DOT/FAA/TC-21/3

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

# Fuel Tank Flammability Assessment Method User's Manual – Updated for Version 11

January, 2021

Final Report

U.S. Department of Transportation
**Federal Aviation Administration**

## NOTICE

Technical Report Documentation Page

**Form DOT F 1700.7** (8-72) Reproduction of completed page authorized

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| DOT/FAA/TC-21/3 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| Fuel Tank Flammability Assessment Method User's Manual – Updated for Version 11 | January 2021 |
| | 6. Performing Organization Code<br>ANG-E21 |
| 7. Author(s)<br>Steven M. Summer | 8. Performing Organization Report No. |
| 9. Performing Organization Name and Address<br>U.S. Department of Transportation<br>William J. Hughes Technical Center<br>Aviation Research Division Fire Safety Branch, ANG-E21<br>Atlantic City International Airport, NJ 0840 | 10. Work Unit No. (TRAIS)<br><br>11. Contract or Grant No.<br>NA |
| 12. Sponsoring Agency Name and Address<br>FAA Northwest Mountain Regional Office<br>1601 Lind Avenue SW<br>Renton, WA 98057 | 13. Type of Report and Period Covered<br>Final Report<br><br>14. Sponsoring Agency Code |

15. Supplementary Notes

16. Abstract

The Fuel Tank Flammability Assessment Method (FTFAM) is a Federal Aviation Administration-developed computer model designed as a comparative analysis tool to determine airplane fuel tank flammability as a requirement of Title 14 Code of Federal Regulations Section 25.981. The model uses Monte Carlo statistical methods to determine the average fuel tank flammability of a fleet of airplanes based upon randomly selecting certain unknown variables over defined distributions for a large number of flights. The FTFAM iterates through each flight, calculating the flammability exposure time of each flight given the data input provided by the user. Calculating this flammability exposure time for a sufficiently large number of flights results in statistically reliable flammability exposure data. These calculations can be performed for fuel tank types utilized in transport airplanes, including body tanks located in the fuselage, wing tanks, and center wing tanks. The program can also be modified by the user to determine fuel tank flammability when a flammability reduction means is employed.

This report serves as a user's manual for this computer model to assist the user in its operation and to discuss the permissible changes that may be made to this model specific to a particular fleet of aircraft. It is updated through version 11 of the FTFAM. The user should reference Advisory Circular 25.981-2A for additional guidance on when to use this model and for a discussion of interpretation of results.

| 17. Key Words | 18. Distribution Statement |
|---|---|
| Flammability, Fuel, Fuel tank, Ignition, Inert, Monte Carlo | This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov. |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 83 | |

## Contents

## Figures

## Tables

## Acronyms

| Acronym | Definition |
|---|---|
| AC | Advisory Circular |
| AFM | Aircraft Flight Manual |
| CWT | Center Wing Tank |
| FAA | Federal Aviation Administration |
| FEET | Flammability Exposure Evaluation Time |
| FRM | Flammability Reduction Means |
| FTFAM | Fuel Tank Flammability Assessment Method |
| LFL | Lower Flammability Limit |
| MEL | Minimum Equipment List |
| MTBF | Mean Time Between Failure |
| nm | Nautical Miles |
| OAT | Outside Air Temperature |
| TAT | Total Air Temperature |
| UFL | Upper Flammability Limit |

## Executive summary

The Fuel Tank Flammability Assessment Method (FTFAM) is a Federal Aviation Administration-developed computer model designed as a comparative analysis tool to determine aircraft fuel tank flammability as a requirement of Title 14 Code of Federal Regulations 25.981. The FTFAM must be used to meet the Section 25.981 requirements for demonstrating the average percentage of time a fuel tank is flammable meets defined limits for both the overall combined operations, and separately, the warm day operations. Warm day operations are defined as those when the ambient temperature at takeoff is above 80°F. In addition, when a flammability reduction means (FRM) is used, Appendix M to part 25 provides further requirements with regards to reliability and performance of the FRM. The FTFAM uses Monte Carlo statistical methods to generate flammability data for certain unknown variables over known distributions for a large number of flights. The FTFAM iterates through each flight, calculating the flammability exposure time of each flight given the data input provided by the user. Calculating this flammability exposure time for a sufficiently large number of flights results in statistically reliable flammability exposure data. These calculations can be performed for fuel tank types utilized in transport airplanes, including body tanks located in the fuselage, wing tanks, and center wing tanks. The program can also be modified by the user to determine fuel tank flammability when a flammability reduction means is employed.

This report serves as a user's manual for the FTFAM to assist the user in its operation and to discuss the permissible changes that may be made to this model specific to a particular fleet of aircraft. It is updated through version 11 of the FTFAM. The user should reference Advisory Circular (AC) 25.981-2A for additional guidance on when to use this model and the interpretation of results.

# 1 Introduction

Users of this manual are encouraged to read AC 25.981-2A before beginning the development of a flammability analysis for their product. A qualitative flammability assessment method is acceptable for conventional unheated aluminum wing tanks. Guidance on performing the qualitative flammability assessment and determining if it is appropriate for the fuel tank under evaluation is provided in AC 25.981-2A. Information in the AC provides guidance on means of compliance with Section 25.981 and explains the user inputs to the Fuel Tank Flammability Assessment Method (FTFAM) such as considerations for developing a fuel tank thermal model for the tank being evaluated. This User's Manual is the required, standardized, method for determining the flammability of fuel tanks. Once the fuel tank flammability has been determined the applicant may need to incorporate a flammability reduction means (FRM) in their design in order to meet the flammability requirements defined in Section 25.981 and Appendix M to part 25. The AC discusses flammability reduction means options such as fuel tank inerting, fuel tank cooling and fuel tank pressurization that may be used. If an FRM is required, the applicant must develop a unique flammability reduction system model, based upon the design specific features, that is incorporated into the FTFAM to calculate the flammability with an FRM installed.

The FTFAM is a computer model that is used as a comparative tool to assist in determining the potential flammability exposure of the fuel tank ullage (vapor space above the fuel) of a fleet of particular aircraft. To accomplish this, the model simulates a large number of flights, comparing the bulk average fuel temperature at each time increment of flight to the lower flammability limit (LFL) and upper flammability limit (UFL). When the bulk average fuel temperature is higher than the LFL and lower than the UFL, the fuel vapor is said to be flammable.

The model uses Monte Carlo statistical methods to determine four unknown variables necessary for the computations. Standardized distributions of these variables, defined in Appendix N of Title 14 Code of Federal Regulations Part 25, are used by the model to randomly select a value for each of the variables that influence the fuel tank flammability during a given flight. These random number methods are used to select the fuel flashpoint temperature, flight mission length, and departing airport ambient and cruise atmospheric temperature. By Monte Carlo theory, generating these random numbers over a sufficiently large number of trials and averaging the combined results will minimize the errors associated with these probabilistic calculations. Typically, as the number of trials are increased within a Monte Carlo model, the error is reduced. This is reflected in Table 5 of Appendix N where the flammability limits were established based upon the number of Monte Carlo flights that are simulated. For example, the average flammability limit for 10,000 flights is 2.91, for 100,000 flights it is 2.98 and for 1,000,000 it is 3 percent.

It should be noted that this model was written as an engineering tool utilizing a Microsoft Excel spreadsheet and visual basic coding. The model's code was written in such a way that the engineers using it could understand and follow the computations and theory involved. Version 11 of the FTFAM Excel spreadsheet can be downloaded from the FAA Technical Center website: (http://www.fire.tc.faa.gov/systems/fueltank/FTFAM.stm). This manual serves as a tool for the user to assist in the operation of the model and is updated through version 11 of the FTFAM.

# 2 Summary of version 11 updates

Version 11 of the FTFAM provides several updates designed to address slight errors or coding bugs that have been identified by users. As changes were made to these areas, a set of standardized inputs were used to determine the overall impact to the Flammability Exposure Evaluation Time (FEET). The standardized inputs that were used for this evaluation are shown in Figure 1. Table 1 provides a summary of the changes made to this version of the FTFAM along with the impact to the FEET based on these standardized inputs. This summary is also provided on the "Intro" worksheet of the FTFAM. Utilizing the standardized inputs with all of these changes implemented, results in a FEET of 55.45% compared to 52.90% prior to the changes.

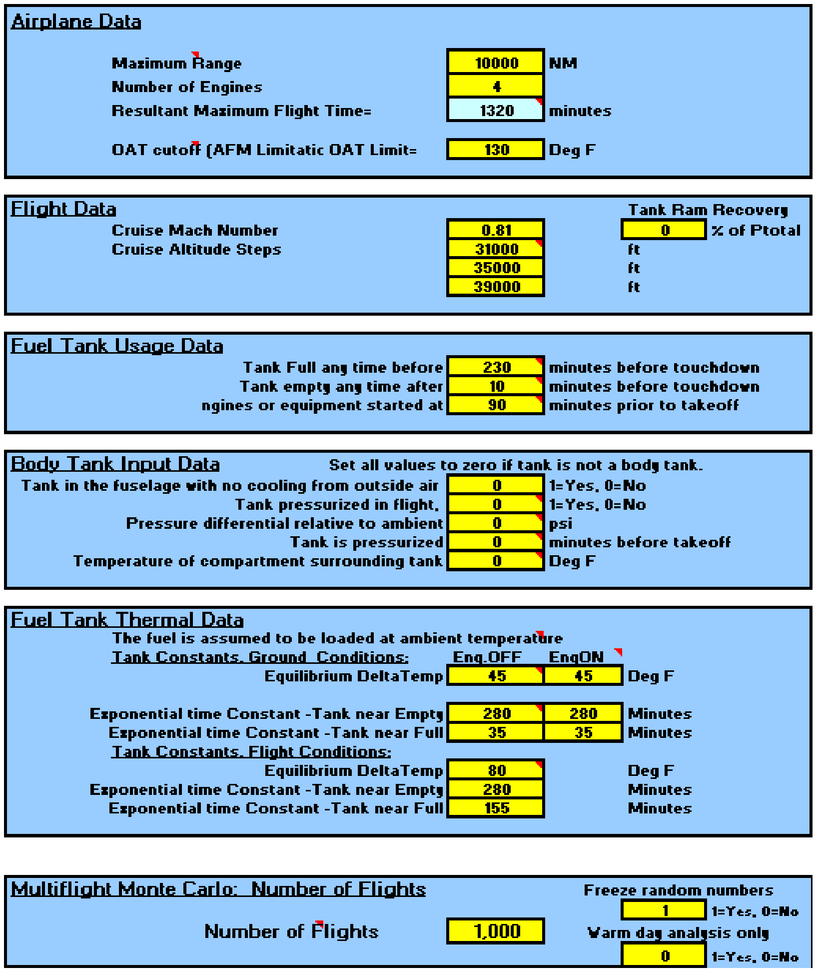


Figure 1. Standardized set of user inputs used to evaluate FTFAM changes

Table 1. Summary of FTFAM V11 changes and their impact to results

| **Summary of Change** | **Standardized Input FEET Prior to Change (1,000 flights)** | **Standardized Input FEET After Change (1,000 flights)** |
|---|---|---|
| Descent profile used in the model was corrected to accurately represent the descent rate that is defined in the regulations. | 52.90% | 55.72% |
| The temperature lapse rate calculations were corrected to consistently utilize a 3.57F/1000 ft lapse rate. | 52.90% | 52.64% |

| **Summary of Change** | **Standardized Input FEET Prior to Change (1,000 flights)** | **Standardized Input FEET After Change (1,000 flights)** |
|---|---|---|
| There was an error in the FRM worksheet which misrepresented the "Contribution to the Whole" in the FRM Performance table. This error was corrected. This calculation was provided for user informational purposes only and has no impact on the results of the model. | NA | NA |
| There was an error in the FRM worksheet, cell 'O14', which misrepresented the Fleet Average Flammability Performance. This error was corrected and has no impact on the results of a flammability analysis. | NA | NA |
| There was an error in the Visual Basic code which made it so that the total FRM flammability time was not transferred onto the 'Summary of n Cases' worksheet. This error was corrected and has no impact on results. | NA | NA |

# 3 Model overview

The Monte Carlo method employed in this model uses random number generation techniques over a standardized range of values for several variables. The results of the multivariable problem are calculated over a large number of trials to compute the average result or range of results.

In a broad view of the analysis, as shown in Figure 2, the main components to the problem are the environment surrounding the fuel tank, flight mission data, fuel properties, and the thermal characteristics of the tank. A detailed description of each of the inputs to these four components is provided in AC 25.981-2A. Once these four components are determined, whether by user input or by Monte Carlo calculations, the model can then determine whether the ullage of the fuel tank for each time increment of flight is flammable or not; and therefore, the percent of the mission time that the fuel tank is flammable. Performing this over a large number of iterations gives the user a value for the fleet average flammability exposure for a specific aircraft model. A simplified flowchart of the major computations of the model is shown in Figure 3. An expanded version of this flowchart, with all user inputs and model preprocessing, is discussed in Section 4.1.

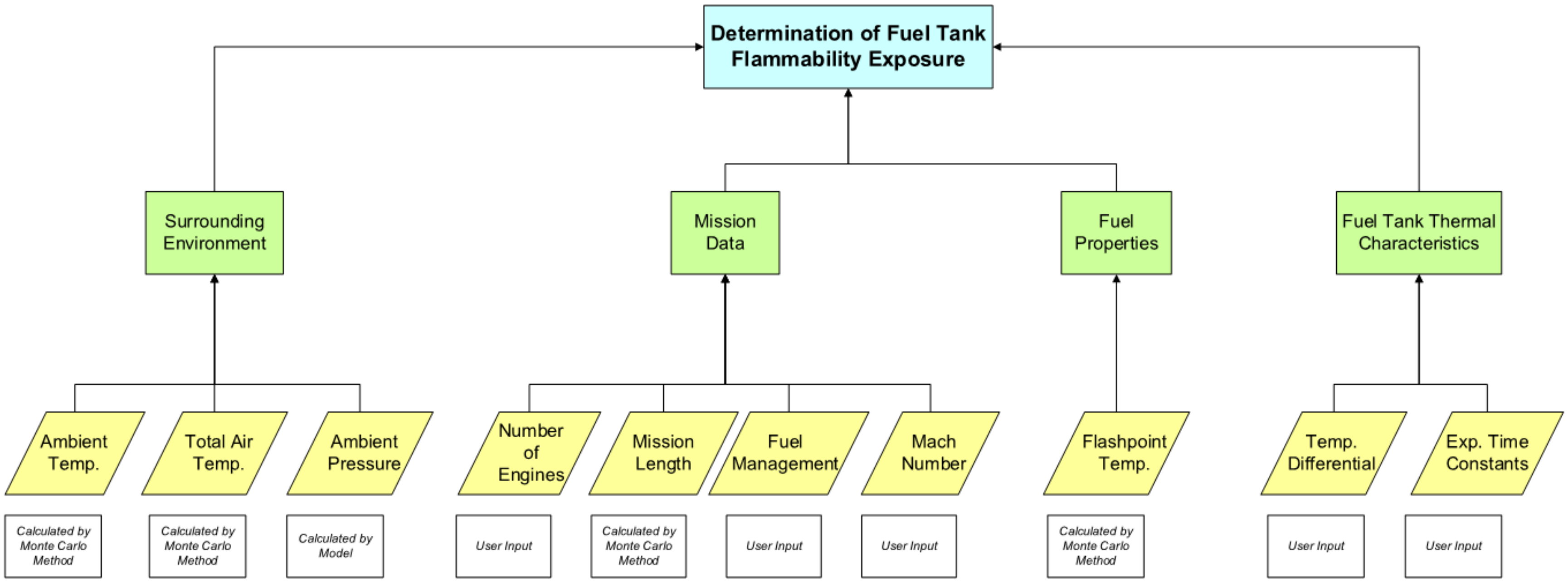

Figure 2. Overview of the main components of the analysis and their associated inputs

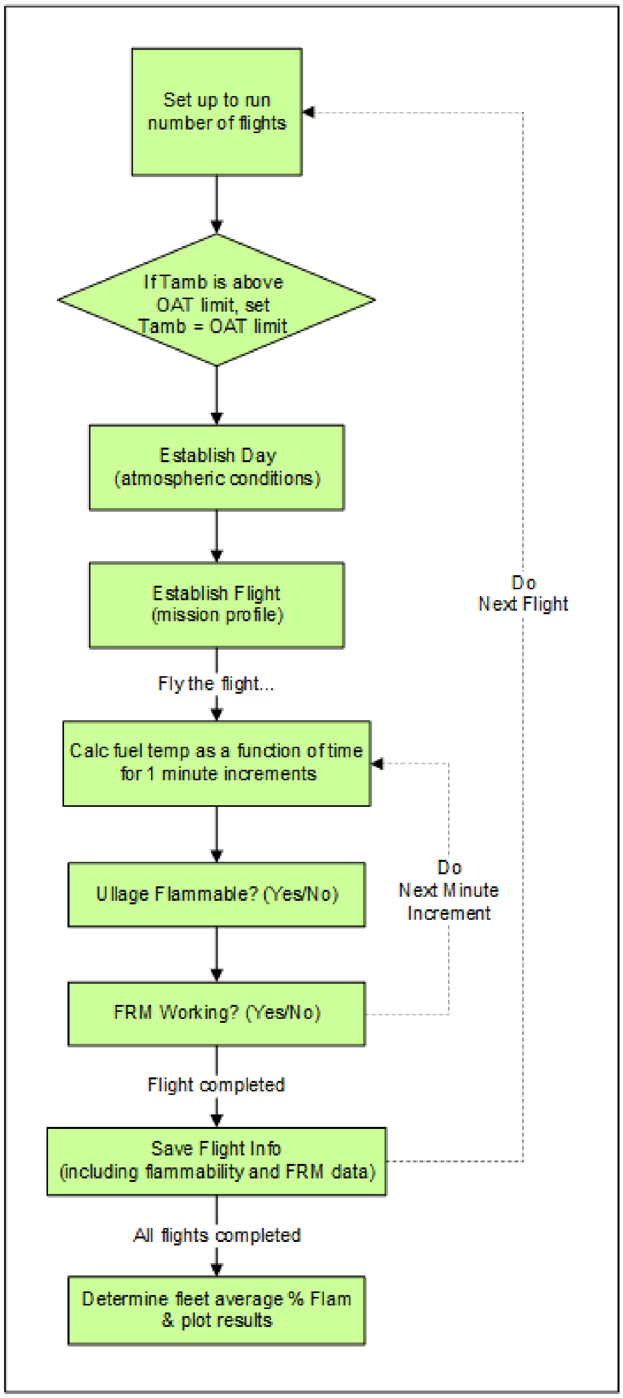


Figure 3. Overview of the computations performed by the Monte Carlo model

The user should note that if the aircraft design uses complex fuel transfer features or other features that significantly influence the thermal characteristics between tanks, the model does not have the capability to analyze these effects. The user should refer to AC 25.981-2 for guidance in this case.

The model was tested using Microsoft Excel 2000®, Microsoft Excel XP®, Microsoft Excel 2003® and Microsoft Excel 2016® on Microsoft Windows 2000®, Microsoft Windows XP® and Microsoft Windows 10® operating systems. Operation on all other versions of Microsoft Excel® and/or Microsoft Windows® is subject to further verification of the model.

## 3.1 Overview and description of model worksheets

The model consists of several separate worksheets that are used for notation to the user, data input, computation, and presentation of results. The worksheets can be separated into the categories shown in Table 2. It should be noted that some worksheets appear in more than one category, as they serve multiple functions. In addition, the user can use several of these worksheets as tools to debug and fix any issues that may arise, such as discrepancies in expected versus actual results. Sections 3.1.1 through 3.1.6 discuss each worksheet and its function in detail.

In all worksheets, a yellow cell denotes a user input cell. Any cell not shaded yellow must remain untouched by the user unless approved by the Federal Aviation Administration (FAA).

Table 2. Model worksheet categories

| **Notation** | **Data Input** | **Computation** | **Results** |
|---|---|---|---|
| Intro | User Inputs and Results<br>FRM | Summary of n Cases<br>Internal Calculations<br>Internal Calc 2<br>Internal Calc 3<br>Internal Calc 4 | User Inputs and Results<br>FRM<br>Summary of n Cases |

### 3.1.1 Intro worksheet

When the computer model is first opened, the user is taken to the Intro worksheet. This worksheet provides a brief statement of the model's intended purpose, as well as several notes and limitations as to its use. On this worksheet, website addresses are provided where the user can find additional information, along with contact information for questions or comments regarding the model is provided. Additionally, the user can view a summary of changes made to the FTFAM from version 5 through the currently released version.

### 3.1.2 User Inputs and Results worksheet

The User Inputs and Results worksheet is the main interface of the FTFAM. It contains all user inputs necessary for performing a Monte Carlo flammability analysis as well as the results from the analysis. The inputs located on this worksheet are divided into six categories, each of which is discussed thoroughly in section 4.1. The results from the Monte Carlo analysis are displayed on this worksheet in the form of a graph depicting the percentage of flight time that the tank was flammable for each of the performed missions.

### 3.1.3 Flammability reduction method worksheet

It should be noted that this worksheet is not needed unless a FRM analysis is conducted. The FRM worksheet allows the user to evaluate the effectiveness of an FRM. The effectiveness of an FRM is separated into reliability and performance factors. The results of the FRM analysis are displayed on this page in the form of several different tables, all of which will be discussed in section 4.4.2.

### 3.1.4 Single-Flight worksheet

The Single-Flight worksheet allows the user to simulate and analyze a particular flight scenario. The user can either perform a single-flight scenario by entering flight time and temperature data specific to that flight or by entering a flight number from the Monte Carlo analysis. The results of the single-flight scenario are displayed in two graphical formats on this worksheet. It should be noted that when performing the Monte Carlo analysis, these plots depict the results of the last flight performed.

### 3.1.5 Summary of n Cases worksheet

The Summary of n Cases worksheet displays the results of each flight in a tabular format, sorted by the percentage of flight time that the fuel tank was flammable. Along with preflight ground time, flight time, and various flight-specific temperatures, this table also includes the amount of time that the tank was flammable, the amount of time that the FRM was maintaining an inert ullage, and the percentage of total flight time for both.

### 3.1.6 Internal Calculation worksheets

There are four Internal Calculation worksheets in the model that contain all the essential information processed by the model. All data inputs, calculated values, and the results are stored here for use by the program. All pertinent data are then copied to other worksheets to be displayed in a user-friendly fashion. These worksheets are only used by the model and should not be modified by the user in any way. They are provided to the user for troubleshooting purposes only.

# 4 Operation of the model

The main interface for general operation of the model is the User Inputs and Results worksheet. This worksheet contains all the input cells and various cells displaying much of the calculated results. Cells in this and all worksheets contained in the model are shaded according to their use. A yellow shaded cell indicates a user input; these are the only cells that should be modified by the user. A light blue shaded cell indicates a computed result, and the green shaded cells located throughout the model are buttons that perform an operation when left-clicked.

## 4.1 Monte Carlo analysis user inputs

Figure 4 shows a flowchart depicting the main computations of the FTFAM, including FRM computations, and the usage of each user input. When the Monte Carlo analysis is being performed without an FRM, there are six user input categories.

- Airplane Data
- Flight Data
- Fuel Tank Usage Data
- Body Tank Input Data
- Fuel Tank Thermal Data
- Multiflight Monte Carlo Data

There are six blocks in the User Inputs and Results worksheet corresponding to these categories, as shown in Figure 5. Sections 4.1.1 through 4.1.6 define and discuss each of the parameters within each category.

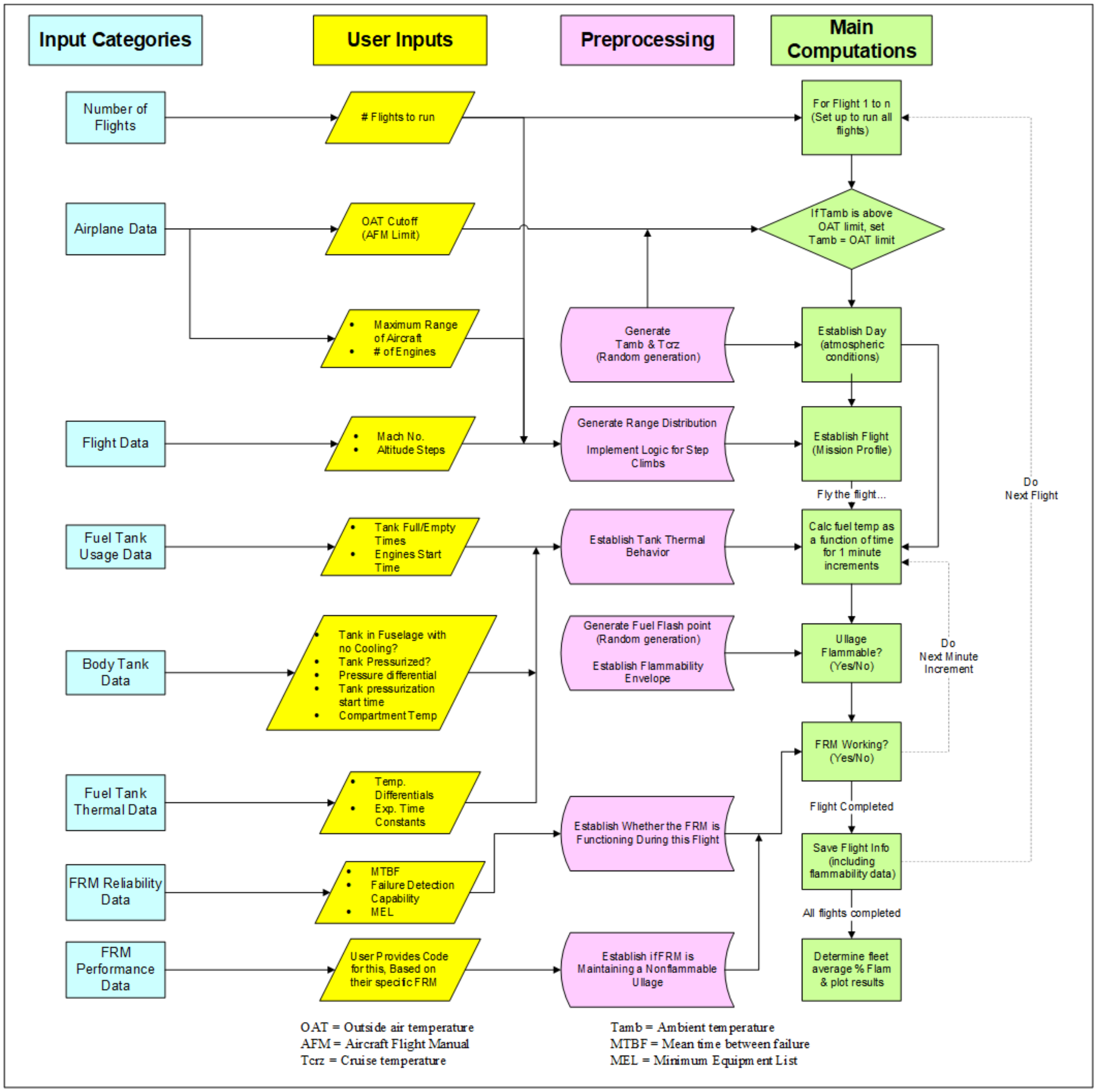


Figure 4. Flowchart of Monte Carlo model operation, including FRM computations

**Airplane Data**

| | | |
|---|---|---|
| Maximum Range | 4500 | NM |
| Number of Engines | 2 | |
| Resultant Maximum Flight Time= | 610 | minutes |
| OAT cutoff (AFM Limitation) OAT Limit= | 130 | Deg F |

**Flight Data**

| | | | Tank Ram Recovery | |
|---|---|---|---|---|
| Cruise Mach Number | 0.81 | | 0.35 | % of Ptotal |
| Cruise Altitude Steps | 31000 | ft | | |
| | 35000 | ft | | |
| | 39000 | ft | | |

**Fuel Tank Usage Data**

| | | |
|---|---|---|
| Tank Full any time before | 610 | minutes before touchdown |
| Tank empty any time after | 500 | minutes before touchdown |
| Engines or equipment started at | 90 | minutes prior to takeoff |

**Body Tank Input Data** Set all values to zero if tank is not a body tank.

| | | |
|---|---|---|
| Tank in the fuselage with no cooling from outside air | 0 | 1=Yes, 0=No |
| Tank pressurized in flight, | 0 | 1=Yes, 0=No |
| Pressure differential relative to ambient | 0 | psi |
| Tank is pressurized | 0 | minutes before takeoff |
| Temperature of compartment surrounding tank | 0 | Deg F |

**Fuel Tank Thermal Data**

The fuel is assumed to be loaded at ambient temperature

| Tank Constants, Ground Conditions: | Eng.OFF | EngON | |
|---|---|---|---|
| Equilibrium DeltaTemp | 60 | 60 | Deg F |
| Exponential time Constant -Tank near Empty | 200 | 200 | Minutes |
| Exponential time Constant -Tank near Full | 400 | 400 | Minutes |
| **Tank Constants, Flight Conditions:** | | | |
| Equilibrium DeltaTemp | 60 | | Deg F |
| Exponential time Constant -Tank near Empty | 200 | | Minutes |
| Exponential time Constant -Tank near Full | 400 | | Minutes |

**Multiflight Monte Carlo: Number of Flights**

| | | | |
|---|---|---|---|
| | | Freeze random numbers | |
| | | 0 | 1=Yes, 0=No |
| Number of Flights | 100,000 | Warm day analysis only | |
| | | 0 | 1=Yes, 0=No |

Figure 5. The six data blocks of the user inputs and results worksheet

### 4.1.1 Airplane data

The airplane data that are required to perform the Monte Carlo calculations include the maximum range of the aircraft, the number of engines, and the outside air temperature (OAT) cutoff limit. These inputs are located in the Airplane Data block on the User Inputs and Results worksheet, as shown in Figure 6.

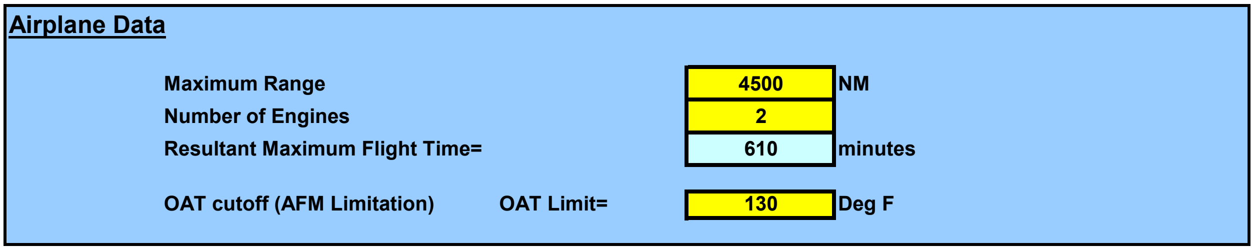


Figure 6. Airplane data block

*4.1.1.1 Maximum range of aircraft*

The maximum range of the aircraft along with the number of flights to be performed (section 4.1.6.1) is used by the model to develop a mission distribution. Figure 7 shows a sample mission distribution for an aircraft with a maximum range of 4,500 nautical miles (nm) and 100,000 total flights. This distribution data is used by the model to randomly select a mission length for each flight that the model generates, while preserving the accuracy in the overall distribution of flights.

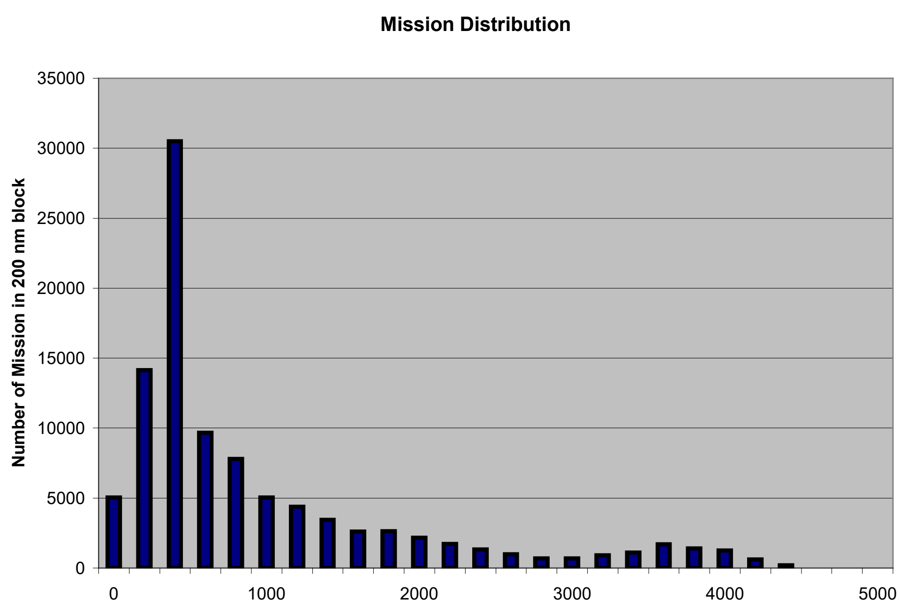


Figure 7. Mission distribution for 100,000 flights with a maximum range of 4,500 nm

It should be noted that this input value is the maximum range of the aircraft for a normal mission during revenue operation. It is not the absolute maximum range of the aircraft without a payload. This maximum range is also used, along with the cruise Mach number (section 4.1.2.1), to determine the maximum flight time. The flight time is also displayed in the Aircraft Data block of the User Inputs and Results worksheet.

To develop an overall mission profile for a particular flight, pre- and post-flight is added as ground time to each randomly selected flight time. The preflight ground time is set by the model as 30 minutes for short flights (less than 3 hours), 45 minutes for mid-length flights (between 3 and 4 hours), and 90 minutes for long flights (greater than 4 hours). The post-flight ground time is set by the model at 30 minutes for all flights. Flights performed by the model are restricted to no less than 15 minutes in duration.

#### *4.1.1.2 Number of engines*

The model uses the number of engines, along with individual flight mission lengths, to determine the appropriate amount of time for the aircraft to climb to its cruise altitude for each flight. Table 3 shows the amount of time for the aircraft to climb to cruise level that is used by the model based on flight mission length and number of engines. The model then uses this and the cruise altitude to determine the climb rate for each flight.

It should be noted that descent rates, unlike climb rates, are fixed in the model and do not vary based on airplane type. The descent time is calculated by the model using descent rates of 2500 ft/min down to 4000 ft and 500 ft/min from 4000 ft to touchdown.

Table 3. Time to climb to cruise altitude (in minutes)

| **Number of Engines** | **Mission Length (Percent of Absolute Maximum Range)** | | | | | |
|---|---|---|---|---|---|---|
| | <20% | <40% | <60% | <80% | <100% | 100% |
| 2 | 20 | 20 | 30 | 30 | 35 | 35 |
| 3 | 25 | 30 | 35 | 35 | 40 | 40 |
| 4 | 25 | 35 | 40 | 40 | 45 | 50 |

#### *4.1.1.3 Outside air temperature cutoff limit*

This cell allows the user to input a temperature cutoff point for the case where the operation of an aircraft is limited to a maximum temperature as specified in the Aircraft Flight Manual (AFM). In any case that the random ambient OAT is above this limit, the model, instead of using the OAT, will use the OAT cutoff limit temperature that was entered by the user. Operating in this manner simulates the aircraft waiting until the OAT cools to within the AFM limits.

If there is no cutoff temperature limit, the user should input into this cell a large number outside the range of ambient temperatures, such as 150°F, so that the OAT cutoff limit will not be activated.

### 4.1.2 Flight data

The flight data that are required to perform the Monte Carlo calculations are the cruise Mach number, altitude steps and tank ram recovery. These inputs are located in the Flight Data block of the User Inputs and Results worksheet, as shown in Figure 8.

| **Flight Data** | | **Tank Ram Recovery** | |
|---|---|---|---|
| **Cruise Mach Number** | **0.81** | **0** | **% of Ptotal** |
| **Cruise Altitude Steps** | **31000** | **ft** | |
| | **35000** | **ft** | |
| | **39000** | **ft** | |

Figure 8. Flight data block

#### *4.1.2.1 Cruise Mach number*

The cruise Mach number input is the Mach number at cruise altitude during typical revenue operation. The Mach number is used by the model to determine each flight profile and flight time.

#### *4.1.2.2 Cruise altitude steps*

This set of altitudes define the step cruise levels used by the program. For airplanes that typically do not use a three-step profile, all three values can be set to be equal. The climb between these steps is treated as instantaneous, as opposed to the climb rate used from takeoff to the first cruise level, as discussed in section 4.1.1.2. The way that the model treats the cruise altitude steps is broken into four categories:

- Flight Times Less Than 50 Minutes—The flight time is divided with 40% allocated for climb and 60% for descent. For very short flights, it is possible that the first altitude step will never be reached. Due to the variable climb rates (discussed in section 4.1.1.2), a short amount of cruise time is possible in these flights. In cases where this occurs, the first altitude step is used.
- Flight Times Between 50 and 100 Minutes—The flight cruises at the first altitude step and does not step to the other levels.
- Flight Times Between 100 and 200 Minutes—Two cruise step altitudes are used, with the step increase occurring midway through cruise time.

- Flight Times Over 200 Minutes—Three cruise step altitudes are used, with the cruise time equally split three ways.

For example, if 31-, 35-, and 39-thousand foot altitude steps were entered, possible flight profiles generated for short, medium-length, and long flights are shown in Figure 9. The 30-, 45-, and 90-minute preflight and 30-minute post-flight ground times (see section 4.1.1.1) are also included in Figure 9.

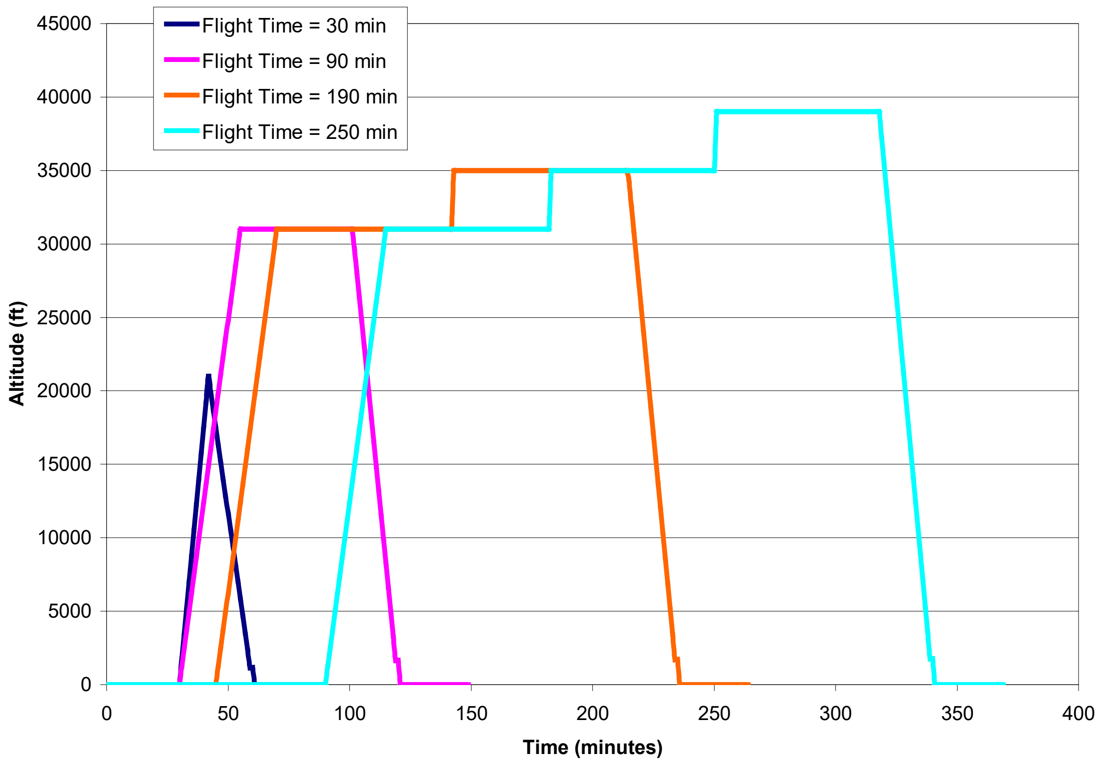


Figure 9. Possible flight profiles generated with 31-, 35-, and 39-thousand foot cruise altitude steps

For flights longer than two hours, the model uses a routine to introduce a different ambient temperature for the latter stages of flight and landing to replicate the aircraft flying into a new climate. This is performed by the model by ramping to the new temperature over a 45-minute period starting just after the midway point of the cruise cycle. Once at the new temperature (climate), the flight continues to land with a new ground ambient temperature.

#### *4.1.2.3 Tank ram recovery*

The ram recovery of the tank vent can have an effect on flammability exposure. As such, the user may input the tank ram recovery as a percentage of the total ram recovery pressure. If ram recovery is used, the input should be verified by analysis and flight test data.

### 4.1.3 Fuel tank usage data

Data concerning fuel tank usage that are required to perform the Monte Carlo calculations are the tank full and empty times and the engine start times. These inputs are located in the Fuel Tank Data block of the User Inputs and Results worksheet, as shown in Figure 10.

**Fuel Tank Usage Data**

| | | |
|---|---|---|
| Tank Full any time before | 610 | minutes before touchdown |
| Tank empty any time after | 500 | minutes before touchdown |
| Engines or equipment started at | 90 | minutes prior to takeoff |

Figure 10. Fuel tank usage data block

#### *4.1.3.1 Tank full and empty times*

The model uses the time prior to touchdown that the tank is full and empty to calculate the lapse rate for the fuel temperature. This lapse rate allows the calculation of the fuel temperature at each time step during the flight.

Figure 11 is a graphical representation of the tank full and empty times measured prior to touchdown for each tank type (center wing tank [CWT], auxiliary tank, and main tank with reserves) for a maximum-range flight.

For a CWT that is used first, the tank full time would be the maximum flight time, while the tank empty time would be the time prior to touchdown that the tank becomes empty. Therefore, for shorter flights, the tank would be empty at takeoff.

For an auxiliary tank, which is used next in the flight cycle, the tank will remain full in the early part of long flights; thus, the tank full time is the time prior to touchdown that the fuel starts to be depleted. Similar to the CWT, the tank empty time would be the time prior to touchdown that the tank becomes empty. However, it should be noted that for a main tank with reserves the term empty is defined as the tank's reserve level, not an actual empty tank. As such, the tank empty time for a main tank with reserves would be 0 minutes prior to touchdown, not the time that the tank runs dry. Similarly, the thermal data (see section 4.1.5) for an empty main tank with reserves would be for a tank at reserve levels, not empty levels.

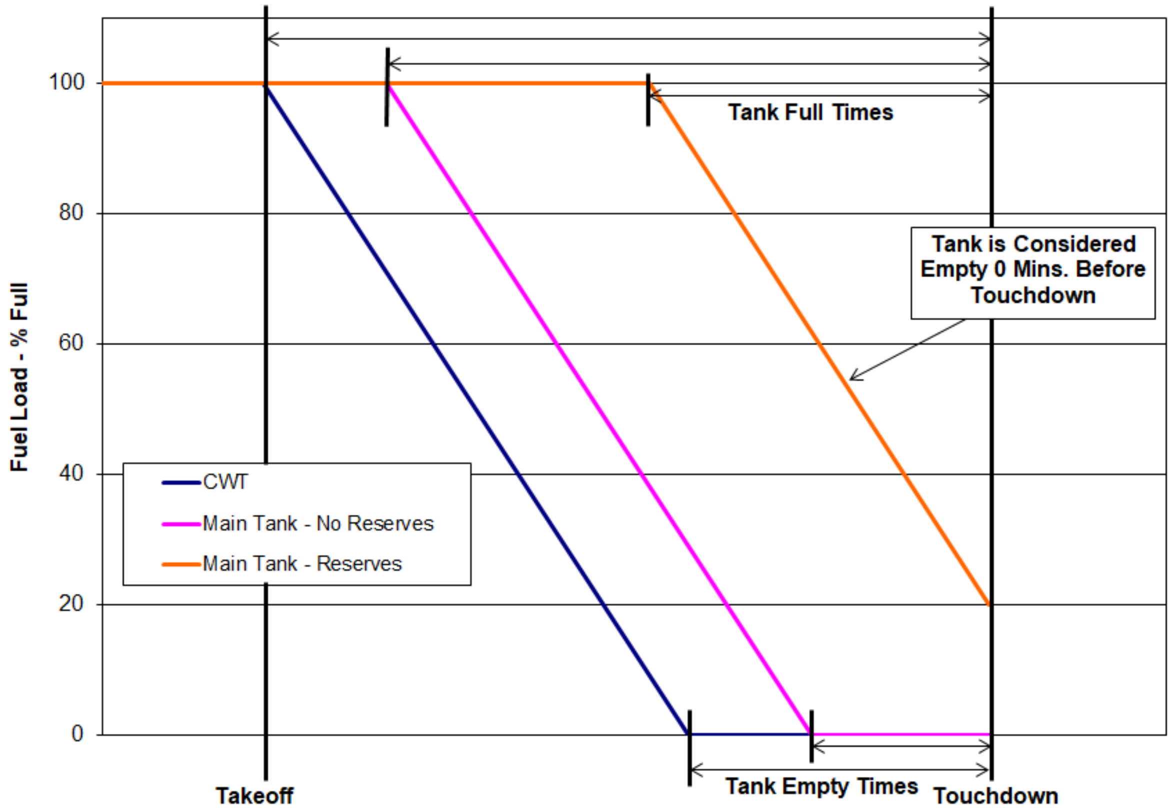


Figure 11. Usage of tank full and empty times for a maximum-range flight

*4.1.3.2 Engine start time*

The program assumes that normal heat loads (e.g., environmental control systems) are operating from the start of the flight. The engine start time cell gives the option to simulate a fuel tank with additional heat input from engines and/or systems. For example, some airplanes have hydraulic heat exchangers, engine oil and fuel recirculation features, or other systems that add heat to the tank.

The input value for this cell is the time prior to takeoff that the engines and/or systems start adding heat to the tank. This value will vary by airplane utilization. For example, longer-range airplanes typically have much longer times, where the engines and/or systems function on the ground. The change in actual thermal effects from operating these engines and/or systems is entered in as a difference in thermal time constants from when the systems are inoperable (see section 4.1.5.2).

### 4.1.4 Body tank input data

The body tank data needed for the model consists of five inputs that allow the user to study tanks that are completely enclosed in the fuselage, or a similar container, with no direct cooling to ambient air. It allows the calculated temperatures of the fuel tank to be controlled by the temperature of the compartment in which it is contained. In addition, the tank may be set to be pressurized in flight, or vented to ambient pressure based on the design of the tank. It should be noted that if the tank of interest is not a body tank, all five input cells should be set to zero. These inputs are located in the Body Tank Input Data block within the User Inputs and Results worksheet, as shown in Figure 12.

| **Body Tank Input Data** | | **Set all values to zero if tank is not a body tank.** |
|---|---|---|
| Tank in the fuselage with no cooling from outside air | 0 | 1=Yes, 0=No |
| Tank pressurized in flight, | 0 | 1=Yes, 0=No |
| Pressure differential relative to ambient | 0 | psi |
| Tank is pressurized | 0 | minutes before takeoff |
| Temperature of compartment surrounding tank | 0 | Deg F |

Figure 12. Body tank input data block

The first input asks the user if the tank of interest is located in the fuselage with no direct cooling from ambient air (i.e., if it is a body tank). The user should enter the numeral 1 here if it is a body tank, and 0 if it is not. Next, the user must indicate if the tank is pressurized in-flight. Again, the user should enter the numeral 1 if it is, 0 if it is not. If 0 is entered in this cell, the tank pressure will then be set to the altitude of the airplane at each time step.

If the tank is pressurized in-flight, the user must then enter the pressure differential, relative to ambient pressure, that the tank is pressurized to. The user must also input the time prior to takeoff at which the tank pressurization system becomes active. These inputs should be established based on validated values, because the pressure altitude of the tank will vary based on tank vent designs and will have a significant effect on flammability.

The final piece of required body tank information is the temperature of the compartment surrounding the tank. At the start of the flight (i.e., time = 0), the fuel tank temperature is set to ambient to represent the fuel loaded from the ground supply. The fuel will then cool (or heat) to the surrounding compartment temperature using the exponential time constants for ground conditions discussed in section 4.1.5.2.

### 4.1.5 Fuel tank thermal data

The fuel tank thermal data required to perform the Monte Carlo calculations are the temperature differential to both the ambient and total air temperature (TAT) as well as a number of exponential time constants.

The model assumes that the fuel is loaded at the start of the mission (i.e., time = 0). The time constants required are then used to determine the manner in which the fuel will heat or cool in response to its surroundings. These time constants are inputted for a near empty and a near full tank, both on the ground and in-flight. In addition, for ground conditions, these values must be entered with and without the engines running. The tank thermal data is critical to achieving accurate results with the FTFAM, and as such, all inputs must be validated through flight test data or a thorough thermal analysis.

These inputs are located in the Fuel Tank Thermal Data block within the User Inputs and Results worksheet, as shown in Figure 13.

| | | | |
|---|---|---|---|
| **Tank Thermal Data** | | | |
| **The fuel is assumed to be loaded at ambient temperature** | | | |
| **Tank Constants, Ground Conditions:** | **Eng.OFF** | **EngON** | |
| **Equilibrium DeltaTemp** | **60** | **60** | **Deg F** |
| **Exponential time Constant -Tank near Empty** | **200** | **200** | **Minutes** |
| **Exponential time Constant -Tank near Full** | **400** | **400** | **Minutes** |
| **Tank Constants, Flight Conditions:** | | | |
| **Equilibrium DeltaTemp** | **60** | | **Deg F** |
| **Exponential time Constant -Tank near Empty** | **200** | | **Minutes** |
| **Exponential time Constant -Tank near Full** | **400** | | **Minutes** |

Figure 13. Fuel tank thermal data block

#### *4.1.5.1 Temperature differential relative to ambient and TAT*

The Equilibrium DeltaTemp input cell is the fuel temperature differential relative to ambient temperature that will be reached given sufficient time. This value must be determined from a thermodynamic analysis of the tank and the surrounding systems. In addition, this value must be entered with and without the engines and/or systems running to better represent the heat input to the fuel tank from the engines or other equipment.

Similarly, for flight conditions, the user must input the fuel temperature differential relative to the TAT that will be reached given sufficient time. If the tank is in the fuselage with no cooling from outside air, then this input would be relative to the surrounding compartment temperature.

It is assumed that the flight thermal data entered includes heat input from the engines and systems.

#### *4.1.5.2 Exponential time constants*

There are six exponential time constants required as input, which define how the fuel in the tank heats or cools in response to heat input:

- Ground condition, tank near empty, engines off
- Ground condition, tank near empty, engines on
- Ground condition, tank near full, engines off
- Ground condition, tank near full, engines on
- In-flight condition, tank near empty
- In-flight condition, tank near full

The fuel is assumed to heat or cool according to a normal exponential transition as shown in equation 1, where *t* is the time constant at the current time step and $T_{final}$ is the long-term equilibrium fuel temperature (discussed in section 4.1.5.1). Additionally, the model assumes a linear change in time constants as fuel is burned and the tank moves from a full to empty state.

$$\frac{T_{fuel,i} - T_{fuel,i-1}}{T_{final} - T_{fuel,i-1}} = 1 - e^{-t/\tau} \qquad 1$$

All time constants must be determined either by flight test or by a thermodynamic analysis backed up by flight tests. If, based on this data, this exponential transition does not satisfactorily model the fuel's temperature change, the user does have the ability to modify the model's code to allow for a more accurate representation of the fuel's temperature profile. This and other allowable code modifications are discussed in section 5.

### 4.1.6 Multiflight Monte Carlo data

The required data specific to the multiflight Monte Carlo analysis are the number of flights to be performed, whether or not the user would like to freeze the random numbers generated by the Monte Carlo analysis for developmental purposes, and whether or not the user would like to perform an analysis of flights operating on warm days only. These inputs are located in the Multiflight Monte Carlo Data Block within the User Inputs and Results worksheet, as shown in Figure 14.

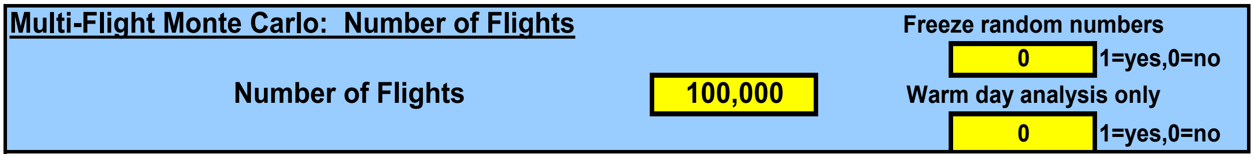


Figure 14. Multi-flight Monte Carlo data block

*4.1.6.1 Number of flights*

Due to the nature of the Monte Carlo analysis, the model's accuracy is increased as the number of flights is increased. As such, in order for the Monte Carlo analysis to be valid for showing compliance with flammability requirements, a minimum number of flights must be performed to ensure the applicable flammability limits are met. Table 4 shows the minimum number of flights and the maximum acceptable levels to meet requirements for 3% and 7% flammability exposure requirements (as set forth in the FRM Notice of Proposed Rulemaking).

Once the number of flights to perform has been determined and it, along with all other inputs, has been entered in the User Inputs and Results worksheet, the Run Monte Carlo button should be pressed to initiate the Monte Carlo analysis. Viewing and interpreting these results will be discussed in section 4.4.

Table 4. Minimum number of flights and acceptable level necessary to meet

| Minimum Number of Flights in Monte Carlo Analysis | Maximum Acceptable Monte Carlo Average Fuel Tank Flammability Exposure (%) to Meet 3% Requirements | Maximum Acceptable Monte Carlo Average Fuel Tank Flammability Exposure (%) to Meet 7% Requirements |
|---|---|---|
| 10,000 | 2.91 | 6.79 |
| 100,000 | 2.98 | 6.96 |
| 1,000,000 | 3.00 | 7.00 |

*4.1.6.2 Random number freeze*

As an option, for development purposes, the user can freeze the random numbers generated by the model by entering a 1 in this input box. This will force the model to use the same set of random numbers during each set of analyses, allowing the user to vary input parameters to better understand their sensitivity effect on the flammability exposure of the fuel tank. This option must be turned off, by entering a 0 in the input box, for final certification analysis.

*4.1.6.3 Warm-day analysis*

As an additional option, the user has the ability to perform a warm-day flammability analysis. By entering a 1 in this input box, the model will analyze only those flights for which the ground ambient temperatures are above 80°F. This allows the user to further analyze the flammability exposure of a particular fuel tank under what would be considered warm-day operations.

## 4.2 Monte Carlo analysis operation when a flammability reduction means is used

When a Monte Carlo analysis is being performed, and an FRM is being used, the user must enter some additional inputs. The effectiveness of an FRM is broken into two parts: (1) the reliability effect of the FRM and (2) the performance effect of the FRM.

Reliability effects of an FRM are determined from analyses of potential malfunctions of the FRM, which would make the system ineffective. To determine these effects, the user must input the following pieces of data into the FRM worksheet in addition to all inputs previously discussed:

- Mean Time Between Failure (MTBF)—This is the number of hours that the FRM is expected to be operational between failure events.
- Failure Detection Capability—The input needed here is the expected number of flights that it will take before the FRM system failure is detected. This value depends on system monitoring and information display as well as operational procedures such as frequency of FRM maintenance checks.
- Minimum Equipment List (MEL) Assumption—The value entered here is the average expected time, in-flight hours, that it will take for the FRM system to be restored once a failure has been detected.

The values used for these three inputs must be shown to be accurate by analysis and testing of the FRM and are entered as inputs on the FRM worksheet.

The user should note that once an FRM evaluation has been performed (cases run), the reliability inputs can be changed, and the results will be displayed without needing to perform the FRM evaluation again.

The performance effects of an FRM relate to the FRM's ability to maintain a nonflammable ullage while operating as expected. The program, as written, contains code to replicate the use of a generic ground-based inerting system. The user must replace this code with programming based on the exact FRM system intended to be used. Performance of the system under all flight conditions must be taken into account in writing this portion of the code and must be shown to be accurate by analyzing and testing the FRM. Further details concerning writing and inserting this code are discussed in section 5.3.

Once the performance aspect of the FRM has been coded and inserted into the model, the user must enter all FRM reliability data and the data values discussed in section 3.1. Once this data is entered fully, the user presses the Run FRM Evaluation button on the FRM worksheet to initiate the FRM analysis. It should be noted that pressing the Run Monte Carlo button on the User Inputs and Results worksheet will perform only a flammability analysis (i.e., with no FRM) regardless of the inputs made on the FRM worksheet.

## 4.3 Single-flight User Inputs

Located in the Single-flight Condition block of the User Inputs and Results worksheet is a set of values that allow the user to input a specific set of conditions for a single flight and examine the results. This could be useful in a troubleshooting operation, as well as looking at various trends, and the effects of changing certain variables. The Single-Flight Condition block is shown in Figure 15.

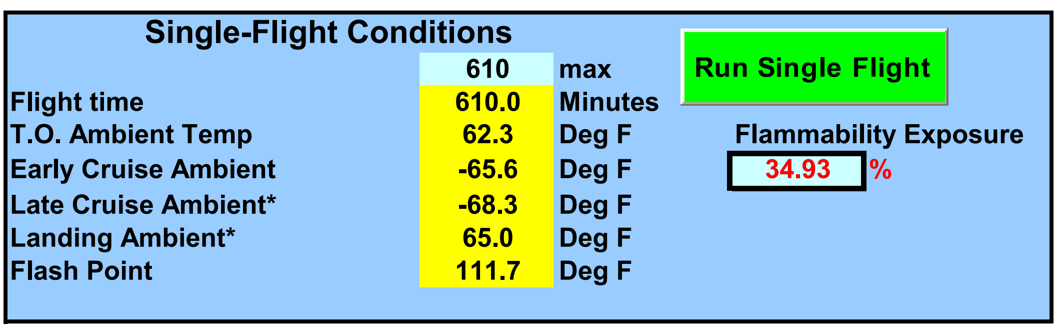


Figure 15. Single-flight condition data block

All inputs needed for the Monte Carlo analysis (discussed in sections 4.1.1 through 4.1.6) are still necessary, except for the number of flights. In addition to these inputs, the user must enter in this data block information specific to the single flight being performed. The needed inputs are the total flight time, ambient temperature (on the ground and at cruise), and the fuel’s flashpoint. Once these inputs have been entered, the Run Single Flight button can be pressed to initiate the flight analysis. It should be noted that if the user should enter a takeoff ambient temperature that is above the OAT limit, the model will use the OAT limit as the ambient temperature and a warning will display to alert the user of this condition.

As an option, rather than inputting these data for a single flight, the user may also want to run a selected flight from the Monte Carlo analysis to view its results in detail. This can be done by entering the selected case number from the Monte Carlo analysis (found in the Summary of n Cases worksheet under column M) in the block shown in Figure 16 and pressing the corresponding Run Selected Flight button.

Viewing and interpreting the results of these individual flights, as well as the Monte Carlo analysis results, is discussed in detail in section 4.4.

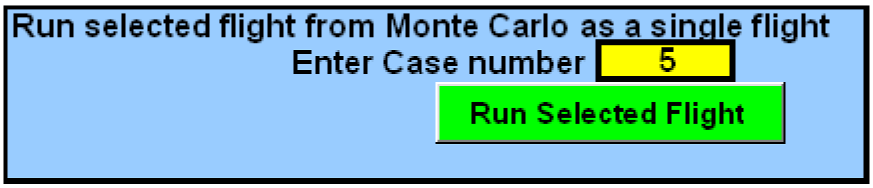


Figure 16. Monte Carlo single-flight condition data block

## 4.4 Viewing and interpreting results

Upon completion of either the Monte Carlo analysis or a single-flight analysis, the model will bring the user to the corresponding results page. For a Monte Carlo analysis, it will be the Summary of n Cases worksheet, and for a single-flight analysis it will be the Result Plots worksheet.

### 4.4.1 Monte Carlo flammability analysis

The Summary of n Cases worksheet contains the results of the Monte Carlo analysis in a tabular format with a graphical representation of the data displayed in the User Inputs and Results worksheet.

The table on the Summary of n Cases worksheet contains all the vital information for each flight simulated and sorts them by the percentage of flight time that they were flammable in decreasing order. Data given in this table includes the following, as calculated by the Monte Carlo model: (1) preflight ground time, (2) flight time, (3) ambient temperature, (4) cruise temperature, (5) fuel flashpoint temperature, (6) the amount of time the flight was flammable, (7) the percentage of flight time that the flight was flammable, and (8) whether or not the FRM was achieving a nonflammable ullage at each time increment.

A sample of the results is shown Figure 17. This graph depicts the percentage of flight time that the tank was flammable for each flight performed in the Monte Carlo analysis, sorted in decreasing order. Also displayed on this graph is a summation of the total percentage of flammable flight time for all flights, termed the Fleet Average Flammability Exposure Percentage. This value is the amount of time the tank was determined to be flammable divided by the total flight time amongst all the performed flights, which is the most important number in determining the flammability exposure of a given fuel tank.

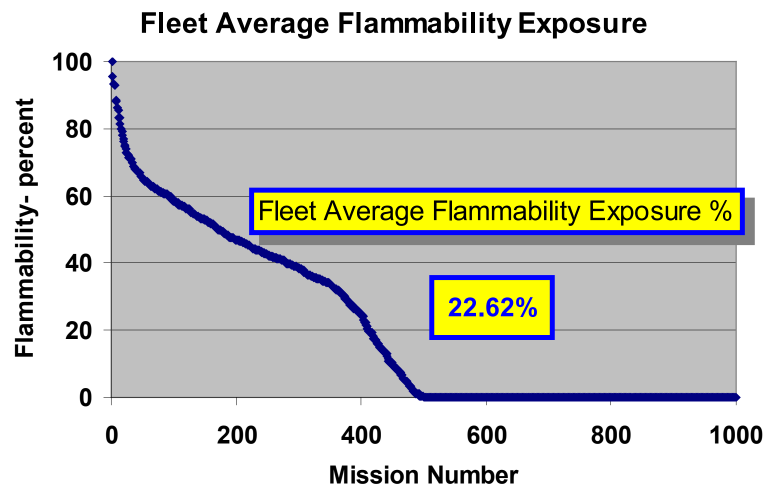


Figure 17. Summary of multi-flight Monte Carlo analsysis results

It should be noted that for Monte Carlo analyses consisting of greater than 5000 flights, both the table of all flights on the Summary of n Cases worksheet and the results graph on the User Inputs and Results worksheet will appear blank, because Excel cannot save all the necessary data.

Directly beneath this graph, the Monte Carlo analysis results are further broken down in the warm day operations results table, as shown in Figure 18. This table displays the results for all flights during which the ambient ground temperature before takeoff was greater than 80°F. The table shows the total flight time, total time during which the fuel tank was flammable, and the resulting flammability exposure percentage for all flights during warm days (i.e., ambient temperature greater than 80°F). In addition, this table displays the corresponding data for each phase of flight (i.e., ground, climb, each of the three cruise levels, descent, and taxi-in.) The table in Figure 18, in this instance, shows the fuel tank was flammable for 72.7% of the time during warm day operations.

If the user, after evaluating the results on this worksheet, would like to further evaluate the data from a particular flight contained in the Monte Carlo analysis table, the flight case number can be entered and performed as a single-flight scenario, as discussed in section 4.3.

| Warm Day (Above 80F) Results | | | |
|---|---|---|---|
| | total time | flam time | % Flam |
| ground | 263 | 95 | 36.1 |
| climb | 160 | 144 | 90.0 |
| Cruise1 | 422 | 363 | 86.0 |
| Cruise2 | 307 | 274 | 89.3 |
| Cruise3 | 0 | 0 | 0.0 |
| descent | 164 | 133 | 81.1 |
| taxi-in | 232 | 116 | 50.0 |
| total | 1548 | 1125 | 72.7 |

Figure 18. Warm day operations results table

### 4.4.2 Monte Carlo with FRM analysis

The results from an FRM analysis are displayed in the FRM worksheet in several different tables. The first of these, shown in Figure 19, displays the baseline flammability data (i.e., with no FRM) and the corresponding flammability data due to reliability failures of the FRM. Values for both of these are shown for the fleetwide average and for the climb and cruise portions of flight for those flights where ground ambient conditions were greater than 80°F.

| Reliability Effect of FRM on Tank Flammability | | |
|---|---|---|
| Baseline Fleet Average Flammability Exposure | 13.2 | % |
| Above 80 Deg F | | |
| Ground flammability | 0.0 | % |
| Climb flammability | 73.3 | % |
| FRM capability (Reliability and MEL cases only) | | |
| Fleet Average Flammability Exposure | 0.86 | % |
| Above 80 Deg F | | |
| Ground flammability | 0.00 | % |
| Climb flammability | 4.79 | % |

Figure 19. FRM table displaying baseline and reliability effects data

There are two larger tables below that allow users to analyze the performance effect of the FRM, as shown in Figure 20. Both tables display flammability data for each phase of flight (i.e., ground, climb, each cruise altitude, descent, and taxi-in), as well as the total time of flight. Data are again presented for the fleetwide average and for flights where ground ambient conditions were greater than 80°F.

**Baseline, NO FRM results**
**Summary data for specific portions of the flights**

**All flights**

| | total time | flam time | % Flam | Contribution to whole |
|---|---|---|---|---|
| ground | 300 | 0 | 0.0 | 0.0 |
| climb | 270 | 28 | 10.4 | 13.3 |
| Cruise1 | 361 | 108 | 29.9 | 51.2 |
| Cruise2 | 168 | 62 | 36.9 | 29.4 |
| Cruise3 | 0 | 0 | 0.0 | 0.0 |
| descent | 205 | 13 | 6.3 | 6.2 |
| taxi-in | 300 | 0 | 0.0 | 0.0 |
| total | 1604 | 211 | 13.2 | 100.0 |

**Flights above 80 Deg F**

| | total time | flam time | % Flam |
|---|---|---|---|
| ground | 31 | 0 | 0.0 |
| climb | 30 | 22 | 73.3 |
| Cruise1 | 47 | 47 | 100.0 |
| Cruise2 | 0 | 0 | 0.0 |
| Cruise3 | 0 | 0 | 0.0 |
| descent | 19 | 13 | 68.4 |
| taxi-in | 29 | 0 | 0.0 |
| total | 156 | 82 | 52.6 |

**FRM Performance results**
**Summary data for specific portions of the flights**

**All flights**

| | total time | flam time | % Flam | Contribution to whole |
|---|---|---|---|---|
| ground | 300 | 0 | 0.0 | 0.0 |
| climb | 270 | 0 | 0.0 | 0.0 |
| Cruise1 | 361 | 0 | 0.0 | 0.0 |
| Cruise2 | 168 | 0 | 0.0 | 0.0 |
| Cruise3 | 0 | 0 | 0.0 | 0.0 |
| descent | 205 | 0 | 0.0 | 0.0 |
| taxi-in | 300 | 0 | 0.0 | 0.0 |
| total | 1604 | 0 | 0.0 | 0.0 |

**Flights above 80 Deg F**

| | total time | flam time | % Flam |
|---|---|---|---|
| ground | 31 | 0 | 0.0 |
| climb | 30 | 0 | 0.0 |
| Cruise1 | 47 | 0 | 0.0 |
| Cruise2 | 0 | 0 | 0.0 |
| Cruise3 | 0 | 0 | 0.0 |
| descent | 19 | 0 | 0.0 |
| taxi-in | 29 | 0 | 0.0 |
| total | 156 | 0 | 0.0 |

Figure 20. FRM table summarizing performance effects on each phase of flight

The last table, as shown in Figure 21, provides the user with a summary of the FRM effectiveness. It includes the percentage of flammable time due to reliability factors as well as performance factors and the total fleet effectiveness due to the FRM. It also includes the total flammable percentage times (due to both reliability and performance factors) for ground and climb times during flights where ground ambient conditions were greater than 80°F.

**FRM Summary**

| | |
|---|---|
| **Fleet Average (all flights)** | |
| **Fleet Average Flamability Performance** | **100.0** |
| **Fleeet Average Reliability Effect** | **1.59** |
| **Total Fleet Efffectiveness** | **101.59** |
| **Above 80 Deg F days** | |
| **Flammability, Ground Conditions** | **12.5** |
| **Flammability, Climb Conditions** | **71.6** |

Figure 21. The FRM table displaying FRM effectiveness

In addition to this data, the baseline Monte Carlo results (i.e., with no FRM) are displayed in the Summary of n Cases and User Inputs and Results worksheets, as discussed in section 4.4.1.

### 4.4.3 Single-flight flammability analysis

The results from a single-flight analysis are shown on the Single-Flight worksheet in two different graphical formats, time- and altitude-based plots.

The time-based plot, as shown in Figure 22, depicts the TAT, fuel temperature, LFL temperature, and UFL temperature as a function of time. In addition, along the secondary y axis, the red bar indicates the percent flammability exposure time for that particular flight. This percent flammability number is also displayed in the User Inputs and Results worksheet in the Single-flight Condition block. The points on this plot where the bulk average fuel temperature falls between the LFL and UFL temperatures are the portions of the flight when the tank was flammable. The points where the bulk average fuel temperature falls below the LFL temperature are the portions of the flight when the tank was lean. Although no points are shown in Figure 22, if the fuel temperature was above the UFL temperature, it would indicate that the tank was too rich. The TAT is also displayed in this plot for reference to the outside ambient conditions of the airplane.

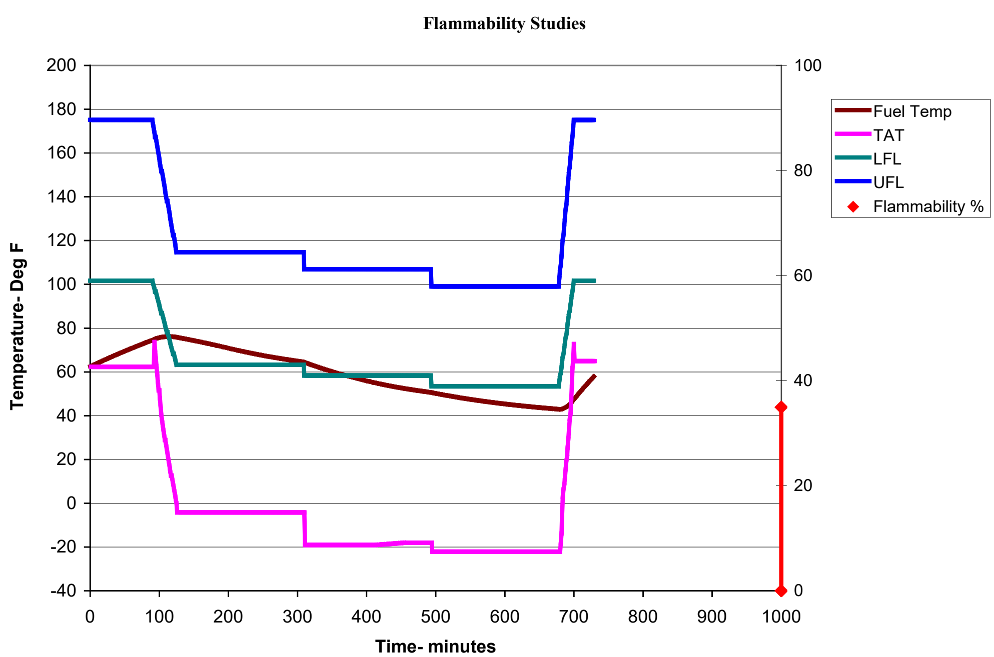


Figure 22. Single-flight results—time-based plot

The altitude-based plot on this worksheet, shown in Figure 23, displays the same information, but in terms of altitude. Again, points where the bulk average fuel temperature falls between the LFL and UFL temperatures are the portions of the flight that the tank was determined to be flammable. Similarly, the points to the left of the LFL temperature indicate that the tank was lean, and points to the right of the UFL temperature indicate that the tank was rich. Some small amount of perturbation from a smooth curve is evident in this figure due to rounding of temperatures to every 1000 ft.

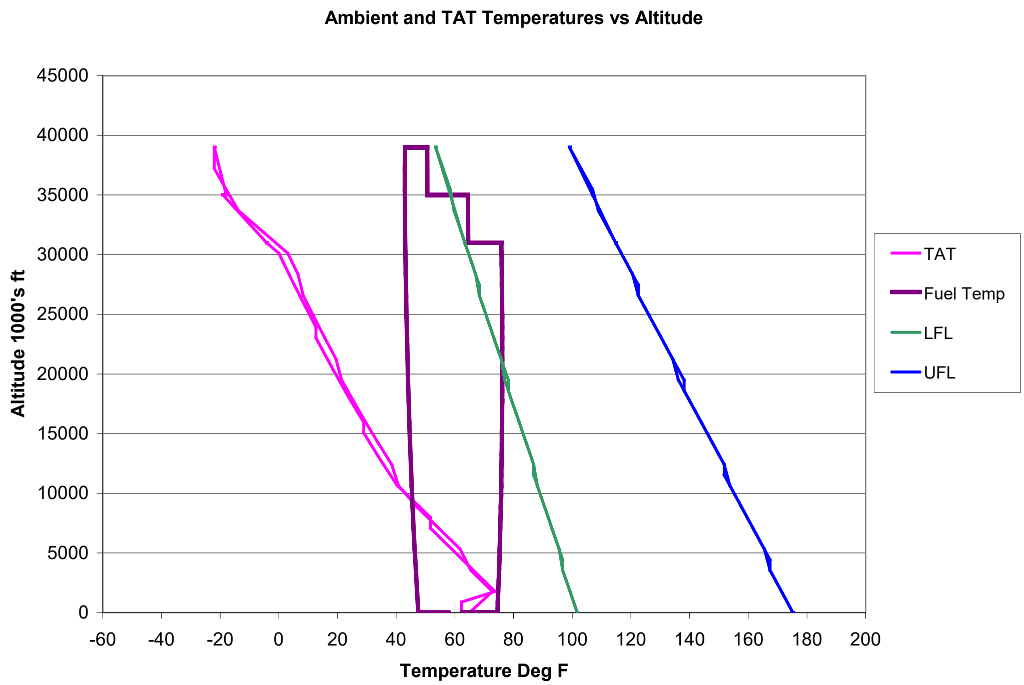


Figure 23. Single-flight results—altitude-based plot

# 5 Permissible user modification of the code

Although the FTFAM is a very effective and versatile tool, there are certain aspects of the model's code that may need to be modified by the user based on acquired aircraft data. The permissible code changes within the model include how the model computes the tank thermal effects, fuel tank usage, and FRM performance effects. Anything outside these computations must not be modified in any way. To adhere to the FAA standards, any changes must be sufficiently documented and validated with flight test data or analysis backed up by flight data.

## 5.1 Fuel tank thermal effects

The thermal behavior of the fuel tank due to its surroundings, as discussed in section 4.1.5, is based on calculations using the fuel temperature differential relative to the ambient temperature and TAT, as well as several exponential time constants. If flight test data or a detailed analysis of the fuel tank's thermal behavior shows that this method cannot yield an accurate representation of the actual fuel tank's fuel temperature profile, then modification of the code is warranted.

## 5.2 Fuel tank usage

The usage of fuel within the tank is calculated at each time increment by the Monte Carlo model using a linear decay from the tank full to tank empty times, as discussed in section 4.1.3.1. Figure 11 displayed this type of fuel usage and it is repeated here for reference as Figure 24. This representation of a fuel tank's usage is meant as a general profile for a generic tank. Depending on the aircraft and fuel tank configuration, such as shifting fuel between tanks, actual fuel usage may be very different from this.

In this situation, the user is able to modify this portion of the code to better represent the actual fuel tank usage. Changes to this portion of the code must be shown, either by flight test data or a detailed analysis of the tank's usage of fuel backed up by data, to provide an accurate representation of fuel usage by the tank in question.

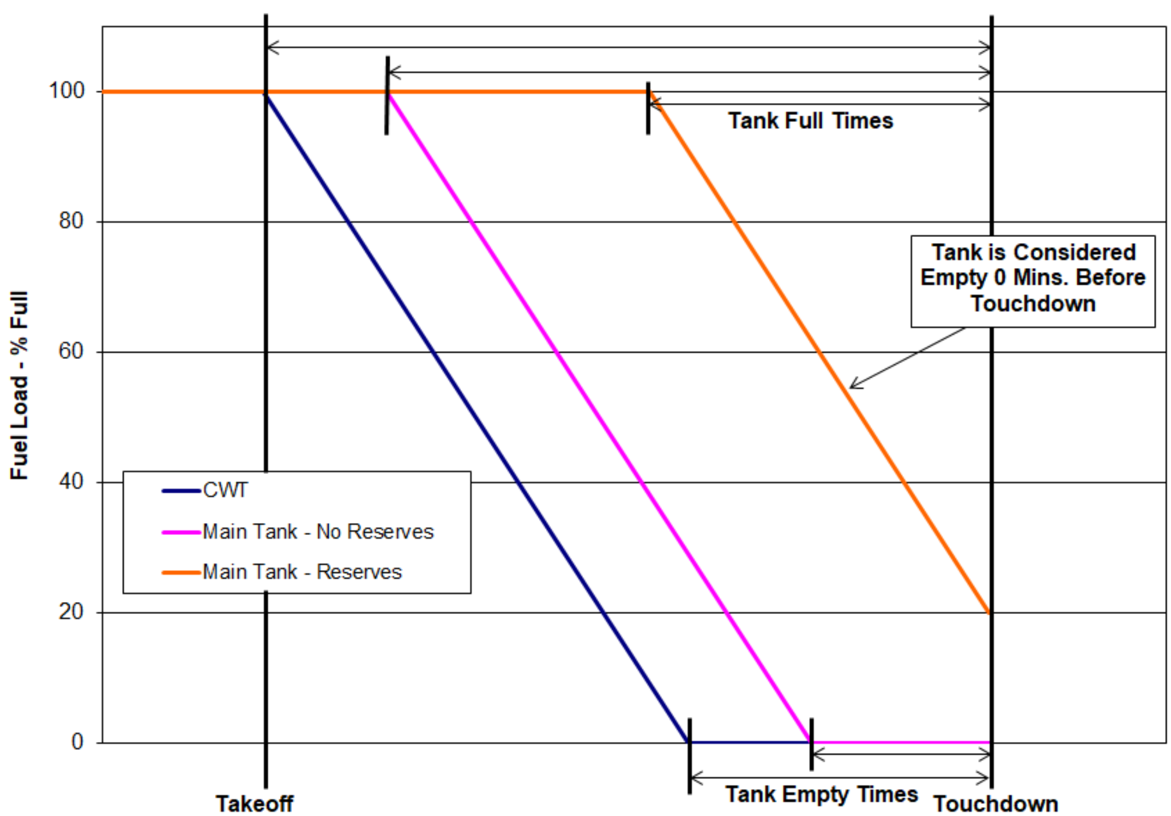


Figure 24. Model's usage of tank full and empty times

## 5.3 Using a flammability reduction means

When an FRM is being evaluated during a Monte Carlo flammability analysis, it is necessary for the user to ensure that the code of the model reflects any performance effects of the FRM. The performance effects of an FRM relate to the FRM's ability to maintain a nonflammable ullage while operating as expected, and should be programmed separately from the reliability effects—those due to the FRM being inoperative due to a system or part failure. The program, as written, contains code to replicate the use of a generic ground-based inerting system. The user must replace this with code based on the exact FRM system intended to be used.

Performance of the system under all flight conditions must be taken into account in writing this portion of the code, and must be shown to be accurate by analyzing and testing the FRM. If the FRM module needs additional data beyond what is provided by the Monte Carlo model, the user is required to compute that information and also show it to be accurate by analysis and/or testing.

The output of the added FRM module should be the parameter FRMyesno, which is set to 0 at each time increment when the FRM is not maintaining the tank from being flammable and is set to 1 at each time increment when it is sustaining a nonflammable ullage.

The placement of this FRM module is clearly identified in the model's code with a comment statement that reads "TEST FRM ONLY". The code that follows this comment statement is the current default FRM performance code that replicates the use of a generic ground-based inerting system. This code must be replaced with the user's own code reflecting the specific FRM in use. This section of the code is shown below for reference, beginning with the comment statement:

```
'====================TEST FRM ONLY =====================
'This is a very simple FRM to test the FRM module and data collecting code. The FRM is
'assumed to be effective after t=10 min and to be effective until the end of cruise for all flights.
If FRMonoff = 0 Then
flammyesnoFRM = flammyesno
GoTo 299
End If
If time1 < phase6 And time1 > 10 Then
FRMyesno = 1
End If
```

When using a nitrogen inerting system as the FRM, it is necessary to take into account oxygen evolution, the release of oxygen from the fuel, when performing a flammability analysis.

Air evolution in fuel is driven by how much air is dissolved in the fuel, what keeps it dissolved, and what drives it out. Air dissolves in fuel until the partial pressure of the dissolved air equals the local partial pressure of the ullage. There is an Ostwald Coefficient that defines the amount of gas the fuel can absorb to reach equilibrium, which varies with temperature. Once the gas is dissolved in the fuel, it will remain dissolved until the ullage partial pressure is changed, with surface tension of the fuel helping to keep the gas in solution. This leads to a condition of supersaturation, where the partial pressure of the dissolved gas can be significantly higher than the ullage partial pressure. This can result in a condition where a triggering mechanism can break the surface tension effect, causing a large amount of gas to be liberated quickly.

The problem is approached by determining the partial pressure of oxygen in both the ullage and the fuel, assuming total saturation of the fuel at the beginning of the flight. It is then assumed that the oxygen release, or absorption, is driven by the difference of the two partial pressures and that there is some exponential mass transfer time constant to reach equilibrium. An effect for supersaturation, where the concentration of oxygen has not reached equilibrium with that of the ullage, can also be added by keeping the gas dissolved in the fuel (i.e., maintaining 100% saturation) until a certain pressure differential is achieved.

The user should replicate this effect by assuming 100% saturation at normal atmospheric conditions at refueling (i.e., the fuel is saturated with air at 21% $O_2$ and has not been prescrubbed of $O_2$). Under quiescent conditions, with no altitude changes, the time constant used for gas transfer is 3500 minutes. For climb conditions, no mass transfer takes place until above 15,000 ft, and during the remainder of climb the time constant used is 100 minutes.

# 6 Documentation and validation of code modifications

All modifications to the model's code must be thoroughly documented and validated both through detailed analysis and flight test data. The modified code must be shown to provide an accurate representation of the aircraft's systems and their interaction with the environment. Guidance for documenting and validating fuel tank thermal models is provided in Appendix 1 of AC 25.981-2A. Similar documentation and validation is required for all other code modifications.

# A The fuel tank flammability assessment method programming code

Below is a copy of the programming code of the FTFAM. This code can be accessed within the FTFAM through the macro feature of Microsoft Excel. The programming code is presented exactly as entered in the Excel macro, and so formatting and/or spelling errors may exist. There is a horizontal line inserted between each sub-routine to aid the reader.

```
Option Explicit

Private Sub RunMCBtn_Click()
'Unlock sheets (added by S. Summer 01/31/2007)
Sheets("User inputs and results").Unprotect
Sheets("FRM").Unprotect
Sheets("Single Flight").Unprotect
Sheets("Summary of n Cases").Unprotect
Sheets("Internal Calculations").Unprotect
Sheets("Internal Calc 2").Unprotect
Sheets("Internal Calc 3").Unprotect
Sheets("Internal Calc 4").Unprotect
Sheets("Internal Calculations").Cells(2, 11) = Sheets("User inputs and results").Cells(49,
6)
Sheets("Internal Calc 4").Cells(2, 10) = 0
montecarlo

'Lock sheets (added by S. Summer 01/31/2007)
Sheets("User inputs and results").Protect
Sheets("FRM").Protect
Sheets("Single Flight").Protect
Sheets("Summary of n Cases").Protect
Sheets("Internal Calculations").Protect
Sheets("Internal Calc 2").Protect
Sheets("Internal Calc 3").Protect
Sheets("Internal Calc 4").Protect

End Sub
```

---

```
Private Sub RunFRMBtn_Click()

'This module runs the montecarlo analysis with the FRM turned on.
'It captures both FRM "ON" and FRM "OFF" data in order to evaluate the FRM relative
to FAR 25.981
'Unlock sheets (added by S. Summer 01/31/2007)
Sheets("User inputs and results").Unprotect
```

```
Sheets("FRM").Unprotect
Sheets("Single Flight").Unprotect
Sheets("Summary of n Cases").Unprotect
Sheets("Internal Calculations").Unprotect
Sheets("Internal Calc 2").Unprotect
Sheets("Internal Calc 3").Unprotect
Sheets("Internal Calc 4").Unprotect

Sheets("Internal Calculations").Cells(2, 11) = Sheets("User inputs and results").Cells(49,
6)
Sheets("Internal Calc 4").Cells(2, 10) = 1
montecarlo

'Lock Sheets (added by S. Summer 01/31/2007
Sheets("User inputs and results").Protect
Sheets("FRM").Protect
Sheets("Single Flight").Protect
Sheets("Summary of n Cases").Protect
Sheets("Internal Calculations").Protect
Sheets("Internal Calc 2").Protect
Sheets("Internal Calc 3").Protect
Sheets("Internal Calc 4").Protect

Sheets("FRM").Select
End Sub
```

---

```
Private Sub ReturntoInputBtn_Click()
Sheets("User inputs and Results").Select
Sheets("User inputs and Results").Range("a1").Select
End Sub
```

---

```
Private Sub CheckBox1_Click()
Sheets("INTRO").Unprotect
Sheets("INTRO").Select
If Sheets("INTRO").Rows("9:1000").Hidden = False Then
   Sheets("INTRO").Rows("9:1000").Hidden = True
   Range("D1").Select
   Else
   Sheets("INTRO").Rows("9:1000").Hidden = False
   Range("D15").Select
End If
Sheets("INTRO").Protect
End Sub
```

---

```
Private Sub RunSingleFltBtn_Click()
```

```
'Unlock sheets (added by S. Summer 01/31/2007)
Sheets("User inputs and results").Unprotect
Sheets("FRM").Unprotect
Sheets("Single Flight").Unprotect
Sheets("Summary of n Cases").Unprotect
Sheets("Internal Calculations").Unprotect
Sheets("Internal Calc 2").Unprotect
Sheets("Internal Calc 3").Unprotect
Sheets("Internal Calc 4").Unprotect
Sheets("Internal Calculations").Cells(2, 11) = 1
montecarlo
'Lock sheets (added by S. Summer 01/31/2007)
Sheets("User inputs and results").Protect
Sheets("FRM").Protect
Sheets("Single Flight").Protect
Sheets("Summary of n Cases").Protect
Sheets("Internal Calculations").Protect
Sheets("Internal Calc 2").Protect
Sheets("Internal Calc 3").Protect
Sheets("Internal Calc 4").Protect

Sheets("Single Flight").Select
'Sheets("Single Flight").Range("a1").Select
End Sub
```

---

```
Private Sub RunSelectedFltBtn_Click()

'Unlock sheets (added by S. Summer 01/31/2007)
Sheets("User inputs and results").Unprotect
Sheets("FRM").Unprotect
Sheets("Single Flight").Unprotect
Sheets("Summary of n Cases").Unprotect
Sheets("Internal Calculations").Unprotect
Sheets("Internal Calc 2").Unprotect
Sheets("Internal Calc 3").Unprotect
Sheets("Internal Calc 4").Unprotect
Sheets("Internal Calculations").Cells(2, 11) = 1
pickedflight

'Lock sheets (added by S. Summer 01/31/2007)
Sheets("User inputs and results").Protect
Sheets("FRM").Protect
Sheets("Single Flight").Protect
Sheets("Summary of n Cases").Protect
Sheets("Internal Calculations").Protect
```

```
Sheets("Internal Calc 2").Protect
Sheets("Internal Calc 3").Protect
Sheets("Internal Calc 4").Protect
End Sub
```

---

```
Private Sub ReturntoInputBtn_Click()
Sheets("User inputs and results").Select
Sheets("User inputs and Results").Range("a1").Select
End Sub
```

---

```
Option Explicit

Private Sub ReturntoInputBtn_Click()
Sheets("User inputs and Results").Select
Sheets("User inputs and Results").Range("a1").Select
End Sub
```

---

```
Option Explicit

Sub montecarlo()
'User interface updated Feb 2005

'Uses control z
'Application.ScreenUpdating = False
Dim Totalflamm As Single
Dim Tlfl As Single
Dim Tufl As Single
Dim GBItime As Single
Dim percentfull As Single
Dim GBIon As Long
Dim RangeIncrNum As Long
Dim CumRangeNum As Long
Dim RangeMean As Single
Dim RangeFlight As Single
Dim timeflight As Single
Dim MaxTimeFlight As Single
Dim Flighttime As Single
Dim Numberengines As Integer
Dim Percentmax As Integer
Dim z As Single
Dim a As Single
Dim b As Single
Dim c As Single
Dim d As Single

Dim Flights As Long
```

```
Dim flightnumber As Long
Dim LFLdelta As Single
Dim LFLSlope As Single
Dim UFLDelta As Single
Dim UFLSlope As Single
Dim tmission As Long
Dim tfuel As Single
Dim Maxrange As Single
Dim e As Single
Dim f As Single
Dim g As Single
Dim h As Single
Dim i As Single
Dim iL As Single
Dim iU As Single
Dim j As Single
Dim k As Single
Dim l As Single
Dim m As Single
Dim n As Single
Dim o As Single
Dim p As Single
Dim q As Single
Dim Rndland As Single
Dim RndOATend As Single
Dim curflamm As Single
Dim Deltafueltemp As Single
Dim CruiseMN As Single
Dim Tamb As Single
Dim Tambend As Single
Dim Tgrdland As Single
Dim tatend As Single
Dim Tgrd As Single
Dim tcrz As Single
Dim tcrzend As Single
Dim Tcruise As Single
Dim Tflashpt As Single
Dim tbf As Single
Dim tflt As Single
Dim Alt1 As Single
Dim Tankalt As Single
        Dim Alt2 As Single
        Dim Alt3 As Single
        Dim mode As Long
        Dim Altstep1 As Single
        Dim Altstep2 As Single
```

```
            Dim Altstep3 As Single

Dim tclb As Single
Dim tcrz1 As Single
Dim tcrz2 As Single
Dim tcrz3 As Single
Dim tdes As Single
Dim Taft As Single
Dim time1 As Single
Dim phase1 As Single
Dim phase2 As Single
Dim phase3 As Single
Dim phase4 As Single
Dim phase5 As Single
Dim phase6 As Single
Dim phase7 As Single
Dim tat As Single
Dim mach As Single
Dim tanktemp As Single
Dim Alt As Single
Dim timeatMT As Long
Dim timetankstart As Long
Dim targTgrd As Single
Dim Timeconstgrd As Single
Dim Timeconstgrdfull As Single
Dim targtflt As Single
Dim timeconstflt As Single
Dim timeconstfltfull As Single
Dim Taugnd As Single

Dim Tauflt As Single
Dim deltatemp As Single
Dim steptemp As Single
Dim lapse As Single
Dim flammtime As Single
Dim ALtvTAT(0 To 65)
Dim AltvsTATend(0 To 65)
Dim Crzmn As Single
Dim Tlapsegrd As Single
Dim tlapseflt As Single
Dim flammyesno As Long
Dim Timetogo As Long
Dim totalflam As Single
Dim totalmissiont As Single
Dim percent As Single
```

```
        Dim enginestarttime As Single
        Dim targettemp_engineON As Single
        Dim timeconstantMT_engineON As Single
        Dim timeconstantfull_engineON As Single

Dim OATLimit As Single


'Added Body tank features
Dim BodyTankYesNO As Long
Dim BodyTankpressyesno As Long
Dim BodyTankPressDiff As Long  'Modified S. Summer 01/31/2007
Dim BodyTankPressTime As Single 'Added S. Summer 01/31/2007
Dim BodyTankTemp As Single

'added mission length/distribution array
Dim MLarray(0 To 53, 3)
Dim MLstep As Integer
Dim MLrnd As Single
Dim MLrange As Single

Dim timein As Single
Dim timeout As Single

Dim check1 As Single
Dim check2 As Single
Dim ramp As Integer
Dim timestartramp As Single
Dim timeendramp As Single
Dim rampslope As Single


Dim flamyesnogrd As Single
Dim flamyesnoclb As Single
Dim flamyesnocrz1 As Single
Dim flamyesnocrz2 As Single
Dim flamyesnocrz3 As Single
Dim flamyesnodes As Single
Dim flamyesnopost As Single

Dim flamyesnogrd80 As Single
Dim flamyesnoclb80 As Single
Dim flamyesnocrz180 As Single
Dim flamyesnocrz280 As Single
Dim flamyesnocrz380 As Single
Dim flamyesnodes80 As Single
```

```
Dim flamyesnopost80 As Single
Dim timegrd80 As Single
Dim timeclb80 As Single
Dim timecrz180 As Single
Dim timecrz280 As Single
Dim timecrz380 As Single
Dim timedes80 As Single
Dim timepost80 As Single



Dim Tottimegrd As Single
Dim Tottimeclb As Single
Dim Tottimecrz1 As Single
Dim Tottimecrz2 As Single
Dim Tottimecrz3 As Single
Dim Tottimedes As Single
Dim Tottimepost As Single


Dim Totflamgrd As Single
Dim Totflamclb As Single
Dim Totflamcrz1 As Single
Dim Totflamcrz2 As Single
Dim Totflamcrz3 As Single
Dim Totflamdes As Single
Dim Totflampost As Single

'Get times for above 80 deg F days
Dim Tottimegrd80 As Single
Dim Tottimeclb80 As Single
Dim Tottimecrz180 As Single
Dim Tottimecrz280 As Single
Dim Tottimecrz380 As Single
Dim Tottimedes80 As Single
Dim Tottimepost80 As Single

Dim Totflamgrd80 As Single
Dim Totflamclb80 As Single
Dim Totflamcrz180 As Single
Dim Totflamcrz280 As Single
Dim Totflamcrz380 As Single
Dim Totflamdes80 As Single
Dim Totflampost80 As Single

Dim FRMonoff As Integer 'This defines is FRM running
```

```
Dim FRMyesno As Integer 'This defines if FRM is effective at any one minute period
'This next set of parameters are used togather the FRM flammability exposure data
Dim flammyesnoFRM As Integer
Dim flammtimeFRM As Single
Dim flamyesnogrdFRM As Single
Dim flamyesnoclbFRM As Single
Dim flamyesnocrz1FRM As Single
Dim flamyesnocrz2FRM As Single
Dim flamyesnocrz3FRM As Single
Dim flamyesnodesFRM As Single
Dim flamyesnopostFRM As Single

Dim flamyesnogrd80FRM As Single
Dim flamyesnoclb80FRM As Single
Dim flamyesnocrz180FRM As Single
Dim flamyesnocrz280FRM As Single
Dim flamyesnocrz380FRM As Single
Dim flamyesnodes80FRM As Single
Dim flamyesnopost80FRM As Single

Dim TotflamgrdFRM As Single
Dim TotflamclbFRM As Single
Dim Totflamcrz1FRM As Single
Dim Totflamcrz2FRM As Single
Dim Totflamcrz3FRM As Single
Dim TotflamdesFRM As Single
Dim TotflampostFRM As Single

Dim Totflamgrd80FRM As Single
Dim Totflamclb80FRM As Single
Dim Totflamcrz180FRM As Single
Dim Totflamcrz280FRM As Single
Dim Totflamcrz380FRM As Single
Dim Totflamdes80FRM As Single
Dim Totflampost80FRM As Single
Dim TotalflammFRM As Single
Dim percentFRM As Single
Dim curflammFRM As Single

Dim Fixflights As Integer 'Fixedflights =1 locks random number generator to repeat list
of random numbers
Dim warm As Integer 'added by S. Summer 05/25/05
'warm =1 generates data for only those flights with ambient temp. greater than 80F
```

```
' = (Time) 'gets start time to track run time
'Sheets("Internal Calculations").Cells(17, 20).Value = Time 'Puts start time in cell to
eventually compute run time
'Get body tank input data
BodyTankYesNO = Sheets("Internal Calculations").Cells(9, 20)
BodyTankpressyesno = Sheets("Internal Calculations").Cells(10, 20)
BodyTankPressDiff = Sheets("Internal Calculations").Cells(11, 20) 'Modified S. Summer
01/31/2007

BodyTankTemp = Sheets("Internal Calculations").Cells(13, 20) 'Modified S. Summer
01/31/2007

'Get engine number
Numberengines = Sheets("User inputs and results").Cells(6, 6)

'Get number of Flights from SS
Flights = Sheets("Internal Calculations").Cells(2, 11)
'Create LFL and UFL values
 'get fix flight option
Fixflights = Sheets("User inputs and results").Cells(48, 8)

    'Get LFL and UFL values
    LFLdelta = Sheets("Internal Calculations").Cells(3, 15)
    LFLSlope = Sheets("Internal Calculations").Cells(4, 15)
    UFLDelta = Sheets("Internal Calculations").Cells(3, 16)
    UFLSlope = Sheets("Internal Calculations").Cells(4, 16)


    'Clear out single flight data for single flight case
    If Flights = 1 Then
    Sheets("Internal Calculations").Select
    Range("H24:S4000").Select
    Selection.ClearContents
    Sheets("Internal Calculations").Select
    GoTo 22
    End If


    'Clear out multi-flight data for cases more than 5000 flights
    If Flights > 1 Then
    Sheets("Summary of n Cases").Select
    Range("b6:l6000").Select
    Selection.ClearContents
    Sheets("User inputs and results").Select
```

```
        End If
'Get all the mean and Std deviation data for Ambient temp and Flash point
h = Sheets("Internal Calculations").Cells(9, 3).Value 'Mean value of ground Temp

iL = Sheets("Internal Calculations").Cells(10, 3).Value ' 1 Sigma value of lower 50% of
ground temp
iU = Sheets("Internal Calculations").Cells(11, 3).Value ' 1 Sigma value of upper 50% of
ground temp

 k = Sheets("Internal Calculations").Cells(9, 4).Value 'Mean value of crz Temp
l = Sheets("Internal Calculations").Cells(10, 4).Value ' 1 Sigma value of crz temp


 o = Sheets("Internal Calculations").Cells(9, 5).Value 'Mean value of Flash Point
p = Sheets("Internal Calculations").Cells(10, 5).Value ' 1 Sigma value of Flash Point
        'Range algrithm to get random missions.
' This gives the user a
'visual indication of the variation in flammability to see if the solution
'is converging quickly or not.

'This process takes the percent of missions in each of the range blocks, picks a random
number,
'finds which block to choose from and gets a flight. repeating this often enough will
generate
'the correct distribution of flight but with a random sequence.

'First fill the array with mission length and corresponding percent of flights
        For MLstep = 0 To 52
        MLarray(MLstep, 0) = MLstep * 200
        MLarray(MLstep, 1) = Sheets("Internal Calc 3").Cells(MLstep + 4, 12)
        Next MLstep
22 ' code jumps to here if doing a single flight


OATLimit = Sheets("User inputs and results").Cells(9, 6)
 Deltafueltemp = Sheets("Internal Calculations").Cells(15, 4) 'Gets correction to Tfuel if
Tfuel is hotter or colder than ambient


FRMonoff = Sheets("Internal Calc 4").Cells(2, 10)

warm = Sheets("User inputs and results").Cells(50, 8) 'added by S. Summer 05/25/05
```

```
'**********************************************************************
****************************
'Flight Loop starts here....Flight Loop starts here....Loop Flight starts here....Flight Loop starts here....Flight Loop starts here....
'**********************************************************************
****************************
   'set collector parameters to zero
   totalmissiont = 0
  Totalflamm = 0
   TotalflammFRM = 0
flammtime = 0
flamyesnogrd = 0
flamyesnoclb = 0
flamyesnocrz1 = 0
flamyesnocrz2 = 0
flamyesnocrz3 = 0
flamyesnodes = 0
flamyesnopost = 0

flamyesnogrd80 = 0
flamyesnoclb80 = 0
flamyesnocrz180 = 0
flamyesnocrz280 = 0
flamyesnocrz380 = 0
flamyesnodes80 = 0
flamyesnopost80 = 0
timegrd80 = 0
 timeclb80 = 0
 timecrz180 = 0
timecrz280 = 0
timecrz380 = 0
timedes80 = 0
timepost80 = 0

flammtimeFRM = 0
flamyesnogrdFRM = 0
flamyesnoclbFRM = 0
flamyesnocrz1FRM = 0
flamyesnocrz2FRM = 0
flamyesnocrz3FRM = 0
flamyesnodesFRM = 0
flamyesnopostFRM = 0
flamyesnogrd80FRM = 0
flamyesnoclb80FRM = 0
flamyesnocrz180FRM = 0
flamyesnocrz280FRM = 0
```

```
flamyesnocrz380FRM = 0
flamyesnodes80FRM = 0
flamyesnopost80FRM = 0


'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 '                  lllll             ooo             ooo             pppppppp
 '                   lll            oo   oo         oo   oo             ppp   ppp
 '                   lll           oo     ooo       oo     ooo            ppp    ppp
 '                   lll           oo     ooo       oo     ooo            ppp   ppp
 '                   lll           oo     ooo       oo     ooo            pppppppp
 '                   lll           oo     ooo       oo     ooo            ppp
 '                   lll           oo     ooo       oo     ooo            ppp
 '                   lll     l      oo   ooo         oo   ooo             ppp
 '                   llllllllll       ooo             ooo              pppp


'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

'Start big loop for number of Flights


'Option for using a fixed set of random numbers for debugging code etc
If Fixflights = 1 Then
Rnd (-1)
Randomize (100)
Else
Randomize
End If
'Start loop for n flights
For flightnumber = 1 To Flights

999 'added by S. Summer 05/25/05

'Randomize the Excel Rnd Function ,and select four values

'Randomize
a = Rnd() 'random numbers used to generate flight profile
b = Rnd()
c = Rnd()
d = Rnd()

MLrnd = Rnd() * 100
Rndland = Rnd()
```

```
RndOATend = Rnd()

CruiseMN = Sheets("Internal Calculations").Cells(13, 4)



'  range loop to match range distribution table
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Calculate Mission length (nautical miles)                                    '
' Program steps through each range increment based on historical flight distribution '
' The program randomly varies the range by +/- 100 within each 200 nm increment      '
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

50

If Flights = 1 Then
GoTo 23
End If

'Now pick random number and find appropriate mission length
'MLrnd = Rnd() * 100
For MLstep = 0 To 52
        If MLrnd < MLarray((MLstep + 1), 1) And MLrnd > MLarray(MLstep, 1) Then
        'edited by S. Summer as per email dated 7/13/06 from Mike Collins
        RangeFlight = MLarray((MLstep + 1), 0) + ((MLarray((MLstep + 1), 0) -
MLarray(MLstep, 0)) _
        / (MLarray((MLstep + 1), 1) - MLarray(MLstep, 1))) * (MLrnd - MLarray((MLstep
+ 1), 1))
        'MLrange = MLarray(MLstep, 1) + 200
        'GoTo 111
        End If
Next MLstep
111
'MLarray(MLstep, 3) = MLarray(MLstep, 3) + 1

'RangeFlight = MLrange + (a - 0.5) * 200 'flight distance for this mission

''''''''''''''''''''''''''''''''
' Determine Mission Flight Time '
''''''''''''''''''''''''''''''''

' Flight Time based on Flight Range
    Flighttime = (RangeFlight - 100) * 60 / 573.6 / CruiseMN + (0.7 * 60)
```

```
23
' Maximum flight time (minutes)
Maxrange = Sheets("User inputs and results").Cells(5, 6)
   MaxTimeFlight = (Maxrange - 100) * 60 / 573.6 / CruiseMN + 0.7 * 60

' Limit Flighttime to not exceed max range
   If Flighttime > MaxTimeFlight Then
      Flighttime = MaxTimeFlight
   End If

' Set Flighttime to a minimum of 15 minutes
   If Flighttime < 15 Then
      Flighttime = 15
   End If




If Flights = 1 Then
Flighttime = Sheets("Internal Calculations").Cells(18, 8)
End If
Flighttime = CInt(Flighttime) 'rounds flight time to integer value and converts to integer

If Flights = 1 Then
GoTo 24
End If
'next piece calculates ground ambient from the random number/mean/std dev
99
'b = Rnd() 'Random number to determine ground ambient temperature

If b < 0.5 Then
i = iL
Else: i = iU
End If
Tgrd = Application.NormInv(b, h, i) 'This is the departure airport ambient temperature
If warm = 1 And Tgrd < 80 Then 'added by S. Summer 05/25/05
   GoTo 999 'added by S. Summer 05/25/05
End If 'added by S. Summer 05/25/05

'Find landing airport ambient temp
'Rndland = Rnd()
If Rndland < 0.5 Then
i = iL
Else: i = iU
```

```
End If
Tgrdland = Application.NormInv(Rndland, h, i) 'This is the landing airport ambient
temperature


24

Percentmax = Int((Flighttime / MaxTimeFlight) * 5)  ' this defines mission length to
determine climb time

If Flights = 1 Then
Tgrd = Sheets("Internal Calculations").Cells(19, 8)
Tgrdland = Sheets("Internal Calculations").Cells(19, 9)
End If



'Test of a upper limit cuttoff for Ambient temperature to reduce flammability

If Tgrd > OATLimit Then
Tgrd = OATLimit 'This sets Tgrnd to the OAT Limit as if the flight as delayed until the
temperature dropped to the OAT limit
End If



tfuel = Tgrd + Deltafueltemp 'sets tfuel to be equal to tamb +/- correction at start of
mission

'Tgrd = Tamb
If Flights = 1 Then
GoTo 25
End If

'next piece calculates Cruise amb from the random number/mean/std dev.




m = Application.NormInv(c, k, l)

tcrz = m

25
If Flights = 1 Then
tcrz = Sheets("Internal Calculations").Cells(20, 8) 'This is the start of cruise OAT
```

```
tcrzend = Sheets("Internal Calculations").Cells(20, 9) 'this is the end of crz OAT for
flights over 120 minutes in crz
GoTo 26
End If



'RndOATend = Rnd()
tcrzend = Application.NormInv(RndOATend, k, l)




'next piece calculates flash point from the random number/mean/std dev.

q = Application.NormInv(d, o, p)

Tflashpt = q



26
If Flights = 1 Then
Tflashpt = Sheets("Internal Calculations").Cells(21, 8)
End If

'Next section calculates the mission profile
'First calculate time before flight tbf
'Assumptions are:
'For flights over 240 minutes, ground time is 90 minutes,
'Between 280 and 240 minutes the ground time is 45 minutes,
'For flights less than 180 minutes, the ground time is 30 minutes.



 If Flighttime > 240 Then
 tbf = 90
   ElseIf Flighttime > 180 Then
   tbf = 45
      Else: tbf = 30

End If
BodyTankPressTime = Sheets("Internal Calculations").Cells(12, 20) 'Added S. Summer
01/31/2007
```

```
If BodyTankPressTime > tbf Then 'Added S. Summer 01/31/2007
   BodyTankPressTime = tbf 'Added S. Summer 01/31/2007
End If 'Added S. Summer 01/31/2007
'Now calculate flight time

tmission = Flighttime + tbf + 30
'the 30 mins added is the time after the flight


'now define flight profiles as short(up and down), single, one step or two step cruise
flights

'Get cruise altitudes from input data
               Altstep1 = Sheets("Internal Calculations").Cells(20, 3)
               Altstep2 = Sheets("Internal Calculations").Cells(21, 3)
               Altstep3 = Sheets("Internal Calculations").Cells(22, 3)



      If Flighttime < 50 Then  'This is a short up and down to an altitude less than cruise
because then isn't enough time to get to cruise altitude
               Alt1 = Flighttime * 20 / 50 * 1750 'This assumes a climb rate of 1750 ft/min
and climb will take 40% of flight time
               Alt2 = Alt1
               Alt3 = Alt1
                        If Alt1 > Altstep1 Then
                        Alt1 = Altstep1
                        Alt2 = Alt1
                        Alt3 = Alt1
                        End If
               mode = 1 'Short up and down flight


                  ElseIf Flighttime < 100.1 Then
                  Alt1 = Altstep1
                  Alt2 = Alt1
                  Alt3 = Alt1
                  mode = 2 'One altitude flight, no step climb

                     ElseIf Flighttime < 200 And Flighttime > 100.1 Then
                     Alt1 = Altstep1
                     Alt2 = Altstep2
                     Alt3 = Alt2
                     mode = 3 'Two altitude flight, one step climb at mid point
```

```
            ElseIf Flighttime > 199.99 Then 'Three altitude flight, two step climbs.
Each altitude flight is for 33% of cruise time.
            Alt1 = Altstep1
            Alt2 = Altstep2
            Alt3 = Altstep3
            mode = 4 'Three altitude flight, two step climbs.
    End If


'This section was commented out by S. Summer on 01/31/2007
'This section resets altitude for a body tank to the cabin altitude IF the tank is pressurized
'If BodyTankYesNO = 1 And BodyTankpressyesno = 1 Then

'Alt1 = BodyTankAlt
'Alt2 = BodyTankAlt
'Alt3 = BodyTankAlt

'End If
'now assign time to altitudes
'Reset all values to Zero
tclb = 0
tdes = 0
tcrz1 = 0
tcrz2 = 0
tcrz3 = 0

If mode = 1 Then
      tclb = 0.4 * Flighttime
      tdes = Flighttime - tclb

        ElseIf mode = 2 Then
        tclb = Sheets("Internal Calc 2").Cells((6 + Numberengines), (9 + Percentmax))
'Climb time is a function of number of engines and mission length
        tdes = Sheets("Internal Calc 2").Cells((Alt3 / 1000 - 6), 10) 'fixed descent
profile for all flights
        tcrz1 = Flighttime - tclb - tdes

          ElseIf mode = 3 Then
          tclb = Sheets("Internal Calc 2").Cells((6 + Numberengines), (9 +
Percentmax))
          tdes = Sheets("Internal Calc 2").Cells((Alt3 / 1000 - 6), 10)
          tcrz1 = (Flighttime - tclb - tdes) / 2
          tcrz2 = tcrz1

            ElseIf mode = 4 Then
```

```
                tclb = Sheets("Internal Calc 2").Cells((6 + Numberengines), (9 +
Percentmax))
                tdes = Sheets("Internal Calc 2").Cells((Alt3 / 1000 - 6), 10)
                tcrz1 = (Flighttime - tclb - tdes) / 3
                tcrz2 = tcrz1
                tcrz3 = tcrz1
End If

Taft = 30



'Now create an array of TAT versus Altitude in 1000' increments that the temp
'calculations can use.



 'Get Mach Number

           Crzmn = Sheets("Internal Calculations").Cells(13, 4)
           'define mach number changes with altitude
     For Alt = 0 To Altstep3 / 1000 'Modified S. Summer 01/31/2007
        If Alt = 0 Then
             mach = 0

                ElseIf Alt < 10 Then
                mach = 0.4

                ElseIf Alt < 30 Then
                mach = ((Alt - 10) * (Crzmn - 0.4) / 20) + 0.4

                Else: mach = Crzmn
        End If


'define tamb versus altitude


                ' Tamb Algorithm, provides cold day inversion, and change of Tropopause
Alt.

                If Alt < 10 Then
```

```
                If Tgrd > 39.9 Then
                    Tamb = Tgrd - 3.57 * Alt 'standard lapse rate
                End If


                If Tgrd < 40 Then
                    Tamb = Tgrd - (Tgrd - 4.3) / 10 * Alt 'Temperature inversion
below 40 deg F, returns to normal lapse rate at 10,000ft
                End If
            End If

            If Alt > 9.5 Then 'Lapse rate here was modified from 3.75 to 3.57 by S.
Summer on 1/11/10
                Tamb = Tamb - 3.57 'This runs temperature down as altitude
climbs using std lapse rate
            End If


            If Alt > 9.5 Then
                If Tamb < tcrz Then 'This cuts off lapse rate when cruise temperature
is reached. This is setting cruise temp as the tropopause temperature
                Tamb = tcrz
                End If
            End If



 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   'define tamb versus altitude for end of flight conditions


            ' Tamb Algorithm, provides cold day inversion, and change of Tropopause
Alt.

            If Alt < 10 Then
                If Tgrdland > 39.9 Then
                    Tambend = Tgrdland - 3.57 * Alt 'standard lapse rate
                End If


                If Tgrdland < 40 Then
                    Tambend = Tgrdland - ((Tgrdland - 4.3) / 10) * Alt
'Temperature inversion below 40 deg F, returns to normal lapse rate at 10,000ft
                End If
            End If
```

```
                If Alt > 9.5 Then 'Lapse rate here was modified from 3.75 to 3.57 by S. Summer on 1/11/10
                        Tambend = Tambend - 3.57 'This runs temperature dwon as altitude climbs using std lapse rate
                End If


                If Alt > 9.5 Then
                     If Tambend < tcrzend Then 'This cuts off lapse rate when cruise temperature is reached. This is setting cruise temp as the tropopause temperature
                     Tambend = tcrzend
                     End If
                End If


'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
'Define TAT (Actually this is Skin recovery temperature not TAT)
'Note:True TAT would use (1+ 0.2*mach^2); using 1.8 lowers TAT a small amount to represent skin temp at tank surface


 'Correction for Body Tank conditions, by setting Tamb and TAT to Cabin Temperature


 tat = ((Tamb + 460) * (1 + 0.18 * mach ^ 2)) - 460
 tatend = ((Tambend + 460) * (1 + 0.18 * mach ^ 2)) - 460
 If BodyTankYesNO = 1 Then
 Tamb = BodyTankTemp
 tat = BodyTankTemp
 tatend = BodyTankTemp 'Added by S. Summer 4/11/2007
End If

'Put TAT and tatend into Arrays
ALtvTAT(Alt) = tat
AltvsTATend(Alt) = tatend

'If Flights = 1 Then
Sheets("Internal Calculations").Cells(35 + Alt, 4) = tat
Sheets("Internal Calculations").Cells(35 + Alt, 2) = Tamb
Sheets("Internal Calculations").Cells(35 + Alt, 3) = mach

Sheets("Internal Calculations").Cells(101 + Alt, 4) = tatend
Sheets("Internal Calculations").Cells(101 + Alt, 2) = Tambend
Sheets("Internal Calculations").Cells(101 + Alt, 3) = mach
'End If
```

```
Sheets("Internal Calculations").Cells(35 + Alt, 5) = ALtvTAT(Alt)
Sheets("Internal Calculations").Cells(101 + Alt, 5) = AltvsTATend(Alt)
Next Alt


'Now calculate flight temp calculations

'define time phases of the mission

phase1 = tbf
phase2 = tbf + tclb
phase3 = tbf + tclb + tcrz1
phase4 = tbf + tclb + tcrz1 + tcrz2
phase5 = tbf + tclb + tcrz1 + tcrz2 + tcrz3
phase6 = tbf + tclb + tcrz1 + tcrz2 + tcrz3 + tdes
phase7 = tbf + tclb + tcrz1 + tcrz2 + tcrz3 + tdes + Taft

check1 = tclb + tcrz1 + tcrz2 + tcrz3 + tdes

'Check for length of crz time to reset ambient temperatures
check2 = tcrz1 + tcrz2 + tcrz3
If check2 > 120 Then
ramp = 1
timestartramp = tbf + tclb + (check2 / 2) + 10
timeendramp = timestartramp + 45
Else: ramp = 0
timestartramp = phase7
timeendramp = phase7
End If


'start time loop going
' get all the tank thermal constants from the spreadsheet



   targTgrd = Sheets("Internal Calculations").Cells(9, 8)
   Timeconstgrd = Sheets("Internal Calculations").Cells(10, 8)
   Timeconstgrdfull = Sheets("Internal Calculations").Cells(11, 8)
   targtflt = Sheets("Internal Calculations").Cells(13, 8)
   timeconstflt = Sheets("Internal Calculations").Cells(14, 8)
   timeconstfltfull = Sheets("Internal Calculations").Cells(15, 8)
   timeatMT = Sheets("Internal Calculations").Cells(14, 4)
   timetankstart = Sheets("Internal Calculations").Cells(16, 4)
```

```
    enginestarttime = Sheets("Internal Calculations").Cells(8, 14)
    targettemp_engineON = Sheets("Internal Calculations").Cells(9, 14)
    timeconstantMT_engineON = Sheets("Internal Calculations").Cells(10, 14)
    timeconstantfull_engineON = Sheets("Internal Calculations").Cells(11, 14)







'START TIME LOOP
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx


'        tttttttttttt    iiiii      mmm       mmm       eeeeeeeee       lll        ooo        ooo
ppppppp
'            ttt        iii       mmmm mm   mm  mmm      eee            lll       oo   oo     oo
oo    ppp   ppp
'            ttt        iii       mmm   mm mm    mmm     eee           lll      oo    ooo   oo
ooo   ppp   ppp
'            ttt        iii       mmm   mmmm     mmm     eeeeeeee       lll     oo     ooo  oo
ooo   ppppppp
'            ttt        iii       mmm    mm     mmm     eee           lll     oo     ooo  oo     oo
ppp
'            ttt        iii       mmm    mm     mmm     eee           lll      oo     oo    oo    oo
ppp
'            ttt        iii       mmm    mm     mmm     eee           lll       oo   oo      oo  oo
ppp
'            ttt       iiiii      mmm    mm     mmm     eeeeeeeee      lllllll     ooo        ooo
ppp

'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx



For time1 = 0 To (tmission - 1)
Timetogo = tmission - time1
flammyesno = 0 'Sets flammyesno to zero at start of loop
```

```
FRMyesno = 0 'Sets FRM to "not effective" at the start of the flight loop until turned to
"effective" by the FRM module
'set time constants to appropriate values


'The following section of code was modified by S. Summer on 4/26/2007


If Timetogo > (timetankstart + 30) Then 'checks if tank is full

        If time1 > (tbf - enginestarttime) Then  'checks if engines are on
           Taugnd = timeconstantfull_engineON
        Else
           Taugnd = Timeconstgrdfull
        End If
           Tauflt = timeconstfltfull
End If


If Timetogo <= (timeatMT + 30) Then  'checks if tank is empty
        If time1 > (tbf - enginestarttime) Then ' checks if engines ON
           Taugnd = timeconstantMT_engineON
        Else
           Taugnd = Timeconstgrd
        End If
           Tauflt = timeconstflt
End If

                    'This code creates linear change in time constants with fuel burn from tank
full to tank empty

If Timetogo <= (timetankstart + 30) And Timetogo > (timeatMT + 30) Then
        Taugnd = ((Timetogo - (timeatMT + 30)) / (timetankstart - timeatMT)) *
(timeconstantfull_engineON - timeconstantMT_engineON) +
timeconstantMT_engineON
         Tauflt = ((Timetogo - (timeatMT + 30)) / (timetankstart - timeatMT)) *
(timeconstfltfull - timeconstflt) + timeconstflt
End If



'This completes the section of code modifed on 04/26/2007 S. Summer


 'Define lapse rate constants as change per minute
               Tlapsegrd = (1 - Exp(-1 / Taugnd))
```

```
        tlapseflt = (1 - Exp(-1 / Tauflt))

'get altitude

If time1 < phase1 Then
            Alt = 0

'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXx
              '"If" statement to change equilibrium temp for engine ON case
              If time1 > (tbf - enginestarttime) Then
              targTgrd = targettemp_engineON
              End If

            'this next code calculates fuel temp from tat and lapse rate- ground case
            z = Int(Alt / 1000)

            tat = ALtvTAT(z) 'gets tat from array altvtat
            steptemp = (tat + targTgrd) - tfuel 'This is the temperature differential
between current fuel temperature and long term equilibrium temperature
            tfuel = tfuel + (steptemp) * Tlapsegrd 'This computes the temperature
change in one minute based on temperature differential and exponential decay rate

ElseIf time1 < phase2 Then
            Alt = Int(Alt1 * (time1 - tbf) / tclb)

            'this next code calculates fuel temp from tat and lapse rate-flt case
            z = Int(Alt / 1000)

            tat = ALtvTAT(z) 'gets tat from array altvtat
            steptemp = (tat + targtflt) - tfuel
            tfuel = tfuel + (steptemp) * tlapseflt

ElseIf time1 < phase3 Then
            Alt = Alt1

            'this next code calculates fuel temp from tat and lapse rate-flt case
            z = Int(Alt / 1000)
```

```
            If time1 < timestartramp Then
            tat = ALtvTAT(z) 'gets tat from array altvtat
            End If
                If ramp = 1 Then 'this loop ramps TAT from start of flight TAT to end of
flight TAT with 45 min ramp
                        If time1 > timestartramp And time1 < timeendramp Then
                        tat = ALtvTAT(z) - (ALtvTAT(z) - AltvsTATend(z)) * (1 -
(timeendramp - time1) / 45)
                        End If
                End If
                        If ramp = 1 And time1 > timeendramp Then
                        tat = AltvsTATend(z)

                        End If

                steptemp = (tat + targtflt) - tfuel
                tfuel = tfuel + (steptemp) * tlapseflt

ElseIf time1 < phase4 Then
                Alt = Alt2


                'this next code calculates fuel temp from tat and lapse rate-flt case
                z = Int(Alt / 1000)

            If time1 < timestartramp Then
            tat = ALtvTAT(z) 'gets tat from array altvtat
            End If
                If ramp = 1 Then 'this loop ramps TAT from start of flight TAT to end of
flight TAT with 45 min ramp
                        If time1 > timestartramp And time1 < timeendramp Then
                        tat = ALtvTAT(z) - (ALtvTAT(z) - AltvsTATend(z)) * (1 -
(timeendramp - time1) / 45)
                        End If
                End If
                        If ramp = 1 And time1 > timeendramp Then
                        tat = AltvsTATend(z)

                        End If

                steptemp = (tat + targtflt) - tfuel
                tfuel = tfuel + (steptemp) * tlapseflt
ElseIf time1 < phase5 Then
                Alt = Alt3
```

```
            'this next code calculates fuel temp from tat and lapse rate-flt case
            z = Int(Alt / 1000)

        If time1 < timestartramp Then
        tat = ALtvTAT(z) 'gets tat from array altvtat
        End If
            If ramp = 1 Then 'this loop ramps TAT from start of flight TAT to end of
flight TAT with 45 min ramp
                  If time1 > timestartramp And time1 < timeendramp Then
                  tat = ALtvTAT(z) - (ALtvTAT(z) - AltvsTATend(z)) * (1 -
(timeendramp - time1) / 45)
                  End If
            End If
                  If ramp = 1 And time1 > timeendramp Then
                  tat = AltvsTATend(z)

                  End If

            steptemp = (tat + targtflt) - tfuel
            tfuel = tfuel + (steptemp) * tlapseflt

ElseIf time1 < phase6 Then
            'Altitude profile modified by S. Summer on 1/11/2010 to properly reflect
the descent profile specified in the rule.
            If mode = 1 Then
              Alt = Int(Alt3 - (Alt3 * (time1 - phase5) / tdes))
            Else
              If Alt > 4000 Then
                Alt = Alt - 2500
                If Alt <= 4000 Then Alt = 4000
              Else
                Alt = Alt - 500
              End If
            End If
            z = Round(Alt / 1000) 'Modified by S. Summer on 4/30/07 per email from
Helen Tsai(RGW Ass.) on 4/19
            If ramp = 1 Then
            tat = AltvsTATend(z) 'gets tat from array altvstatend for long flights
            Else
            tat = ALtvTAT(z) 'gets tat from array altvtat for short flights
            End If

            steptemp = (tat + targtflt) - tfuel
            tfuel = tfuel + (steptemp) * tlapseflt
```

```
ElseIf time1 >= phase6 Then  'Modified by S. Summer on 4/30/07 per email from Helen
Tsai(RGW Ass.) on 4/19
                Alt = 0
                z = 0
                If ramp = 1 Then
                tat = AltvsTATend(z)
                Else
                tat = ALtvTAT(z)
                End If
                steptemp = (tat + targTgrd) - tfuel
                tfuel = tfuel + (steptemp) * Tlapsegrd

End If


'Calculate Flammability Limits

'*****************************************************************
                'LFL and UFl Definitions

'*****************************************************************

             'if tank is body tank,then bypass the ram pressure recovery effect 'Modified
out by S. Summer 01/31/2007
             'If BodyTankYesNO = 1 Then
             'Tankalt = Alt
             'GoTo 777
             'End If
             'This next if statement determines if the tank is pressurized or not
             'added S. Summer 01/31/2007
             If time1 <= (tbf - BodyTankPressTime) Then
             Tankalt = Alt
             GoTo 777
             End If
             Tankalt = Sheets("Internal Calculations").Cells(101 + CInt(Alt / 1000), 7)

777
                Tlfl = (Tflashpt + LFLdelta) - Tankalt / (LFLSlope)
                Tufl = (Tflashpt + UFLDelta) - Tankalt / (UFLSlope)




' Sheets("Internal Calculations").Cells(time1 + 24, 14) = tflashptflt
             If tfuel > Tlfl Then
```

```
                flammyesno = 1
                Else
                flammyesno = 0
                End If

                If tfuel > Tufl Then
                flammyesno = 0
                End If
'.........FRM.................FRM...................FRM.............................FRM

'This point is where the user can add a module to compute FRM performance at this time
of the flight.
'The temperature, altitude and flight profile are available from this montecarlo module.
'IF the FRM module needs more data than provided by the montecarlo module, the user
will be required to compute that information.
'The output of the FRM module should be the parameter "FRMyesno" being set to a 0 if
the FRM is NOT keeping the tank from being flammable and
'FRMyesno is set to 1 if the FRM is keeping the tank from being flammable at this time
of flight



'========================TEST FRM
ONLY=====================================================
'This is a very simple FRM to test the FRM module and data collecting code. The FRM
is assumed to be effective after t=10 min and to be effective until the end of cruise for all
flights.
If FRMonoff = 0 Then
flammyesnoFRM = flammyesno
GoTo 299
End If
If time1 < phase6 And time1 > 10 Then
FRMyesno = 1
End If





'========================TEST FRM
ONLY=====================================================

'This loop will change flammyesno to 0 if FRM is working at this time period
                flammyesnoFRM = flammyesno 'This sets the FRM flammability to be the
same as the no FRM case until changed by the FRM module
```

```
        If FRMyesno = 1 Then
        flammyesnoFRM = 0

        End If
 '.........FRM.................FRM...................FRM.............................FRM

'Put values to Sheet Internal Calculations for a single flight
299
If Flights > 1 Then
GoTo 333
End If
Sheets("Internal Calculations").Cells(time1 + 24, 11) = Tamb
Sheets("Internal Calculations").Cells(time1 + 24, 8) = time1
Sheets("Internal Calculations").Cells(time1 + 24, 9) = Alt
Sheets("Internal Calculations").Cells(time1 + 24, 10) = tat
Sheets("Internal Calculations").Cells(time1 + 24, 12) = tfuel
Sheets("Internal Calculations").Cells(time1 + 24, 14) = Tlfl
Sheets("Internal Calculations").Cells(time1 + 24, 19) = Tufl
Sheets("Internal Calculations").Cells(time1 + 24, 16) = tfuel
Sheets("Internal Calculations").Cells(time1 + 24, 15) = flammyesno
Sheets("Internal Calculations").Cells(time1 + 24, 20) = flammyesnoFRM
333
flammtime = flammtime + flammyesno
flammtimeFRM = flammtimeFRM + flammyesnoFRM
' Sheets("Internal Calculations").Cells(19, 15) = flammtime

curflamm = flammtime
curflammFRM = flammtimeFRM 'Added by S. Summer 01/13/2010
'loops back for the next time increment

'*********************************************************************
****************

'Section inserted to capture the flammability exposure by phase and for hot days only

'*********************************************************************
****************

'This section was modified on April 30, 2007 by S. Summer per email from Helen Tsai
(RGW Cherry & Associates) on 04/19

        If time1 < phase1 Then 'Ground preflight
        flamyesnogrd = flamyesnogrd + flammyesno
        flamyesnogrdFRM = flamyesnogrdFRM + flammyesnoFRM
        End If
```

```
If time1 < phase2 And time1 >= phase1 Then 'Climb
flamyesnoclb = flamyesnoclb + flammyesno
flamyesnoclbFRM = flamyesnoclbFRM + flammyesnoFRM
End If
'stop set to find flam case
'If flammyesno = 1 Then
'Stop
'End If

If time1 < phase3 And time1 >= phase2 Then 'Cruise1
flamyesnocrz1 = flamyesnocrz1 + flammyesno
flamyesnocrz1FRM = flamyesnocrz1FRM + flammyesnoFRM
End If

If time1 < phase4 And time1 >= phase3 Then 'Cruise2
flamyesnocrz2 = flamyesnocrz2 + flammyesno
flamyesnocrz2FRM = flamyesnocrz2FRM + flammyesnoFRM
End If

If time1 < phase5 And time1 >= phase4 Then 'Cruise3
flamyesnocrz3 = flamyesnocrz3 + flammyesno
flamyesnocrz3FRM = flamyesnocrz3FRM + flammyesnoFRM
End If

If time1 < phase6 And time1 >= phase5 Then 'Descent
flamyesnodes = flamyesnodes + flammyesno
flamyesnodesFRM = flamyesnodesFRM + flammyesnoFRM
End If

If time1 < phase7 And time1 >= phase6 Then 'Taxi-in
flamyesnopost = flamyesnopost + flammyesno
flamyesnopostFRM = flamyesnopostFRM + flammyesnoFRM
End If


'888888888888888888888888888888888888888888888888888888888888888888
8888888888888
'For Days above 80 Deg F

'check if day is above 80 Deg F , If not skip this section

If Tgrd < 80 Then
GoTo 1111
End If

If time1 < phase1 And Tgrd > 80 Then 'Ground preflight
```

```
flamyesnogrd80 = flamyesnogrd80 + flammyesno
flamyesnogrd80FRM = flamyesnogrd80FRM + flammyesnoFRM
timegrd80 = timegrd80 + 1
End If

If time1 < phase2 And time1 >= phase1 And Tgrd > 80 Then 'Climb
flamyesnoclb80 = flamyesnoclb80 + flammyesno
flamyesnoclb80FRM = flamyesnoclb80FRM + flammyesnoFRM
timeclb80 = timeclb80 + 1
End If
'stop set to find flam case
'If flammyesno = 1 Then
'Stop
'End If

If time1 < phase3 And time1 >= phase2 And Tgrd > 80 Then 'Cruise1
flamyesnocrz180 = flamyesnocrz180 + flammyesno
flamyesnocrz180FRM = flamyesnocrz180FRM + flammyesnoFRM
timecrz180 = timecrz180 + 1
End If

If time1 < phase4 And time1 >= phase3 And Tgrd > 80 Then 'Cruise2
flamyesnocrz280 = flamyesnocrz280 + flammyesno
flamyesnocrz280FRM = flamyesnocrz280FRM + flammyesnoFRM
timecrz280 = timecrz280 + 1
End If

If time1 < phase5 And time1 >= phase4 And Tgrd > 80 Then 'Cruise3
flamyesnocrz380 = flamyesnocrz380 + flammyesno
flamyesnocrz380FRM = flamyesnocrz380FRM + flammyesnoFRM
timecrz380 = timecrz380 + 1
End If

If time1 < phase6 And time1 >= phase5 And Tgrd > 80 Then  'Descent
flamyesnodes80 = flamyesnodes80 + flammyesno
flamyesnodes80FRM = flamyesnodes80FRM + flammyesnoFRM
timedes80 = timedes80 + 1
End If

If time1 < phase7 And time1 >= phase6 And Tgrd > 80 Then 'Taxi-in
flamyesnopost80 = flamyesnopost80 + flammyesno
flamyesnopost80FRM = flamyesnopost80FRM + flammyesnoFRM
timepost80 = timepost80 + 1
End If
```

```
'collect time above 80deg F

'For Days above 80 Deg F
'If time1 < phase1 + 1 And Tgrd > 80 Then 'Ground preflight
' timegrd80 = timegrd80 + 1
'End If

'If time1 < phase2 + 0.1 And time1 > phase1 And Tgrd > 80 Then 'Climb
'timeclb80 = timeclb80 + 1
'End If
'stop set to find specific case
'If timeclb80 = 1 Then
'Stop
'End If

'If time1 < phase3 + 0.1 And time1 > phase2 And Tgrd > 80 Then 'Cruise1
'timecrz180 = timecrz180 + 1
'End If

' If time1 < phase4 + 0.1 And time1 > phase3 And Tgrd > 80 Then 'Cruise2
' timecrz280 = timecrz280 + 1
'End If

'If time1 < phase5 + 0.1 And time1 > phase4 And Tgrd > 80 Then 'Cruise3
'timecrz380 = timecrz380 + 1
'End If

'If time1 < phase6 + 0.1 And time1 > phase5 And Tgrd > 80 Then  'Descent
'timedes80 = timedes80 + 1
'End If

'If time1 < phase7 + 0.1 And time1 > phase6 And Tgrd > 80 Then 'Taxi-in
'timepost80 = timepost80 + 1
'End If

1111

Next time1

'This completes the loop for one case

'Now collect data for the sum total of all cases
```

```
Totalflamm = Totalflamm + flammtime 'This adds flammable time of last case onto total
flammable time counter
totalmissiont = totalmissiont + Int(phase7) 'This adds time of last case onto total time
counter
percent = Totalflamm / totalmissiont * 100
 Sheets("Internal Calculations").Cells(3, 11) = Totalflamm
 Sheets("Internal Calculations").Cells(4, 11) = totalmissiont


 TotalflammFRM = TotalflammFRM + flammtimeFRM 'This adds flammable time of
last case onto total flammable time counter

percentFRM = TotalflammFRM / totalmissiont * 100
 Sheets("Internal Calculations").Cells(3, 11) = Totalflamm
 Sheets("Internal Calculations").Cells(7, 11) = TotalflammFRM


 'Send percent number to single flight cell
 If Flights = 1 Then
 Sheets("Internal Calculations").Cells(6, 8) = percent
 Else
Sheets("Internal Calculations").Cells(5, 11) = percent
End If
If Flights > 1 Then
 Sheets("User inputs and results").Cells(6, 15) = flightnumber 'Puts flight number to ss as
check of progress
End If
If Flights = 1 Then
Sheets("Internal Calculations").Cells(24, 3) = tbf
Sheets("Internal Calculations").Cells(25, 3) = tclb
Sheets("Internal Calculations").Cells(26, 3) = tcrz1
Sheets("Internal Calculations").Cells(27, 3) = tcrz2
Sheets("Internal Calculations").Cells(28, 3) = tcrz3
Sheets("Internal Calculations").Cells(29, 3) = tdes
Sheets("Internal Calculations").Cells(30, 3) = Taft

GoTo 444
End If



'*************************************************************
' Section to output flammability by phase
'*************************************************************
```

```
'Jump if flights >4999, don't need to capture all this data
'If Flights > 4999 Then
'GoTo 499
'End If

Tottimegrd = Tottimegrd + tbf
Tottimeclb = Tottimeclb + tclb
Tottimecrz1 = Tottimecrz1 + tcrz1
Tottimecrz2 = Tottimecrz2 + tcrz2
Tottimecrz3 = Tottimecrz3 + tcrz3
Tottimedes = Tottimedes + tdes
Tottimepost = Tottimepost + Taft


Totflamgrd = Totflamgrd + flamyesnogrd
Totflamclb = Totflamclb + flamyesnoclb
Totflamcrz1 = Totflamcrz1 + flamyesnocrz1
Totflamcrz2 = Totflamcrz2 + flamyesnocrz2
Totflamcrz3 = Totflamcrz3 + flamyesnocrz3
Totflamdes = Totflamdes + flamyesnodes
Totflampost = Totflampost + flamyesnopost

'Get times for above 80 deg F days
Tottimegrd80 = Tottimegrd80 + timegrd80
Tottimeclb80 = Tottimeclb80 + timeclb80
Tottimecrz180 = Tottimecrz180 + timecrz180
Tottimecrz280 = Tottimecrz280 + timecrz280
Tottimecrz380 = Tottimecrz380 + timecrz380
Tottimedes80 = Tottimedes80 + timedes80
Tottimepost80 = Tottimepost80 + timepost80

Totflamgrd80 = Totflamgrd80 + flamyesnogrd80
Totflamclb80 = Totflamclb80 + flamyesnoclb80
Totflamcrz180 = Totflamcrz180 + flamyesnocrz180
Totflamcrz280 = Totflamcrz280 + flamyesnocrz280
Totflamcrz380 = Totflamcrz380 + flamyesnocrz380
Totflamdes80 = Totflamdes80 + flamyesnodes80
Totflampost80 = Totflampost80 + flamyesnopost80


TotflamgrdFRM = TotflamgrdFRM + flamyesnogrdFRM
TotflamclbFRM = TotflamclbFRM + flamyesnoclbFRM
Totflamcrz1FRM = Totflamcrz1FRM + flamyesnocrz1FRM
Totflamcrz2FRM = Totflamcrz2FRM + flamyesnocrz2FRM
Totflamcrz3FRM = Totflamcrz3FRM + flamyesnocrz3FRM
TotflamdesFRM = TotflamdesFRM + flamyesnodesFRM
```

```
TotflampostFRM = TotflampostFRM + flamyesnopostFRM

Totflamgrd80FRM = Totflamgrd80FRM + flamyesnogrd80FRM
Totflamclb80FRM = Totflamclb80FRM + flamyesnoclb80FRM
Totflamcrz180FRM = Totflamcrz180FRM + flamyesnocrz180FRM
Totflamcrz280FRM = Totflamcrz280FRM + flamyesnocrz280FRM
Totflamcrz380FRM = Totflamcrz380FRM + flamyesnocrz380FRM
Totflamdes80FRM = Totflamdes80FRM + flamyesnodes80FRM
Totflampost80FRM = Totflampost80FRM + flamyesnopost80FRM

499

If Flights < 5000 Then
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 1) = flightnumber
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 2) = tbf
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 3) = Flighttime
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 4) = Tgrd
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 5) = tcrz
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 6) = Tflashpt
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 7) = curflamm
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 8) = (curflamm / tmission) *
100
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 11) = Tgrdland
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 12) = tcrzend


Sheets("Summary of n Cases").Cells((flightnumber) + 5, 9) = curflammFRM
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 10) = (curflammFRM /
tmission) * 100
Sheets("Summary of n Cases").Cells((flightnumber) + 5, 13) = flightnumber




Else: GoTo 425
End If
425
flammtime = 0 'Resets flammtime to zero for next case


flamyesnogrd = 0
flamyesnoclb = 0
flamyesnocrz1 = 0
flamyesnocrz2 = 0
flamyesnocrz3 = 0
```

flamyesnodes = 0
flamyesnopost = 0
tbf = 0
tclb = 0
tcrz1 = 0
tcrz2 = 0
tcrz3 = 0
tdes = 0
Taft = 0
flamyesnogrd80 = 0
flamyesnoclb80 = 0
flamyesnocrz180 = 0
flamyesnocrz280 = 0
flamyesnocrz380 = 0
flamyesnodes80 = 0
flamyesnopost80 = 0
timegrd80 = 0
timeclb80 = 0
timecrz180 = 0
timecrz280 = 0
timecrz380 = 0
timedes80 = 0
timepost80 = 0

flamyesnogrdFRM = 0
flamyesnoclbFRM = 0
flamyesnocrz1FRM = 0
flamyesnocrz2FRM = 0
flamyesnocrz3FRM = 0
flamyesnodesFRM = 0
flamyesnopostFRM = 0

flamyesnogrd80FRM = 0
flamyesnoclb80FRM = 0
flamyesnocrz180FRM = 0
flamyesnocrz280FRM = 0
flamyesnocrz380FRM = 0
flamyesnodes80FRM = 0
flamyesnopost80FRM = 0

Next flightnumber

```
'Section to output data by flight phase


        Sheets("FRM").Cells(50, 5) = Tottimegrd
        Sheets("FRM").Cells(51, 5) = Tottimeclb
        Sheets("FRM").Cells(52, 5) = Tottimecrz1
        Sheets("FRM").Cells(53, 5) = Tottimecrz2
        Sheets("FRM").Cells(54, 5) = Tottimecrz3
        Sheets("FRM").Cells(55, 5) = Tottimedes
        Sheets("FRM").Cells(56, 5) = Tottimepost


        Sheets("FRM").Cells(50, 6) = Totflamgrd
        Sheets("FRM").Cells(51, 6) = Totflamclb
        Sheets("FRM").Cells(52, 6) = Totflamcrz1
        Sheets("FRM").Cells(53, 6) = Totflamcrz2
        Sheets("FRM").Cells(54, 6) = Totflamcrz3
        Sheets("FRM").Cells(55, 6) = Totflamdes
        Sheets("FRM").Cells(56, 6) = Totflampost

        Sheets("FRM").Cells(50, 11) = Tottimegrd80
        Sheets("FRM").Cells(51, 11) = Tottimeclb80
        Sheets("FRM").Cells(52, 11) = Tottimecrz180
        Sheets("FRM").Cells(53, 11) = Tottimecrz280
        Sheets("FRM").Cells(54, 11) = Tottimecrz380
        Sheets("FRM").Cells(55, 11) = Tottimedes80
        Sheets("FRM").Cells(56, 11) = Tottimepost80


        Sheets("FRM").Cells(50, 12) = Totflamgrd80
        Sheets("FRM").Cells(51, 12) = Totflamclb80
        Sheets("FRM").Cells(52, 12) = Totflamcrz180
        Sheets("FRM").Cells(53, 12) = Totflamcrz280
        Sheets("FRM").Cells(54, 12) = Totflamcrz380
        Sheets("FRM").Cells(55, 12) = Totflamdes80
        Sheets("FRM").Cells(56, 12) = Totflampost80


        Sheets("FRM").Cells(71, 5) = Tottimegrd
        Sheets("FRM").Cells(72, 5) = Tottimeclb
        Sheets("FRM").Cells(73, 5) = Tottimecrz1
        Sheets("FRM").Cells(74, 5) = Tottimecrz2
        Sheets("FRM").Cells(75, 5) = Tottimecrz3
        Sheets("FRM").Cells(76, 5) = Tottimedes
```

```
        Sheets("FRM").Cells(77, 5) = Tottimepost


        Sheets("FRM").Cells(71, 6) = TotflamgrdFRM
        Sheets("FRM").Cells(72, 6) = TotflamclbFRM
        Sheets("FRM").Cells(73, 6) = Totflamcrz1FRM
        Sheets("FRM").Cells(74, 6) = Totflamcrz2FRM
        Sheets("FRM").Cells(75, 6) = Totflamcrz3FRM
        Sheets("FRM").Cells(76, 6) = TotflamdesFRM
        Sheets("FRM").Cells(77, 6) = TotflampostFRM

        Sheets("FRM").Cells(71, 11) = Tottimegrd80
        Sheets("FRM").Cells(72, 11) = Tottimeclb80
        Sheets("FRM").Cells(73, 11) = Tottimecrz180
        Sheets("FRM").Cells(74, 11) = Tottimecrz280
        Sheets("FRM").Cells(75, 11) = Tottimecrz380
        Sheets("FRM").Cells(76, 11) = Tottimedes80
        Sheets("FRM").Cells(77, 11) = Tottimepost80


        Sheets("FRM").Cells(71, 12) = Totflamgrd80FRM
        Sheets("FRM").Cells(72, 12) = Totflamclb80FRM
        Sheets("FRM").Cells(73, 12) = Totflamcrz180FRM
        Sheets("FRM").Cells(74, 12) = Totflamcrz280FRM
        Sheets("FRM").Cells(75, 12) = Totflamcrz380FRM
        Sheets("FRM").Cells(76, 12) = Totflamdes80FRM
        Sheets("FRM").Cells(77, 12) = Totflampost80FRM



'added algorithm to output flight length mixture
For MLstep = 0 To 52
Sheets("Internal Calc 3").Cells(MLstep + 2, 14) = MLarray(MLstep, 3)
Next MLstep
444
'Sheets("Internal Calculations").Cells(16, 20).Value = Time
Sheets("Summary of n Cases").Select
Datasort
Sheets("User inputs and results").Select
'timeout = Time
'Sheets("Internal Calculations").Cells(17, 19).Value = timeout
'Application.ScreenUpdating = True

End Sub
```

---

```
Sub Datasort()
```

```
'
' Datasort Macro
' Macro recorded 2/22/2002 by NM117NIT
'

'
    Range("a5:l1005").Select
    ActiveWindow.ScrollRow = 1
    Selection.Sort Key1:=Range("H6"), Order1:=xlDescending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom
        Range("m1").Select

End Sub
```

---

```
Sub pickedflight()

Dim fltnumber As Long
Dim Flighttime As Single
Dim Groundtemp As Single
Dim Crztemp As Single
Dim Flashpoint As Single
Dim tcrzend As Single
Dim Tgrdland As Single
fltnumber = Sheets("Internal Calculations").Cells(16, 8)
Flighttime = Sheets("Summary of n Cases").Cells(fltnumber + 5, 3)
Groundtemp = Sheets("Summary of n Cases").Cells(fltnumber + 5, 4)
Crztemp = Sheets("Summary of n Cases").Cells(fltnumber + 5, 5)
Flashpoint = Sheets("Summary of n Cases").Cells(fltnumber + 5, 6)
tcrzend = Sheets("Summary of n Cases").Cells(fltnumber + 5, 12)
Tgrdland = Sheets("Summary of n Cases").Cells(fltnumber + 5, 11)


Sheets("Single Flight").Cells(7, 5) = Flighttime
Sheets("Single Flight").Cells(8, 5) = Groundtemp 'Sheets("summary of n
Cases").Cells(fltnumber + 5, 4)
Sheets("Single Flight").Cells(9, 5) = Crztemp 'Sheets("summary of n
Cases").Cells(fltnumber + 5, 5)
Sheets("Single Flight").Cells(12, 5) = Flashpoint 'Sheets("summary of n
Cases").Cells(fltnumber + 5, 6)
Sheets("Single Flight").Cells(10, 5) = tcrzend
Sheets("Single Flight").Cells(11, 5) = Tgrdland

Sheets("Internal Calculations").Cells(2, 11) = 1
montecarlo
Sheets("Single Flight").Select
Sheets("Single Flight").Range("a1").Select
```

```
'Sheets("Single Flight").Cells(17, 11) = Sheets("Summary of n Cases").Cells(fltnumber +
5, 2)
'Sheets("Single Flight").Cells(17, 12) = Sheets("Summary of n Cases").Cells(fltnumber +
5, 3)
'Sheets("Single Flight").Cells(17, 13) = Sheets("Summary of n cases").Cells(fltnumber +
5, 7)

End Sub
```