible','off')
set(handles.uitable_aforcedconvection,'Visible','off')
set(handles.uitable_velocityflowfield,'Visible','off')
set(handles.uitable_jnodal,'Visible','on')
set(handles.uitable_errlim,'Visible','on')
set(handles.uitable_controlparameter,'Visible','on')
set(handles.text5,'Visible','on')
set(handles.text6,'Visible','on')
set(handles.text7,'Visible','on')
set(handles.text8,'Visible','on')
set(handles.text9,'Visible','on')
set(handles.errormsg,'Visible','off')

else
set(handles.uitable_inodal,'Visible','off')
set(handles.uitable_conduction,'Visible','off')
set(handles.uitable_radiation,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')

set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_forcedconvection,'Visible','on')
set(handles.uitable_userdefined,'Visible','off')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_mforcedconvection,'Visible','off')
set(handles.uitable_muserdefined,'Visible','off')
set(handles.uitable_aforcedconvection,'Visible','off')
set(handles.uitable_velocityflowfield,'Visible','off')
set(handles.uitable_jnodal,'Visible','off')
set(handles.uitable_errlim,'Visible','off')
set(handles.uitable_controlparameter,'Visible','off')
set(handles.text5,'Visible','off')
set(handles.text6,'Visible','off')
set(handles.text7,'Visible','off')
set(handles.text8,'Visible','off')
set(handles.text9,'Visible','off')
set(handles.errormsg,'Visible','on')
end

% --- Executes during object creation, after setting all properties.
function popupmenu_connectionheat_CreateFcn(hObject, eventdata, handles)

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

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

% --- Executes on selection change in popupmenu_connectionair.
function popupmenu_connectionair_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_connectionair (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu_connectionair contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu_connectionair

get(handles.popupmenu_network,'Value');
a=get(handles.popupmenu_network,'Value');

get(handles.popupmenu_inode,'Value');
b=get(handles.popupmenu_inode,'Value');

get(handles.popupmenu_connectionair,'Value');
d=get(handles.popupmenu_connectionair,'Value');

%Air
c14=(a==1) && (d==1);\phantom{\text{With energy conversion}}
c15=(a==1) && (d==2);\phantom{\text{Without energy conversion}}
c16=(a==1) && (d==3);\phantom{\text{User defined-Momentum}}
c17=(a==1) && (d==4);\phantom{\text{User defined-Air-mass flux}}
c18=(a==1) && (d==5);\phantom{\text{User defined-Air velocity}}

\%AIR

if c14||c15||c16
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','off')
    set(handles.uitable_radiation,'Visible','off')

    set(handles.uitable_controlledconvection,'Visible','off')
    set(handles.uitable_forcedconvection,'Visible','off')
    set(handles.uitable_userdefined,'Visible','off')
    set(handles.uitable_mconduction,'Visible','off')
    set(handles.uitable_mfreeconvection,'Visible','off')

    set(handles.uitable_mcontrolledconvection,'Visible','off')
    set(handles.uitable_mforcedconvection,'Visible','off')
    set(handles.uitable_muserdefined,'Visible','off')
    set(handles.uitable_aforcedconvection,'Visible','on')
    set(handles.uitable_velocityflowfield,'Visible','on')
    set(handles.uitable_jnodal,'Visible','on')
    set(handles.uitable_errlim,'Visible','on')
    set(handles.uitable_controlparameter,'Visible','on')
    set(handles.text5,'Visible','on')
    set(handles.text6,'Visible','on')
    set(handles.text7,'Visible','on')
    set(handles.text8,'Visible','on')
elseif c17||c18
    set(handles.uitable_inodal,'Visible','on')
    set(handles.uitable_conduction,'Visible','off')
    set(handles.uitable_radiation,'Visible','off')

    set(handles.uitable_controlledconvection,'Visible','off')
    set(handles.uitable_forcedconvection,'Visible','off')
    set(handles.uitable_userdefined,'Visible','off')
    set(handles.uitable_mconduction,'Visible','off')
    set(handles.uitable_mfreeconvection,'Visible','off')

    set(handles.uitable_mcontrolledconvection,'Visible','off')
    set(handles.uitable_mforcedconvection,'Visible','off')
    set(handles.uitable_muserdefined,'Visible','off')
    set(handles.uitable_aforcedconvection,'Visible','off')
    set(handles.uitable_velocityflowfield,'Visible','on')
    set(handles.uitable_jnodal,'Visible','on')
    set(handles.uitable_errlim,'Visible','on')
    set(handles.uitable_controlparameter,'Visible','on')
    set(handles.text5,'Visible','on')
    set(handles.text6,'Visible','on')
    set(handles.text7,'Visible','on')
    set(handles.text8,'Visible','on')
    set(handles.text9,'Visible','on')
    set(handles.errormsg,'Visible','off')

else
set(handles.uitable_inodal,'Visible','off')
set(handles.uitable_conduction,'Visible','off')
set(handles.uitable_radiation,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_controlledconvection,'Visible','off')
set(handles.uitable_forcedconvection,'Visible','off')
set(handles.uitable_userdefined,'Visible','off')
set(handles.uitable_mconduction,'Visible','off')
set(handles.uitable_mfreeconvection,'Visible','off')
set(handles.uitable_mcontrolledconvection,'Visible','off')
set(handles.uitable_mforcedconvection,'Visible','off')
set(handles.uitable_muserdefined,'Visible','off')
set(handles.uitable_aforcedconvection,'Visible','off')
set(handles.uitable_velocityflowfield,'Visible','off')
set(handles.uitable_jnodal,'Visible','off')
set(handles.uitable_errlim,'Visible','off')
set(handles.uitable_controlparameter,'Visible','off')
set(handles.text5,'Visible','off')
set(handles.text6,'Visible','off')
set(handles.text7,'Visible','off')
set(handles.text8,'Visible','off')
set(handles.text9,'Visible','off')
set(handles.errormsg,'Visible','on')
end

% --- Executes on selection change in popupmenu_network.
function popupmenu_network_Callback(hObject, eventdata, handles)
get(handles.popupmenu_network,'Value');
a=get(handles.popupmenu_network,'Value');

if a==2
    set(handles.popupmenu_connectionair,'Visible','off')
    set(handles.popupmenu_connectionheat,'Visible','on')

elseif a==1
    set(handles.popupmenu_connectionair,'Visible','on')
    set(handles.popupmenu_connectionheat,'Visible','off')
end

% --- Executes during object creation, after setting all properties.
function popupmenu_network_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu_inode.
function popupmenu_inode_Callback(hObject, eventdata, handles)
function popupmenu_inode_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on selection change in popupmenu_jnode.
function popupmenu_jnode_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function popupmenu_jnode_CreateFcn(hObject, eventdata, handles)

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

% --- Executes when entered data in editable cell(s) in uitable_inodal.
function uitable_inodal_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_jnodal.
function uitable_jnodal_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);
% --- Executes when entered data in editable cell(s) in uitable_userdefined.

function uitable_userdefined_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_conduction.
function uitable_conduction_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_radiation.
function uitable_radiation_CellEditCallback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton_next.
data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_controlledconvection.
functionuitable_controlledconvection_CellEditCallback(hObject,
eventdata, handles)

data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_forcedconvection.
functionuitable_forcedconvection_CellEditCallback(hObject,
eventdata, handles)

data = get(hObject,'data');
disp(data);

functionpushbutton_next_Callback(hObject, eventdata,
handles)

global DUCH DUCHg CODh CODhg COCh COChg FOCh FOChg T_tm
TG_tm RADh RADhg;
global DUCm DUCmg CODm CODmg COCm COCmg FOCm FOCmg FRCm
FRCmg VFFa VFFag;
global FOCa FOCag;

g = get(handles.popupmenu_network,'Value');
a = get(handles.popupmenu_network,'Value');

get(handles.popupmenu_inode,'Value');
b = get(handles.popupmenu_inode,'Value');
get(handles.popupmenu_connectionheat,'Value');
c=get(handles.popupmenu_connectionheat,'Value');

getch(handles.popupmenu_connectionair,'Value');
d=get(handles.popupmenu_connectionair,'Value');

%Heat

cl=(a==2) && (c==7);          %User-defined
cl2=(a==2) && (c==1);          %Conduction - parallel plates
cl3=(a==2) && (c==2);          %Conduction - cylindrical surfaces
cl4=(a==2) && (c==3);          %Convection - free
cl5=(a==2) && (c==4);          %Convection - forced
cl6=(a==2) && (c==5);          %Convection - controlled
cl7=(a==2) && (c==6);          %Radiation

%Moisture

c8=(a==3) && (c==7);          %User-defined
cl9=(a==3) && (c==1);          %Conduction - parallel plates
cl10=(a==3) && (c==2);         %Conduction - cylindrical surfaces
cl11=(a==3) && (c==3);         %Convection - free
cl12=(a==3) && (c==4);         %Convection - forced
cl13=(a==3) && (c==5);         %Convection - controlled

%Air

c14=(a==1) && (d==1);          %With energy conversion
cl15=(a==1) && (d==2);         %Without energy conversion
cl16=(a==1) && (d==3);         %User defined-Momentum
cl17=(a==1) && (d==4);         %User defined-Air-mass flux
c18=(a==1) && (d==5); %User defined-Air velocity

%Temperature
table1=get(handles.uitable_inodal,'Data');
u1=table1(6);
table2=get(handles.uitable_jnodal,'Data');
u2=table2(6);
yes=get(handles.checkbox_generator,'Value'); %Generator check box status
data1=get(handles.uitable_inodal,'Data'); %Get value from i and j nodal tables
data2=get(handles.uitable_jnodal,'Data');

i=data1(1); %getting 'i' node
j=data2(1); %getting 'j' node
if c1
    data3=get(handles.uitable_userdefined,'Data');
    ir=1/data3(1);
    irg=1/data3(1);

    if yes
        DUCHg=[DUChg; 700, i, i, irg];
        TG_tm(1,i)=u1;
    else
        DUCH=[DUCh; 700, i, j, ir];
        T_tm(1,i)=u1;
    end
elseif c2
    data4=get(handles.uitable_conduction,'Data');
ai=data4(1);  
L=data4(2);  
K=data4(3);  
m=data4(4);  
ri=0;  
ro=0;

if yes
  CODhg=[CODhg; 500 i j ai ri ro L K m];
  TG_tm(1,i)=u1;
else
  %CODh=[CODh; 500 i1 i2 100*ds1(i) 0 0 Prd/2 K 1];
  CODh=[CODh; 500 i j ai ri ro L K m];
  T_tm(1,i)=u1;
  T_tm(1,j)=u2;
end

%c3 to be done
%c4 to be done

elseif c5
  data5=get(handles.uitable_forcedconvection,'Data');
  k=data5(1);
  l=data5(2);
  ai=data5(3);
  ri=data5(4);
  ro=data5(5);
  L=data5(6);
  kod=data5(7);
  m=data5(8);
  led=0;  % not yet developed in the current MF version
if yes
   FOChg=[FOChg; 302 i j k l ai ri ro L kod led m];
   TG_tm(1,i)=u1;
else
   FOCh=[FOCh; 302 i j k l ai ri ro L kod led m];
   T_tm(1,i)=u1;
   T_tm(1,j)=u2;
end

elseif c6
   data6=get(handles.uitable_controlledconvection,'Data');
   k=data6(1);
   l=data6(2);
   ai=data6(3);
   m=data6(4);
   if yes
      COChg=[COChg; 400, i, j, k, l, ai, m];
      TG_tm(1,i)=u1;
   else
      COCh=[COCh; 400, i, j, k, l, ai, m];
      T_tm(1,i)=u1;
      T_tm(1,j)=u2;
   end

elseif c7
   data5=get(handles.uitable_radiation,'Data');
   ai=data5(1);
   e=data5(2);
   edd=1/(1/e+1/e-1);
   vf=data5(3);
m=data5(4);
if yes
    RADhg=[RADhg; 600 i j ai e vf m ];
%RADhg=[RADhg; 600 i j ai e vf m ];
    TG_tm(1,i)=u1;
else
    RADh=[RADh; 600 i j ai e vf m ];
    T_tm(1,i)=u1;
    T_tm(1,j)=u2;
end
elseif c8
    data8=get(handles.uitable_muserdefined,'Data');
    ir=1/data8(1);
    irg=1/data8(1);
    if yes
        DUCmg=[DUCmg; 700, i, i, irg];
        TG_tm(1,i)=u1;
    else
        DUCm=[DUCm; 700, i, j, ir];
        T_tm(1,i)=u1;
        T_tm(1,j)=u2;
    end
elseif c9
    data9=get(handles.uitable_mconduction,'Data');
    ai=data9(1);
    L=data9(2);
    K=data9(3);
m=data9(4);
ri=0;
ro=0;

if yes
CODmg=[CODmg; 500 i j ai ri ro L K m];
TG_tm(1,i)=u1;
else
%CODh=[CODh; 500 i1 i2 100*dsl(i) 0 0 Prd/2 K 1];
CODm=[CODm; 500 i j ai ri ro L K m];
T_tm(1,i)=u1;
T_tm(1,j)=u2;
end

%c10 - Conduction - cylindrical to be done

elseif c11
data11=get(handles.uitable_mfreeconvection,'Data');
a=data11(1);
Be=data11(2);
Bc=data11(3);
L=data11(4);
m=data11(5);
if yes
FRCmg=[FRCmg; 20 i j a Be Bc L m];
%frcm=[frcm; i*0+20 ilinerw(i)' ilinerw(i)'
Prd*dsl(i)*(wetness) i*0+h_evap*lewm i*0+h_cond*lewm i*0
i*0+1];
TG_tm(1,i)=u1;
else
FRCm=[FRCm; 20 i j a Be Bc L m];
elseif c12
    data12=get(handles.uitable_mforcedconvection,'Data');
k=data12(1);
l=data12(2);
ai=data12(3);
ri=data12(4);
ro=data12(5);
L=data12(6);
kod=data12(7);
m=data12(8);
led=0; % not yet developed in the current MF version
    if yes
        FOCmg=[FOCmg; 302 i j k l ai ri ro L kod led m];
        TG_tm(1,i)=u1;
    else
        FOCm=[FOCm; 302 i j k l ai ri ro L kod led m];
        T_tm(1,i)=u1;
        T_tm(1,j)=u2;
    end

elseif c13
    data13=get(handles.uitable_mcontrolledconvection,'Data');
k=data13(1);
l=data13(2);
ai=data13(3);
m=data13(4);

    if yes
        COCmg=[COCmg; 400, i, j, k, l, ai, m];
        TG_tm(1,i)=u1;
    else
        COCm=[COCm; 400, i, j, k, l, ai, m];
        T_tm(1,i)=u1;
        T_tm(1,j)=u2;
    end

elseif c14

data14=get(handles.uitable_aforcedconvection,'Data');
zi=data14(1);
zj=data14(2);
ai=data14(3);
aj=data14(4);
dhij=data14(5);
L=data14(6);
kod=data14(7);
K=data14(8);
m=data14(9);

    if yes
        FOCag=[FOCag; 300, i, j, zi, zj, ai, aj, dhij, L,
        kod, K, m];
        TG_tm(1,i)=u1;
    else
        FOCa=[FOCa; 300, i, j, zi, zj, ai, aj, dhij, L, kod,
        K, m];
        T_tm(1,i)=u1;
T_tm(1,j)=u2;
end

elseif c15

data14=get(handles.uitable_aforcedconvection,'Data');
zi=data14(1);
zj=data14(2);
ai=data14(3);
aj=data14(4);
dhij=data14(5);
L=data14(6);
kod=data14(7);
K=data14(8);
m=data14(9);
if yes
    FOCag=[FOCag; 310, i, j, zi, zj, ai, aj, dhij, L, kod, K, m];
    TG_tm(1,i)=u1;
else
    FOCa=[FOCa; 310, i, j, zi, zj, ai, aj, dhij, L, kod, K, m];
    T_tm(1,i)=u1;
    T_tm(1,j)=u2;
end

elseif c16

data14=get(handles.uitable_aforcedconvection,'Data');
zi=data14(1);
zj=data14(2);
ai=data14(3);
aj=data14(4);
dhij=data14(5);
L=data14(6);
kod=data14(7);
K=data14(8);
m=data14(9);
if yes
    FOCag=[FOCag; 320, i, j, zi, zj, ai, aj, dhij, L, kod, K, m];
    TG_tm(1,i)=u1;
else
    FOCa=[FOCa; 320, i, j, zi, zj, ai, aj, dhij, L, kod, K, m];
    T_tm(1,i)=u1;
    T_tm(1,j)=u2;
end

elseif c17

data17=get(handles.uitable_velocityflowfield,'Data');
k=data17(1);
l=data17(2);
v=data17(3);
L=data17(4);
H=data17(5);
Dh=data17(6);
ai=data17(7);
aj=data17(8);
K=data17(9);
m=data14(10);
if yes
elseif c18

data17 = get(handles.uitable_velocityflowfield,'Data');
k=data17(1);
l=data17(2);
v=data17(3);
L=data17(4);
H=data17(5);
Dh=data17(6);
ai=data17(7);
aj=data17(8);
K=data17(9);
m=data17(10);
if yes
    VFFag=[VFFag; 912, i, j, k, l, v, L, H, Dh, ai, aj, kod, m]
    TG_tm(1,i)=u1;
else
    VFFa=[VFFa; 912, i, j, k, l, v, L, H, Dh, ai, aj, kod, m]
    T_tm(1,i)=u1;
    T_tm(1,j)=u2;
end
T_tm(1,i)=u1;
T_tm(1,j)=u2;
end
end

% --- Executes on button press in pushbutton_save.
function pushbutton_save_Callback(hObject, eventdata, handles)

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

global DUCh DUChg T_tm TG_tm M FOCa FOBa DUCa VFFa FOCag
FOBag DUCag VFFag;

global FRCh FOCh COCh CODh RADh DUCh FRChg FOChg COChg
CODhg RADhg DUChg;

global FRCm FOCm COCm CODm DUCm FRCmg FOCmg COCmg CODmg
DUCmg;
wd='.';

errlim=get(handles.uitable_errlim,'Data');
disac_in=get(handles.uitable_controlparameter,'Data');

%required files for MF
TN=1; % Temporary, set to 1 for number of time divisions
M=size(T_tm,2); % M = Maximum number of nodes
P_tm=zeros(TN,M);
tmp=TG_tm;
TG_tm=zeros(TN,M);
ii=find(abs(tmp)>0);
TG_tm(ii)=tmp(ii);

PG_tm=zeros(TN,M);
qc_tm=zeros(TN,M);
qab_tm=zeros(TN+2,M);
v_tm=zeros(TN+2,M);
Mtmi_tm=zeros(TN,M);
time=zeros(TN+1,1);
qh_tm=zeros(TN,M);
qm_tm=zeros(TN,M);
wf_tm=zeros(TN,M);
Pb_tm=zeros(TN,M);
Pbg_tm=zeros(TN,M);
qabg_tm=zeros(TN,M);
vg_tm=zeros(TN,M);
qa_tm=zeros(TN,M);

% save results
mwrite([wd '/disac_d/disac_in.dat'], disac_in);
mwrite([wd '/disac_d/errlim.dat'], errlim);
mwrite([wd '/disac_d/T_tm.dat'], T_tm);
mwrite([wd '/disac_d/P_tm.dat'], P_tm);
mwrite([wd '/disac_d/TG_tm.dat'], TG_tm);
mwrite([wd '/disac_d/PG_tm.dat'], PG_tm);
mwrite([wd '/disac_d/qc_tm.dat'], qc_tm);
mwrite([wd '/disac_d/qab_tm.dat'], qab_tm);
mwrite([wd '/disac_d/v_tm.dat'], v_tm);
mwrite([wd '/disac_d/Mtmi_tm.dat'], Mtmi_tm);
mwrite([wd '/disac_d/time.dat'], time);
mwrite([wd '/disac_d/qh_tm.dat'], qh_tm);
mwrite([wd '/disac_d/qm_tm.dat'],qm_tm);
mwrite([wd '/disac_d/wf_tm.dat'],wf_tm);
mwrite([wd '/disac_d/Pb_tm.dat'],Pb_tm);
mwrite([wd '/disac_d/Pbg_tm.dat'],Pbg_tm);
mwrite([wd '/disac_d/qabg_tm.dat'],qabg_tm);
mwrite([wd '/disac_d/vg_tm.dat'],vg_tm);
mwrite([wd '/disac_d/qa_tm.dat'],qa_tm);

mwrite([wd '/cfd_d/VFFa.dat'], VFFa);
mwrite([wd '/cfd_d/VFFag.dat'], VFFag);
mwrite([wd '/cfd_d/FOCa.dat'], FOCa);
mwrite([wd '/cfd_d/FOCag.dat'], FOCag);
mwrite([wd '/cfd_d/foba.dat'], FOBa);
mwrite([wd '/cfd_d/fobag.dat'], FOBag);
mwrite([wd '/cfd_d/DUCa.dat'], DUCa);
mwrite([wd '/cfd_d/DUCag.dat'], DUCag);

mwrite([wd '/cfd_d/frch.dat'], FRCh);
mwrite([wd '/cfd_d/foch.dat'], FOCh);
mwrite([wd '/cfd_d/coch.dat'], COCh);
mwrite([wd '/cfd_d/codh.dat'], CODh);
mwrite([wd '/cfd_d/radh.dat'], RADh);
mwrite([wd '/cfd_d/duch.dat'], DUCH);

mwrite([wd '/cfd_d/frchg.dat'], FRChg);
mwrite([wd '/cfd_d/fochg.dat'], FOChg);
mwrite([wd '/cfd_d/cochg.dat'], COChg);
mwrite([wd '/cfd_d/codhg.dat'], CODhg);
mwrite([wd '/cfd_d/radhg.dat'], RADhg);
mwrite([wd '/cfd_d/duchg.dat'], DUCHg);

mwrite([wd '/cfd_d/frcm.dat'], FRCm);
mwrite([wd '/cfd_d/focm.dat'], FOCm);
mwrite([wd '/cfd_d/cocm.dat'], COCm);
mwrite([wd '/cfd_d/codm.dat'], CODm);
mwrite([wd '/cfd_d/ducm.dat'], DUCm);

mwrite([wd '/cfd_d/frcmg.dat'], FRCmg);
mwrite([wd '/cfd_d/focmg.dat'], FOCmg);
mwrite([wd '/cfd_d/cocmg.dat'], COCmg);
mwrite([wd '/cfd_d/codmg.dat'], CODmg);
mwrite([wd '/cfd_d/ducmg.dat'], DUCmg);

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

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

yes = get(hObject, 'Value');
if yes
    set(handles.popupmenu_jnode,'Enable','Off')
set(handles.uitable_jnodal,'Enable', 'Off')
else
    set(handles.popupmenu_jnode,'Enable','On')
    set(handles.uitable_jnodal,'Enable', 'On')
end

% --- Executes when entered data in editable cell(s) in uitable_errlim.
function uitable_errlim_CellEditCallback(hObject, eventdata, handles)

% --- Executes when entered data in editable cell(s) in uitable_controlparameter.
function uitable_controlparameter_CellEditCallback(hObject, eventdata, handles)

% --- Executes on key press with focus on pushbutton_next and none of its controls.
function pushbutton_next_KeyPressFcn(hObject, eventdata, handles)

% --- Executes on key press with focus on pushbutton_save and none of its controls.
function pushbutton_save_KeyPressFcn(hObject, eventdata, handles)

% --- Executes when entered data in editable cell(s) in uitable_mconduction.
function uitable_mconduction_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_mcontrolledconvection.
function
uitable_mcontrolledconvection_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_mforcedconvection.
function
uitable_mforcedconvection_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);

% --- Executes when entered data in editable cell(s) in uitable_muserdefined.
function uitable_muserdefined_CellEditCallback(hObject, eventdata, handles)

data = get(hObject,'data');
disp(data);
% --- Executes when entered data in editable cell(s) in
uitable_mfreeconvection.
function uitable_mfreeconvection_CellEditCallback(hObject,
eventdata, handles)

% --- Executes when entered data in editable cell(s) in
uitable_velocityflowfield.
function
uitable_velocityflowfield_CellEditCallback(hObject,
eventdata, handles)

% --- Executes when entered data in editable cell(s) in
uitable_aforcedconvection.
function
uitable_aforcedconvection_CellEditCallback(hObject,
eventdata, handles)

% --- Executes during object creation, after setting all
properties.
function popupmenu_connectionair_CreateFcn(hObject,
eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
14.2 Appendix 2: Diesel equipment emission calculations

This appendix summarizes a study in which the relation between the amount of air consumption by any diesel machinery in underground mines and the amount of CO₂, water vapor and heat generated by the engine, is calculated. These calculations are based on the information provided from Barrick Gold Strike mine’s annual diesel consumption report.

Diesel is not a single compound, but a mixture of hydrocarbons ranging from C₁₀H₂₀ to C₁₅H₂₈. Diesel is represented by an “average” of the mixture, which is C₁₂H₂₃. The chemical equation for the complete combustion of a diesel molecule is given by:

\[
4 \text{C}_{12}\text{H}_{23} + 71 \text{O}_2 \rightarrow 48 \text{CO}_2 + 46 \text{H}_2\text{O}
\]

**AFR mixture:**

For gasoline, the stoichiometric air/fuel mixture is approximately 14.95:1, i.e, the approximate mass of the air is 14.7 times the fuel’s. Any mixture less than this ratio is a rich mixture, and one with more than this ratio is considered to be a lean mixture.

\[
\text{C}_x\text{H}_y + a (\text{O}_2 + 0.79 / 0.21\text{N}_2) \rightarrow x \text{CO}_2 + y/2 \text{H}_2\text{O} + a (0.79/0.21)\text{N}_2
\]

Where, \(a = (x+y)/4\)

So,

\[
\text{AFR} = a \left(\frac{\text{MW} \text{O}_2 + (0.79/0.21) \text{MW} \text{N}_2}{\text{MWfuel}}\right)
\]

AFR value for petrol is 15.06, and 14.95 for diesel. 18:1 is the ideal ratio for optimum power and minimal smoke emissions, as per experimental results. The annual fuel consumption data summarized from Meikle and Rodeo mines is presented in Table A1.

Table A1. Meikle and Rodeo mines annual fuel consumption.

<table>
<thead>
<tr>
<th></th>
<th>Meikle mine – for 360 days</th>
<th>Rodeo mine – for 360 days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gallons per year</td>
<td>577422</td>
<td>347947</td>
</tr>
<tr>
<td>Liters per second</td>
<td>0.070266</td>
<td>0.042341</td>
</tr>
</tbody>
</table>
Mine equipment – LDH:

According to the proposed ideal air-fuel ratio, Meikle mine needs to be supplied with 1.265 liters/sec of air for a consumption of 0.0703 liters/sec of fuel. A typical LHD in the mine uses the Duetz engine (DUETZ) with 40 HP. The engine specifications of Duetz 912 family are listed in Table A2.

Table A2. Duetz 912 family – Construction engine specifications.

<table>
<thead>
<tr>
<th>Engine</th>
<th>Duetz F3L912</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>40 kW</td>
</tr>
<tr>
<td>Fuel consumption</td>
<td>225 gr/kWh or 0.225 lit/kWh</td>
</tr>
</tbody>
</table>

From the information on fuel consumption in Table A2, and from the fact that the energy produced from combustion of a kg diesel is 44,800 kJ/kg, we can calculate the efficiency of the machine, which comes to be 35%. So, the remaining energy is liberated as heat, which is 28,800 kJ per kg of diesel.

From the equation above, one can also say that 1 liter of diesel produces 2.65 kg of CO₂ and 2.352 kg of water vapor up on combustion.