

**ALGORITHMS FOR INCORPORATION OF DYNAMIC RECOVERY IN
ESTIMATING FREQUENCY OF CRITICAL STATION BLACKOUT**

A Thesis

by

PAUL J. RODI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2012

Major Subject: Nuclear Engineering

Algorithms for Incorporation of Dynamic Recovery in Estimating Frequency of Critical

Station Blackout

Copyright 2012 Paul J. Rodi

**ALGORITHMS FOR INCORPORATION OF DYNAMIC RECOVERY IN
ESTIMATING FREQUENCY OF CRITICAL STATION BLACKOUT**

A Thesis

by

PAUL J. RODI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Paul Nelson Jr.
Committee Members,	Pavel V. Tsvetkov
	Graham Donald Allen
Head of Department,	Yassin A. Hassan

May 2012

Major Subject: Nuclear Engineering

ABSTRACT

Algorithms for Incorporation of Dynamic Recovery in Estimating Frequency of Critical Station Blackout. (May 2012)

Paul J. Rodi, B.S., Virginia Commonwealth University

Chair of Advisory Committee: Dr. Paul Nelson Jr.

This thesis involves exploring enhancement of estimating the probability of a critical station blackout in nuclear power plant operations by the use of direct numerical evaluation of multidimensional nonrecovery integrals. This requires development of computational methods with data provided from South Texas Project Nuclear Operating Company (STPNOC). Several methods that are currently used in the industry to estimate such probabilities often overestimate the value substantially. The computational integral method developed in the thesis will reduce excess conservatism while maintaining plant safety standards. This computational integral is calculated using a MATLAB research code referred to generally as “STP-TAMIL” which is for South Texas Project – Texas A&M Improved LOOP. The code itself (along with the user manual) was developed in conjunction with this Thesis. STP-TAMIL is successful in reducing the estimated probability of critical station blackout by a significant amount (about 88.47%) with the incorporation of recovery of offsite and onsite power for South Texas Project’s nuclear plants, and results were verified. This thesis also describes an asymptotic justification for to the non-recovery integral used. Applications to the industry, or STPNOC, which will use the “TAMIL” code are addressed. Some assumptions used throughout the problem suggest that if more dynamic rates or distributions are used then more recovery can be obtained, which will decrease the probability of critical station blackout. Methodology developed in this thesis will be used in future work to develop this STP-TAMIL research code into a model used industry wide in commercial nuclear power plants.

DEDICATION

I would like to dedicate this work to my father, Paul James Rodi Jr. He has always inspired the way I choose to live my life and my career path. My dad helped me make my first computer at the age of three, create a program at five, and build competitive robots all throughout grade school. Camping with me at the nuclear power plants, taking tours of coal plants, letting me trip hydro-electric dams, and showing me around the nuclear submarines growing up did not hurt either. I hope this work will make you proud.

I would also like to dedicate this thesis to the memory of LCpl. Karl Linn, whose ultimate sacrifice has always been an inspiration for me. As a teammate and co-founder of Synergy Robotics, an engineer from Virginia Commonwealth University, a marine, and a friend, he will always be remembered.

ACKNOWLEDGEMENTS

I would like to thank South Texas Project Nuclear Operating Company for all their support in the creation and development of this thesis. I especially want to note the assistance of Shawn Rodgers, Coral Betancourt, and Ernie Kee. Their continued assistance has been invaluable. In addition, I wish to thank Dr. Paul Nelson for his support and encouragement throughout the project. The time he has taken to advise me these past two years have been truly appreciated. I want to express my thanks to the committee members, Dr. G. Donald Allen and Dr. Pavel Tsvetkov, for their advice and assistance. Dr. Tsvetkov's classes and labs have been so helpful to me in gaining the knowledge needed to perform this thesis work. The professors at Texas A&M University have been exceptional in knowledge and teaching ability.

I would like to thank all of my professors from Virginia Commonwealth University. Most notably, I would like to express my thanks to Dr. Ross Anderson for his assistance with my senior design project along with instruction in Probabilistic Risk Assessment, and to Dr. Afroditi V. Filippas for advising in both my senior design and da Vinci Center Projects.

I would like to thank my co-workers from Dominion-Virginia Power most especially Ed Shore, who allowed me to take on real engineering assignments during my undergraduate years. I would like to thank my high school physics teacher, Mr. Timothy Couillard, for his hours of unpaid volunteer work in mentoring the robotics team, Synergy #975. He was one of the few teachers in high school who really inspired creativity and learning.

I would also like to thank my family who taught me the values of working together as a team and promoting excellence.

I would most importantly like to thank God for everything he has given me and strength to continue with my education.

“A wise heart shall acquire knowledge: and the ear of the wise seeketh instruction”

(Proverbs 18:15)

NOMENCLATURE

2CCF	Common-Cause Doublet Failure
3CCF	Common-Cause Triplet Failure
AC	Alternating Current
CCF	Common-Cause Failure
CDF	Core Damage Frequency
cdf	Cumulative Distribution Function
CSBO	Critical Station Blackout
DC	Direct Current
EDG	Emergency Diesel Generator
icdf	Inverse Cumulative Distribution Function
IE	Initiating Event
LOOP	Loss of Offsite Power
NPP	Nuclear Power Plant
NRC	Nuclear Regulatory Commission
pdf	Probability Density Function
PRA	Probabilistic Risk Assessment
SBO	Station Blackout
STPNOC	South Texas Project Nuclear Operating Company
STP-TAMIL	South Texas Project Texas A&M Improved LOOP

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
NOMENCLATURE	vi
TABLE OF CONTENTS.....	vii
LIST OF FIGURES	x
LIST OF TABLES.....	xi
CHAPTER I INTRODUCTION.....	1
Section 1.1 - Objective	1
Section 1.2 - Background	2
Subsection 1.2.a – Significance of Critical Station Blackout	2
Subsection 1.2.b – Existing Methods for Estimating Probability of CSBO	4
Section 1.3 - Overview	8
CHAPTER II STP-TAMIL USER MANUAL.....	12
Section 2.1 – Version Information	12
Section 2.2 – Description	13
Section 2.3 – Development.....	13
Section 2.4 – Installation	13
Section 2.5 – Building the Input Deck	14
Subsection 2.5.a – Details for Types of Distributions	23
Subsection 2.5.b – Entering a cdf Type of Distribution.....	26
Subsection 2.5.c – Entering an Inverse cdf Type of Distribution	26
Subsection 2.5.d – Entering a pdf Type of Distribution.....	27
Subsection 2.5.e – Creating a New MATLAB “.m” File for Distributions	27
Section 2.6 – Running STP-TAMIL	28
Section 2.7 – Output File.....	35
Section 2.8 – End Notes	36
CHAPTER III THEORY	37
Section 3.1 –Setting Up the Problem.....	37
Subsection 3.1.a – Simplified Two-EDG Markov Model.....	38
Subsection 3.1.b – Advanced Three-EDG Markov Model	41
Subsection 3.1.c – List of Variables.....	43
Section 3.2 – State-Transition Equations for Two-EDG Case	45
Section 3.3 – Asymptotic Solution for Two-EDG Case.....	46
Subsection 3.3.a – For the n=0 Case for Two-EDG Case.....	48
Subsection 3.3.b – For the n=1 Case for Two-EDG Case	49
Subsection 3.3.c – For the n=2 Case for Two-EDG Case.....	52

	Page
Section 3.4 – Interpretation as a Nonrecovery Integral for Two-EDG Case.....	53
Section 3.5 – Common-Cause Failure Contributions for Two-EDG Case	56
Section 3.6 – The General Form for the Two-EDG Case	59
Section 3.7 – For the Three-EDG Case	60
Section 3.8 – Advanced Three-EDG State-Transition Differential Equations	65
CHAPTER IV APPLICATIONS.....	67
Section 4.1 – Types of Recovery	67
Subsection 4.1.a – No Credit (1).....	68
Subsection 4.1.b – Offsite Only Credit (2).....	68
Subsection 4.1.c – Offsite Credit and Onsite Only After Full Failure (3)	68
Subsection 4.1.d – Full Credit of Offsite and Onsite Failure (4)	69
Section 4.2 – Number of Trains	69
Section 4.3 – Mission and Critical Times	70
Section 4.4 – CC Triplet, CC Doublet, Independent Failures	71
Section 4.5 – Distribution Types	71
Section 4.6 – Repair	72
Section 4.7 – Equation Types	73
CHAPTER V RESULTS AND BENCHMARKING.....	74
Section 5.1 – Verification of STPQUAD2	75
Subsection 5.1.a – No Credit for Recovery (Offsite or Onsite) for STPQUAD2	78
Subsection 5.1.b – Dynamic Credit for Offsite Recovery (Only) for STPQUAD2... ..	80
Subsection 5.1.c – Dynamic Offsite Recovery, Onsite Repair Post Failure (STPQUAD2).....	84
Subsection 5.1.d – Comparison of Methods (STPQUAD2)	85
Section 5.2 –Verification of STPQUAD3 (Without Common-Cause)	86
Subsection 5.2.a – No Credit for Recovery (Offsite or Onsite) for STPQUAD3	90
Subsection 5.2.b – Dynamic Credit for Offsite Recovery (Only) for STPQUAD3... ..	91
Subsection 5.2.c – Dynamic Offsite Recovery, Onsite Repair Post Failure (STPQUAD3).....	97
Subsection 5.2.d – Comparison of Methods (STPQUAD3)	98
Section 5.3 – Verification of STPQUAD3CCF (with Common-Cause)	101
Subsection 5.3.a – 3/3 Common-Cause Failures.....	102
Subsection 5.3.b – Inclusion of Independent Failures	107
Subsection 5.3.c – 2/3 Common-Cause Failures.....	113
Subsection 5.3.d – Industrial Average Benchmark	117
Subsection 5.3.e – Alpha Factor Method	117
Subsection 5.3.f – Values Used in Industrial Average Case	124
Section 5.4 –Summary of Verified Results	131
CHAPTER VI IMPLEMENTATION	133
Section 6.1 – STPNOC	133
Section 6.2 – Other Commercial Nuclear Operators.....	133

	Page
CHAPTER VII SUMMARY AND CONCLUSIONS	134
Section 7.1 – Summary.....	134
Section 7.2 – Significance	135
Section 7.3 – Future Work.....	136
REFERENCES	138
APPENDIX A STP-TAMIL CODE	141
Appendix A.1 – STPQUAD2 Code Package	143
Appendix A.2 – STPQUAD3 Code Package	148
Appendix A.3 – STP-TAMIL (STPQUADCCF) Code Package	155
APPENDIX B RUNFILE EXAMPLE	195
APPENDIX C LETTER FROM STPNOC.....	202
VITA.....	204

LIST OF FIGURES

FIGURE	Page
2.1 Example of RunFile with Generic Data Used.....	15
2.2 Example of How to Run MATLAB from the Editor Window	29
2.3 Example of Change Folder Prompt in MATLAB.....	30
2.4 Example of Program Working and Running the Code	31
2.5 Example of Filling Default Cells Prompt in MATLAB	32
2.6 Example of No Empty Cells Prompt in MATLAB	32
2.7 Example of Code Working and Finishing Showing “P_CSBO”	33
2.8 Example of Code Working and Finishing Showing Multiple “P_CSBO”s	34
2.9 Example of OutputFile After Code is Executed	36
3.1 State-Transition Diagram for the Two-EDG Markov Model	39
3.2 State-Transition Diagram for the Three-EDG Markov Model	42
5.1 Complemented Cumulative Distribution Function for One Month (30 Days), for the Lognormal Distribution Employed Here for Offsite Recovery	100
5.2 Various cdfs for Time of SBO, Conditional Upon SBO Associated to Specific Types of EDG Failures Occurring During a 24-Hour Mission Time	112
B.1 First Part of RunFile Example	195
B.2 Second Part of RunFile Example.....	195
B.3 Third Part of RunFile Example.....	196
B.4 Fourth Part of RunFile Example.....	196
B.5 Fifth Part of RunFile Example.....	197
B.6 Sixth Part of RunFile Example	197
B.7 Seventh Part of RunFile Example.....	198
B.8 Eighth Part of RunFile Example	198
B.9 Ninth Part of RunFile Example.....	199
B.10 Tenth Part of RunFile Example	199
B.11 Eleventh Part of RunFile Example	200
B.12 Twelfth Part of RunFile Example	200
B.13 Thirteenth Part of RunFile Example	201

LIST OF TABLES

TABLE	Page
1.1 Nonrecovery Integral Functions	10
2.1 First Part of Title Names and Descriptions for RunFile	16
2.2 Second Part of Title Names and Descriptions for RunFile	17
2.3 Third Part of Title Names and Descriptions for RunFile	18
2.4 Fourth Part of Title Names and Descriptions for RunFile	19
2.5 Fifth Part of Title Names and Descriptions for RunFile	20
2.6 Sixth Part of Title Names and Descriptions for RunFile	21
2.7 First Part of Function Names and Parameters for Type of Distributions for CDF	24
2.8 Second Part of Function Names and Parameters for Type of Distributions for CDF	25
3.1 First Part of Variables and Definitions for the Derivation of the Non-Recovery Integral.....	44
3.2 Second Part of Variables and Definitions for the Derivation of the Non-Recovery Integral.....	45
3.3 List of the Minimal Cut Sets.....	63
5.1 Results for STPQUAD2.....	76
5.2 Data Employed in [22], as Used Here for STPQUAD2 Verification.....	78
5.3 PCSBO Values While Varying Absolute Error Tolerances with Case C vs Case D (STPQUAD2).....	83
5.4 PCSBO Values While Varying Absolute Error Tolerances with Case E vs Case F (STPQUAD2).....	85
5.5 Comparison of Estimates of PCSBO from Various Methods (STPQUAD2)	86
5.6 Results for STPQUAD3.....	87
5.7 Data Employed in [22], as Used Here for STPQUAD3 Verification.....	89
5.8 pdf and cdf Terms (STPQUAD3).....	93
5.9 PCSBO Values While Varying Absolute Error Tolerances with Case C vs Case D (STPQUAD3).....	95
5.10 PCSBO Values While Varying Absolute Error Tolerances with Case C vs Case D with QUADL (STPQUAD3).....	96

TABLE	Page
5.11 Comparison of Estimates of PcsBO from Various Methods (STPQUAD3)	99
5.12 Top Four Cut Sets in the STP PRA Model	102
5.13 Probability of Critical Station Blackout from Common-Cause Triplet Failure of EDGs or Associated Fans, Given a LOOP Initiating Event has Occurred.....	105
5.14 Important Cut Sets for Independent Failure Based on the Industry Average	107
5.15 Probabilities of Critical Station Blackout from Common-cause Triplet Failure of EDGs or Associated Fans, and from Independent Failures of the EDGs, Given a LOOP Event.	110
5.16 Important Cut Sets for CC Doublet Failure Based on the Industry Average	114
5.17 Probabilities of CSBO from Common-cause Triplet, CC Doublet or Independent Failures of EDGs Along with the Total, Given a LOOP Event	116
5.18 NUREG-6929 Component Type Failure Averages [24]	125
5.19 Common-Cause Component Database Alpha Factors (CCCG=3) [18, 19]	126
5.20 Cut Set Probabilities and Totals	128
5.21 Difference Between Pseudo-PRA and Code Results.....	129
5.22 Probabilities of CSBO from the Code Using the Industrial Average and Pseudo-PRA Model	130
A.1 M-File Names and Descriptions for STPQUAD2 Verification.....	141
A.2 M-File Names and Descriptions for STPQUAD3 Verification.....	142
A.3 M-File Names and Descriptions for STP-TAMIL (STPQUADCCF Verification).....	143

CHAPTER I

INTRODUCTION

This chapter will discuss the significance of the elimination of excessive conservatism in critical station blackout, reasons for studying it, and the driving forces behind it as well as an outline of the goals that this thesis intends to achieve. This chapter will also contain a brief overview of the methods and techniques utilized to obtain the results. This chapter will be broken into three sections, Objective (Section 1.1), Background (Section 1.2), and Overview (Section 1.3).

Section 1.1 – Objective

The objective of this thesis is to develop, test, and document a MATLAB^a research code that incorporates and enhances a novel methodology to eliminate excessive conservatism in the evaluation of probability of critical station blackout in a nuclear power plant. The methodology employed involves direct numerical evaluation of multidimensional nonrecovery integrals, as developed in detail in work previously reported [1]. This approach will require development, verification and application of efficient computational methods for evaluation of multidimensional integrals, as will be documented in the thesis. Existing approaches used, or developed for use; in the industry involve various approximations to this approach. These estimations are generally excessively conservative. The code documented in this thesis provides the capability also to generate some of these previous overly conservative approximations to the probability of critical station blackout, thereby facilitating a comparison of results to those obtained via numerical evaluation of the multidimensional recovery integral.

This thesis follows the style of *Safety Science*.

^a MATLAB is a registered trademark of The MathWorks, Inc.; cf. MathWorks - Trademarks

Section 1.2 – Background

This section is divided into two subsections. In Subsection 1.2.a the event and importance of critical station blackout to a nuclear power plant (NPP) is summarized. In Subsection 1.2.b the present status of methods used in the nuclear industry to estimate the probability of critical station blackout is reviewed. This latter subsection serves as a baseline that any proposed new methodology must seek to improve upon from the current methodologies.

Subsection 1.2.a – Significance of Critical Station Blackout

As of 2012, there are 104 operating commercial nuclear power plants in the United States [2]. All of these are required by 10 CFR 50.69 [3] to develop and employ plant-specific probabilistic risk assessments (PRAs) to estimate the (typically very low) probability of a plant accident involving damage to the nuclear core. The event tree approach to PRA models begins with various initiating events (IEs). As analyzed by South Texas Project Nuclear Operating Company (STPNOC), 25% of the total core damage frequency comes from an initiating event known as a Loss of Offsite Power (LOOP). This LOOP IE leads the plant to lose all external power feeding into the plant from the electrical switchyard.

A nuclear power plant has several emergency safety systems that require power for full operation to work. U.S. plants have pumps to provide water to the reactor core even after it has been shut down; this makes up the emergency core cooling system. Cooling is necessary due to the build-up of decay heat, which immediately after reactor shutdown typically is about six to seven percent of total reactor power. The cooling systems can include different types of safety pumps and spray systems. High head and low head safety injection pumps help cool down the unit by injecting water straight into the reactor coolant system from the refueling water storage tank [4]. The containment spray pumps located at the top of the containment area spray down the reactor vessel with water into a sump area below where this water is recovered and reintroduced into the system. This not only cools down the reactor coolant system but also depressurizes it [4].

In case of a LOOP event, needed AC power can be restored using Emergency Diesel Generators (EDGs). Such generators normally provide the first line of defense against damage to the reactor core from the decay heat. There is some possibility that these generators could themselves fail. Failure causes could include, among others, a failure to start, failure to run, failure to load and run, or generator maintenance (preventive or corrective). If there is a common reason underlying failure of two or more of the EDGs, this is known as a common-cause failure. This could be due to multiple factors. One instance is a flood in the diesel generator room, which could cause a failure to start in both generators, thus increasing the core damage frequency (CDF) from the same accident scenario. Common-cause failures strike at the heart of the “defense in depth” philosophy that underlies the engineered approach to safety in NPPs, so must be carefully considered in PRA analyses for NPPs.

Operators also need lights, gauges, and computers, which normally run on offsite AC power, for instrumentation and control to help better understand and respond to plant events. With no AC power, a battery system can be used as a coping mechanism, but it only lasts for a short amount of time. [5]

As described in CR-6890 [6], power systems are divided into three separate classes. These include systems powered by offsite power (normal AC power), systems powered by Emergency Power Systems (EPS) that provide onsite AC power (such as the EDGs), and Station Blackout (SBO) coping systems that do not depend on electrical power trains (such as the decay-heat steam-turbine driven pump). Each case has an associated risk for core damage but if even one is still operating then the others can fail while the plant will be in a safe mode to shut down.

LOOP initiating events and PRA model are not specific to STPNOC, but rather are an important generic issue to US and international NPPs. In this regard, a recent United States Nuclear Regulatory Commission (NRC) report states that:

“... risk analyses performed for NPPs (nuclear power plants) indicate that the loss of all AC power can be a significant contributor to the risk associated with plant operation, contributing more than 70 percent of the overall risk at some plants. Therefore, a loss of offsite power (LOOP) and its subsequent restoration are important inputs to plant risk models, and these inputs must reflect current industry performance in order for

plant risk models to accurately estimate the risk associated with LOOP-initiated scenarios.” [6]

The critical station blackout (CSBO) probability is defined as the probability that during a LOOP IE, within some mission time (typically ~ 24 hours), the plant will experience a loss of all emergency power for a time exceeding some critical time. Mission time is an upper limit on the amount of time that it takes to restore outside power, which effectively ends the loss of the offsite power event. The critical time is the maximum amount of time that a component can be offline before damage to the core is inevitable. It is generally assumed that within 24 hours, workers can bring in another component that will replace that component; therefore, disregarding the failure of the original components effect on the CDF. In the recent developments at Fukushima Dai-ichi, this expected 24 hour value used as a mission time did not adequately address the situation. No access by road and lack of infrastructure worsened the critical incident post tsunami and prevented protective actions from being taken. The use of 24 hours as a mission time needs to be addressed by the industry; and until that is done, a logical value will be placed in this report. The Fukushima accident also was the driving force behind a NRC task force that looked at ways to enhance nuclear reactor safety. One recommendation given was that the NRC should require emergency plans to address SBO and multiunit events [7]. If implemented, this recommendation could have a great effect on how station black outs are evaluated by type and associated mission time.

Subsection 1.2.b – Existing Methods for Estimating Probability of CSBO

As discussed in more detail in Rodgers *et al.* [1], CSBO means the following events have occurred:

1. A LOOP event occurs at some initial time, which typically is taken as $t = 0$.
2. Offsite power is not recovered within a subsequent time equal to the mission time, denoted as T .

3. At some time within this same mission time, all emergency diesel generators (EDGs) (or other sources of emergency power) become unavailable, and remain so within a subsequent elapsed time equal to the designated critical time.

Data related to frequency of occurrence of LOOP events [5], to rate of recovery of offsite power [8], to various sources of unavailability for diesel generators [9] and to expected repair rate for EDGs [10] are available. Therefore, it is possible, at least in principle, to calculate the frequency (probability per unit time) for occurrence of CSBO.

A digression is appropriate here to note that equivalence of CSBO to the conjunction of the preceding list of events invokes a weak form of the rare events approximation, to the effect that it ignores the possibility of CSBO occurring through events that involve any single EDG failing, being repaired, and subsequently failing again, within the designated mission time. Should at some point mission times become sufficiently large, it will be necessary to evaluate this assumption more critically; but for the present work, it is assumed to be valid.

To return to the main thread, varieties of approximations that conservatively overestimate probability of CSBO are conceivable. These are discussed in this subsection, beginning with the most obvious, which tend to be grossly overconservative, and ending with more complex schemes that are currently applied in the industry. All of these achieve conservatism by either ignoring or underestimating the impact of recovery, either of offsite power or of onsite power (i.e., repair of EDGs), or both.

The most obvious overconservative approximation is simply to ignore the effect of recovery, of either offsite or onsite power. Today, this approach is seldom used in the industry, but it is nonetheless useful as a measure of gains attained by employing improved methods of estimation that nonetheless remain conservative.

For more complex (and therefore realistic) approaches, note that probability of recovery (of either offsite power or a failed EDG) is a function of elapsed time since failure and, for onsite power, of the repair doctrine at the particular plant.

The most commonly used way to determine risk in a plant is by the use of fault-trees, which are logic trees for a way systems can fail, and event trees, which are logic trees for the paths to core damage. This systematic approach sums up every possible cut set along with a path of probable scenarios that would lead to core damage within the plant to

create a total core damage probability. These cut sets include the initiating event probability along with each failure probability of a component and human error that would lead to core damage. This method insured that failure events are independent of each other but does not explicitly address time dependencies, and therefore recovery or repair is not fully addressed [11]. Time sequencing is considered in the present approach via recovery integrals or in other approaches to “dynamic reliability” [12].

In the “convolution integral” approach pioneered by Lloyd and Anoba [11], the offsite and emergency power train “nonrecovery factors” are given constant values, independent of the duration from time of failure of the particular system to onset of CSBO. In this method, the different cut sets are examined and each cut set has a specific sequence of events. The time during the mission for which the component might fail because of a “type-2” event (dependent of mission time) [11] is reduced by the time it takes for the prior component in the sequence to fail. For example, assuming a loss of offsite power and a single diesel generator failure, the two cut sets could be a possibility. In case one, the second emergency diesel generator would fail. Then the turbine driven pump, running off decay heat from the plant, would fail. In case two, the turbine driven pump would fail first, and then the second EDG would fail. Therefore, in case one the time is reduced for the turbine drive pump to fail by the time it took for the EDG to fail and vice versa in case 2 [11].

The general solution for the convolution integral provides the point values to the nonrecovery events consistent with the point-estimate of probabilities that underlies fault-tree analysis. In other words, core damage frequency that is calculated by the Lloyd and Anoba approach will be assumed to be close to that from a fault-tree approach. [11]

The values for nonrecovery of offsite power defined by Lloyd and Anoba and the fault-tree method used are overconservative. Lloyd and Anoba, themselves, note the following in regard to the offsite nonrecovery term:

“.. it must be supplied by the analyst. If the term has a mission-time dependence, establishing a probability value for the term may be difficult. As a result, this value may be significantly overconservative.” [11]

One way of approaching this issue would be to take as the nonrecovery factor that corresponds to the critical time as the elapsed time between failure and core damage. One always has at least that much time to recover either offsite or onsite power.

Lloyd [8] later demonstrated how to incorporate the dynamic variation in offsite recovery into the solution. The solution assumes a constancy of the nonrecovery factors only for the emergency power trains.

An alternative method developed by Read and Fleming [14] is conservative in that the approach does not take into account any repair work on any of the emergency trains until all trains have failed. This conservatism is somewhat reduced due to the parallel “infinite workforce” assumption that once repairs start, the repairs start on all trains simultaneously with equal effectiveness [15]. With Lloyd and Anoba this repair conservatism is negated somewhat by the use of the nonrecovery time distribution that “applies to situations involving a high urgency for diesel generator repairs” [15].

The Read-Fleming method is equivalent to taking the (onsite) recovery factor in the Lloyd-Anoba approach equal to that at the critical time. This relatively simple fact does not seem to have previously been explicitly noted in the literature.

An alternative method using the Markov chain model has also been proposed by STPNOC for finding the CSBO [5]. The Markov model addressed by STPNOC was used only as a tool to see how different conservatisms would respond which could determine how the CSBO probabilities differed but not give exact solutions as it was not intended of having that degree of realism. It was determined that the approach could take into account both the failure and repair factors of the EDGs as compared to the convolution integral method that assumes that once an EDG fails it could not be repaired. This also shows that if the diesel repair rate is determined by the time of diesel failure, then only the offsite power recovery factor is a function of calendar time. This is due to repair time being a function of time required for workers to arrive at the scene and effect repairs. However, due to this clock only really being determined by standard clock, the diesel repair rate is set to a constant rate [5].

Section 1.3 – Overview

Current models for LOOP recovery are very conservative because of the limited credit taken for the various possible forms of recovery of different previously failed sources of power. The models in existence today also take into account static reliability as opposed to a more dynamic approach for recovery. The approach implemented in the MATLAB research code that will be described in the proposed thesis seeks to remove this excess conservatism by taking full credit for any recovery of (onsite or offsite) power that occurs following onset of the hypothesized LOOP event, but prior to a time period equal to the critical time elapsing following loss of all forms of power – that is, prior to CSBO. The MATLAB code will be referred to generally as “STP-TAMIL” which is for South Texas Project – Texas A&M Improved LOOP. Use of this approach could permit plant resources to be directed toward more realistic safety concerns while maintaining a low core damage probability needed for operation of a nuclear power plant.

This full credit for recovery will be implemented by numerical evaluation of the “nonrecovery integral” found in Equation (1.1).

$$\begin{aligned}
 F_{CSBO}(T) &= \int_0^T \int_0^T \int_0^T \left[1 - G(\max\{\tau_1, \tau_2, \tau_3\} + T_c) \right] \prod_{j=1}^3 \left[\frac{1 - R(\max\{\tau_1, \tau_2, \tau_3\})}{+T_c - \tau_j; \tau_1, \tau_2, \tau_3} \right] dF(\tau_1) dF(\tau_2) dF(\tau_3) \\
 &+ 3 \int_0^T \int_0^T \left[1 - G(\max\{\tau_{12}, \tau_3\} + T_c) \right] \left[1 - R(\max\{\tau_{12}, \tau_3\} + T_c - \tau_{12}; \tau_{12}, \tau_3) \right]^2 \times \\
 &\left[1 - R(\max\{\tau_{12}, \tau_3\} + T_c - \tau_3; \tau_{12}, \tau_3) \right] dF_2(\tau_{12}) dF(\tau_3) + \\
 &\int_0^T \left[1 - G(\max\{\tau_{123} + T_c) \right] \left[1 - R(T_c; \tau_{123}) \right]^2 dF_3(\tau_{123}) \\
 &= \text{Prob}\{T_{CSBO} \leq T\}, \text{ for } T \geq 0.
 \end{aligned} \tag{1.1}$$

Here F_{CSBO} is the cumulative density function (cdf) for the occurrence of critical station blackout, F is the cdf for independent failure of the EDGs, F_{12} is the cdf for 2/3 common-cause failures of the first and second EDGs, F_3 is the cdf for 3/3 common-cause failure of the EDGs, G is the cdf for recovery of offsite power, R is the cdf for repair of

an EDG, T is the critical time; the value under currently applicable U.S. regulations is 24 hours, T_c is the critical time, nominally about one hour, τ_i is dummy variable representing the (stochastic) time of failure of the i th EDG, τ_{12} is a dummy variable representing the (stochastic) time of common-cause failure of the first and second EDGs, τ_{123} is a dummy variable representing the (stochastic) time of failure of the i th EDG.

The actual evaluation of Equation (1.1) will be made in the code using numerical methods to evaluate the “nonrecovery integral” that is the last line of this equation. A significant element of the research component of this report will be determining the computational effort and technology appropriate in evaluation of this multidimensional nonrecovery integral.

The functions in the integrand of the integral, Equation (1.1), will be obtained from a combination of different sources. Some functions will come from data analyzed by all of the commercial nuclear power plants during loss of offsite power events which will come from Nuclear Regulatory Commission Condition Report 6890 [6] or other industry sources. Other data in this integrand will come from STPNOC, which in turn comes from operating licenses, experimental data, and analytical calculations. The functions, which are not directly called for, i.e. mission time, will have to come from engineering judgment, experience and adding in conservatisms as no definitive standards are available.

All of the functions used in the nonrecovery integral along with associated meanings and sources are compiled in Table 1.1.

Table 1.1

Nonrecovery Integral Functions.

Function (meaning)	Data Source	Reference
G (cdf for offsite recovery)	Section III, esp. Tables 3-1 and 3-2, of NUREG/CR-6890, Vol. 2.	[5]
F (cdf for independent failure of the EDGs)	Section III, esp. Table 3-3 of NUREG/CR-6890, Vol. 2.	[8]
R (cdf for repair of the EDGs)	Page 31 of NUREG/CR-6890, Vol. 2.	[5]
f (pdf for failure of the EDGs)*	Tables 4-2, of NUREG/CR-6890, Vol. 2.	[10]

* Not directly shown in Equation (1.1) but used in its asymptotic justification

The structural plan for this thesis is to include the following chapters. This introductory chapter, Chapter I, will discuss significance of project, reasons for studying it, and the driving forces behind it as well as an outline of the goals that the thesis achieves. This chapter will also contain a brief overview of the methods and techniques utilized to obtain the results. Chapter II will be a user manual for the STP-TAMIL code. This will act to serve as a readme for setting up the program, how to use the program and associated input file, and what limitations are found in the program. Chapter III, theory, will include a formal statement of the problem, and an asymptotic approach for the derivation of the nonrecovery integral as an approximate solution to that problem. Chapter IV, applications, will discuss special cases that cause deviations in results which will be discussed along with the use of different equations to determine results. The differences can included but are not limited to how many safety trains are used, what type of repair is executed, and what recovery methods are implemented, what time in the accident repairs begin, and how much man power is needed for the repairs. Common-cause failure implementation will also be discussed in the Chapter IV. Chapter V, benchmarking results, will outline the results and there will be a discussion of how these results are realistic, meaningful, and useable. This chapter will involve benchmarking to provide verification of results. An implementation chapter, Chapter VI, will then be used

to discuss how this could be implemented into the PRA models at STPNOC as well as around the nation. The final chapter, Chapter VII, will summarize the project and give suggestions for improvement. In this concluding chapter, the significance of the problem and suggestions for resolution will be evaluated along with a plan for future action.

The Appendixes listed will include the code itself (Appendix A) and an example of the input deck used to execute the program in its final version (Appendix B). Appendix C will contain a letter from STPNOC discussing confidentially of results as well as rights to use the differences observed by the results.

CHAPTER II

STP-TAMIL USER MANUAL

This chapter is intended to serve as the user manual for the STP-TAMIL code. The general information in this user manual is written as a standalone document from this thesis, and therefore is presented as such in this chapter. However, for advanced users there are pointers to the thesis for more information.

This manual acts to:

- a. serve as a readme for setting up the program,
- b. describe how to use the program and associated input file,
- c. describe what limitations are found in the program, and
- d. explain how to read the output file.

Section 2.1 refers to version information and distribution rights control. Section 2.2 talks about the description of the code while Section 2.3 discusses the development of the code. Section 2.4 discusses the set-up and installation of STP-TAMIL. Section 2.5 talks about how to build the input deck or “RunFile.xlsx” for the program. This also has subsections that discuss the types of distributions that can be used as well as how to set up a custom distribution program. Section 2.6 explains how to run the code itself from inside the MATLAB environment, while Section 2.7 discusses for to read the results or output file generated from the code. End notes for the code are found in Section 2.8.

Section 2.1 – Version Information

This user manual is for version 1.0 of STP-TAMIL developed during the Spring of 2012.

The distribution and usage rights of this code belong to Dr. Paul Nelson and Paul Rodi of Texas A&M University. Please contact Dr. Nelson at “p-nelson@tamu.edu” for questions on rights access. Technical questions for general use of the software can be directed towards Paul Rodi at “rodipj@vcu.edu”.

Section 2.2 – Description

This code is used to help eliminate excessive conservatism in the evaluation of the probability of critical station blackout in a commercial nuclear power plant.

The code has been written in the MATLAB language for use in the MATLAB (Version 7.10.0 R 2010a) environment. This is ***required*** for running the code correctly, as it is intended. However, the Excel input file can be sent to a MATLAB computer and an output file will be generated in excel, which could be sent back to the originator.

The MATLAB code will be referred to as “STP-TAMIL” which stands for South Texas Project Nuclear Operating Company – Texas A&M Improved LOOP. LOOP refers to Loss Of Offsite Power at a nuclear power plant.

Section 2.3 – Development

STP-TAMIL was developed for use by Texas A&M University under contract to the South Texas Project Nuclear Operating Company. The theory, methodology, and verification studies are documented in the graduate thesis work entitled “Algorithms for Incorporation of Dynamic Recovery in Estimating Frequency of Critical Station Blackout” by Paul Rodi. This thesis also includes this User Manual as Chapter II.

During development of the software, the term 3-way common-cause and 2-way common-cause failure were used to describe 3/3 common-cause failures or 2/3 common-cause failures, respectively. A 3/3 common-cause failure may also be referred to as a common-cause triplet (3CCF) and a 2/3 common-cause failure may be referred to as a common-cause doublet (2CCF). A 1/3 common-cause failure would be known as an independent failure and is not really a common-cause failure at all. Therefore, the codes titles, variable names, and comment sections may refer to anyone of these naming conventions.

Section 2.4 – Installation

The following Software must exist for STP-TAMIL to function properly:

- 1) Microsoft Windows XP/2000/7 – Other Operating Systems are not tested
- 2) MATLAB Version 7.10.0 R 2010a – Newer/Previous Versions have not been tested
- 3) Microsoft Excel 2010 – Files need to be saved in “.xlsx” format

Create a folder called “STP-TAMIL” directly in the “C:” or for advanced users in the directory of one’s choosing.

Put all of the following files in the STP-TAMIL folder (for Version 1.0):

- 1) STPQUAD3CCF.m
- 2) FTR6890.m
- 3) FTR6890inv.m
- 4) FTR6890pdf.m
- 5) INTEGRAND2way.m
- 6) INTEGRAND3way.m
- 7) INTEGRANDv1pt0.m
- 8) RunFile.xlsx – Excel file to be filled in the next step “Building the Input Deck”
- 9) OutputFile.xlsx – Intentionally left blank excel file

Note: Other Previous Versions of STP-TAMIL, “STPQUAD2” and “STPQUAD3” code packages or suites, shown used in Chapter V are not listed above, as they are not needed to run the current software. However, they are included in the Appendix of this thesis.

Section 2.5 – Building the Input Deck

The Microsoft Excel file “RunFile.xlsx” serves as the input to the MATLAB code. This may be referred to as RunFile, input deck, or as the run file throughout this user manual. Therefore, RunFile.xlsx should be updated with values before running the software.

Figure 2.1 is a picture of what RunFile should look like when opened. Note filler values are used and may not be the same for all cases.

Table 2.1
First Part of Title Names and Descriptions for RunFile.

Title	Description
Data Row	This is the number of data rows that are entered in by the user. Note: it should start with 1 and increase by 1 with each row after. For example, if Excel Row 3 had data in it, then Data Row for that row would be 3.
Number of Trains	This is the number of safety trains used in the analysis. For STP, the number will likely be 3. Note, that if it is less than 3 then only the variables up to the number will be used. For Example, if 2 trains are input, then 2 Train Variables will be read as well as 1-Train variables but 3-Train Variables will not. Therefore, a value like P (FTS Train 3) would not be entered into the system. Also, any number of trains entered above 3 will cause an error.
Mission Time	This is the time is the amount of time assumed taken to restore outside power, which effectively ends the loss of the offsite power event (typically 24 hours).
Critical Time	This is the maximum amount of time that a component can be offline before damage to the core is inevitable.
Recovery of Offsite Power Alpha Parameter of G	This is the Alpha parameter for the cdf of offsite recovery.
Recovery of Offsite Power Beta Parameter of G	This is the Beta parameter for the cdf of offsite recovery.
Recovery of Offsite Power Gamma Parameter of G	This is the Gamma parameter for the cdf of offsite recovery.
Recovery of Offsite Power Delta Parameter of G	This is the Delta parameter for the cdf of offsite recovery.
Probability of Failure to Start (3CCF)	This is the probability of a Failure to Start in a common-cause triplet event.

Table 2.2
Second Part of Title Names and Descriptions for RunFile.

Title	Description
Probability of Failure to Load and Run (3CCF)	This is the probability of a Failure to Load and Run (fails to operate for at least 1 hour after the initiating event) in a common-cause triplet event.
Error Tolerance for Single Integral to Compute CC Triplet Failure Probability	This is user defined error tolerance used for the single integral (QUAD) for the common-cause triplet failure portion of probability of critical station blackout.
Alpha Parameter for CC Triplet	This is the Code's Alpha Parameter for the CC Triplet cdf related to the three Trains. This is generally the CC Triplet mean time to failure, up to the Beta Parameter.
Beta Parameter for CC Triplet	This is the Code's Beta Parameter for the CC Triplet cdf related to the three Trains. This is the CC Triplet cutoff time or critical time.
Gamma Parameter for CC Triplet	This is the Code's Gamma Parameter for the CC Triplet cdf related to the three Trains. This is generally the CC Triplet mean time to failure, during the rest of the mission time after the first hour.
Delta Parameter for CC Triplet	This is the Code's Delta Parameter for the CC Triplet cdf related to the three Trains.
Probability of CC Doublet Failure to Start (2CCF)	This is the Probability of a Common-Cause Doublet Failure to Start for a pair of trains.
Probability of CC Doublet Failure to Load and Run (2CCF)	This is the Probability of a Common-Cause Doublet Failure to Load and Run (fails to operate for at least 1 hour after the initiating event) for a pair of trains.
Error Tolerance for Double Integral to Compute CC Doublet Failure Probability	This is user defined error tolerance used for the double integral (DBLQUAD) for the Common-Cause Doublet failure portion of probability of critical station blackout.
Alpha Parameter for CC Doublet	This is the Code's Alpha Parameter for the CC Doublet cdf related to the two Trains. For most distributions, this is generally the mean time to failure, up to the Beta Parameter.

Table 2.3
Third Part of Title Names and Descriptions for RunFile.

Title	Description
Beta Parameter for CC Doublet	This is the Code's Beta Parameter for the CC Doublet cdf related to the two Trains. This is the cutoff time or critical time.
Gamma Parameter for CC Doublet	This is the Code's Gamma Parameter for the CC Doublet cdf related to the two Trains. This is generally the mean time to failure, during the rest of the mission time after the first hour.
Delta Parameter for CC Doublet	This is the Code's Delta Parameter for the CC Doublet cdf related to the two Trains.
Probability of Independent Failure to Start (Ind.)	This is the Probability that an Independent Failures Start will occur for any given train.
Probability of Independent Failure to Load and Run (Ind.)	This is the Probability that an Independent Failures to Load and Run (fails to operate for at least 1 hour after the initiating event) will occur for any given train.
Error Tolerance for Triple Integral to Compute Independent Failure Probability	This is user defined error tolerance used for the triple integral (TRIPLEQUAD) for the Independent failure portion of probability of critical station blackout.
Alpha Parameter for Independent	This is the Code's Alpha Parameter for the Independent cdf related to the single Train. For most distributions, this is the mean time to failure, up to the Beta Parameter.
Beta Parameter for Independent	This is the Code's Beta Parameter for the Independent cdf related to the single Train. For most distributions, This is the cutoff time or critical time.
Gamma Parameter for Independent	This is the Code's Gamma Parameter for the Independent cdf related to the single Train. This is generally the mean time to failure, during the rest of the mission time after the first hour.
Delta Parameter for Independent	This is the Code's Delta Parameter for the Independent cdf related to the single Train.
Recovery of Train 1 Alpha Parameter for R{1}	This is the Code's Alpha Parameter for the Onsite Recovery Rate related to Train 1.
Recovery of Train 1 Beta Parameter for R{1}	This is the Code's Beta Parameter for the Onsite Recovery Rate related to Train 1.

Table 2.4

Fourth Part of Title Names and Descriptions for RunFile.

Title	Description
Recovery of Train 1 Gamma Parameter for R{1}	This is the Code's Gamma Parameter for the Onsite Recovery Rate related to Train 1.
Recovery of Train 1 Delta Parameter for R{1}	This is the Code's Delta Parameter for the Onsite Recovery Rate related to Train 1.
Recovery of Train 2 Alpha Parameter for R{2}	This is the Code's Alpha Parameter for the Onsite Recovery Rate related to Train 2.
Recovery of Train 2 Beta Parameter for R{2}	This is the Code's Beta Parameter for the Onsite Recovery Rate related to Train 2.
Recovery of Train 2 Gamma Parameter for R{2}	This is the Code's Gamma Parameter for the Onsite Recovery Rate related to Train 2.
Recovery of Train 2 Delta Parameter for R{2}	This is the Code's Delta Parameter for the Onsite Recovery Rate related to Train 2.
Recovery of Train 3 Alpha Parameter for R{3}	This is the Code's Alpha Parameter for the Onsite Recovery Rate related to Train 3.
Recovery of Train 3 Beta Parameter for R{3}	This is the Code's Beta Parameter for the Onsite Recovery Rate related to Train 3.
Recovery of Train 3 Gamma Parameter for R{3}	This is the Code's Gamma Parameter for the Onsite Recovery Rate related to Train 3.
Recovery of Train 3 Delta Parameter for R{3}	This is the Code's Delta Parameter for the Onsite Recovery Rate related to Train 3.
Type of Repair for 3CCF: Enter a number 1-4 (1 for No Credit (offsite/onsite), 2 for Offsite Credit Only, 3 for Offsite + Onsite Repair only after Full Failure, 4 for Full Recovery (NORMALLY/DEFAULT))	This is for the Type of Repair for Common-Cause Triplet events that the user would like to use. One enters a whole integer value of "1" through "4". "1" leads to No Credit for Offsite or Onsite Power, "2" takes Credit for Offsite power Only, "3" takes Credit for Offsite and Onsite only after Full Failure has Occurred, "4" takes Credit for Full Recovery (Onsite + Offsite). Note: It is "4" by default and if an error value is placed in this cell. These Cases are described more fully in Chapter IV.

Table 2.5
Fifth Part of Title Names and Descriptions for RunFile.

Title	Description
<p>Type of Repair for 2CCF: Enter a number 1-4 (1 for No Credit (offsite/onsite), 2 for Offsite Credit Only, 3 for Offsite + Onsite Repair only after Full Failure, 4 for Full Recovery (NORMALLY/DEFAULT))</p>	<p>This is for the Type of Repair for Common-Cause Doublet events that the user would like to use. One enters a whole integer value of “1” through “4”. “1” leads to No Credit for Offsite or Onsite Power, “2” takes Credit for Offsite power Only, “3” takes Credit for Offsite and Onsite only after Full Failure has Occurred, “4” takes Credit for Full Recovery (Onsite + Offsite). Note: It is “4” by default and if a error value is placed in this cell. These Cases are described more fully in Chapter IV.</p>
<p>Type of Repair for Independent: Enter a number 1-4 (1 for No Credit (offsite/onsite), 2 for Offsite Credit Only, 3 for Offsite + Onsite Repair only after Full Failure, 4 for Full Recovery (NORMALLY/DEFAULT))</p>	<p>This is for the Type of Repair for Independent events that the user would like to use. One enters a whole integer value of “1” through “4”. “1” leads to No Credit for Offsite or Onsite Power, “2” takes Credit for Offsite power Only, “3” takes Credit for Offsite and Onsite only after Full Failure has Occurred, “4” takes Credit for Full Recovery (Onsite + Offsite). Note: It is “4” by default and if an error value is placed in this cell. These Cases are described more fully in Chapter IV.</p>
<p>Type of Distribution for G, cdf in time for recovery of offsite power</p>	<p>This is MATLAB function handler for the type of distribution for G, which is the cdf for the time of recovery of offsite power. Details of inputs in Subsection 2.5.a.</p>
<p>Type of Distribution for R, cdf in time for repair for the ith train</p>	<p>This is MATLAB function handler for the type of distribution for R, which is the cdf in time for repair of a train 1. Details of inputs in Subsection 2.5.a.</p>
<p>Type of Distribution for Failure to Load and Run cdf in time of ith train</p>	<p>This is MATLAB function handler for the type of distribution used for Failure to Load and Run which is computed in the custom distribution files defined in the next cells or created by the user. FTR6890 and FTR6890inv is an example of these files. Details of inputs in Subsection 2.5.a.</p>
<p>File Used for CC Triplet Failure to Load and Run cdf</p>	<p>MATLAB “.m” File Written for use to find the cdf of a Failure to Load and Run for a Common-Cause Triplet event. Details in Subsection 2.5.e. FTR6890 is an example and default for this file.</p>

Table 2.6
Sixth Part of Title Names and Descriptions for RunFile.

Title	Description
File Used for CC Triplet Failure to Load and Run inverse cdf	MATLAB “.m” File Written for use to find the inverse cdf of a Failure to Load and Run for a Common-Cause Triplet event. Details in Subsection 2.5.e. FTR6890inv is an example and default for this file.
File Used for CC Doublet Failure to Load and Run cdf	MATLAB “.m” File Written for use to find the cdf of a Failure for to Load and Run a Common-Cause Doublet event. Details in Subsection 2.5.e. FTR6890 is an example and default for this file.
File Used for CC Doublet Failure to Load and Run inverse cdf	MATLAB “.m” File Written for use to find the inverse cdf of a Failure to Load and Run for a Common-Cause Doublet event. Details in Subsection 2.5.e. FTR6890inv is an example and default for this file.
File Used for Independent Failure to Load and Run cdf	MATLAB “.m” File Written for use to find the cdf of a Failure for Independent events. Details in Subsection 2.5.e. FTR6890 is an example and default for this file.
File Used for Independent Failure to Load and Run inverse cdf	MATLAB “.m” File Written for use to find the inverse cdf of a Failure to Load and Run for Independent events. Details in Subsection 2.5.e. FTR6890inv is an example and default for this file.

Note the parameters or input parameters (alpha, beta, gamma, and delta) mentioned in the various preceding tables all have specific meanings based on the type of distribution employed. The parameters are defined and described for each type of distribution in Tables 2.7-2.8. The parameters for the common-cause triplet, common-cause doublet, and independent events are clarified better in subsequent paragraphs.

For common-cause triplet failure events defined by an exponential distribution, the alpha parameter is the mean time to failure up to time defined by the beta parameter, which is the cutoff time in hours. The beta parameter is a user input in the spreadsheet where as the alpha parameter is based on the user input of both the probability of a 3CC failure to start event and the probability of a 3CC failure to load and run. The gamma parameter is the subsequent exponential mean time (hours) conditional on reaching the cutoff time (beta parameter). This is based upon the probability of a 3CC Failure to run as

well as the probabilities defined previously in the code by alpha parameter automatically. The delta function is meant to be empty, left as a “0” value, unless using a more advanced type of distribution. The description of how the parameters relate to each other is described fully in subsection 5.3.a.

For common-cause doublet failure events, the alpha parameter is the exponential mean up to time defined by the beta parameter, which is the cutoff time in hours. The beta parameter is a user input in the spreadsheet. For the two-regime exponential, the alpha parameter is based on of the user input of both the probability of a single independent failure to run in conjunction with the probability of 2CC failure to start and the probability of a single independent failure to run in conjunction with the probability of 2CC failure to load and run. The gamma parameter is the subsequent exponential mean conditional on reaching the cutoff time (beta parameter). This is based on the probability of a single independent failure to run in conjunction with a CC doublet failure to run as well as the probabilities defined previously in the code by alpha parameter automatically. The delta function is meant to be empty, left as a “0” value, unless using a more advanced type of distribution. The description of how the parameters relate to each other is shown in subsection 5.3.b.

For independent failure events, the alpha parameter is the exponential mean up to time defined by the beta parameter, which is the cutoff time in hours. The beta parameter is a user input in the spreadsheet where as the alpha parameter is based off of the user input of both the probability of a three independent failure to load and run event and the probability of single independent failure to start in conjunction with two independent load and run failures. The gamma parameter is the subsequent exponential mean conditional on reaching the cutoff time (beta parameter). The gamma parameter is based upon the probability of an independent failure to run as well as the probabilities defined previously in the code by alpha parameter automatically. The delta function is meant to be empty, left as a “0” value, unless using a more advanced type of distribution. The description of how the parameters relate to each other is described in subsection 5.3.c.

Once all the data for a row is entered into the run file, then either one can choose to have that be the only case run by the program or one can choose to add another row of data. To achieve this, start at the new row and enter all data again, repeating steps in this

subsection. Call the data row by a different number to distinguish it from the output file. Running multiple cases, or data rows, can be used as a comparison or as a way to test different scenarios.

All entered data should be rechecked for accuracy. Generally, all of the trains within one case, or data row, will have the same values for the same types of rates for failure and recovery. Additionally, it is important to make sure there are no blanks in the row, as it will return an error in the code.

It is important to note that if any cell is left blank it will be filled in automatically with a default value so the code can still run. These default values are described in more detail in Section 2.6.

An example of a full input deck is shown in Appendix B for a better reference. Note that that RunFile entered correctly are the values for the industrial average case from Chapter V of the thesis. This means that the output will also be the same as data presented in Chapter V. It can also be noted that the default values are the same values that appear in “Data Row” 1 of Appendix B.

The file should be saved under the title “RunFile.xlsx” in the folder STP-TAMIL or the folder in which the MATLAB environment is running.

Subsection 2.5.a – Details for Types of Distributions

The spreadsheet contains many cells that should contain specification of a function handler for the type of distribution desired. These input cells are compiled and shown in the Tables 2.1-2.6 found earlier in this section.

It is important to note that the exact text (as entered) into those cells will be used as a function in the MATLAB script. Therefore, it is imperative to remove any extra spaces or any difference from what would appear in MATLAB. It is also important to make sure the alpha, beta, gamma, and delta parameters that are being passed into the computer code using the certain distribution are used correctly.

From the MATLAB Documentation [16], the types of distributions for cdf, inverse cdf, and pdf that can be used are shown in Tables 2.7 and 2.8.

Table 2.7

First Part of Function Names and Parameters for Type of Distributions for CDF.

Distribution Type	Function Name	Input Parameter Alpha (cdf,inv,pdf)	Input Parameter Beta	Input Parameter Gamma	Input Parameter Delta
Beta	beta	(X,P,X)	a	b	—
Binomial	bino	(X,P,X)	n: number of trials	p: probability of success for each trial	—
Chi-Square	chi2	(X,P,X)	v: degrees of freedom	—	—
Exponential	exp	(X,P,X)	μ : mean	—	—
Extreme Value	ev	(X,P,X)	μ : location parameter	σ : scale parameter	—
F	f	(X,P,X)	v1: numerator degrees of freedom	v2: denominator degrees of freedom	—
Gamma	gam	(X,P,X)	a: shape parameter	b: scale parameter	—
Generalized Extreme Value	gev	(X,P,X)	k: shape parameter	σ : scale parameter	μ : location parameter
Generalized Pareto	gp	(X,P,X)	k: tail index (shape) parameter	σ : scale parameter	μ : threshold (location) parameter
Geometric	geo	(X,P,X)	p: probability parameter	—	—
Hypergeometric	hyge	(X,P,X)	M: size of the population	K: number of items with the desired characteristic in the population	n: number of samples drawn

Table 2.8

Second Part of Function Names and Parameters for Type of Distributions for CDF.

Distribution Type	Function Name	Input Parameter Alpha (cdf,inv,pdf)	Input Parameter Beta	Input Parameter Gamma	Input Parameter Delta
Lognormal	logn	(X,P,X)	μ	σ	—
Negative Binomial	nbin	(X,P,X)	r: number of successes	p: probability of success in a single trial	—
Noncentral F	ncf	(X,P,X)	v1: numerator degrees of freedom	v2: denominator degrees of freedom	δ : noncentrality parameter
Noncentral t	nct	(X,P,X)	v: degrees of freedom	δ : noncentrality parameter	—
Noncentral Chi-Square	ncx2	(X,P,X)	v: degrees of freedom	δ : noncentrality parameter	—
Normal	norm	(X,P,X)	μ : mean	σ : standard deviation	—
Poisson Distribution	poiss	(X,P,X)	λ : mean	—	—
Rayleigh	rayl	(X,P,X)	b: scale parameter	—	—
Student's t	t	(X,P,X)	v: degrees of freedom	—	—
Uniform (Continuous)	unif	(X,P,X)	a: lower endpoint (minimum)	b: upper endpoint (maximum)	—
Uniform (Discrete)	unid	(X,P,X)	N: maximum observable value	—	—
Weibull	wbl	(X,P,X)	a: scale parameter	b: shape parameter	—

For the alpha parameter for the cdf and pdf, the X variable represents numeric arrays of values where one would evaluate the cdf or pdf, respectively. For example in FTR6890, it represent mission time. For the alpha parameter for inverse cdf, P is a probability so it would be a numeric array of values from 0 to 1 where one would evaluate the inverse cdf. In FTR6890inv, this parameter would be represented by the ratio of the difference between the current time step failure probability initial probability of failure of the *i*th train by the difference between one and the same initial failure probability. However, in the current version of the code the alpha parameter used only requires input for cdf as the inverse cdf automatically converts the parameters used as well as holding the distribution constant.

Subsection 2.5.b – Entering a cdf Type of Distribution

Enter into the desired cell of the spreadsheet “RunFile” the function name followed by “cdf” without quotes.

For example, if a Log-Normal Distribution was required then one would enter “logncdf” without quotes into the cell, where “logn” is found in the Tables in this section under Function Handler. The “cdf” is the cumulative distribution function.

Note that the input parameters are entered as per the definitions of the distributions following the input parameter definitions in Tables 2.8-2.9. If a certain distribution does not contain a certain parameter, then in the preceding tables the input parameter will be shown as a “—”. A “0” value is then placed in the associated cell in the run file without quotes in the cell.

Subsection 2.5.c – Entering an Inverse cdf Type of Distribution

Enter into the desired cell of the spreadsheet “RunFile” the function name followed by “inv” without quotes.

For example, if a Log-Normal Distribution was required then one would enter “logninv” without quotes into the cell, where “logn” is found in Tables 2.1-2.7 in this

section under Function Handler. The “inv” is the inverse cumulative distribution function.

Note that the input parameters must also be entered as per the definitions of the distributions following the input parameter definitions in Tables 2.8-2.9. If a certain distribution does not contain a certain parameter, then in the preceding tables the input parameter will be shown as a “—”. A “0” value is then placed in the associated cell in the run file without quotes in the cell.

Subsection 2.5.d – Entering a pdf Type of Distribution

Enter into the desired cell of the spreadsheet “RunFile” the function name followed by “pdf” without quotes.

For example, if a Log-Normal Distribution was required then one would enter “lognpdf” without quotes into the cell, where “logn” is found in Tables 2.1-2.7 found in this section under Function Handler. The “pdf” is the probability density function.

Note that the input parameters must also be entered as per the definitions of the distributions following the input parameter definitions in Tables 2.8-2.9. If a certain distribution does not contain a certain parameter, then in the preceding tables the input parameter will be shown as a “—”. A “0” value is then placed in the associated cell in the run file without quotes in the cell.

It can be noted, in the current version of the code, that the pdf type distributions planned use in the code were replaced by using inverse cdf distributions as they are related.

Subsection 2.5.e – Creating a New MATLAB “.m” File for Distributions

This subsection is for advanced users. The input run file allows the capability of one to build a new MATLAB file. These files will have the extension “.m” (also known as m-files). Please note that if the file does not work, STP-TAMIL will also no longer function if one entered data in that file’s input cells.

This m-file can be used to create a new type of cdf, inverse cdf, or pdf that does not appear in the Tables 2.8-2.9. It can be noted that by default, STP-TAMIL has a set of codes that consist of “FTR6890.m”, “FTR6890inv.m”, and “FTR6890pdf.m” which generate cdf, inverse cdf, and pdf respectively. Also by default, “FTR6890pdf.m” is not used within the code.

These codes all use four parameters (alpha, beta, gamma, and delta) to work correctly just as the other distributions found in MATLAB. Inside of STPQUAD3CCF.m, the alpha and gamma parameters for common-cause triplet, common-cause doublet, and independent failures to load and run are calculated. If the user decides to make a new distribution type, then the user can change the alpha and gamma parameters to the desired values. The code will use the values that appear in the input deck instead of the values based off of equations seen in Chapter V for those specific parameters.

After the code is created, save the code in the STP-TAMIL folder as the name the user would choose to call the file. Make certain to save the file with a “.m” extension. Then inside the run file, set the parameters, or parameters, then go to the specific columns where the user wants to run this new distribution. Put the name of the file inside the box exactly as it appears as saved without the “.m” part of the name. The system may require that the file name be entered in multiple locations in the run file. Check Section 2.4 to identify all places that this entry is required.

Section 2.6 – Running STP-TAMIL

After completion of the input file, the code can be run in MATLAB. However, this does require a MATLAB license. If one is not available, after meeting proper requirements and receiving approval from Dr. Paul Nelson (“p-nelson@tamu.edu”), the file may be run and outputs sent back to the originator. After approval, this involves emailing the “RunFile.xlsx” file and then the originator may receive the “OutputFile.xlsx” file in response.

At this point, it is necessary to save the file in that computer’s STP-TAMIL folder where all “.m” files listed in the Installation section are housed.

To run the code, Open the folder STP-TAMIL, and then double click on the file STPQUAD3CCF.m.

This will open MATLAB automatically (showing the command window); and the window editor will open. The MATLAB main window must show under “current folder” the correct folder, directory and files as shown in Installation as shown in Section 2.4. If all is correct then click the green button in the Editor Window as shown in Figure 2.2.

It is important at this point to have both the input and output Excel files closed during the programs execution or MATLAB will return an error.

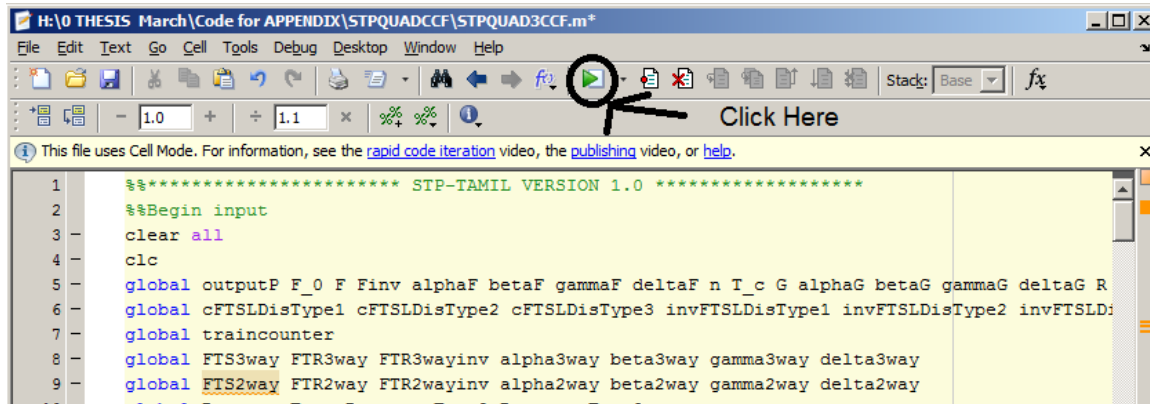


Figure 2.2 – Example of How to Run MATLAB from the Editor Window

If a window pops up at anytime prompting the user that STPQUAD3CCF.m does not exist within the current path/folder. This indicates that MATLAB is pointing to the wrong folder so the button labeled “Change Folder” or “Change Path” must be clicked. An example of this error is shown in Figure 2.3.

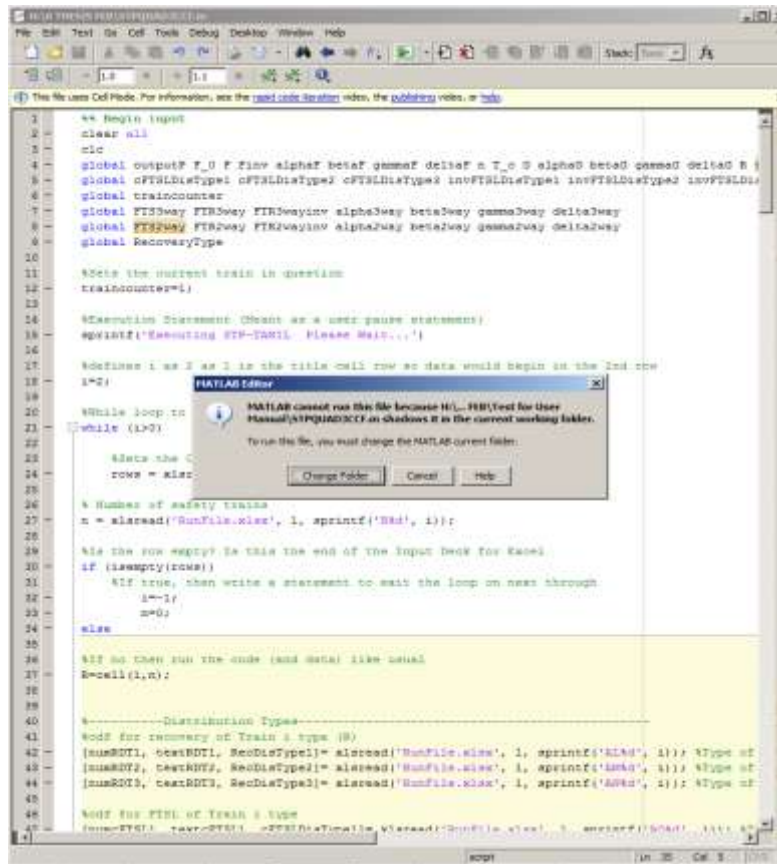


Figure 2.3 – Example of Change Folder Prompt in MATLAB

Once that is done, the program will take a few minutes to execute. A screenshot of the executed program is shown following in Figure 2.4.

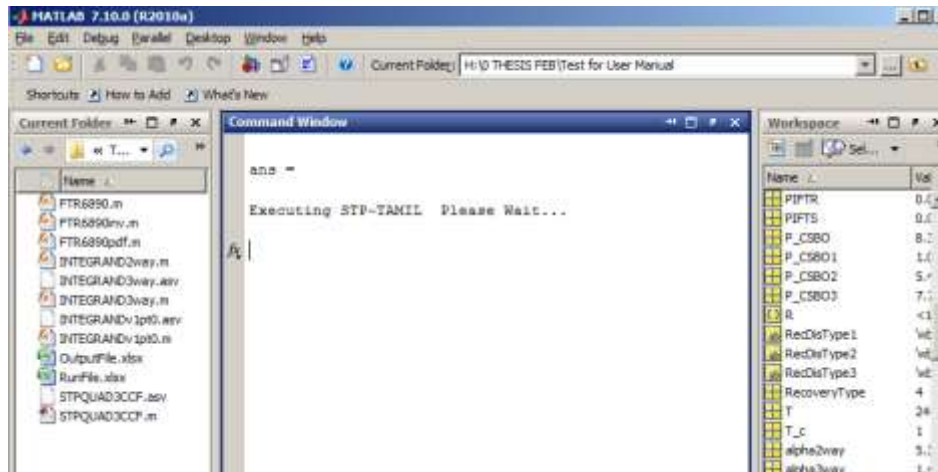
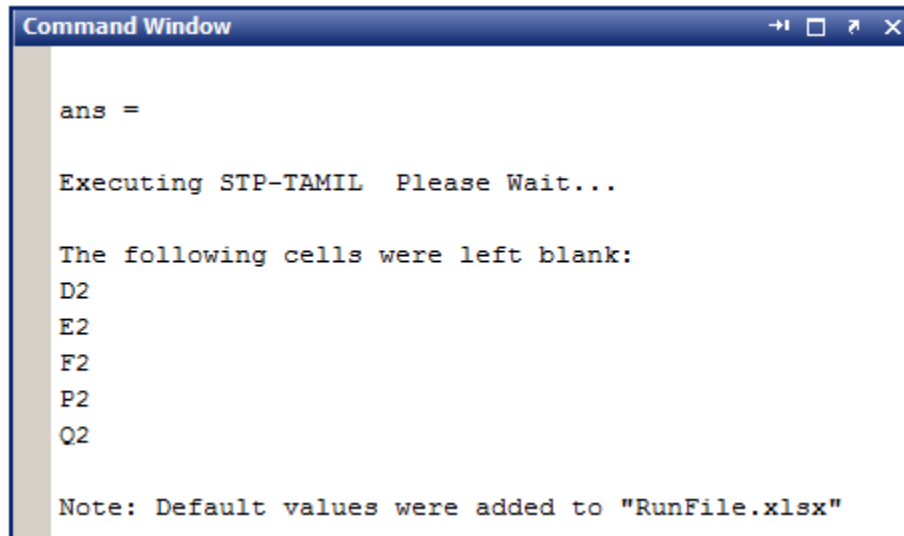


Figure 2.4 – Example of Program Working and Running the Code

If there is an error, check the input deck for problems or the folder to ensure all files required are indeed in the correct place. Advanced errors may need to be diagnosed further.

The code then will determine if any cells were left blank when the input Excel file was created. This will prompt a message that is shown in Figure 2.5 if there were blanks in the input deck and Figure 2.6 if there were no blanks in the input deck. When this prompt from Figure 2.5 is occurring, “RunFile.xlsx” is being written to and the empty cells are replaced with the default values. These default values come from the industrial average method shown in Chapter V and are also shown in the Data Row of Appendix B. In this particular case, cells “D2”, “E2”, “F2”, “P2” and “Q2” were filled in with default values.



```

Command Window

ans =

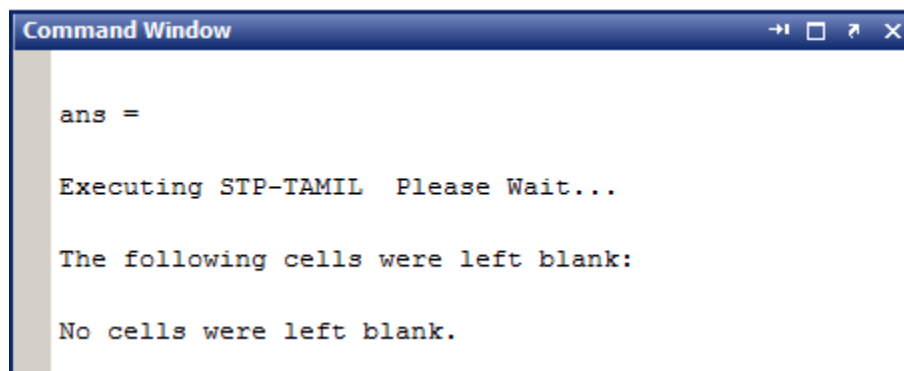
Executing STP-TAMIL Please Wait...

The following cells were left blank:
D2
E2
F2
P2
Q2

Note: Default values were added to "RunFile.xlsx"

```

Figure 2.5 – Example of Filling Default Cells Prompt in MATLAB



```

Command Window

ans =

Executing STP-TAMIL Please Wait...

The following cells were left blank:

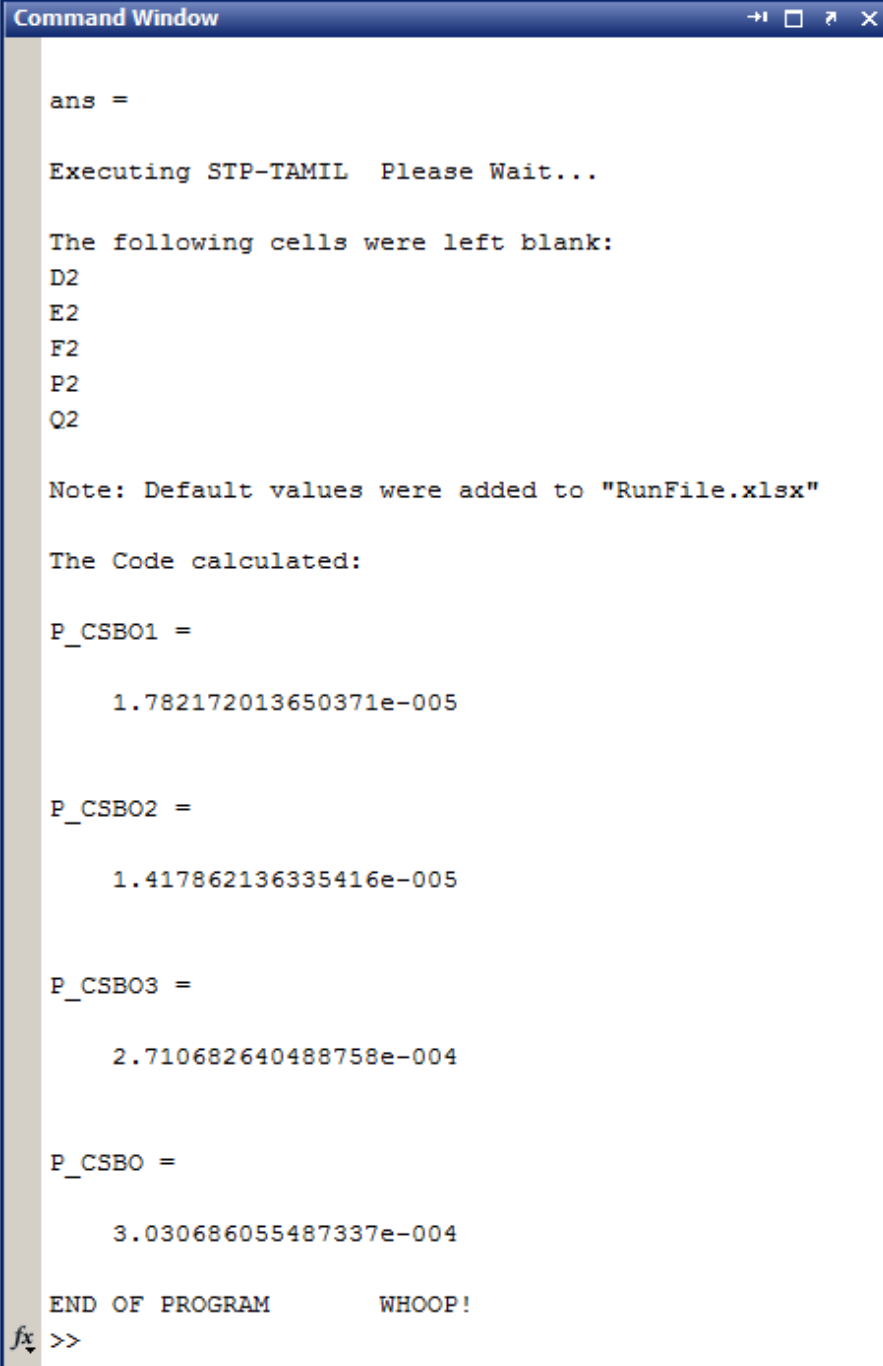
No cells were left blank.

```

Figure 2.6 – Example of No Empty Cells Prompt in MATLAB

After the program finishes correctly, the code will print out several probability of critical station blackout scenarios. “P_CSBO1” is the probably of critical station blackout for only independent failures, “P_CSBO2” is the probably of critical station blackout considering common-cause doublet failures, “P_CSBO3” is the probably of critical station blackout considering common-cause triplet failures, and “P_CSBO” is this total probability of critical station blackout. “P_CSBO” should be the sum of “P_CSBO1”, “P_CSBO2”, and “P_CSBO3”. Multiple “P_CSBO” may be printed out. The output is a result of how many data rows one inputted into the RunFile.xlsx file. See Figure 2.7 for

an example of one row of data run through STP-TAMIL that has values filled in on the excel file.



```
Command Window

ans =

Executing STP-TAMIL Please Wait...

The following cells were left blank:
D2
E2
F2
P2
Q2

Note: Default values were added to "RunFile.xlsx"

The Code calculated:

P_CSBO1 =

    1.782172013650371e-005

P_CSBO2 =

    1.417862136335416e-005

P_CSBO3 =

    2.710682640488758e-004

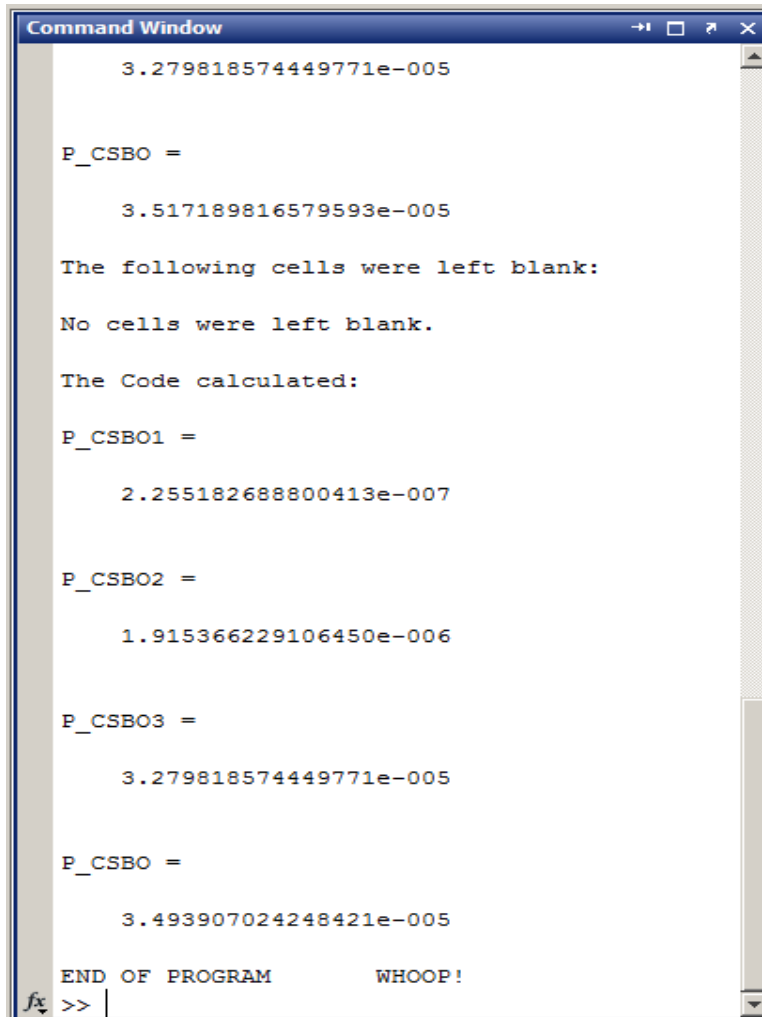
P_CSBO =

    3.030686055487337e-004

END OF PROGRAM      WHOOP!
fx >>
```

Figure 2.7 – Example of Code Working and Finishing Showing “P_CSBO”

Figure 2.8 shows an example of the code after multiple cases are entered into the data rows of the run file.



```
Command Window
3.279818574449771e-005

P_CSBO =
3.517189816579593e-005

The following cells were left blank:
No cells were left blank.

The Code calculated:
P_CSBO1 =
2.255182688800413e-007

P_CSBO2 =
1.915366229106450e-006

P_CSBO3 =
3.279818574449771e-005

P_CSBO =
3.493907024248421e-005

END OF PROGRAM      WHOOP!
fx >>
```

Figure 2.8 – Example of Code Working and Finishing Showing Multiple “P_CSBO”s

Note that there are multiple data rows, four in this example, input into the RunFile.xlsx file, so there are multiple “P_CSBO” values that are generated and outputted to the Output.xlsx file. See Section 2.7 on how to read the output file. When the

program finishes correctly, there will be an “END OF PROGRAM” message and the program will terminate.

Section 2.7 – Output File

There are two ways to read the output of this file. The first way is to read it straight off of the command window from MATLAB. The “P_CSBO” values will print out in order from the lowest numbered Data Row to the highest.

The second way, the preferred way, is that the MATLAB script after execution will create an Microsoft Excel file that will store all the values. This file will be in the same folder “STP-TAMIL” and will be initially titled, “OutputFile.xlsx”. After running the code, save the program created under a new file name to avoid overwriting the original programming. It can also be noted that the Excel file will show more digits on the answer than the MATLAB command window. This could be useful for comparison or for future work.

It is also important to note that only the amount of data rows that are input will be overwritten in the output. If one ran a case of three data rows and later ran a case with only two data rows, then only the first two data rows would be overwritten in the output file. However, the output file would still show the third data row values. Therefore, it is important *to delete everything in the output file* before each code execution. The output file itself can also be deleted as well; and it will create a new one the next time the code is executed.

The OutputFile.xlsx can later be sent to the originator of the input file, probably South Texas Project, by email or any other form of data transfer.

An Example of “OutputFile” is shown in Figure 2.9. It also shows one of the cells highlighted, showing the degree of precision that variables hold in the output file. It can be noted that this is the output file generated from the case of Figure 2.8, which had four input data rows.

	A	B	C	D	E	F	G	H	I
1	Data Row	P_CSBO	P_CSBO1	P_CSBO2	P_CSBO3				
2	1	0.000303	1.78E-05	1.42E-05	0.000271				
3	2	5.19E-05	5.03E-07	2.65E-06	4.87E-05				
4	3	3.52E-05	3.39E-07	2.04E-06	3.28E-05				
5	4	3.49E-05	2.26E-07	1.92E-06	3.28E-05				

Figure 2.9 – Example of OutputFile After Code is Executed

It might be interesting to note that the only thing switched between each data row was the type of recovery used. Where column “B” leads to No Credit for Offsite or Onsite Power, column “C” takes Credit for Offsite power Only, column “D” takes Credit for Offsite and Onsite only after Full Failure has Occurred, column “E” takes Credit for Full Recovery (Onsite + Offsite). These Cases are described further in Chapter IV.

Section 2.8 – End Notes

It is always a good idea to keep a backup of a working copy of the program if one chooses to edit it. Should a problem arise with the program not listed in this user manual, one could first check the MATLAB documentation [12]. If this does not help, then with proper rights one should contact Dr. Paul Nelson or Paul Rodi and attempt to get a new copy or seek assistance. Contact information is listed earlier in this chapter.

CHAPTER III

THEORY

This chapter will include a formal statement of the CSBO problem for simplified Markov models, and a derivation of the nonrecovery integral as an approximate solution to that problem. This derivation employs the methodology of asymptotic analysis.

The main purpose of this thesis is to develop an algorithm that will compute the probability of critical station blackout after a loss of offsite power (LOOP) has occurred. This is done through the non-recovery integral; therefore, it is critical to fully explain the parts that contribute to that integral.

Section 3.1 will discuss setting up the problem for the non-recovery integral as well as introduce the Markov models for both a two-EDG system and the three-EDG system. This section will also include a subsection that contains a list of variable names and definitions used throughout this chapter. Section 3.2 goes into the two-EDG case and shows the state-transition equations while Section 3.3 describes the asymptotic solution method employed throughout the next few sections. This solution is broken up into different order solutions described in this section and shown in multiple subsections. After the solution is determined, a subsection discusses how this result leads to the non-recovery integral. Section 3.5 discusses how common-cause failures contribute to this model. Section 3.6 gives a general form of the non-recovery integral and Section 3.7 gives adds a third EDG to this two-EDG model and shows how the new non-recovery equation would look as well as describing what the various terms entail. Section 3.8 goes back to the three-EDG Markov Model and shows the initial conditions and the state-transition differential equations that could be used to solve this more simplified approach.

Section 3.1 – Setting Up the Problem

This chapter treats two different models. The first is a two-EDG simplified model and in that model the non-recovery integral is fully developed, which includes common-cause failures. This model is described in Subsection 3.1 and the associated asymptotic

analysis is described in Sections 3.2 through 3.7. The code developed for this thesis, STP-TAMIL version 1.0, uses the non-recovery integral justified in these sections.

The second model is similar, but incorporates a three EDGs. This model may be referred to as the advanced three EDG model. It is described in Subsection 3.1.b and the associated asymptotic analysis approach is mentioned in Section 3.8. The code in this thesis does not fully represent the result, but even description of the model highlights some important issues.

Both models provide useful insights into capabilities and limitations of the non-recovery integral. The difference should lie in the simplicity of the integration forms and the usage of known constants.

This section also contains a table for the list of variables used throughout the thesis as a reference. This should be used as a quick reference but not as an absolute reference as variables are used multiple times so the specific section of the thesis should be reviewed before assuming a meaning for a variable.

Subsection 3.1.a – Simplified Two-EDG Markov Model

The first step to deriving the nonrecovery integral will be to start with a simple two emergency-diesel-generator case and work on from there. The main purpose of this is to show the development as well as show how to go from the model developed in previous work [1], and incorporate common-cause failure.

To incorporate either i) failure or repair that are governed by statistical distributions, or ii) positive critical times, will require reentry time models as found in Nelson and Wang [17]. This will require future development, so for now only a simple case for the system is considered. As seen in Figure 3.1, the system consists of two EDGs connected by an OR gate, as an event tree, or as an AND gate, as a fault tree. The idea for this methodology comes from previous work done by Texas A&M and South Texas Project [1].

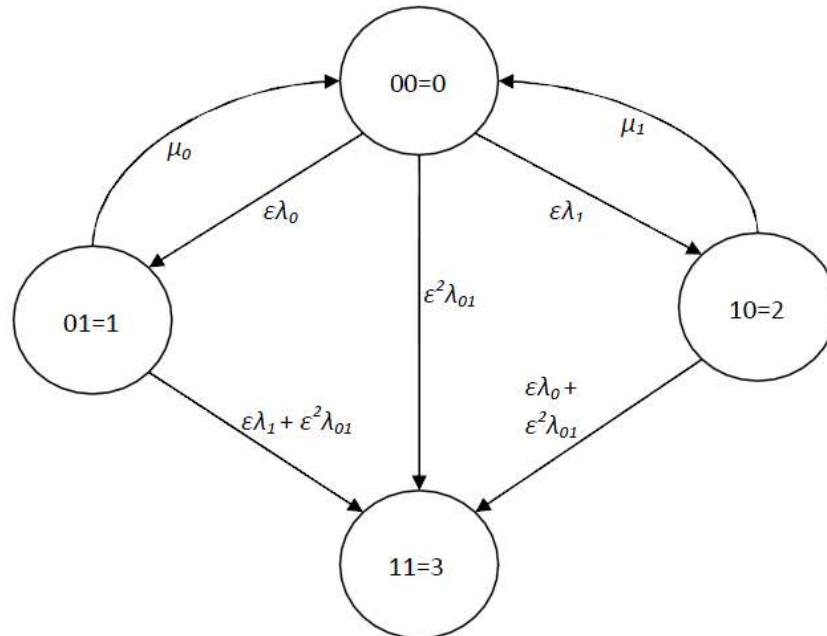


Figure 3.1 – State-Transition Diagram for the Two-EDG Markov Model

The two emergency diesel generators are labeled as “0” and “1”. A system blackout (SBO) or the system failing could occur if the two EDGs fail independently (in either order) or by both failing simultaneously while running. The latter is known generally as a common-cause failure. These failures can occur at either start-up, classified as a failure on demand, or while in operation, known as a failure to run (FTR). Independent failures of the i th EDG are distributed exponentially with a constant rate λ_i (mean time to failure, $1/\lambda_i$) and common-cause failures are distributed exponentially with a constant rate λ_{01} . It is assumed that if only one EDG has failed then the other one can still operate, and the plant would not enter a station black out. It is also assumed that the one that failed has a chance to be repaired. Repairs of EDG i are taken as distributed exponentially in time, with a constant rate μ_i , for EDG i . When both EDGs have failed simultaneously and the station has entered its blackout scenario, the model assumes that repair is not possible.

Common cause events are classified as x/n type events where x is the number of failures out of n , the total amount that could fail. For example, if two EDGs fail out of a possibility of three EDGs, then this scenario would be a $2/3$ common cause event.

In Figure 3.1, the system is presented and the binary index of a state is to be interpreted as follows. If the m th bit, as read from the right starting at $m=0$, is zero then the m th EDG is operating in that state, and if it is one then that EDG is failed in that particular state. The failure rates (λ s) and repair rates (μ s) are indexed according to the EDG, zero or one, respectively it fails or it is repaired to effect the transition. The 2/2 common cause failure rate, from the system modeled, is then indexed “01” because both EDGs would fail, if they were in operation, from any such event. Note that common-cause failures contribute to the transitions from states from which only one EDG has failed, either states “01” or “10”, to the station blackout state 3. This is because any such common-cause event would cause station blackout, even if one of the two EDGs had already failed.

The idea of the use of epsilon, ε , is to reflect the belief that the independent failure rates are relatively small compared to reciprocal of the mission time ($1/T$). The failure rates can be an order of small parameter epsilon, $O(\varepsilon)$, or, for the common-cause failure rates, an order of extremely small parameter epsilon $O(\varepsilon^2)$. In this description, “small” is used as more of a relative term, so it is necessary to specify what the basis of the comparison is for these values. In this case it is the repair rates. The basis for comparison represents terms (mean times to repair) that are conventionally taken as order of unity, $O(1)$. Therefore, there is no epsilon associated with the repair rates.

The assumption here is that the independent and 1/2 failure rates are small relative to the repair rates. The 2/2 common-cause failure rate is also to be much smaller as compared to independent and 1/2 failure rates; therefore, common-cause is assumed to be extremely small compared to the repair rate. These are in practice reasonable assumptions. For example, as shown in the Results and Benchmarking section (Chapter V), the failure rates were taken as about $1.0e-3$ per hour, mean time to failure is then about 1000 hour. This is about one and a half orders of magnitude smaller than the repair rate of 0.065 per hour, mean time to repair is about 15.5 hours, corresponding to Weibull distribution ($\alpha=0.739$, $\beta=15.50$ hours) that was used for the distribution of repairs.

Common-cause failures typically occur with about 1% of the frequency of independent and 1/n failures [18, 19]. Common-cause failure frequencies are about two orders of magnitude smaller than the $O(\varepsilon)$ independent and 1/m failure rates; since ε was

already established as about $1.0e-1.5$ then that is very consistent with λ_{012} is about equal to $O(\varepsilon^2)$.

Note that the typical mission time of 24 hours, when expressed as a frequency (1/24 per hour or 0.04 inverse hours) is an $O(1)$ quantity. $O(1)$ represents an order of one value. This will be important in the following developments described in the next sections.

Subsection 3.1.b – Advanced Three-EDG Markov Model

The idea in this subsection is to describe a simplified three-EDG model to be used as a basis for development of the non-recovery integrals. For this advanced three-EDG case, the figure and equations become more complex. An example of how it would appear can be seen in Figure 3.2.

The state numbers are defined by the binary number defined by the specific EDGs that have failed. For example, if EDG 1 and EDG 3 have failed, then it would be in state 5 due to the failure of the EDGs represented in binary as “101”.

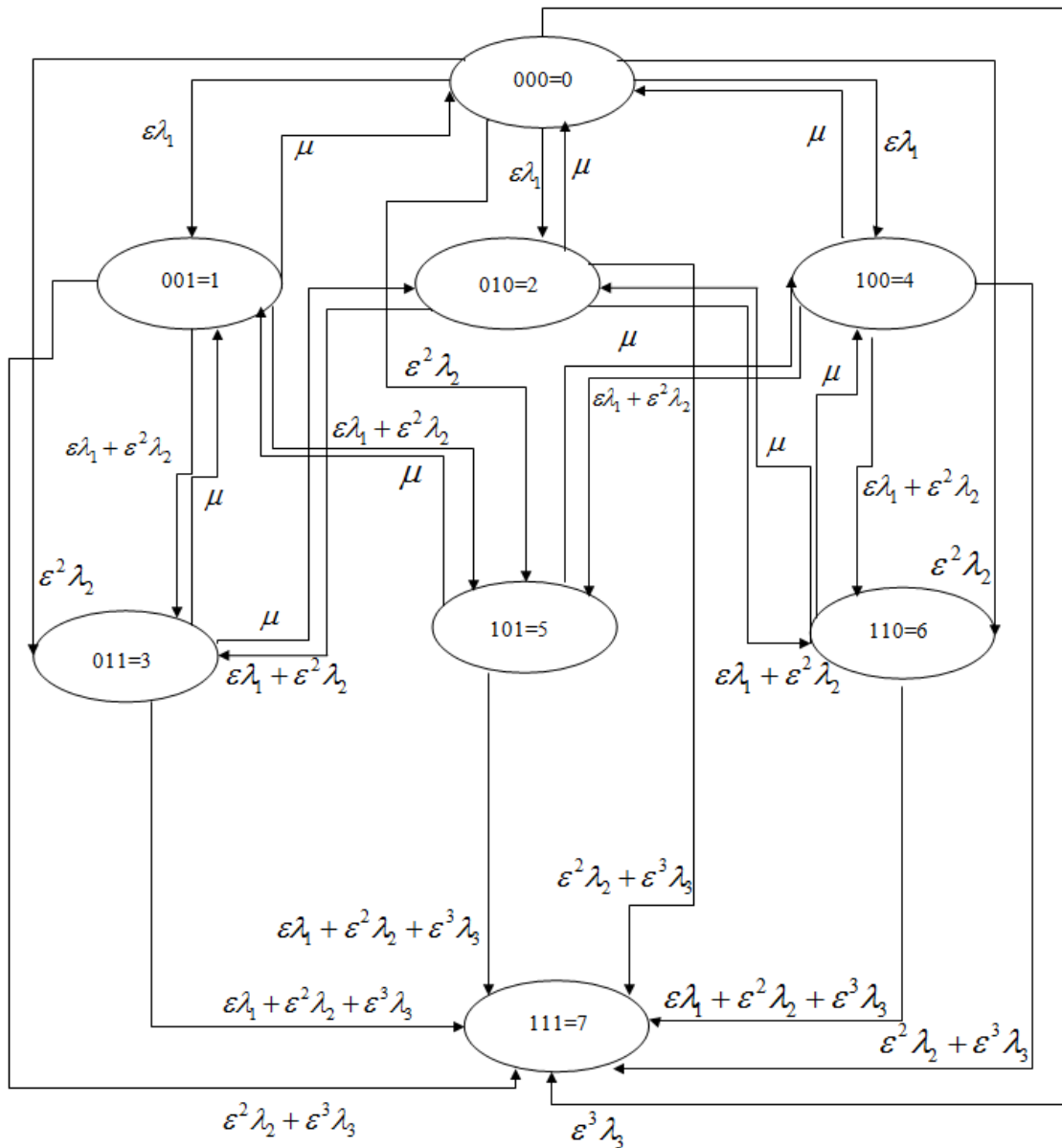


Figure 3.2 – State-Transition Diagram for the Three-EDG Markov Model

It can be noted that the repair rates, μ (μ s), are constant throughout the model. If two EDGs are failed and one is still in operation, then either of the EDGs have a chance for repair. However, both cannot be repaired at the exact same time. The model assumes that one EDGs has the chance to be repaired, and then the other one has a chance

to be repaired. For example, in Figure 3.2 if EDGs were to be repaired, State 4, which has a failure of EDGs 1 and 2, can either repair EDG 1 and go to state 2, where EDG 2 is still failed, or repair EDG 2 and go to state 1, where EDG 1 is still failed. Therefore, the model still assumes no common-cause repair rate.

For the three-EDG state-transition differential equation, this methodology no longer looks to set up in the exact same non-recovery integral found in the Rodgers et al. [1]. Therefore, there are a few assumptions that can be made. The first is that all of the single failure rates, λ , are the same, and therefore does not depend on a specific EDG, defined as λ_1 . This is the same case for all double common cause, now defined as λ_2 , and triple common cause failure rates, now defined as λ_3 . In addition, the repair rates are constant throughout the model, defined by μ .

Note that this assumption of constant rates is not necessarily valid during cut sets that requires one or multiple EDGs to be offline due to maintenance. This would vary the repair rates as well as the failure rates. This requires further research to develop the dynamic repair rate.

Subsection 3.1.c – List of Variables

Table 3.1-3.2 shows the variables and definitions that will be used in this chapter/thesis.

Table 3.1

First Part of Variables and Definitions for the Derivation of the Non-Recovery Integral.

Variable	Definition
f_i	Probability density function for failure of jth backup train
$F_i(0)$	Probability of failure to start for ith backup generator
G	Cumulative distribution function for recovery of offsite power
R_j	Cumulative distribution function for repair of jth generator
T	Mission time
T_c	Critical time
T_{CSBO}	Time of critical station blackout
$\hat{\tau}$	Time of failure of last train (same as T_{CSBO})
τ_j	Time of failure of jth train
P_i	Probability that the system is in state i
\tilde{P}_i	Regarded as a known constant for a system in state i
n	*order
$P^{(n)}$	Probability of order n
μ	Repair Rate
μ_i	Repair Rate of ith backup generator
λ	Failure Rate
λ_i	*Failure Rate of ith backup generator
λ_{ij}	*Failure Rate of ith and jth backup generator (two-way)
λ_{ijk}	*Failure Rate of ith, jth and kth backup generator (three-way)
λ_1	*Failure Rate of Independent Failures
λ_2	*Failure Rate of common cause doublet Failures
λ_3	*Failure Rate of common cause triplet Failures

* Certain variables might be used multiple times with different meaning. See associated section to make sure correct variable meaning is used.

Table 3.2

Second Part of Variables and Definitions for the Derivation of the Non-Recovery Integral.

Variable	Definition
ε	Small parameter for order of failure rates
$O(\varepsilon)$	Order of epsilon
$p_i(\tau_i)$	Net rate at which EDG i is failing at time τ_i . Where is defined as the probability density function for independent failures while the i th EDG is operating.
τ_i	Time of failure of EDG i
τ_{ij}	Time of common-cause doublet failure of EDG i and j
τ_{ijk}	Time of common-cause doublet failure of EDG i , j and k
p_i	Probability density function for independent failures while the i th EDG is operating.
p_{ij}	Probability density function for common-cause doublet failure of EDG i and j with both are operating
p_{ijk}	Probability density function for common-cause triplet failure of EDG i , j and k with both are operating
H	Non-Recovery Function

* Certain variables might be used multiple times with different meaning. See associated section to make sure correct variable meaning is used.

Section 3.2 – State-Transition Equations for Two-EDG Case

From Figure 3.1, the state-transition differential equations are shown in Equations (3.1) though (3.4).

$$\frac{dP_0}{dt} = -\varepsilon\lambda_0 P_0 - \varepsilon\lambda_1 P_0 - \varepsilon^2\lambda_{01} P_0 + \mu_0 P_1 + \mu_1 P_2 \quad (3.1)$$

$$\frac{dP_1}{dt} = -\mu_0 P_1 - \varepsilon \lambda_1 P_1 - \varepsilon^2 \lambda_{01} P_1 + \varepsilon \lambda_0 P_0 \quad (3.2)$$

$$\frac{dP_2}{dt} = -\mu_1 P_2 - \varepsilon \lambda_0 P_2 - \varepsilon^2 \lambda_{01} P_2 + \varepsilon \lambda_1 P_0 \quad (3.3)$$

$$\frac{dP_3}{dt} = -\varepsilon \lambda_1 P_1 + \varepsilon^2 \lambda_{01} P_1 + \varepsilon \lambda_0 P_2 + \varepsilon^2 \lambda_{01} P_2 + \varepsilon^2 \lambda_{01} P_0 \quad (3.4)$$

Here $P_i = P_i(t)$ is the probability that the system is in state i at time t . Manipulations of these equations would allow this system of four first order differential equations (ODEs) in four unknowns to be written more compactly. It is more convenient to leave it in this form for the manipulation following.

The accompanying initial conditions for this system are shown in Equations (3.5) through (3.8).

$$P_0(0) = 1 - \varepsilon \tilde{P}_1^{(1)} - \varepsilon \tilde{P}_2^{(1)} - \varepsilon^2 \tilde{P}_3^{(2)} \quad (3.5)$$

$$P_1(0) = \varepsilon \tilde{P}_1^{(1)} \quad (3.6)$$

$$P_2(0) = \varepsilon \tilde{P}_2^{(1)} \quad (3.7)$$

$$P_3(0) = \varepsilon^2 \tilde{P}_3^{(2)} \quad (3.8)$$

For the moment the quantities $\tilde{P}_1^{(1)}$, $\tilde{P}_2^{(1)}$, and $\tilde{P}_3^{(2)}$ are to be regarded merely as known constants; however, the parenthetical superscripts have been judiciously selected to provide a comfortable fit into the notational framework to be introduced shortly. Also, the epsilons reflect the assumption that a given EDG is available but the other either is unavailable (e.g., for maintenance) or fails to start at time of the initiation LOOP event is small ($O(\varepsilon)$) and that the probability that each is either unavailable or fails to start is even smaller yet ($O(\varepsilon^2)$).

Section 3.3 – Asymptotic Solution for Two-EDG Case

It is rather trivial to solve the initial Equations ((3.1)-(3.8)) numerically; however, the calculation of the contribution to the time-dependent probability of station blackout,

$P_3(t)$, from the minimum possible number of EDG failures that could cause a blackout, two in this model, is needed. While that computation could be carried out analytically, by evaluating all of the integrals that would arise, it is intended to leave those integrals unevaluated. This is to arrive at a result having the form of the nonrecovery integral found in Rodgers et al. [1], except for the inclusion of common-cause contributions associated with the rate constant λ_{12} . The hope is that the form of this common-cause contribution will suggest how common-cause failures should be incorporated into the nonrecovery integral in the more general setting.

The parameter ε introduced in the state-transition equations will be indispensable to this calculation, which will require the use of an asymptotic expansion process. The solution should be in the form of Equation (3.9).

$$P_i(t) = P_i^{(0)}(t) - \varepsilon P_i^{(1)}(t) - \varepsilon^2 P_i^{(2)}(t) + \dots = \sum_{n=0}^{\infty} \varepsilon^n P_i^{(n)}(t), \quad i = 0, \dots, 3 \quad (3.9)$$

For all values of n up to the smallest, $P_i^{(n)}(t)$ is not identically zero. The anticipated value of this index is $n=2$, but this needs more proof. The procedure by which the solution will be obtained is to substitute the asymptotic expansions (3.9) into the initial value problem (3.1)-(3.8) and systematically equate the coefficients of terms of like order n in ε , for first $n=0$, then $n=1$ and etc. At a given value of n , this will give a determined initial value problem in the four unknowns $P_0^{(n)}(t)$, $P_1^{(n)}(t)$, $P_2^{(n)}(t)$, and $P_3^{(n)}(t)$. The hope is for the initial value problems to be sufficiently simpler than state-transition equations to permit them to be solved analytically, at least up to the value of interest n , assumed as a value of two.

There is an interpretation of the asymptotic solution to order n that can serve as a useful check on the computations done in MATLAB. The asymptotic solution up to order n should agree with the solution that would prevail if a maximum of n failure occurs. This is where single-point failures, independent and one-way common-cause failures as incorporated in the failure rates λ_i are counted as single failures which is with multiplicity

of one. By contrast, common-cause doublet failures, as incorporated in the failure rate λ_{01} , are counted with multiplicity of two.

Subsection 3.3.a – For the n=0 Case for Two-EDG Case

The results of carrying out the preceding procedure at order zero, equating coefficients of ε^0 in the state-transition equations lead to the equations seen in Equations (3.10)-(3.13). The initial conditions for the previous set of equations can be seen in (3.14) through (3.17).

$$\frac{dP_0^{(0)}}{dt} = -\mu_0 P_1^{(0)} + \mu_0 P_2^{(0)} \quad (3.10)$$

$$\frac{dP_1^{(0)}}{dt} = -\mu_0 P_1^{(0)} \quad (3.11)$$

$$\frac{dP_2^{(0)}}{dt} = -\mu_1 P_2^{(0)} \quad (3.12)$$

$$\frac{dP_3^{(0)}}{dt} = 0 \quad (3.13)$$

$$P_0^{(0)}(0) = 1 \quad (3.14)$$

$$P_1^{(0)}(0) = 0 \quad (3.15)$$

$$P_2^{(0)}(0) = 0 \quad (3.16)$$

$$P_3^{(0)}(0) = 0 \quad (3.17)$$

This system can easily be solved. The solution of Equations (3.13) and (3.17) is obvious to see as it is zero throughout, or $P_3^{(0)}(t) \equiv 0$. The solutions from Equations (3.11) and (3.12) can then be readily solved and shown as $P_1^{(0)}(t) \equiv 0$ and $P_2^{(0)}(t) \equiv 0$, respectively. After, that these can be input into Equation (3.10) and the solution can be seen as $P_0^{(0)}(t) \equiv 1$. Therefore, for the lowest order in ε , the emergency AC system analyzed performs perfectly. That is in agreement with the interpretation suggested above of the order n as the maximum number of failures allowed for, counting number by

multiplicities as described in the first part of Section 3.3. At present that order is zero, meaning zero failures. However, if there are no failures, then the initial conditions shown in Equations (3.5) through (3.8) assert that the system initially is in state 0, both EDGs running, with certainty, probability of one, and if there are no subsequent failures, then that certainty persists for an indefinite time. The results obtained are comforting but used as mostly a check on our assumptions and analysis. The interpretation suggested for the solution of order n is also checked by the result; although, that is hardly surprising.

Subsection 3.3.b – For the $n=1$ Case for Two-EDG Case

For the lower order case $n=0$, it is expected that the solution of the differential equation for $P_3^{(1)}$ to be the catalyst that permits analytic solution of the remainder of the first-order system of ODEs. If the first order terms in Equation (3.4) were equated with the initial conditions, the result would be Equations (3.18) and (3.19) which is where this methodology would start.

$$\frac{dP_3^{(1)}}{dt} = \lambda_1 P_1^{(0)} + \lambda_0 P_2^{(0)} \quad (3.18)$$

$$P_3^{(1)}(0) = 0 \quad (3.19)$$

Except from the results from the $n=0$ case, it is known that $P_1^{(0)}(t) = P_2^{(0)}(t) \equiv 0$, so these equations can simplify to Equation (3.20).

$$\frac{dP_3^{(1)}}{dt} = 0 \quad (3.20)$$

This equation still using Equation (3.19) as the initial condition and the solution is $P_3^{(1)}(t) \equiv 0$, is expected. This result already shows that no “single-point failures”, i.e. no failure of a single EDG, where failures are counted with multiplicities as indicated previously, will lead to station blackout, which is a design criterion.

Similarly, the first-order counterpart of Equation (3.3) and of the associated initial conditions result in Equations (3.21) and (3.22).

$$\frac{dP_2^{(1)}}{dt} = -\mu_1 P_2^{(1)} - \lambda_0 P_2^{(0)} + \lambda_1 P_0^{(0)} \quad (3.21)$$

$$P_2^{(1)}(0) = \tilde{P}_2^{(1)} \quad (3.22)$$

From the results at order $n=0$, it is known that $P_2^{(0)}(t) \equiv 0$ and $P_0^{(0)}(t) \equiv 1$, and from preceding computation we know that $P_3^{(1)}(t) \equiv 0$. Therefore, Equation (3.21) becomes Equation (3.23) which still uses Equation (3.22) as the initial value.

$$\frac{dP_2^{(1)}}{dt} = -\mu_1 P_2^{(1)} + \lambda_1 \quad (3.23)$$

The solution of this linear first order initial value problem, Equation (3.23), is then Equation (3.24).

$$P_2^{(1)}(t) = \tilde{P}_2^{(1)} e^{-\mu_1 t} + \lambda_1 \int_0^t e^{-\mu_1(t-\tau_0)} d\tau_0 = \tilde{P}_2^{(1)} e^{-\mu_1 t} + \left(\frac{\lambda_1}{\mu_1} \right) (1 - e^{-\mu_1 t}) \quad (3.24)$$

Similarly, the first-order contribution to $P_1^{(1)}(t)$ is seen in Equation (3.25).

$$P_1^{(1)}(t) = \tilde{P}_1^{(1)} e^{-\mu_0 t} + \lambda_0 \int_0^t e^{-\mu_0(t-\tau_0)} d\tau_0 = \tilde{P}_1^{(1)} e^{-\mu_0 t} + \left(\frac{\lambda_0}{\mu_0} \right) (1 - e^{-\mu_0 t}) \quad (3.25)$$

The rightmost forms of Equations (3.24) and (3.25) show the fully analytic forms for the probabilities, as a function of time, of the states having a signal failed EDG. The intermediate forms, in terms of the definite integrals, will be most convenient for the purposes of reproducing the lowest nonzero order of $P_3^{(n)}$ in the form of the nonrecovery integral in Rodgers et al. [1].

The first-order differential Equation (3.1) for $P_0^{(1)}$ and associated initial condition are shown in Equation (3.26).

$$\frac{dP_0^{(1)}}{dt} = \tilde{P}_1^{(1)} \mu_0 e^{-\mu_0 t} - \lambda_0 e^{-\mu_0 t} + \tilde{P}_2^{(1)} \mu_1 e^{-\mu_1 t} - \lambda_1 e^{-\mu_1 t} \quad (3.26)$$

The results from the $n=0$ order case give $P_0^{(0)} \equiv 1$, and the prior results from the current first order case give the expressions for $P_1^{(1)}$ and $P_2^{(1)}$ are found in Equations (3.24) and (3.25) respectively. Substituting these previously listed equations, (3.24) and (3.25), into Equation (3.26) one can solve for Equations (3.27) and (3.28)

$$\frac{dP_0^{(1)}}{dt} = -\lambda_0 - \lambda_1 - \mu_0 \left\{ \tilde{P}_1^{(1)} e^{-\mu_0 t} + \left(\frac{\lambda_0}{\mu_0} \right) (1 - e^{-\mu_0 t}) \right\} + \mu_1 \left\{ \tilde{P}_2^{(1)} e^{-\mu_1 t} + \left(\frac{\lambda_1}{\mu_1} \right) (1 - e^{-\mu_1 t}) \right\} \quad (3.27)$$

$$P_0^{(1)}(0) = -\tilde{P}_1^{(1)} - \tilde{P}_2^{(1)} \quad (3.28)$$

Algebraically Equation (3.27) simplifies to Equation (3.29).

$$\frac{dP_0^{(1)}}{dt} = \tilde{P}_1^{(1)} \mu_0 e^{-\mu_0 t} - \lambda_0 e^{-\mu_0 t} + \tilde{P}_2^{(1)} \mu_1 e^{-\mu_1 t} - \lambda_1 e^{-\mu_1 t} \quad (3.29)$$

The initial value problem is an evaluation of a definite integral so the solution to Equation (3.29) can be seen in Equation (3.30).

$$P_0^{(1)}(t) = \left[\frac{\lambda_0}{\mu_0} - \tilde{P}_1^{(1)} \right] e^{-\mu_0 t} - \left[\frac{\lambda_1}{\mu_1} - \tilde{P}_2^{(1)} \right] e^{-\mu_1 t} - \frac{\lambda_0}{\mu_0} - \frac{\lambda_1}{\mu_1} \quad (3.30)$$

This is simplified as shown in Equation (3.31).

$$P_0^{(1)}(t) = -\tilde{P}_1^{(1)} e^{-\mu_0 t} - \left(\frac{\lambda_0}{\mu_0} \right) (1 - e^{-\mu_0 t}) - \tilde{P}_2^{(1)} e^{-\mu_1 t} - \left(\frac{\lambda_1}{\mu_1} \right) (1 - e^{-\mu_1 t}) \quad (3.31)$$

Equation (3.31) can be written in terms of Equations (3.24) and (3.25). This is shown in Equation (3.32).

$$P_0^{(1)}(t) = -P_1^{(1)}(t) - P_2^{(1)}(t) \quad (3.32)$$

With $P_3^{(1)}(t) = 0$, this implies that the net sum of all the first order probabilities is zero. It would be of some interest to note whether this is true for the probabilities of all finite orders, other than the zero case for which the net sum would be equal to one. If this is the case, then the total net sum of all probabilities up to any finite order is unity. This exercise should be considered for future work.

This is not our primary interest that is to obtain the analytic expression for $P_3^{(2)}$, which will not require knowledge of $P_0^{(1)}$ to obtain.

Subsection 3.3.c– For the n=2 Case for Two-EDG Case

As for the $n=1$ order case, the appropriate order initial-value problem P_3 for [Equations (3.33) and (3.34)] is started with and subsequent developments ensue.

$$\frac{dP_3^{(2)}}{dt} = -\lambda_1 P_1^{(1)} + -\lambda_{01} P_1^{(0)} + \lambda_0 P_2^{(1)} + \lambda_{01} P_2^{(0)} + \lambda_{01} P_0^{(0)} \quad (3.33)$$

$$P_3^{(2)}(0) = \tilde{P}_3^{(2)} \quad (3.34)$$

With the knowledge of the results from the previous two subsections, the results from Equations (3.25), $P_1^{(0)}(t) = 0$, (3.24), $P_2^{(0)}(t) = 0$, and (3.33), $P_0^{(0)}(t) = 1$, this becomes Equation (3.35).

$$\begin{aligned} \frac{dP_3^{(2)}}{dt} = & -\lambda_1 \left\{ \tilde{P}_1^{(1)} e^{-\mu_0 t} + \lambda_0 \int_0^t e^{-\mu_0(t-\tau_0)} d\tau_0 \right\} \\ & -\lambda_0 \left\{ \tilde{P}_2^{(1)} e^{-\mu_1 t} + \lambda_1 \int_0^t e^{-\mu_1(t-\tau_1)} d\tau_1 \right\} + \lambda_{01} \end{aligned} \quad (3.35)$$

If Equation (3.35) is integrated from 0 to the time constant t , the result can be evaluated for $P_3^{(2)}(t)$ as seen in Equation (3.36).

$$\begin{aligned} P_3^{(2)}(t) = & \tilde{P}_3^{(2)} + \tilde{P}_1^{(1)} \int_0^t \lambda_1 e^{-\mu_0 \tau_1} d\tau_1 \\ & + \int_0^t \lambda_1 \int_0^{\tau_1} \lambda_0 e^{-\mu_0(\tau_1-\tau_0)} d\tau_0 d\tau_1 \\ & + \tilde{P}_2^{(1)} \int_0^t \lambda_0 e^{-\mu_1 \tau_0} d\tau_0 \\ & + \int_0^t \lambda_0 \int_0^{\tau_1} \lambda_1 e^{-\mu_1(\tau_0-\tau_1)} d\tau_1 d\tau_0 + \lambda_{01} \int_0^t d\tau_{01} \end{aligned} \quad (3.36)$$

The various integrals on the right-hand side of Equation (3.36) can be evaluated analytically; however, for the purposes of this thesis, this form is preferred because integrands have a probabilistic interpretation that connects to a possible general form of a non-recovery integral. Using that interpretation, it is left open to future work if the integrals need to be evaluated further.

Section 3.4 – Interpretation as a Nonrecovery Integral for Two-EDG

Case

The contributions of the right hand side of Equation (3.36) can be rearranged and rewritten in a form that groups together terms of similar interpretations. If it is written for some specific mission time T , the result can be written as Equation (3.37).

$$\begin{aligned}
P_3^{(2)}(T) = & \tilde{P}_3^{(2)} + \tilde{P}_1^{(1)} \int_0^T \lambda_1 e^{-\mu_0 \tau_1} d\tau_1 + \tilde{P}_2^{(1)} \int_0^T \lambda_0 e^{-\mu_1 \tau_0} d\tau_0 \\
& + \int_0^T \int_0^T \lambda_1 \lambda_0 e^{-\tilde{\mu}(\hat{\tau} - \tilde{\tau})} d\tau_0 d\tau_1 + \lambda_{01} \int_0^T d\tau_{01}
\end{aligned} \tag{3.37}$$

In Equation (3.37), $\hat{\tau} = \max\{\tau_0, \tau_1\}$, $\tilde{\tau} = \min\{\tau_0, \tau_1\}$, and $\tilde{\mu} = \mu_i$, whenever $\tilde{\tau} = \tau_i$ for the i th electrical train or EDG. The double integral in Equation (3.37) is comprised of two iterated integrals in Equation (3.36). This resulted from noticing that the first of the iterated integrals in Equation (3.36) is over the portion of the square defined by $\{(\tau_0, \tau_1) | 0 \leq \tau_0 \leq t, 0 \leq \tau_1 \leq t\}$ such that $\tau_0 \leq \tau_1$, which is to refer to above the diagonal defined by $\tau_0 = \tau_1$ in the $\tau_0 \tau_1$ plane. Above that diagonal the model relates to $\tilde{\tau} = \tau_1$, $\hat{\tau} = \tau_0$, and $\tilde{\mu} = \mu_0$. Therefore, in this region, the integrand in the first of the iterated integrals in Equation (3.36) is the same as that of the double integral found in Equation (3.37). Similarly, the region of integration in the second iterated integral in Equation (3.36) is the same part of the square above the diagonal in the $\tau_0 \tau_1$ plane, and in that region the integrand in that iterated integral is the same as that double integral found in Equation (3.37). The sum of the two iterated integrals from Equation (3.36) is then the same thing as the double integral found in Equation (3.37).

Further interpretation of Equation (3.37) as the nonrecovery integral, with common-cause failure contribution included, it is useful to consider what the form of this equation would be without common-cause failures. The difference between these two equations would then show the attributions of common-cause failure. This would represent what must be added to the nonrecovery integral from Rodgers et al. [1] in order to incorporate common-cause failure.

In the absence of common-cause failures, $\lambda_{12} = 0$, therefore, the last term in Equation (3.37) would disappear. Further it is noted that $\tilde{P}_3^{(2)} = \hat{P}_0^{(1)} \hat{P}_1^{(1)}$, where $\hat{P}_1^{(1)}$ is the $O(\varepsilon)$ probability that EDG i initially is unavailable, independently of the status of the

other EDG. Note that the \tilde{P} variables are indexed to a particular EDG. This is logical but can be somewhat confusing as the remaining singly subscripted variants of P are indexed to Markov states. Additionally, in the present case $\tilde{P}_i^{(1)} = \hat{P}_i^{(1)}$, for i equals 1 or 2, is the independent probability that respectively EDG 0 or 1 initially unavailable. With these parameters taken into account, the equation for the absence of common-cause failures becomes Equation (3.38).

$$P_3^{(2)}(T) = \varepsilon \hat{P}_0^{(1)} \hat{P}_1^{(1)} + \hat{P}_0^{(1)} \int_0^T \lambda_1 e^{-\mu_0 \tau_1} d\tau_1 + \hat{P}_1^{(1)} \int_0^T \lambda_0 e^{-\mu_1 \tau_0} d\tau_0 + \int_0^T \int_0^T \lambda_1 \lambda_0 e^{-\tilde{\mu}(\hat{\tau}-\tilde{\tau})} d\tau_0 d\tau_1 \quad (3.38)$$

As specialized to the present case, the corresponding expanded form of the nonrecovery integral as presented in Rodgers et al. [1] is shown in Equation (3.39).

$$\varepsilon^2 P_3^{(2)}(T) = \varepsilon \hat{P}_0^{(1)} \varepsilon \hat{P}_1^{(1)} + \hat{P}_1^{(1)} \int_0^T (1 - \varepsilon \hat{P}_1^{(1)}) \varepsilon \lambda_1 e^{-\mu_0 \tau_1} d\tau_1 + \varepsilon \hat{P}_1^{(1)} \int_0^T (1 - \varepsilon \hat{P}_0^{(1)}) \varepsilon \lambda_0 e^{-\mu_1 \tau_0} d\tau_0 + \int_0^T \int_0^T \varepsilon \lambda_1 \varepsilon \lambda_0 e^{-\tilde{\mu}(\hat{\tau}-\tilde{\tau})} d\tau_0 d\tau_1 \quad (3.39)$$

The asymptotic order parameter, ε , has temporarily been reinserted adjacent to the term it is associated with in order to allow the following observation: To the leading order of ε^2 , the terms $\hat{P}_i^{(1)}$ in the two single integrals of Equation (3.39) are negligible. Therefore, to this order, and with the deletion of the asymptotic order parameter, Equation (3.39) becomes equivalent to Equation (3.38). Then the leading order of the nonrecovery integral reported by Rodgers et al. [1], with critical time set to zero, agrees with the leading order approximation of the probability of station blackout as determined from the current asymptotic expansion.

However, both of these results ignore contributions from common-cause failures. It is now important to look at the difference between Equations (3.38) and (3.37), the latter of which does contain contributions from common-cause failures. The idea is to

look at the interpretation of the differences in terms of common-cause failures. The hope is that this interpretation will suggest an appropriate form for the nonrecovery integral, with common-cause failures included, with more general failure and repair distributions uses and with a larger number of EDGs (at least three).

Section 3.5– Common-Cause Failure Contributions for Two-EDG Case

Equation (3.37) can be written as Equation (3.40).

$$\begin{aligned}
P_3^{(2)}(T) = & \hat{P}_0^{(1)} \hat{P}_1^{(1)} + \hat{P}_0^{(1)} \int_0^T \lambda_1 e^{-\mu_0 \tau_1} d\tau_1 + \hat{P}_1^{(1)} \int_0^T \lambda_0 e^{-\mu_1 \tau_0} d\tau_0 \\
& + \int_0^T \int_0^T \lambda_1 \lambda_0 e^{-\tilde{\mu}(\hat{\tau} - \tilde{\tau})} d\tau_0 d\tau_1 + \left(\tilde{P}_3^{(2)} - \hat{P}_0^{(1)} \hat{P}_1^{(1)} \right) + \left(\tilde{P}_1^{(1)} - \hat{P}_0^{(1)} \right) \int_0^T \lambda_1 e^{-\mu_0 \tau_1} d\tau_1 \\
& + \left(\hat{P}_2^{(1)} - \hat{P}_1^{(1)} \right) \int_0^T \lambda_0 e^{-\mu_1 \tau_0} d\tau_0 + \lambda_{01} \int_0^T d\tau_{01}
\end{aligned} \tag{3.40}$$

The first four terms in the right hand side of Equation (3.40) are exactly the same as those found in Equation (3.38). These terms exist due to the contribution of the probability of station blackout, within some mission time T , caused by independent failures of the two EDGS. Then the contributions from common-cause events must be the other four terms of Equation (3.40).

More specifically Equation (3.41) shows the difference between the total probability that both of the EDGS are initially unavailable and the probability of the two EDGs being initially unavailable due to independent random events.

$$\hat{P}_{01}^{(2)} = \tilde{P}_3^{(2)} - \hat{P}_0^{(1)} \hat{P}_1^{(1)} \tag{3.41}$$

Therefore, it must be the probability of both being unavailable due to a common-cause event.

Similarly, Equation (3.42) has the interpretation of the probability that EDG 0 initially is unavailable because of common-cause event shock that caused it to fail, but

not EDG 1. As a result, EDG 1 would still be in operation. Note that a shock is a common-cause event that causes a system to fail. EDG A fails which in turn causes EDG B to fail so B would fail by result of a shock.

$$\widehat{P}_0^{(1)} = \widetilde{P}_1^{(1)} - \widehat{P}_0^{(1)} \quad (3.42)$$

Correspondingly, Equation (3.43) is the probability that EDG 0 is available at the time of the LOOP but EDG 1 is not, even though a common-cause event occurred that had the potential to cause both to be unavailable.

$$\widehat{P}_1^{(1)} = \widetilde{P}_2^{(1)} - \widehat{P}_1^{(1)} \quad (3.43)$$

Finally, the last term of the right hand side of Equation (3.40) is, to second order in epsilon, the probability at the end of the mission time that the station blackout has occurred. Both EDGs initially were available but became unavailable during the mission time because of the same common-cause basic event. This knowledge leads Equation (3.40) to become Equation (3.44).

$$\begin{aligned}
P_7^{(2)}(T) = & \left\{ \begin{aligned} & \prod_{i=1}^3 \tilde{P}_i + \sum_{i=1}^3 \tilde{P}_i (1 - \tilde{P}_i \tilde{P}_{i''}) \hat{P}_{i''} \\ & + \left[1 - \prod_{i=1}^3 \tilde{P}_i + \sum_{i=1}^3 \tilde{P}_i (1 - \tilde{P}_i \tilde{P}_{i''}) \hat{P}_{i''} \right] \hat{P}_{012} \end{aligned} \right\} H_3(T_c, T_c, T_c) \\
& + \sum_{i=1}^3 \left\{ \tilde{P}_i \tilde{P}_{i''} + (1 - (\tilde{P}_i \tilde{P}_{i''})) \hat{P}_{i''} \right\} \int_0^T p_i(\tau_i) H_3(T_i = T_c, T_{i'} = \tau_i + T_c, T_{i''} = \tau_i + T_c) d\tau_i \\
& + \sum_{i=1}^3 \tilde{P}_i \left\{ \begin{aligned} & \int_0^T \int_0^T p_{i'}(\tau_{i'}) p_{i''}(\tau_{i''}) * H_3 \left(\begin{aligned} & T_i = \max\{\tau_{i'}, \tau_{i''}\} + T_c, \\ & T_{i'} = \max\{\tau_{i'}, \tau_{i''}\} + T_c - \tau_{i'}, \\ & T_{i''} = \max\{\tau_{i'}, \tau_{i''}\} + T_c - \tau_{i''} \end{aligned} \right) d\tau_{i'} d\tau_{i''} \\ & + \int_0^T p_{i''}(\tau_{i''}) H_3(T_i = \tau_{i''} + T_c, T_{i'} = T_{i''} = T_c) d\tau_{i''} \end{aligned} \right\} \quad (3.44) \\
& + \int_0^T \int_0^T \int_0^T p_1(\tau_1) p_2(\tau_2) p_3(\tau_3) H_3 \left(\begin{aligned} & \max_{j=1}^3 \{\tau_j\} + T_c - \tau_1, \max_{j=1}^3 \{\tau_j\} + \\ & T_c - \tau_2, \max_{j=1}^3 \{\tau_j\} + T_c - \tau_3 \end{aligned} \right) d\tau_1 d\tau_2 d\tau_3 \\
& + \sum_{i=1}^3 \int_0^T \int_0^T p_i(\tau_i) p_{i''}(\tau_{i''}) H_3 \left(\begin{aligned} & T_i = \max\{\tau_i, \tau_{i''}\} + T_c - \tau_i, \\ & T_{i'} = T_{i''} = \max\{\tau_i, \tau_{i''}\} + T_c - \tau_{i''} \end{aligned} \right) d\tau_i d\tau_{i''} \\
& + H_3(T_c, T_c, T_c) \int_0^T p_{012}(\tau_{012}) d\tau_{012}
\end{aligned}$$

It should be noted that both terms on the right hand side of Equation (3.41) are of order epsilon squared, $O(\varepsilon^2)$ so their difference has the order of ε^2 or higher. A better understanding of how to approach this problem would be to write the initial condition for state 3 as Equation (3.45).

$$P_3(0) = \varepsilon^2 \hat{P}_0^{(1)} \hat{P}_1^{(1)} + \varepsilon^n \hat{P}_{01}^{(n)} \quad (3.45)$$

The terms on the right hand side have the same interpretation as the preceding discussion above, but with the asymptotic order of the common-cause doublet failure chosen as appropriate to the problem at hand. For three EDGs, $n=2$ probably, although ideal use would require $n=3$, similar remarks apply to Equations (3.42) and (3.43). These would also apply to Equation (3.46).

$$\varepsilon\lambda_0 = \varepsilon\hat{\lambda}_0^{(1)} + \varepsilon^2\hat{\lambda}_0^{(2)} \quad (3.46)$$

In Equation (3.46), the first term on the right hand side refers to independent failure (hazard) rate and the second is the one-way common-cause failure rate for EDG 0.

Section 3.6– The General Form for the Two-EDG Case

In the first and second lines of Equation (3.44), the identification shown in Equation (3.47) is made.

$$\begin{aligned} p_0(\tau_0) &:= \lambda_0(P_0(\tau_0) + P_2(\tau_0)) = \lambda_0(P_0^{(0)}(\tau_0) + P_2^{(0)}(\tau_0)) + O(\varepsilon^2) \\ &= \lambda_0(1+0) + O(\varepsilon^2) = \lambda_0 + O(\varepsilon^2) \end{aligned} \quad (3.47)$$

Similarly, Equation (3.48) can be determined and written.

$$\begin{aligned} p_1(\tau_1) &:= \lambda_1(P_0(\tau_1) + P_1(\tau_1)) = \lambda_1(P_0^{(0)}(\tau_1) + P_1^{(0)}(\tau_1)) + O(\varepsilon^2) \\ &= \lambda_1(1+0) + O(\varepsilon^2) = \lambda_1 + O(\varepsilon^2) \end{aligned} \quad (3.48)$$

Note that $p_i(\tau_i)$, via the defining expression, is the net rate at which EDG i is failing at time τ_i . This p_i is defined as the probability density function for independent failures while the i th EDG is operating.

Similarly, Equation (3.49) used the fact that λ_i itself is order of epsilon, $O(\varepsilon)$.

$$\lambda_0\lambda_1 = p_0(\tau_0)p_1(\tau_1) + O(\varepsilon^3) \quad (3.49)$$

Likewise, in the last four terms in Equation (3.44) on the right hand side, identification needs to be made which is shown in Equation (3.50). Note that p_{01} is the pdf for common-cause doublet failures.

$$p_{01}(\tau_{01}) := \lambda_{01} P_0(\tau_{01}) = \lambda_{01} P_0^{(0)} + O(\varepsilon^3) = \lambda_{01} + O(\varepsilon^3) \quad (3.50)$$

The net result of substituting the identifications into Equation (3.44) results in Equation (3.51).

$$\begin{aligned} P_3^{(2)}(T) &= \hat{P}_0^{(1)} \hat{P}_1^{(1)} + \hat{P}_0^{(1)} \int_0^T p_1(\tau_1) e^{-\mu_0 \tau_1} d\tau_1 + \hat{P}_1^{(1)} \int_0^T p_0(\tau_0) e^{-\mu_1 \tau_0} d\tau_0 \\ &\quad + \int_0^T \int_0^T p_0(\tau_0) p_1(\tau_1) e^{-\tilde{\mu}(\hat{t}-\tilde{\tau})} d\tau_0 d\tau_1 + \hat{P}_{01}^{(2)} + \hat{P}_0^{(1)} \int_0^T p_1(\tau_1) e^{-\mu_0 \tau_1} d\tau_1 \\ &\quad + \hat{P}_1^{(1)} \int_0^T p_0(\tau_0) e^{-\mu_1 \tau_0} d\tau_0 + \int_0^T p_{01}(\tau_{01}) d\tau_{01} + O(\varepsilon^3) \end{aligned} \quad (3.51)$$

If the $O(\varepsilon^3)$ correction is neglected, then the result from Equation (3.47) is the two-EDG case in the form of the non-recovery integral with common-cause failures incorporated.

Section 3.7– For the Three-EDG Case

For the three emergency diesel generator case, the conjectured form of the probability of a station blackout occurring within the mission time, while also allocating an addition critical time T_c for recovery following SBO before coping systems must be involved in Equation (3.52).

$$\begin{aligned}
P_7^{(2)}(T) &= \left\{ \begin{aligned} &\prod_{i=1}^3 \tilde{P}_i + \sum_{i=1}^3 \tilde{P}_i (1 - \tilde{P}_i, \tilde{P}_{i''}) \hat{P}_{i',i''} \\ &+ \left[1 - \prod_{i=1}^3 \tilde{P}_i + \sum_{i=1}^3 \tilde{P}_i (1 - \tilde{P}_i, \tilde{P}_{i''}) \hat{P}_{i',i''} \right] \hat{P}_{012} \end{aligned} \right\}^* \\
H_3(T_c, T_c, T_c) &+ \sum_{i=1}^3 \left\{ \tilde{P}_i, \tilde{P}_{i''} + (1 - (\tilde{P}_i, \tilde{P}_{i''})) \hat{P}_{i',i''} \right\} \int_0^T p_i(\tau_i)^* \\
H_3 \left(\begin{array}{l} T_i = T_c, T_{i'} = \tau_i + T_c, \\ T_{i''} = \tau_i + T_c \end{array} \right) &d\tau_i \\
+ \sum_{i=1}^3 \tilde{P}_i &\left\{ H_3 \left(\begin{array}{l} T_i = \max\{\tau_{i'}, \tau_{i''}\} + T_c, \\ T_{i'} = \max\{\tau_{i'}, \tau_{i''}\} + T_c - \tau_{i'}, \\ T_{i''} = \max\{\tau_{i'}, \tau_{i''}\} + T_c - \tau_{i''} \end{array} \right) d\tau_{i'} d\tau_{i''} \right. \\
&\left. + \int_0^T p_{i',i''}(\tau_{i',i''}) H_3(T_i = \tau_{i',i''} + T_c, T_{i'} = T_{i''} = T_c) d\tau_{i',i''} \right\} \\
+ \int_0^T \int_0^T \int_0^T &p_1(\tau_1) p_2(\tau_2) p_3(\tau_3)^* \\
H_3 \left(\begin{array}{l} \max_{j=1}^3 \{\tau_j\} + T_c - \tau_1, \\ \max_{j=1}^3 \{\tau_j\} + T_c - \tau_2, \\ \max_{j=1}^3 \{\tau_j\} + T_c - \tau_3 \end{array} \right) &d\tau_1 d\tau_2 d\tau_3 \\
+ \sum_{i=100}^3 \int_0^T \int_0^T &p_i(\tau_i) p_{i',i''}(\tau_{i',i''}) H_3 \left(\begin{array}{l} T_i = \max\{\tau_i, \tau_{i',i''}\} + T_c - \tau_i, \\ T_{i'} = T_{i''} = \max\{\tau_i, \tau_{i',i''}\} + T_c - \tau_{i',i''} \end{array} \right) d\tau_i d\tau_{i',i''} \\
+ H_3(T_c, T_c, T_c) &\int_0^T p_{012}(\tau_{012}) d\tau_{012}
\end{aligned} \tag{3.52}$$

Here in Equation (3.52), $\tilde{P}_i^{(1)}$ is the probability that EDG i is unavailable at the onset of LOOP, from either independent or a one-way common-cause event; $i'(i'')$ is the smaller (larger) of the two indices $\{0,1,2\}$ different from i ; \hat{P}_{ij} is the probability EDGs i

and j initially are unavailable because of a 2CCF; \hat{P}_{012} is the probability that all three EDGs initially are unavailable because of a three-way CCF; $H_3(T_1, T_2, T_3)$ is the probability that none of the three EDGs recovers after respective (co-terminating) durations of T_i from failure of the i th EDG; p_i is the probability density function for independent or one-way common-cause failures of EDG i while running; $p_{ij} (i < j)$ is the probability density function for common-cause doublet failures of EDGs i and j ; and p_{012} is the probability density function for common-cause triplet failures.

Equation (3.52) can be derived in detail as follows. First, the six different types of events that would cause failure of an emergency diesel generator, which are labeled respectively and in order as S3, R3, S2, R2, S1, and R1, are distinguished. S3 and R3 represent common-cause triplet failures to respectively start or run, S2 and R2 represent common-cause doublet failures to respectively start or run, and S1 and R1 represent one-way common-cause and independent failures to respectively start or run. A minimal cut set then corresponds to any unordered string formed from these symbols that represents failure of all three EDGs. Such a deletion of any individual symbol from that string cannot represent failure of all three EDGs. By convention, we then list the symbols in any individual string in the order of the above list. In this lexicographic order defined by this convention there are then the following 13 classes of minimal cut sets shown in Table 3.3. These thirteen classes of cut sets defined in Table 3.3, are represented by the ten terms in Equation (3.52) as also shown in the second column of Table 3.3. Note that the other three terms that appear in Table 3.1 as “*N/A” are labeled in this way due to the following reason. If independent and one-way common-cause failures are assumed to be $O(\varepsilon)$, then the lowest-order contributions to the right-hand side of Equation (3.52) are $O(\varepsilon^2)$; however, if common-cause doublet failures are $O(\varepsilon^3)$, then the contributions from events of any of the classes 2S2, S2R2, or 2R2 would be $O(\varepsilon^4)$, and therefore higher-order. Situations for which either of these order assumptions is invalid seem unlikely, but if such arise then Equation (3.52) could easily, but tediously, be modified to incorporate appropriate contributions from any or all of these three classes of system failures.

Table 3.3

List of the Minimal Cut Sets.

Definition	Represented from Equation (3.52) as
(S3)	$\left\{ \left[1 - \prod_{i=1}^3 \tilde{P}_i + \sum_{i=1}^3 \tilde{P}_i (1 - \tilde{P}_i \tilde{P}_{i''}) \tilde{P}_{i''} \right] \hat{P}_{012} \right\} H_3(T_c, T_c, T_c)$
(R3)	$H_3(T_c, T_c, T_c) \int_0^T p_{012}(\tau_{012}) d\tau_{012}$
(S2)(S2)=2(S2)	*N/A
(S2)(R2)	*N/A
(S2)(S1)	$\left\{ \sum_{i=1}^3 \tilde{P}_i (1 - \tilde{P}_i \tilde{P}_{i''}) \tilde{P}_{i''} \right\} H_3(T_c, T_c, T_c)$
(S2)(R1)	$\sum_{i=1}^3 \left\{ (1 - (\tilde{P}_i \tilde{P}_{i''})) \tilde{P}_{i''} \right\} \int_0^T p_i(\tau_i) H_3 \left(\begin{matrix} T_i = T_c, T_{i'} = \tau_i + T_c, \\ T_{i''} = \tau_i + T_c \end{matrix} \right) d\tau_i$
(R2)(R2)=2(R2)	*N/A
(R2)(S1)	$\sum_{i=1}^3 \tilde{P}_i \left\{ \int_0^T p_{i''}(\tau_{i''}) H_3(T_i = \tau_{i''} + T_c, T_{i'} = T_{i''} = T_c) d\tau_{i''} \right\}$
(R2)(R1)	$\sum_{i=1}^3 \int_0^T \int_0^T p_i(\tau_i) p_{i''}(\tau_{i''}) H_3 \left(\begin{matrix} T_i = \max\{\tau_i, \tau_{i''}\} + T_c - \tau_i, \\ T_{i'} = T_{i''} = \max\{\tau_i, \tau_{i''}\} + T_c - \tau_{i''} \end{matrix} \right) d\tau_i d\tau_{i''}$
(S1)(S1)(S1)=3(S1)	$\left\{ \prod_{i=1}^3 \tilde{P}_i \right\} H_3(T_c, T_c, T_c)$
(S1)(S1)(R1)=2(S1)(R1)	$\sum_{i=1}^3 \left\{ \tilde{P}_i \tilde{P}_{i''} \right\} \int_0^T p_i(\tau_i) H_3(T_i = T_c, T_{i'} = \tau_i + T_c, T_{i''} = \tau_i + T_c)$
(S1)(R1)(R1)=(S1)2(R1)	$\sum_{i=1}^3 \tilde{P}_i \left\{ \int_0^T \int_0^T p_{i'}(\tau_{i'}) p_{i''}(\tau_{i''}) * H_3 \left(\begin{matrix} T_i = \max\{\tau_{i'}, \tau_{i''}\} + T_c, \\ T_{i'} = \max\{\tau_{i'}, \tau_{i''}\} + T_c - \tau_{i'}, \\ T_{i''} = \max\{\tau_{i'}, \tau_{i''}\} + T_c - \tau_{i''} \end{matrix} \right) d\tau_{i'} d\tau_{i''} \right\}$
(R1)(R1)(R1)=3(R1)	$\int_0^T \int_0^T \int_0^T p_1(\tau_1) p_2(\tau_2) p_3(\tau_3) H_3 \left(\begin{matrix} \max_{j=1}^3 \{\tau_j\} + T_c - \tau_1, \\ \max_{j=1}^3 \{\tau_j\} + T_c - \tau_2, \\ \max_{j=1}^3 \{\tau_j\} + T_c - \tau_3 \end{matrix} \right) d\tau_1 d\tau_2 d\tau_3$

*See note in paragraph preceding Table 3.3

From Equation (3.52), one can simplify it assuming that all of non-recovery integral repairs and offsite recovery are assumed to be mutually exclusive as opposed to independent, which was originally presumed. It is also assumed that the failure and repair rates for each trains are the exact same as they are identical components. Therefore, Equation (3.53) can be determined which is also displayed as Equation (1.1) earlier in Chapter I.

$$\begin{aligned}
F_{\text{CSBO}}(T) &= \int_0^T \int_0^T \int_0^T [1 - G(\max\{\tau_1, \tau_2, \tau_3\} + T_c)] \prod_{j=1}^3 \left[\frac{1 - R(\max\{\tau_1, \tau_2, \tau_3\})}{1 + T_c - \tau_j; \tau_1, \tau_2, \tau_3} \right] dF(\tau_1) dF(\tau_2) dF(\tau_3) \\
&+ 3 \int_0^T \int_0^T [1 - G(\max\{\tau_{12}, \tau_3\} + T_c)] [1 - R(\max\{\tau_{12}, \tau_3\} + T_c - \tau_{12}; \tau_{12}, \tau_3)]^2 \times \\
&\left[1 - R(\max\{\tau_{12}, \tau_3\} + T_c - \tau_3; \tau_{12}, \tau_3) \right] dF_2(\tau_{12}) dF(\tau_3) + \\
&\int_0^T [1 - G(\max\{\tau_{123} + T_c)] [1 - R(T_c; \tau_{123})]^2 dF_3(\tau_{123}) = \text{Prob}\{T_{\text{CSBO}} \leq T\}, \text{ for } T \geq 0.
\end{aligned} \tag{3.53}$$

Here F_{CSBO} is the cumulative density function (cdf) for the occurrence of critical station blackout, F is the cdf for independent failure of the EDGs, F_{12} is the cdf for 2/3 common-cause failures of the first and second EDGs, F_3 is the cdf for 3/3 common-cause failure of the EDGs, G is the cdf for recovery of offsite power, R is the cdf for repair of an EDG, T is the critical time; the value under currently applicable U.S. regulations is 24 hours, T_c is the critical time, nominally about one hour, τ_i is dummy variable representing the (stochastic) time of failure of the i th EDG, τ_{12} is a dummy variable representing the (stochastic) time of common-cause failure of the first and second EDGs, τ_{123} is a dummy variable representing the (stochastic) time of failure of the i th EDG.

Equation (3.53) is the final result for the solution that is required to code the algorithm in MATLAB. It contains all pathways to critical station that does not require more than three independent failures or more than one common-cause failure. It also should not require one common-cause failure and more than one independent failure. If independent failures scale as order of epsilon or higher, than the common-cause failures will be an order of epsilon squared or higher. This will cause Equation (3.53) to contain at least all contributions to the probability of CSBO during the mission time up to order

of epsilon squared. Depending upon further details of the particular scaling employed, Equation (3.53) may contain contributions of even high order but it should not miss any contributions of order three or lower for this analysis.

Section 3.8 – Advanced Three-EDG State-Transition Differential Equations

The state-transition differential equations for Figure 3.2 are shown in Equations (3.54) through (3.61). It can be noted that the placement of equations of state 4 and 3 are switched from chronological order. This is because these states are in binary order defined earlier in this chapter. State 4 is on the same order and almost equivalent to states 1 and 2 where as state 3 is on the same order and almost equivalent to states 5 and 6.

$$\begin{aligned} \frac{dP_0}{dt} = & -\varepsilon\lambda_1 P_0 - \varepsilon\lambda_1 P_0 - \varepsilon\lambda_1 P_0 - \varepsilon^2 \lambda_2 P_0 - \varepsilon^2 \lambda_2 P_0 - \varepsilon^2 \lambda_2 P_0 - \varepsilon^3 \lambda_3 P_0 \\ & + \mu P_1 + \mu P_2 + \mu P_4 \end{aligned} \quad (3.54)$$

$$\begin{aligned} \frac{dP_1}{dt} = & \varepsilon\lambda_1 P_0 + \mu P_3 + \mu P_5 - \mu P_1 - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_1 - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_1 \\ & - (\varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_1 \end{aligned} \quad (3.55)$$

$$\begin{aligned} \frac{dP_2}{dt} = & \varepsilon\lambda_1 P_0 + \mu P_3 + \mu P_6 - \mu P_2 - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_2 - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_2 \\ & - (\varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_2 \end{aligned} \quad (3.56)$$

$$\begin{aligned} \frac{dP_4}{dt} = & \varepsilon\lambda_1 P_0 + \mu P_5 + \mu P_6 - \mu P_4 - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_4 - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_4 \\ & - (\varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_4 \end{aligned} \quad (3.57)$$

$$\begin{aligned} \frac{dP_3}{dt} = & \varepsilon^2 \lambda_2 P_0 + (\varepsilon\lambda_1 + \varepsilon^2 \lambda_3) P_1 - \mu P_3 - \mu P_3 + (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_2 \\ & - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_3 \end{aligned} \quad (3.58)$$

$$\begin{aligned} \frac{dP_5}{dt} = & \varepsilon^2 \lambda_2 P_0 + (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_0 - \mu P_5 - \mu P_5 + (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2) P_4 \\ & - (\varepsilon\lambda_1 + \varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_5 \end{aligned} \quad (3.59)$$

$$\begin{aligned} \frac{dP_6}{dt} = & \varepsilon^2 \lambda_2 P_0 + (\varepsilon \lambda_1 + \varepsilon^2 \lambda_2) P_4 - \mu P_6 - \mu P_6 + (\varepsilon \lambda_1 + \varepsilon^2 \lambda_2) P_2 \\ & - (\varepsilon \lambda_1 + \varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_6 \end{aligned} \quad (3.60)$$

$$\begin{aligned} \frac{dP_7}{dt} = & \varepsilon^3 \lambda_3 P_0 + (\varepsilon \lambda_1 + \varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_5 + (\varepsilon \lambda_1 + \varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_3 \\ & + (\varepsilon \lambda_1 + \varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_6 + (\varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_1 + (\varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_2 \\ & + (\varepsilon^2 \lambda_2 + \varepsilon^3 \lambda_3) P_4 \end{aligned} \quad (3.61)$$

Here $P_i = P_i(t)$ is the probability that the system is in state i at time t . Manipulations of these equations would allow this system of eight first order differential equations (ODEs) in eight unknowns to be written more compactly; but it is more convenient to leave it in this form for the manipulation following.

The accompanying initial conditions for this system are shown in Equations (3.61) through (3.69).

$$P_0(0) = 1 - \varepsilon \tilde{P}_1^{(1)} - \varepsilon \tilde{P}_2^{(1)} - \varepsilon \tilde{P}_4^{(1)} - \varepsilon^2 \tilde{P}_3^{(2)} - \varepsilon^2 \tilde{P}_5^{(2)} - \varepsilon^2 \tilde{P}_6^{(2)} - \varepsilon^3 \tilde{P}_7^{(3)} \quad (3.62)$$

$$P_1(0) = \varepsilon \tilde{P}_1^{(1)} \quad (3.63)$$

$$P_2(0) = \varepsilon \tilde{P}_2^{(1)} \quad (3.64)$$

$$P_4(0) = \varepsilon \tilde{P}_4^{(1)} \quad (3.65)$$

$$P_3(0) = \varepsilon^2 \tilde{P}_3^{(2)} \quad (3.66)$$

$$P_5(0) = \varepsilon^2 \tilde{P}_5^{(2)} \quad (3.67)$$

$$P_6(0) = \varepsilon^2 \tilde{P}_6^{(2)} \quad (3.68)$$

$$P_7(0) = \varepsilon^3 \tilde{P}_7^{(3)} \quad (3.69)$$

The quantities $\tilde{P}_1^{(1)}, \tilde{P}_2^{(1)}, \tilde{P}_4^{(1)}, \tilde{P}_3^{(2)}, \tilde{P}_5^{(2)}, \tilde{P}_6^{(2)}$ and $\tilde{P}_7^{(3)}$ are to be regarded merely as known constants. The asymptotic solution for the Three-EDG case is left to future work to develop as well as check against the current non-recovery integral currently used within the MATLAB code. It is also left to see if the epsilon squared or epsilon cubed is a better representation for the third order parameter used within the Markov model.

CHAPTER IV APPLICATIONS

This chapter will serve to the applications that the code can compute. It will talk about special cases that cause deviations in expected results, which will be discussed along with the use of different equations to determine results. The differences will likely be caused by how many safety trains are used, what type of repair is executed, and what recovery methods are implemented, what time in the accident repairs begin, and how much manpower is available for the repairs. Common-cause failure implementation will also be discussed.

The normal application of the code is to generate a value for the probability of critical station blackout; however, there are many factors that can affect that probability. Section 4.1 discusses Types of Recovery along with the various forms of repair are represented in Subsections 4.1.a-4.1.d. The deviation in the number of emergency trains, mission and critical times, and common-cause inclusions are discussed in Sections 4.2, 4.3, and 4.4, respectively. Section 4.5 discusses the use of different distribution types for the cdf. Section 4.6 talks about repair and how it could better be handled in future work which would later cause a deviation from results. It can also be noted that the specific libraries used in MATLAB may also have a slight effect on the overall probability of CSBO which is explained in Section 4.7.

The actual results from the code suite are shown in Chapter V, but the idea of this chapter was to serve as an introduction to those various small changes that can occur in the code through the use of the input file.

Section 4.1 – Types of Recovery

This section will outline the different types of recovery that are capable inside the STP-TAMIL code. These are selectable via the input file. Further details are discussed in Chapter II, the user manual.

Subsection 4.1.a – No Credit (1)

The first type of recovery is not having any recovery at all. This assumes that offsite power will not come back within the mission time. This could make sense if the grid has suffered sufficient damage, which was the case in Fukushima Dai-ichi [20]. The case assumes that the emergency design generators cannot be repaired within the mission time. This was also the case for that disaster as the EDGs flooded out and swept out to sea [20]; therefore, there was no chance for any type of normal recovery. This is the most conservative case used in the model, which is the most common justification for its use in the industry.

Subsection 4.1.b – Offsite Only Credit (2)

The second type of recovery assumes only offsite power can be restored. That is, there is a probability that offsite power will come back within the mission time. It also assumes that the emergency design generators cannot be repaired within the mission time. This could make sense if the EDGs have suffered sufficient damage or have a fuel problem. This is a likely case if it's a normal LOOP event that does not last an extended period of time, but the EDGs have failed for some reason, again the most common justification for the assumption is conservatism.

Subsection 4.1.c – Offsite Credit and Onsite Only After Full Failure (3)

The third type of assumption regarding recovery and repair is that both offsite and onsite power can be restored, but onsite repairs are not initiated until all power has failed (SBO has occurred). This assumes that there is a probability that offsite power will come back within the mission time. It also assumes that the emergency design generators have a probability of repair within the mission time, but only after all of the EDGs have failed. This scenario would indicate that the repairs did not start until all failed or there existed a common cause triplet event. This method therefore also adds in additional conservatism.

Subsection 4.1.d – Full Credit of Offsite and Onsite Failure (4)

The fourth type of recovery is assuming both offsite and onsite power can be recovered. This assumes that there is a probability that offsite power will come back within the mission time. It also assumes that the emergency design generators have a probability to be repaired within the mission time. This takes into account repair as soon as something fails. This is the most generous case in term of likelihood of core damage. It also is the most realistic case as repairs would generally start right after initial failure or as soon as possible.

Section 4.2 – Number of Trains

This section discusses the impact that different number of trains can have on the result of the problem. For STP, three independent EDG trains are used for each of the two units. Many other plants only have two trains of EDGs, so they cannot afford to lose as many generators as STP and still be able to run their safety systems. The input deck for the program allows the user to enter any number of trains (one through three) for the plant. The idea is that one could see the benefit of adding a third EDG, if a plant only has two, based solely on probability of critical station blackout. South Texas Project could also use the program to see what the probability of having only two trains if they ever had one under maintenance or if they were considering a change in their design certification document. This code could also be used to determine the probability of CSBO if an independent train is operating for a short period while having the two other diesels brought offline.

Currently, the code only fully supports three trains. However, it can work with two or one emergency trains but it is not fully verified, unlike the three train version. Also, the code version cannot handle any more than three trains as this would introduce a common-cause quadruplet term among others.

Section 4.3 – Mission and Critical Times

This section discusses that using different mission and critical times can change the results of the problem. Normally, most PRA estimates have been using a 24-hour mission time. Yet, as discussed in Chapter I, the post-Fukushima-Daiichi nuclear industry may change mission time to a longer period of time. This would mean that there would be a larger failure probability because of the increased time to fail of a particular system.

The capability of this program can also be used if there is a split mission time. For example, one could use a mission time of 24 hours with certain values and then on a different data row use the new values with the extended mission time. However, it should be noted that if there is a split mission time, then the fail to load and run probability will not work properly for the second mission time. This is due to the fact that the parameters, load and run, come from starting and operating for the first hour. After that hour has passed, the probability of a failure would then be zero as it would be impossible to have it fail in the first hour. There are still a few things to consider. If the mission time split is less than 1 hour then one would need to split the mission time again to have it last for the rest of the hour at a decreased probability of failure to load and run. The other split would then be after the first hour with a zero probability of failure to load and run. This might occur in a real world scenario if one had to prove that the plant did not enter a certain level risk finding during an event and one knew the different mission times led to different probabilities. A split mission time could also occur if you exceed the allotted mission time and assume that the failure probabilities for devices have now changed after day one, assuming a twenty four hour mission time.

One can also change the critical time, which is normally one hour. One can try a various amounts of scenarios where the plant may be without emergency systems following a LOOP-IE for a certain amount of time. This could also be the case if the plant had a feature that let critical time be increased. This could be due to passive cooling or from capabilities that arise from the fuel itself, like a pebble-bed reactor. It is hypothesized that a pebble-bed reactor will not melt down in the absence of cooling [21]. This would mean that no core damage would ever occur, or that it could happen within

some extended critical time. That would decrease the probability of critical station blackout.

Section 4.4 – CC Triplet, CC Doublet, and Independent Failures

This section describes how common-cause and independent failures can change the result of the program. The program already breaks up the total probability of critical station blackout into three parameters, corresponding to common-cause triplet, common-cause doublet, and independent failures. One can also determine which probabilities that the code should compute and mix and match which ones can be added together from the output file. This also gives a representation of how the total failure is broken up and divided amongst the three types of failures. However, the program does not do only CC doublet failures with three trains. This is because there is the third generator would not, most realistically, cause full station blackout without independent failures considered. This assumption does not take into account the repair of one of the two units, followed by a CC doublet failure of the last two EDG's in operation.

Section 4.5 – Distribution Types

Section 2.5 describes how to use different distribution types within the software along with how to create an m-file corresponding to a new distribution that is not listed in the MATLAB generic libraries.

These could change results slightly as each distribution is different; however, results should stay relatively close to each other, provided the distribution parameters are determined from the same (or similar) data. This relative closeness is due to the distributions using mostly the same input values and just manipulating them a little differently to lead to a more “accurate” answer in terms of that distribution.

Section 4.6 – Repair

The type of repair is also a factor in changing the result. If repair does not start until after all the trains have failed, then the probability of critical station blackout would be higher because there would be less time to repair the EDGs. Also, the repair rates themselves can be increased or decreased which will then inversely affect the probability of critical station blackout.

The test of the current code, in Chapter V, uses an average (aggregate) repair rate. A better approach would be to divide basic events into repairable and nonrepairable, and employ an aggregate repair rate only for the former. This would involve data collection and analysis (for the division, and the determination of a suitable repair distribution for repairable basic events). It also would require extension of the current coding. For example, there now would be four classes of SBOs associated to three independent failures:

- i) all failures repairable,
- ii) two repairable and one nonrepairable,
- iii) one repairable and two nonrepairable, and
- iv) all three nonrepairable.

Repair rates could also be changed for each type of failure, common-cause shock events versus independent type failure events. This could have an advantage in getting a more realistic repair for common-cause but would not affect independent, as that is how it is currently computed. The repair rate for common-cause could be much simpler than the repair of each one individually on average. An example would be a common-cause triplet caused by lack of fuel. This is easily repaired for all three by adding fuel to the diesel tank which would be vastly different than a mechanical breakdown of each EDG independently.

Another factor is number of persons performing repairs. If more people are repairing failures, then repair rate may be higher/faster. Other factors affecting repair time, may be type of damage done to cause the initial problem and continuing damage from outside sources (aftershocks, flooding, etc.). Repair rates that are more dynamic could be more adequately addressed in future work.

Section 4.7 – Equations Types

The certain equations used or not used may change the result. The advanced three-EDG model seen in Chapter III may lead to the use of a single integral equation which could change the values slightly. Also the use of the QUAD, DBLQUAD and TRIPLEQUAD MATLAB libraries against other known integration MATLAB libraries may have a slight, almost negligible effect. Also, as seen in Chapter V, the other methods employed will get different results which are used as a more simple form or parts of the non-recovery integral to better verify the program.

CHAPTER V

RESULTS AND BENCHMARKING

This chapter will outline the results and there will be a discussion of how these results are realistic, meaningful, and useable. This chapter will involve a benchmarking section to provide verification of results.

The code suite is broken up into many different suites. The “STPQUAD2.m” code suite, or STPQUAD2, looks at the model with only two EDGs failing independently. It was useful to first benchmark and then verify the results from this program before the code was advanced to three EDGs. The verification of “STPQUAD2.m” appears in Section 5.1. “STPQUAD3.m” looked at the three-EDG case, for only independent failure considered. It is important to note that before adding full common-cause effects into the program, it was verified from expected results. The verification for “STPQUAD3.m” is shown in Section 5.2. Subsections in both Sections 5.1 and 5.2 discuss various types of recovery and how the probability of critical station blackout is affected numerically. These subsections also contain a comparison of the different methods employed. The common-cause contribution, three EDG case, as well as the finalized code is verified in Section 5.3. This final program is entitled “STPQUAD3CCF.m” and was used as the basis for development of STP-TAMIL. STP-TAMIL version 1.0 is the “STPQUAD3CCF.m” code suite. It is verified using two different methods. The first, shown in Subsections 5.3.a through 5.3.c, is by STP data, shown in Appendix C. This assumes values based off of the STP PRA system notebook and is kept confidential other than the reduction in CSBO probability. Subsections 5.3.d through 5.3.f discusses how the code is verified using the industry average data in combination with the alpha factor method. This method is also discussed within the presented text. Section 5.4 then summarizes the data, compares the two verifications, and discusses results.

Section 5.1 – Verification of STPQUAD2

This section describes the verification of the code “STPQUAD2.m”. The code itself as described in Appendix A.1. The results are shown in Table 5.1 and the process is described within the rest of this section. As a broad overview, this version of the code takes into account the use of only two independent electrical trains (or two EDGs) in the model as well as assuming only independent type failures. Therefore, this version ignores common-cause failures.

Case G, the last row of Table 5.1, contains the results from the full two EDG version of the code (“STPQUAD2.m”). The rows containing Case A, Case C, and Case E show results from slightly modified versions of STPQUAD2 that carry out the same numerical integration of a two-dimensional recovery integral. However, these cases use a slightly modified integrand that either are computed en route to the full integrand (Case A and Case C) or can be computed (Case E) from the last-line modification of the subroutine used to compute the full integrand. This last-line modification refers to commenting out a line of code for a different type of repair within the specific m-file. These types are covered in Chapter II and Chapter IV.

Table 5.1

Results for STPQUAD2.

Case	Method	<i>Probability of Critical Station Blackout (P_{CSBO})</i>
Case A	No credit for recovery (offsite or onsite), from STPQUAD2 (default absolute error tolerance = 1.0e-5, default relative error tolerance per MATLAB documentation for the subroutine quad2d)	1.074439162179359e-3
Case B	No credit for recovery (offsite or onsite), per hand calculation (see below)	1.074439162179330e-3
Case C	Credit for offsite recovery only, computed dynamically via STPQUAD2 double integration (default error tolerances), but with partial integrand for nonrecovery integral that is computed en route to full integrand in subroutine INTEGRAND	2.206757664289611e-4
Case D	Credit for offsite recovery only, from separate code, computed numerically (and “dynamically”) as a single integral (per “dynamic” method of Lloyd [13])	2.195427574125606e-4
Case E	Computed dynamically via STPQUAD2 double integration, with integrand for nonrecovery integral that is fully dynamic for offsite recovery but incorporates onsite repair only after full failure of onsite power, per integrand computed via slight modification at last part of INTEGRAND code	1.867979788131047e-4
Case F	Dynamic credit for offsite recovery, plus credit for onsite repair only after full failure of onsite power, computed numerically as a single integral (per “dynamic” method of Lloyd [13])	1.858389319818259e-4
Case G	Full credit for recovery of offsite and onsite power, double integral evaluated per STPQUAD2	1.70e-4

It can be noted that Case E’s method refers to the last part of the INTEGRAND code. This refers to the type of non-recovery involved in the code itself. In this code the

part has a very different non-recovery factors (H) which are determined different ways based on the type of recovery and when repair starts. These lines are uncommented or commented when certain types of recovery happen.

The reason for including these partial results (Cases A, C and E) is that they can be obtained by independent methodologies that have already been developed. Therefore, these partial cases make a good comparison between the case created by the code and the cases shown by the prior method in a hand calculation. The calculations are compared and the new result is shown directly underneath each associated case in Table 5.1. These prior method hand calculations are shown as Cases B, D, and F. Therefore, Case A is compared to the Case B, Case C to Case D, and Case E to Case F. In Table 5.1 it can be noted that the underlined digits are the first three digits that do not agree with the value of the digits immediately above when comparing these Cases that were just outlined. Thus, a good comparison between each new Case to the prior method case results should give greater confidence in the version of STPQUAD2 ability to create results. As obtaining the prior method case results are a majority of the actual write-up of the program itself in STPQUAD2, this will verify if the code is written correctly in the vast majority of STPQUAD2.

The base problem used here to verify the scheme behind the code is from previous work developed for ICONE17 [22]. In more detail, an emergency power system consisting of two EDGs is contemplated, and the various functions appearing in the integrand of the non-recovery integral are shown in Table 5.2.

Table 5.2

Data Employed in [22], as Used Here for STPQUAD2 Verification.

Name	Description	Value
G	cdf for offsite recovery	Exponentially distributed
T_{rec}	Mean time to Recovery	4 hours
$F_i(0)$	Probability of Failure of i th EDG to start	0.0132
f_i	pdf for failure of i th EDG to run, conditional upon starting	Exponentially distributed, with mean time to failure of about $1/8.35e-4$ hours or about 1200 hours
R_i	cdf for repair of i th EDG	Exponentially distributed
T_{rep}	Mean time to Repair	12 hours

The results, for the data of Table 5.2, compared are values obtained by various methodologies, show in Table 5.1, for the probability of critical station blackout that will occur within a mission time of 24 hours subsequent to a LOOP event. For this initial verification problem no differentiation is made between the various sources of types of loss of offsite power, as ultimately will be necessary. Some of these differences could include the LOOP being caused by weather, the grid, or the switchyard failure [23].

The results from the various methods not employing STPQUAD2 (i.e. Cases B, D, and F) are shown in subsections 5.1.a, 5.1.b, and 5.1.c.

Subsection 5.1.a – No Credit for Recovery (Offsite or Onsite) for STPQUAD2

This subsection will describe the methodology used to receive results for Case B of the STPQUAD 2 verification. This method involves not using any credit for any time of recovery (whether onsite or offsite). This is the most conservative approach to the problem that is defined by critical station blackout. In the case, $P_{csbo} = P^2$, where P is the probability that an EDF will fail during the mission time. With the mission time and failures probabilities given in Table 5.2 one can create Equation (5.1).

$$\begin{aligned}
P &= F_i(0) + \int_0^T f_i(t) dt = F_i(0) + \int_0^T \frac{e^{-t/\tau}}{\tau} dt = F_i(0) + \left\{ 1 - e^{-T/\tau} \right\} \\
&= 0.0132 + \left\{ 1 - e^{\left(-24 * 8.35 * 10^{-4} \right)} \right\} = 0.0330405338560285
\end{aligned} \tag{5.1}$$

In this equation, (5.1), T is the mission time of 24 hours. Therefore $P_{CSBO} = P^2 = 0.001091676877914 = 1.0917e-3$ to five figures of accuracy. The value initially computed by double numerical integration, Case A, shows a discrepancy, relative to this “exact result”, of $0.018e-3$ or about $2e-5$. This is certainly conceivably due to the error in the numerical integration of the two-dimensional nonrecovery integral. This error appears to be controlled by the default value of $1.0e-5$ for the absolute error tolerance MATLAB code QUAD2D that was used to execute this numerical integration. If so, then confirmation should be found by varying this absolute error tolerance. However, this value of the nonrecovery integral appears to be invariant, to 16 significant figures and at $1.074439162179359e-3$, to the value used for this absolute error tolerance. The value had an exact value of $1.0e-15$ for absolute error tolerance of 1.0. However, this value was not expected based on the result from the MATLAB result shown in Case A of Table 5.1, so the equation was changed to diagnose the error. This meant changing the terms from independent to mutually exclusive. The new equation can be seen in Equation (5.2).

$$\begin{aligned}
P &= F_i(0) + (1 - F_i(0)) \int_0^T f_i(t) dt = F_i(0) + (1 - F_i(0)) \int_0^T \frac{e^{-t/\tau}}{\tau} dt \\
&= F_i(0) + (1 - F_i(0)) \left\{ 1 - e^{-T/\tau} \right\} = 0.0132 + 0.9868 * \left\{ 1 - e^{\left(-24 * 8.35 * 10^{-4} \right)} \right\} \\
&= 0.0327786388091289
\end{aligned} \tag{5.2}$$

The corresponding value of $P_{CSBO} = P^2$ is then equal to $0.00107074439162179330 = 1.0707e-3$ as shown in the Case B of Table 5.1 and calculated in Equation (5.2). This

value differs from that of Case A, the numerical integration method through MATLAB, by only about $3e-13$. This exact result has a very high degree of agreement with that from numerical quadrature.

However, it is still important to show why the value is invariant to the absolute error tolerance. From on-line documentation from MATLAB for `quad2` [16], the error tolerance in `quad2` is $\max\{\text{AbsTol}, \text{RelTol} * |Q|\}$ where Q is the value of the integral, `AbsTol` is the absolute error tolerance, and `RelTol` is the relative error tolerance. Also the `RelTol` minimum value, and default for MATLAB, is $100 * \text{eps}(\text{class}(Q))$. In the usage of `quad2d` in `STPQUAD2`, Q is of class “single precision” and $100 * (2^{-23}) * 1.0e-3$ which is about equal to $1.0e-8$. This is several orders of magnitudes greater than the actual error attained at the default error tolerances; therefore, no further changes of trying to reduce the error should be expected from decreasing the absolute error tolerance further. The result did not change even when `AbsTol` increased by several orders of magnitude. As determined later, it turns out that the MATLAB code starts to ignore the error tolerance because the integral code starts coming up with values based on a certain result and it believe this result to be correct so it just repeats those the entire time. The code is basically taking an integration of a constant value so that a change in error tolerance will have no effect.

Subsection 5.1.b – Dynamic Credit for Offsite Recovery (Only) for STPQUAD2

This subsection will describe the methodology explored to receive results for Case D of the `STPQUAD 2` verification. It uses the method of taking dynamic credit for offsite recovery, not onsite repair, using a single-integral method outlined by Lloyd [13] and described on the top of page 8 of Rodgers et al. [1]. Equation (5.3) shows this single-integral method equation relating to the specific problem at hand.

$$\begin{aligned}
F_{\text{CSBO}}(T) &= \prod_{i=1}^n \hat{R}_i \left\{ \left[1 - G(T_c) \right] \prod_{i=1}^n F_i(0) + \int_0^T \left[1 - G(T' + T_c) \right] \times \right. \\
&\quad \left. \sum_{i=1}^n f_i(T') \int_0^{T'} \dots \int_0^{T'} \left[\prod_{\substack{j=1, \\ j \neq i}}^n f_j(\tau'_j) \right] d\tau'_1 \dots d\tau'_{i-1} d\tau'_{i+1} \dots d\tau'_n dT' \right\} \\
&= \prod_{i=1}^n \hat{R}_i \left\{ \left[1 - G(T_c) \right] \prod_{i=1}^n F_i(0) \right. \\
&\quad \left. + \int_0^T \left[1 - G(T' + T_c) \right] \sum_{i=1}^n \left[f_i(T') \prod_{\substack{j=1, \\ j \neq i}}^n F_j(T') \right] dT' \right\} \\
&= \text{Prob}\{T_{\text{CSBO}} \leq T\}.
\end{aligned} \tag{5.3}$$

Note that the variables used in Equation (5.3) are summarized in Table 3.2 of this Thesis. It also important to note that the following corrections of the equation listed on the top of page 8 of Rodgers et al. [1] are used in Equation (5.3). These corrections include: i) the replacing of $\hat{F}_i(0)$ due to an error of the use of “^” in the wrong place with $F_i(0)$, and ii) the insertion of a sum over I with the factor $f_i(T')$ inclusions in line 3. Also, $\hat{R}_i=1$ is the non recovery factor for onsite repair that is appropriate to the current conservative approximation of no credit for onsite recovery. By contrast, this is the best estimate approach to taking credit for offsite recovery. Computationally, this estimate is shown in Equation (5.4).

$$\begin{aligned}
F_{CSBO}(T) &= \text{Prob}\{T_{CSBO} \leq T\} \\
&= \prod_{i=1}^n \hat{R}_i \left\{ \left[1 - G(T_c) \right] \prod_{i=1}^n F_i(0) + \int_0^T \left[1 - G(T' + T_c) \right]^* \right. \\
&\quad \left. \sum_{i=1}^n f_i(T') \int_0^{T'} \dots \int_0^{T'} \left[\prod_{\substack{j=1, \\ j \neq i}}^n f_j(\tau'_j) \right]^* \right. \\
&\quad \left. d\tau'_i \dots d\tau'_{i-1} d\tau'_{i+1} \dots d\tau'_n dT' \right\} \\
&= 1^2 \left[e^{-T_c/12} \right] (0.0132)^2 \\
&\quad + 2 \int_0^T e^{-(T'+T_c)/12} (1-0.0132)(8.35 \times 10^{-4}) e^{-(8.35 \times 10^{-4})T'_x} \\
&\quad \left\{ 0.0132 + (1-0.0132) \left[1 - e^{-(8.35 \times 10^{-4})T'} \right] \right\} dT'
\end{aligned} \tag{5.4}$$

A MATLAB code (“Lloyd.m”) was written to evaluate Equation (5.4), with the integration carried out numerically using the existing (one-dimensional numerical integration) MATLAB QUAD, with the integrand evaluated by means of a second written MATLAB code “INTEGRAND1.m”. The result, with the integration effected at the default error tolerance for QUAD (absolute error tolerance of 1.0e-6), as shown in Case D of Table 5.1.

The difference between the two computer values of Pcsbo in the Case C and Case D of Table 5.1 is about 1.0e-6. This is smaller than the absolute error tolerance of 1.0e-5 used for in STPQUAD2 for the numerical integration that resulted in the value of Case B. This tolerance is the same as the absolute error tolerance used in “Lloyd” for the numerical integration giving the value in Case C. If the difference were due to numerical integration, one would therefore expect that the value from STPQUAD2 is less accurate, but that the results from the two codes would tend to agree to increasingly more digits as the error tolerances decrease. This behavior is indeed observed in Table 5.3, which

confirms the hypothesis that the difference is entirely due to the inevitable error arising in the numerical integrations as opposed to a mathematical or programming error.

Table 5.3

PCsBO Values While Varying Absolute Error Tolerances with Case C vs Case D (STPQUAD2).

<i>Absolute Error Tolerance</i>	STPQUAD2 <i>PCsBO</i>	Lloyd <i>PCsBO</i>
1.0e-5	2.206757664289611e-4	2.195427574125606e-4
1.0e-6	2.207338404320751e-4	2.195427574125606e-4
1.0e-7	2.196012454513116e-4	2.195427574125606e-4
1.0e-8	2.195441635112097e-4	2.195427574125606e-4
1.0e-9	2.195427744571024e-4	2.195424291367534e-4
1.0e-10	2.195426380224189e-4	2.195424186377920e-4
1.0e-11	2.195426264491898e-4	2.195424213673031e-4
1.0e-12	2.195426250034820e-4	2.195424212590672e-4
1.0e-13	2.195426249851043e-4*	2.195424212811546e-4
1.0e-14	2.195426249851043e-4#	2.195424212816738e-4
1.0e-15	-----	2.195424212816448e-4
1.0e-16	-----	2.195424212816491e-4
1.0e-17	-----	2.195424212816491e-4
1.0e-18	-----	2.195424212816491e-4
1.0e-20	-----	2.195424212816492e-4
1.0e-22	-----	2.195424212816492e-4
1.0e-23	-----	2.195424213102547e-4
1.0e-25	-----	2.195429103006130e-4

* - MATLAB reported: “Warning: Reached the maximum number of function evaluations (2000). The results passes the global error test.” Obtaining this result required minutes of computational time.

- MATLAB reported: “Warning: Reached the maximum number of function evaluations (2000). The results fails the global error test.” Obtaining this result required minutes of computational time.

The results in Table 5.3 contain much more detail than needed to reach a satisfactory conclusion to show that STPQUAD2 can be verified with Lloyd, which was the immediate objective. However, some of the detail could be useful subsequently, for

presently unforeseen purposes and should be left to future work. For example, the results could be used to verify subsequent significantly revised versions of STPQUAD2. The underlined digits are the first two digits that do not agree with the value immediately above that cell in Table 5.3. Reduction of the absolute error tolerance was terminated when the values reported by “Lloyd” began to display the classic mathematical symptoms of round off error.

**Subsection 5.1.c- Dynamic Offsite Recovery, Onsite Repair Post Failure
(STPQUAD2)**

The subsection will describe the methodology explored to receive results for Case F of the STPQUAD 2 verification. It uses the method of taking dynamic credit for offsite recovery, using a single-integral method outlined by Lloyd [13] and described on the top of page 8 of Rodgers et al. [1]. It also credits repair only after a full failure of onsite power. In this case the probability of critical station black out is given by Equation (5.4), but now with the repair term defined in Equation (5.5).

$$\hat{R}_i = 1 - R_i(T_c) = e^{-T_c/12} = e^{-1/12} = 0.920044146292230 \quad (5.5)$$

The corresponding result shown in Equation (5.5), as obtained by Lloyd with the default error tolerance appears in Case F of Table 5.1.

The difference, like in the previous subsection, between the computer value for the probability of critical station blackout from STPQUAD2, Case E, and that from Lloyd, Case F, is about 1.0e-6. The hypothesis is that the difference is still due to the inevitable integration error, primarily from STPQUAD2. The results in Table 5.4, which is an abbreviated version of Table 5.3, are consistent with that hypothesis. The underlined digits are the first two digits that do not agree with the value immediately above that cell in Table 5.4.

Table 5.4

PCSBO Values While Varying Absolute Error Tolerances with Case E vs Case F (STPQUAD2).

<i>Absolute Error Tolerance</i>	<i>STPQUAD2 PCSBO</i>	<i>Lloyd PCSBO</i>
1.0e-5	1.867979788131047e-4	1.858389319818259e-4
1.0e-6	1.868471619906726e-4	1.858389319818259e-4
1.0e-8	1.858404483442557e-4	1.858389319818259e-4
1.0e-10	1.858388331469148e-4	1.858386452151755e-4
1.0e-12	1.858388199254640e-4	1.858386474340370e-4
1.0e-13	1.858388198844045e-4 [*]	1.858386474527336e-4
1.0e-14	1.858388198800244e-4 [#]	1.858386474531731e-4
1.0e-15	1.858388198802952e-4 [@]	1.858386474531485e-4
1.0e-16	-----	1.858386474531522e-4

^{*}MATLAB reported: “Warning: Reached the maximum number of function evaluations (2000). The results passes the global error test.” Obtaining this result required minutes of computational time.

[#]MATLAB reported: “Warning: Reached the maximum number of function evaluations (2000). The results fails the global error test.” Obtaining this result required minutes of computational time.

[@]MatLab reported: “Warning: Reached the maximum number of function evaluations (2000). The result fails the global error test.” Obtaining this result required minutes of computational time.

Subsection 5.1.d – Comparison of Methods (STPQUAD2)

This section has focused on verification of the MATLAB research code STPQUAD2, which was indeed the primary purpose. However, the results also provide a basis for comparing the relative accuracy of various methods that were developed for estimating probability of critical station blackout. The results are relevant to this objective and displayed in Table 5.5.

Table 5.5

Comparison of Estimates of P_{CSBO} from Various Methods (STPQUAD2).

Method	P_{CSBO}
No credit for recovery (from Table 5.1)	1.07e-3
Credit for recovery (onsite and offsite) only following SBO (i.e., Read-Fleming [14] variant of Lloyd-Anoba method [11]), computed using code Lloyd, with default error tolerance for integration	7.08e-4
Dynamic offsite recovery, credit for onsite recovery only following SBO (i.e., approach of Lloyd [13]), from Table 5.1	1.86e-4
Best estimate, with fully dynamic treatment of both onsite and offsite recovery (from Table 5.1)	1.70e-4

In this case, the big improvement in the estimate of probability of critical station blackout seemed to arise from taking credit for offsite recovery, even after SBO occurs. That is from Case B to Case C. It is perhaps a little surprising that dynamic onsite recovery, or repair of one EDG while the other is operating, contributes only about a 10% further reduction. It remains to be seen how general this is and could be something to look into for future work.

Section 5.2 – Verification of STPQUAD3 (Without Common-Cause)

This section describes verification of the code “STPQUAD3.m”. The code itself can be seen in Appendix A.2. The results are shown in Table 5.6 and the process is described within the rest of this section.

Table 5.6

Results for STPQUAD3.

Case	Method	Probability of Critical Station Blackout (P_{CSBO})
Case A	No credit for recovery (offsite or onsite), from STPQUAD3 (default absolute error tolerance = $1.0e-5$, default relative error tolerance per MATLAB documentation for the subroutine quad2d).	$4.068467294820226e-5$
Case B	No credit for recovery (offsite or onsite), per hand calculation (see below)	$4.068467294820230e-5$
Case C	Credit for offsite recovery only, computed dynamically via STPQUAD3 triple integration (default error tolerances), but with partial integrand for nonrecovery integral that is computed en route to full integrand in subroutine INTEGRANDv1pt0	$3.517725266608635e-6^*$ ($3.447045307250793e-6$)#
Case D	Credit for offsite recovery only, from separate code (Lloyd3), computed numerically (and “dynamically”) as a single integral (per “dynamic” method of Lloyd [13])	$2.760537818226182e-6^*$ ($3.435769452659052e-6$)#
Case E	Computed dynamically via STPQUAD3 triple integration, with integrand for nonrecovery integral that is fully dynamic for offsite recovery but incorporates onsite repair only after full failure of onsite power, per integrand computed via slight modification at last part of INTEGRANDv1pt0 code	$2.367947043525971e-6^*$ ($2.320369049126797e-6$)
Case F	Dynamic credit for offsite recovery, plus credit for onsite repair only after full failure of onsite power, computed numerically (code Lloyd3) as a single integral (per “dynamic” method of Lloyd [13])	$1.829911359629647e-6@$ ($2.319879832241328e-6$)\$
Case G	Full credit for recovery of offsite and onsite power, triple integral evaluated per STPQUAD3	$2.040433328944350e-6^*$ ($1.967858943170821e-6$)#

* - Default error tolerance for numerical integrations, $TOL=1.0e-6$.# - More stringent error tolerance for numerical integration, $TOL=1.0e-10$.@ - Default error tolerance for numerical integrations, $TOL=1.0e-6$, QUADL (rather than QUAD) as numerical integrator\$ - More stringent error tolerance for numerical integrations, $TOL=1.0e-10$, QUADL (rather than QUAD) as numerical integrator

Case G, the last row of Table 5.6, contains the results from the full three EDG version of the code (“STPQUAD3.m”). The rows containing Case A, Case C, and Case E show results from slightly modified versions of STPQUAD3 that carry out the same numerical integration of a two-dimensional recovery integral. However, these cases use a slightly modified integrand that either are computed en route to the full integrand (Case A and Case C) or can be computed (Case E) from the last-line modification of the subroutine used to compute the full integrand. Where two values are given, the topmost result was obtained using the default error tolerance of $TOL=1.0e-6$, whereas the second (parenthetical) value was obtained using a more stringent tolerance of $1.0e-10$. The rest of this section will contain a further discussion of the impact of the choice of error tolerance.

It can be noted that Case E’s method refers to the last part of the INTEGRANDv1pt0code. This refers to the type of non-recovery involved in the code itself. In this code the part has a very different non-recovery factors (H) which are determined different ways based on the type of recovery and when repair starts. These lines are uncommented or commented when certain types of recovery or repair happen.

The idea behind including these partial results (Cases A, C and E), as in the preceding section, is that they can be obtained by independent methodologies that have already been developed. Therefore, they make a good comparison between the case created by the code and the cases shown by the prior method in a hand calculation. These values are compared and both are displayed with one on top of the other in Table 5.6. These prior method hand calculations are shown as Cases B, D, and F. Therefore, Case A should be compared to the Case B, Case C to Case D, and Case E to Case F. In Table 5.6 it can be noted that the underlined digits are the first two digits that do not agree with the value of the digits immediately above when comparing the cases that were just outlined. Thus, a good comparison between each new Case to the prior method case results should give greater confidence in the version of STPQUAD3 ability to create results. As obtaining the prior method case results are a majority of the actual write-up of the program itself in STPQUAD3, this will also verify if the code is written correctly in the vast majority of STPQUAD3.

The base problem used here to verify the scheme behind the code is from previous work developed for ICONE17 [22]. In more detail, an emergency power system consisting of three EDGs is contemplated, and the various functions appearing in the integrand of the non-recovery integral are shown in Table 5.7.

Table 5.7

Data Employed in [22], as Used Here for STPQUAD3 Verification.

Name	Description	Value
G	cdf for offsite recovery	Lognormal Cumulative Distribution
$G(t, \mu, \sigma)$	cdf for offsite recovery	$= \Phi \left(\frac{\ln(t) - \mu}{2\sigma} \right)$
Φ	Error function	
t	Time in hours	
μ	Mean time to Recovery	0.300 (Table 4-1 from [6])
σ	Sigma Deviation	1.064 (Table 4-1 from [6])
$F_i(0)$	Probability of Failure of i th EDG to start	1.4e-2 (sum of mean values of FTS (failure to start) and UA (unavailability) per Table 4-2 from [6])
f_i	pdf for failure of i th EDG to run, conditional upon starting	Exponentially distributed, with λ_1 and λ_2
λ_1	Failure Rate during first hours (mean value of FTLR, failure to load and run, failure rate)	2.5e-3 per hour (Table 4-2 of [6])
λ_2	Failure Rate during hour thereafter the first hour (mean value of FTLR, failure to load and run, failure rate)	8.0e-4 per hour (Table 4-2 of [6])
R_i	cdf for repair of i th EDG	Weibull cdf
α	Alpha function	0.739
β	Beta Function	15.50 hours

The results from Cases A, C, E and F from Table 5.6 were obtained using the default error tolerance in the MATLAB triple integration code TRIPLEQUAD. $1.0e-6$ is set as the default error tolerance. MATLAB documentation does not specify whether that value is an absolute or relative error tolerance or rather some combination of the two.

The results, from Table 5.7, compared are values obtained by various methodologies, shown in Table 5.6, for the probability of critical station blackout that will occur within a mission time of 24 hours subsequent to a LOOP event. For this second verification problem again no differentiation is made between the various sources of loss of offsite power [23], as ultimately will be necessary for the full development of the STPQUAD3 code suite. The distribution of offsite recovery times differs depending upon the nature of the loss of offsite power whether the cause is weather related, grid related, or related to some other factor that would shut down power that supplies the station. (Chapter IV of [6]).

The results from the various methods not employing STPQUAD3 (i.e. Cases B, D, and F) are shown in subsections 5.2.a, 5.2.b, and 5.2.c.

Subsection 5.2.a – No Credit for Recovery (Offsite or Onsite) for STPQUAD3

This subsection will describe the methodology explored to receive results for Case B of the STPQUAD 2 verification. This method involves not using any credit for any time of recovery (onsite or offsite). This is the most conservative approach to the problem of critical station black out. In this case, $P_{CSBO} = P^3$. P is the probability that an EDG will fail during the mission time. With the mission time and failure probabilities given in Table 5.7, one can create Equation (5.6). It can be noted that in Equation (5.6), T is the mission time denoted as 24 hours where the rate of failure for the first hour and the time thereafter are different and need to be split up in the equation.

$$\begin{aligned}
P &= F_i(0) + (1 - F_i(0)) \int_0^T f_i(t) dt \\
&= F_i(0) + (1 - F_i(0)) \int_0^1 \lambda_1 e^{-\lambda_1 t} dt \\
&\quad + \left\{ 1 - F_i(0) - \left[(1 - F_i(0)) \left((1 - F_i(0)) \int_0^1 \lambda_1 e^{-\lambda_1 t} dt \right) \int_1^{24} \lambda_2 e^{-\lambda_2(t-1)} dt \right] \right\} \\
&= F_i(0) + (1 - F_i(0)) \left\{ 1 - e^{-\lambda_1} \right\} + (1 - F_i(0)) \left\{ e^{-\lambda_1} \right\} \left\{ 1 - e^{-23\lambda_2} \right\} \\
&= 0.014 + (1 - 0.014) \left(1 - e^{-2.5e-3} \right) + (1 - 0.14) \left\{ e^{-2.5e-3} \right\} \left\{ 1 - e^{-23 * 8e-4} \right\} \\
&= 3.439354511689590e-2
\end{aligned} \tag{5.6}$$

The corresponding value of $P_{CSBO} = P^3$ is then equal to $4.068467294820230e-5$, to sixteen significant figures as shown in the Case B of Table 5.6 and calculated in (5.6). This value agrees with that of Case A, the numerical integration method through MATLAB, up to fifteen significant figures.

Subsection 5.2.b – Dynamic Credit for Offsite Recovery (Only) for STPQUAD3

This subsection will describe the methodology explored to receive results for Case D of the STPQUAD3 verification. It uses the method of taking dynamic credit for offsite recovery, not onsite, using a single integral method outlined by Lloyd [13] and described on the top of page 8 of Rodgers et al. [1]. The equation for this Lloyd approximation is shown as Equation (5.7).

$$\begin{aligned}
F_{\text{CSBO}}(T) &= \prod_{i=1}^n \hat{R}_i \left\{ [1 - G(T_c)] \prod_{i=1}^n F_i(0) + \int_0^T [1 - G(T' + T_c)] \times \right. \\
&\quad \left. \sum_{i=1}^n f_i(T') \int_{0-}^{T'} \dots \int_{0-}^{T'} \left[\prod_{\substack{j=1, \\ j \neq i}}^n f_j(\tau'_j) \right] d\tau'_1 \dots d\tau'_{i-1} d\tau'_{i+1} \dots d\tau'_n dT' \right\} \\
&= \prod_{i=1}^n \hat{R}_i \left\{ \left[1 - G(T_c) \right] \prod_{i=1}^n F_i(0) \right. \\
&\quad \left. + \int_0^T [1 - G(T' + T_c)] \sum_{i=1}^n \left[f_i(T') \prod_{\substack{j=1, \\ j \neq i}}^n F_j(T') \right] dT' \right\} \\
&= \text{Prob}\{T_{\text{CSBO}} \leq T\}.
\end{aligned} \tag{5.7}$$

It is important to note that Equation (5.3) and Equation (5.7) are identical; however, different values of n are used in the previous section. Therefore, it is also important to note that the variables used in Equation (5.7) are summarized in Table 3.1 and Table 3.2. Equation (5.7) also has the same corrections shown for Equation (5.3) which are listed as: i) the replacing of $\hat{F}_i(0)$ to $F_i(0)$, and ii) the insertion of a sum over I with the factor $f_i(T')$ inclusions in line 3. Also $\hat{R}_i=1$ is the non-recovery factor for onsite repair that is appropriate to the current conservative approximation of no credit for onsite recovery. By contrast, this is the best estimate approach to taking credit for offsite recovery. Computationally, this new recovery-based failure probabilities of critical station blackout is shown in Equation (5.8) where f_i and F_i are respectively the pdf for failure to run and the cdf for failure of any one of the generators.

$$\begin{aligned}
F_{\text{CSBO}}(T) &= \text{Prob}\{T_{\text{CSBO}} \leq T\} \\
&= \prod_{i=1}^n \hat{R}_i \left\{ \begin{aligned} & \left[1 - G(T_c) \right] \prod_{i=1}^n F_i(0) \\ & + \int_0^T \left[1 - G(T' + T_c) \right] * \\ & \sum_{i=1}^n f_i(T') \int_0^{T'} \dots \int_0^{T'} \left[\prod_{j=1, j \neq i}^n f_j(\tau'_j) \right] * \\ & d\tau'_i \dots d\tau'_{i-1} d\tau'_{i+1} \dots d\tau'_n dT' \end{aligned} \right\} \tag{5.8} \\
&= 1^3 [1 - G(1; 0.0300, 1.604)] (0.014)^3 \\
&\quad + 3 \int_0^{24} [1 - G(T' + 1; 0.0300, 1.604)] f(T') F(T')^2 dT'
\end{aligned}$$

The pdf and cdf terms are shown more explicitly in Table 5.8.

Table 5.8

pdf and cdf terms (STPQUAD3).

Term	Equation	Bound
$f_i(T') =$	$(1 - 0.014)(2.5e - 3)e^{-(2.5e-3)T'}$	for $0 \leq T' \leq 1$
$F_i(T') =$	$0.014 + (1 - 0.014)(1 - e^{-(2.5e-3)T'})$	for $0 \leq T' \leq 1$
$f_i(T') =$	$(1 - 0.014)e^{(2.5e-3)}(8.0e - 4)e^{-(8.0e-4)(T'-1)}$	for $T' \geq 1$
$F_i(T') =$	$(1 - 0.014)e^{(2.5e-3)}(1 - e^{-(8.0e-4)(T'-1)})$	for $T' \geq 1$

A MATLAB code (“Lloyd3.m”) was written to evaluate (5.8), with the integration carried out numerically using the existing (one-dimensional numerical integration) MATLAB QUAD, with the integrand evaluated by means of a second written MATLAB code “INTEGRAND3.m”. The result, with the integration effected at the default error tolerance for QUAD (absolute error tolerance of $1.0e-6$), is shown in the first value of Case D of Table 5.6. This value is about 30% lower than the corresponding value from STPQUAD3, which is the first value of Case C of Table 5.6. The two corresponding results, those immediately below the values in Table 5.6, are for a more stringent error tolerance for the numerical integration. These values have a discrepancy in reality due to an inadequately converged numerical error in the respective numerical integrations, especially for “Lloyd3”.

This lack of convergence at the default error tolerance of $TOL=1.0e-6$ should not be totally unexpected as the value for PcsBO itself is of the same order. But this does suggest the error tolerance is absolute, rather than relative. Therefore, if anything, the impact of the error tolerance for the numerical integration appears even more significant than in the previously studied two emergency diesel generator cases. This error tolerance may be even more crucial due to the triple integration within the STPQUAD3 script that sometimes required a noticeable amount of time for computation. Just as in the 2-EDG case, it seems worthwhile to digress briefly for a systematic study of the impact of error tolerance in the respective numerical integrations.

Table 5.9 displays the values of PcsBO as obtained from both STPQUAD3 and Lloyd3 with various values of the error tolerance used for their respective numerical integration codes. . The results in Table 5.9 contain much more detail than required to reach a satisfactory conclusion. As in the preceding section, some of the detail could be useful later for presently unforeseen purposes and should be left to future research. For example, the results could be used to verify subsequent significantly revised versions of STPQUAD3. The underlined digit is the first digit that does not agree with the value immediately above that cell in Table 5.9 when both values are rounded. Reduction of the absolute error tolerance was terminated when STPQUAD3, with use of TRIPLEQUAD, began to require an excessive time to compute.

Table 5.9

PCSBO Values While Varying Absolute Error Tolerances with Case C vs Case D (STPQUAD3).

<i>Absolute Error Tolerance</i>	STPQUAD3 <i>PCSBO</i>	Computational Time	Lloyd3 <i>PCSBO</i>
1.0e-5	3.511717430419162e-6	Negligible	2.760537818226182e-6
1.0e-6	3.517725266608635e-6	Negligible	2.760537818226182e-6
1.0e-7	3.484986939479499e-6	Negligible	2.760537818226182e-6
1.0e-8	3.475247269638392e-6	Negligible	2.671103400838021e-6
1.0e-9	3.450868292445484e-6	a few seconds	3.390520692122483e-6
1.0e-10	3.447045307250793e-6	~ 20 seconds	3.435769452659052e-6
1.0e-11	3.446884257000039e-6	~80 seconds	3.441698096826442e-6
1.0e-12	3.446885343382636e-6	~2 minutes 30 seconds	3.446668725886406e-6
1.0e-13	3.446887508231133e-6	~6 minutes 50 seconds	3.446825680272363e-6
1.0e-14	3.446889053301027e-6	~27 minutes 40 seconds	3.446886944545614e-6

Based upon the results shown in Table 5.9, it appears that STPQUAD3 requires an error tolerance on the order of 1.0e-8 or less to obtain even two digits of accuracy in the resulting estimate for PCSBO. On the other hand, tolerances much less than 1.0e-10 seem to require significant amounts of time. For TRIPLEQUAD, as employed within STPQUAD3, it is recommended that 1.0e-10 is used as the standard error tolerance. For a similar analysis of systems with more than three onsite AC generators, the low probabilities corresponding to such a high level of defense would require such high levels of absolute accuracy in the numerical integration that the time required for that integration could become a significant issue.

STPQUAD3 ultimately achieves six digits of accuracy, albeit at the cost of considerable integration time. In contrast, Lloyd3 converges more slowly as the error tolerance decreases, and never achieves more than four digits of accuracy, even at the smallest error tolerance displayed in Table 5.9. The MATLAB documentation says of QUAD, the numerical single-integration employed within Lloyd3 to obtain the results of Table 5.9, "QUAD may be most efficient for low accuracies with nonsmooth integrands"

[16]. The integrand occurring in Lloyd3 is relatively smooth, but high absolute accuracy is required because of the small value of the integrand. This suggests the use of one of the alternative single-integration codes available within MATLAB might be more appropriate within Lloyd3.

Table 5.10, displays the corresponding results, with the MATLAB single-integration code QUADL employed for the numerical integration within Lloyd3, rather than QUAD. The QUADL documentation in MATLAB claims that it “may be more efficient than QUAD at high accuracies with smooth integrands” [16].

Table 5.10

PC_{SBO} Values Varying Absolute Error Tolerances with Case C vs Case D with QUADL (STPQUAD3).

<i>Absolute Error Tolerance</i>	<i>STPQUAD3 P_{CSBO}</i>	<i>Computational Time</i>	<i>Lloyd3 w/QUADL P_{CSBO}</i>
1.0e-5	3.511717430419162e-6	Negligible	2.760537818226182e-6
1.0e-6	3.517725266608635e-6	Negligible	2.760537818226182e-6
1.0e-7	3.484986939479499e-6	Negligible	2.760537818226182e-6
1.0e-8	3.475247269638392e-6	Negligible	2.671103400838021e-6
1.0e-9	3.450868292445484e-6	a few seconds	3.390520692122483e-6
1.0e-10	3.447045307250793e-6	~ 20 seconds	3.435769452659052e-6
1.0e-11	3.446884257000039e-6	~80 seconds	3.441698096826442e-6
1.0e-12	3.446885343382636e-6	~2 minutes 30 seconds	3.446668725886406e-6
1.0e-13	3.446887508231133e-6	~6 minutes 50 seconds	3.446825680272363e-6
1.0e-14	3.446889053301027e-6	~27 minutes 40 seconds	3.446886944545614e-6

The Lloyd3 results in Table 5.10 are somewhat better converged (six digits) than the corresponding results in Table 5.9, (four digits). More importantly for present purposes, the most highly converged Lloyd3 results in Table 5.9 agree with the corresponding results from STPQUAD3 to seven digits of accuracy rounded. These along

with the similar seven digits of agreement in Table 5.9 are taken as the further verification of STPQUAD3.

Subsection 5.2.c - Dynamic Offsite Recovery, Onsite Repair Post Failure (STPQUAD3)

This subsection will describe the methodology explored to receive results for Case F of the STPQUAD3 verification. Case F uses the method of taking dynamic credit for offsite recovery, using a single integral method outlined by Lloyd [13] and described on the top of page 8 of Rodgers et al. [1]. It also credits repair only after a full failure of onsite power. In this case, the probability of critical station blackout is given by Equation (5.8) but now with the repair term defined in (5.9).

$$\hat{R}_i = 1 - R_i(T_c) = 1 - W(1; 0.739, 15.50) = 0.8764019916337390 \quad (5.9)$$

The corresponding result shown in (5.9), as obtained by Lloyd3 with the default error tolerance of 1.0e-6 and the recommended more stringent error tolerance of 1.0e-10, are shown in Case F of Table 5.7

If the default error tolerances of 1.0e-6 are used in the upper values of Table 5.7 for Case E and F, the difference between the computed values (as in the previous subsection, between the computer value for the probability of critical station blackout from STPQUAD3, Case E, and that from Lloyd, Case F) is about 0.5e-6. This is comparable to the reported values of Pcsbo. The hypothesis is that the difference is still due to the inevitable integration error, primarily from STPQUAD3, but perhaps mostly in Lloyd3. The bottommost (“parenthetical”) entries in Table 5.7 for these Cases, which are obtained using a more stringent error tolerance of 1.0e-10, show a difference of only about 0.4e-9. This seems to provide a strong enough support to justify the formulated hypothesis. It is left to future work to look more closely into the impact of the error tolerance for this application as analogous to Table 5.5 of the previous section.

Subsection 5.2.d – Comparison of Methods (STPQUAD3)

This subsection is focused on verification of the MATLAB research code STPQUAD3 in showing the overall comparison between methods.

Following the events of Fukushima Dai-ichi many a Near-term Task Force was created to review the current fleet of commercial nuclear power plants in the United States. This task force was made up of NRC officials and representatives for the utilities as well as experts in the field of nuclear energy. Recommendation number nine of the recent Near-Term Task Force review reads as follows:

“The Task Force recommends that the NRC require that facility emergency plans address prolonged station blackout and multiunit events.” [7]

This suggests applying STPQUAD3 to investigate the sensitivity of the predicted values of PCSBO to the chosen mission time. The results displayed in Table 5.11 indicate a very high degree of insensitivity. In effect, the various estimates of the probability of critical station blackout are fully converged at a mission time of 24 hours. In the present context, the 24-hour timeframe is an excellent approximation to infinity for the probability of critical station blackout.

Table 5.11Comparison of Estimates of P_{CSBO} from Various Methods (STPQUAD3).

Method	P_{CSBO} $T = 24$ hours	P_{CSBO} $T = 48$ hours	P_{CSBO} $T = 1$ week
No credit for recovery	4.07e-5	1.47e-4	2.71e-3
Credit for recovery (onsite and offsite) only following SBO (i.e., Read-Fleming [14] variant of Lloyd-Anoba method), computed using STPQUAD3, with integration error tolerance of 1.0e-10 for integration	3.45e-6	3.56e-6	3.62e-6
Dynamic offsite recovery, credit for onsite recovery only following SBO, computed using STPQUAD3, with integration error tolerance of 1.0e-10 for integration	2.32e-6	2.39e-6	2.44e-6
Best estimate, with fully dynamic treatment of both onsite and offsite recovery, computed using STPQUAD3, with integration error tolerance of 1.0e-10 for integration	1.97e-6	1.98e-6	1.98e-6

However, the results from Table 5.11 seem largely associated with the use of a lognormal distribution for offsite recovery, which provides a distribution function that dies away very rapidly with duration of the hypothesized event seen in Figure 5.1.

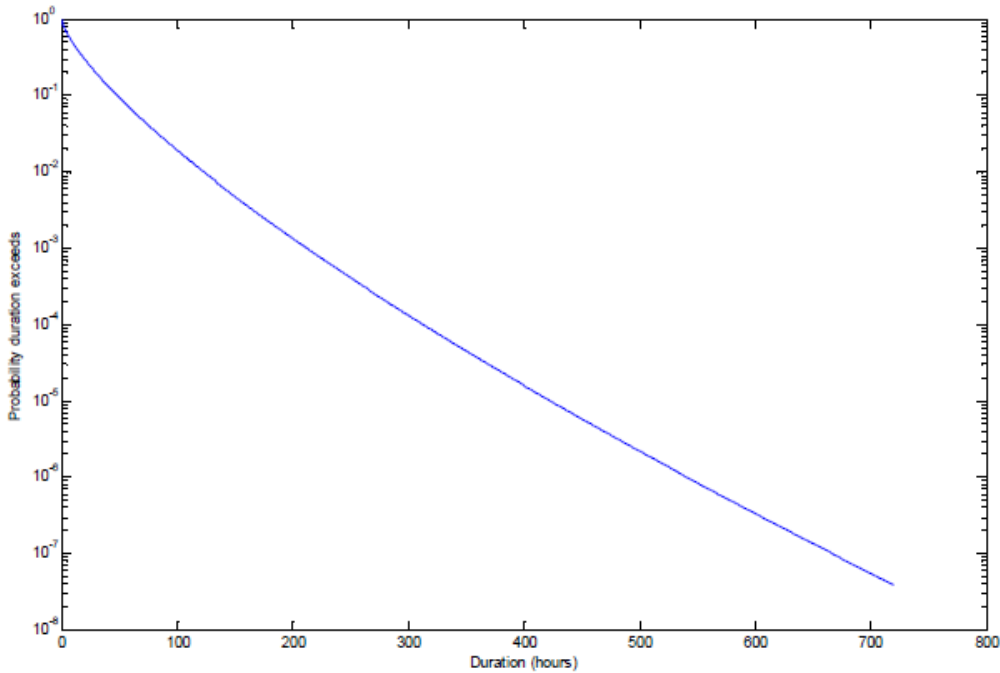


Figure 5.1 – Complemented Cumulative Distribution Function for One Month (30 Days), for the Lognormal Distribution Employed Here for Offsite Recovery

It seems unlikely that the tacitly imbedded hypothesis in this distribution that the probability of a LOOP event extending beyond a duration of 30 days is less than one in ten million which is supported by data. A quick estimate suggests that the lognormal distribution describing non-recovery of offsite power behaves asymptotically in duration (t) as in Equation (5.10).

$$\frac{1}{t \ln(t)} \quad (5.10)$$

This solution converges more slowly than any exponential decay, but faster than any power law. A more conservative, “heavy-tailed”, fit to the data and one that seems to have some support in data for various sorts of catastrophic events, would be a power law.

Section 5.3 – Verification of STPQUAD3CCF (with Common-Cause)

This section describes the attempted verification of the code “STPQUAD3CCF.m”. The code itself is listed in Appendix A.3. One big change for the STPQUAD3CCF is the required input file described more in Chapter II. This run file is provided as an example, screen shots from Microsoft Excel, shown in Appendix B. This code suite, fully displayed in Appendix A, is the final version of the STP-TAMIL code suite to date. It is benchmarked a few different ways which include verification against STPNOC specific values that are kept confidential (see Appendix C) and against average PRA industry numbers [18, 19, 24]. Subsections 5.3.a-5.3.c cover the STPNOC case; and Subsections 5.3.d-5.3.g cover the methodology and the use of the industrial average.

STP-TAMIL uses a dynamic stochastic model of events occurring during a Loss of Offsite Power (LOOP) event to compute nonrecovery integrals, which include:

- a. no recovery (which is generally done in a typical “static” PRAs);
- b. dynamic credit only for recovery of offsite power (Lloyd-Anoba [11, 13]);
- c. credit for recovery of offsite power and for repair following onset of Station Blackout (SBO) (Read-Fleming [14]); and,
- d. credit for recovery of offsite power and for repair of any EDG train occurring at any time following its failure

Previous sections in this chapter, describe verification studies carried out under the assumption of only independent failure, and two and three EDG trains. The version verified in this section considered three EDG trains as well as common-cause failures for 2/3 (two of three) and 3/3 (three of three) trains. Here 2/3 train common-cause failures are failures that lead two EDG trains to fail simultaneously for the same reason. To reach station blackout, the last EDG would fail independently. The 3/3 train common cause failures are failures that would cause three EDG trains to fail instantaneously for the same reason. Therefore, the station would enter blackout simultaneous with failure.

As written, STP-TAMIL allows arbitrary user-specified distributions in time of failures of EDG trains (independent, 2/3 and 3/3 failures), of recoveries of offsite power and of repair of the EDG trains. Therefore, a verification or validation requires some specific choice of those various distributions. For recovery of offsite power and repair,

the same specific distributions previously documented in the verified studies are employed. For the failure distributions, the same two-regime exponential distribution likewise documented was employed. However, the probabilities of all types of failure to start and rates of failure to load and run (failures during the first hour following a LOOP event), and of failures to run (during the subsequent 23 hours of a presumed 24 hour mission time) were inferred from cut sets developed for the EDG cases using the STP PRA model (see Appendix C). The manner of these inferences will be discussed in more detail in the rest of this section.

Subsection 5.3.a – 3/3 Common-Cause Failures

The four most important minimal cut sets in the STPNOC PRA model, see Appendix C, are all examples of 3/3 common-cause failures. They are shown in Table 5.12. They were, with associated values of probabilities over the 24-hour mission time, conditional on a LOOP occurring.

Table 5.12

Top Four Cut Sets in the STP PRA Model.

Cut set	Description	Variable
1	Probability of 3/3 failure to Run of EDGs	P3FTR
2	Probability of 3/3 failure to Start of EDGs	P3FTSEDG
3	Probability of 3/3 failure to Load and Run of EDGs	P3FTLR
4	Probability of 3/3 failure to Start of Fans	P3FTSFans
2+4	Probability of 3/3 failure to Start (Total)	P3FTS

Note that cut set “2+4” in Table 5.12 acts to serve as a total failure to start probability combining, both the EDGs and the associated fans that are generally used for cooling of the EDGs.

If λ_1 is the failure rate during the first hour, then the probability of failure during that hour, i.e. failure to start and run for one hour, is conditional upon start occurring shown in Equation (5.11).

$$1 - e^{-\lambda_1} \quad (5.11)$$

Then the total probability of failure to load and run is expressed in Equation (5.12).

$$(1 - P_{3FTS}) \left(1 - e^{-\lambda_1} \right) = P_{3FTLR} \quad (5.12)$$

PNORMAL is equal to about 0.99159 [23], the probability (fraction of time) that the system is in normal alignment. This can also be solved for λ_1 as shown in Equation (5.13).

$$\lambda_1 = -\ln \left(1 - \frac{P_{3FTLR}}{(1 - P_{3FTS})} \right) \quad (5.13)$$

The parameter required as an input to the code is the associated mean, μ_1 , which is found in (5.14).

$$\mu_1 = 1 / \lambda_1 = \frac{1}{\ln \left(1 - \frac{P_{3FTLR}}{(1 - P_{3FTS})} \right)} \quad (5.14)$$

Similarly, fail to run probability can also be determined by the same approach that was previously used to calculate fail to start probability. This assumes that λ_2 is the rate of failure during the final 23 hours of the mission time. Then the probability of a common-cause triplet failure to run is shown in Equation (5.15).

$$(1 - P_{3FTS})(1 - P_{3FTLR})(1 - e^{-23\lambda_2}) = P_{3FTR} \quad (5.15)$$

Equation (5.15) can be solved for the corresponding mean time to failure (conditional upon the first hour being reached without failure), μ_2 , as shown in Equation (5.16)

$$\mu_2 = 1 / \lambda_2 = - \frac{23}{\ln \left(1 - \frac{P_{3FTR}}{[(1 - P_{3FTS})(1 - P_{3FTLR})]} \right)} \quad (5.16)$$

All of the minimal cut sets considered fall above this value, within the normal alignment. Within the dynamic framework, unavailability due to maintenance, is to be treated as a failure to start. Thus verification against some class of minimal cut sets, including some deriving from other than normal alignment, would require that all of the probabilities and rates be re-determined, with the corresponding value of P3FTS including maintenance-related failures to start. For the current common-cause triplet failure, considerations would involve only very rare events in which all three generators are unavailable due to maintenance. However, for the independent failures, these considerations would be more common as one emergency diesel generator could briefly go offline and be under preventative maintenance without causing big safety concerns; in fact with enchantment of safety, over a long period.

The results for common-cause triplet failures are shown in Table 5.13; underlying the process is described within the rest of this subsection. Due to confidentiality, the results are shown as some constant probability X, or some other variable and the change in results are presented as X times some delta constant. For example, if the results for Case A's probability were actually 1.0E-4, then X would equal 1.0E-4. Therefore, if

another case claims the probability is “ $=X*20\%$ ”, then the probability for that particular case would be $2E-5$. This would mean that this case would have an 80% reduction due to recovery from Case A which has no recovery.

Table 5.13

Probability of Critical Station Blackout from a Common-cause Triplet Failure of EDGs or Associated Fans, Given a LOOP Initiating Event has Occurred.

Case	Method	Probability of Critical Station Blackout (P_{CSBO3})
Case A	No credit for recovery (offsite or onsite), from STPQUAD3CCF (with failure probabilities and rates inferred from STP PRA Model, see Appendix C, as described previously in this subsection.	X
Case B	PRA result from STP PRA Model using four most significant minimal cut sets, see Appendix C.	$\approx X*1.0$
Case C	Credit for offsite recovery only, computed dynamically via STPQUAD3CCF single integration (default error tolerance), but with partial integrand for nonrecovery integral that is computed en route to full integrand in subroutine INTEGRANDv1pt0 (Lloyd-Anoba [11])	$= X*0.17976$
Case D	Computed dynamically via STPQUAD3CCF single integration, with integrand for nonrecovery integral that is fully dynamic for offsite recovery but incorporates onsite repair only after full failure of onsite power, per integrand computed via slight modification at last part of INTEGRANDv1pt0 code (H) (READ-FLEMING [14])	$= X*0.12101$
Case E	Full dynamic credit for recovery of offsite and onsite power, single integral evaluated per STPQUAD3CCF	$= X*0.12101$

Case A of Table 5.13 shows that STP-TAMIL reports the value of P_{CSBO3} , with only common-cause triplet failures of the EDGs and fans considered, as X . All results have been considered using five significant figures, which is the maximum warranted. This level of significant figures is used because the probabilities from the model, see Appendix C, that were used to extract probabilities and rates of failure only contained few significant figures. Case B shows that this is exactly the value from the corresponding, most probable four, minimal cut sets of the STPNOC PRA model. See Appendix C. Preliminary calculations as above, to extract the CC triplet failure probabilities and rates, employed the approximation shown in (5.17).

$$1 - e^{-\lambda t} \approx \lambda t \quad (5.17)$$

The corresponding result from STP-TAMIL was $\approx X*1.0$, which differed by a significant figure in the fifth place digits.

With credit for restoration of offsite power, Case C, the reported value of critical station blackout becomes $P_{CSBO3} = X*0.17976$. This is a reduction of about 82.024% of Case A, which takes no credit for recovery. With additional credit for repair of the EDG systems, P_{CSBO3} becomes $= X*0.12101$. The reported values in Case D and Case E agree and are identical. Both are a reduction of Case A of about 87.8989%. In both cases, all EDGs fail simultaneously in a common-cause triplet shock event, so that time of initial failure of a train is the same as time of failure of all three trains. Therefore, there is no distinction between repair beginning at the time of initial failure of a train and repair beginning at the time that the third train fails.

It can be noted that Case D's method refers to the last part of the INTEGRANDv1pt0code. This refers to the type of non-recovery involved in the code itself. In this code the part has a very different non-recovery factors (H) which are determined different ways based on the type of recovery and when repair starts. This idea of recovery is true for all versions of the code These lines are uncommented or commented when certain types of recovery or repair happen based on the input deck.

Subsection 5.3.b– Inclusion of Independent Failures

The next logical step is to extract the failure probabilities and rates for independent failures of the EDGs, then verify STP-TAMIL results against the PRA results for the minimal cut sets that involve both common-cause triplet and independent failures. These cut sets with descriptions are shown in Table 5.14.

Table 5.14

Important Cut Sets for Independent Failure Based on the Industry Average

Cut set	Description	Variable
52	Probability Independent Failure to Load and Run of all three EDGs	P_{IFTLR}^3
$\sqrt[3]{52}$	Probability of a Single Independent failure to load and run (contingent upon start occurring which includes the particular EDG being available)	P_{IFTLR}
81-83	Probability of Independent failure to start of some particular EDG, accompanied by independent failures to load and run of the two remaining EDGs. ($P_{IFTS}P_{IFTLR}^2$)	$P_{1IFTS2IFTLR}$
N/A	$P_{IFTS} = P_{1IFTS2IFTLR} / P_{IFTLR}^2$	P_{IFTS}
5	Probability all three EDG trains fail to run independently	P_{3IFTR}
$\sqrt[3]{5}$	Probability of a Single Independent failure to run (contingent upon start occurring which includes the particular EDG being available)	P_{IFTR}

Note that P_{IFTLR} can be written as a function of the independent failure to load and run over the first hour rate, λ_1 , as shown in Equation (5.18) which can then be solved as shown in Equation (5.19).

$$P_{IFTLR} = (1 - P_{IFTS})(1 - e^{-\lambda_1}) \quad (5.18)$$

$$\lambda_1 = -\ln\left(1 - \frac{P_{IFTLR}}{1 - P_{IFTS}}\right) \quad (5.19)$$

The parameter required as an input for the code is the associated mean, μ_1 , and can now be solved and shown in Equation (5.20).

$$\mu_1 = 1 / \lambda_1 = -\frac{1}{\ln\left(1 - \frac{P_{IFTLR}}{1 - P_{IFTS}}\right)} \quad (5.20)$$

Note that P_{3IFTR} can be written as a function of the independent failure run over the final 23 hours of operation following a LOOP (considering a 24 hour mission time) rate, λ_2 , as shown in Equation (5.21) which can then be solved as shown in Equation (5.22). Note that λ_2 is contingent upon starting and loading/running for the first hour.

$$P_{IFTR} = \sqrt[3]{P_{3IFTR}} = (1 - P_{IFTS})(1 - P_{IFTLR})(1 - e^{-\lambda_2}) \quad (5.21)$$

$$\lambda_2 = -\ln\left(1 - \frac{P_{IFTR}}{(1 - P_{IFTS})(1 - P_{IFTLR})}\right) \quad (5.22)$$

The parameter required as an input for the code is the associated mean, μ_2 , and can now be solved and shown in Equation (5.23).

$$\mu_2 = 1/\lambda_2 = -\frac{1}{\ln\left(1 - \frac{P_{IFTR}}{(1-P_{IFTS})(1-P_{IFTLR})}\right)} \quad (5.23)$$

Table 5.15 contains results with the inclusion of independent failures, in addition to the common-cause triplet failures considered in the earlier in this section. The results largely reflect the trends that were seen in Table 5.13. Note: That X, Y, and T are the probabilities held confidential from STPNOC, see Appendix C. The description above Table 5.13 describes more fully how these constants are used.

Table 5.15 differs in two senses from the trends from Table 5.13, which both warrant further dialogue and are discussed below.

First, the value of probability of critical station blackout due to independent failures only, P_CSBOI, and with no credit for recovery, was found from STPQUAD3CCF to be Y. By contrast, the value found by summing the probabilities of the corresponding minimal cut sets in the 100 most probable minimal cut sets was found to be $\approx Y \cdot 0.9930$. It is reasonable to assume that the discrepancy was due to some relevant minimal cut sets not being among the “top 100.” On further inquiry, those not included were:

- i) the three minimal cut sets corresponding to independent failure of some EDG to load and run, and independent failure of the other two to start; and
- ii) the minimal cut set corresponding to independent failure of all three EDGs to start.

Table 5.15

Probabilities of Critical Station Blackout from Common-cause Triplet Failure of EDGs or Associated Fans, and from Independent Failures of the EDGs, Given a LOOP Event.

Case	Method	P_{CSBO3}	P_{CSBO1}	$P_{CSBO} = P_{CSBO3} + P_{CSBO1}$
A	No credit for recovery (offsite or onsite), from STPQUAD3CCF, with failure probabilities and rates inferred from STP system notebook, shown in Appendix C, as described in above text.	X	Y	W
B	PRA result, from STP system notebook (relevant minimal cut sets among the 100 most probable). See Appendix C.	$\approx X*1.0$	$=Y*0.9930^a$	$=W*0.99930$
C	Credit for offsite recovery only, computed dynamically via STPQUAD3ccf integration, with partial integrand for nonrecovery integral that is computed en route to full integrand in subroutine INTEGRANDv1pt0 and INTEGRAND3way (Lloyd-Anoba [11])	$=X*0.17976$	$=Y*0.03219$	$=W*0.1649$
D	Computed dynamically via STPQUAD3CCF single integration, with integrand for nonrecovery integral that is fully dynamic for offsite recovery but incorporates onsite repair only after full failure of onsite power, per integrand computed via slight modification at last part of relevant INTEGRAND codes (Read-Fleming [14])	$=X*0.12101$	$=Y*0.02167$	$=W*0.11103$
E	Full dynamic credit for recovery of offsite and onsite power, single integral evaluated per STPQUAD3CCF	$=X*0.12101$	$=Y*0.01476$	$=W*0.11033$

^a If all relevant minimal cut sets are included, as opposed to just those in the top 100, see Appendix C, then this value becomes $=Y*1.000014$, which agrees with the value from STPQUAD3CCF to within one digit in the fifth place. See text for further related discussion.

These minimal cut sets were identified respectively as numbers 105-107 and 174 in an expanded STP PRA Model, see Appendix C, incorporating a list of the top 250 most probable minimal cut sets. Their respective probabilities were determined from the expanded system notebook to be $Y*0.00203$, for each of the minimal cut sets of class i) above, and $Y*9.1786E-4$ for that in class ii). When these probabilities were added to that associated with minimal cut sets of the relevant type that were among the top 100, the associated probability was found to be $Y*1.000014$, which agrees with the value from STPQUAD3CCF to within one in the fifth to last digit. The probabilities and rates were determined above and used as input to STPQAD3CCF, with only five digits of accuracy. This level of discrepancy could well be due to round off error. We therefore take these results as confirmatory of the hypothesis described earlier in this paragraph.

Second, if one compares Cases B and C of Table 5.15, then it is clear that the probability of critical station blackout due to three independent failures is reduced by about a factor of 30 by taking credit for offsite recovery. The corresponding reduction for common-cause triplet failures is approximately a factor of five. This is accounted for by the fact that a much higher proportion of the putative SBOs associated to three independent failures would occur within the mission time. These would be absent of any kind of recovery and occur late in the mission time, as compared to such putative SBOs associated to common-cause triplet failures. This is illustrated in Figure 5.2, where the blue and the red curves correspond to SBOs (more precisely, putative SBOs, absent recovery) associated to respectively common-cause triplet failures of the EDGs or three independent failures. The solid lines are the corresponding cdfs for the respective types of SBOs, conditional upon such an SBO occurring during a (24-hour) mission time, absent recovery. The dotted lines are the cdfs for the SBOs occurring and not being prevented by recovery of offsite power. This is conditioned upon occurrence of failure during the mission time, absent of any form of recovery. The crucial point is that the dotted red line (SBOs due to three independent EDG failures not prevented by recovery) is much lower than all of the other curves (and constant, to the eye, except very near the initial occurrence of LOOP). Almost all of these unrecovered SBOs associated to three independent failures stem from failures to start. The vast majority of the other such putative SBOs would occur after the first hour, and hence are precluded because offsite

recovery is highly likely (65% chance) by the end of two hours. After 3 and 4 hours the corresponding likelihoods are 77% and 85%, respectively.

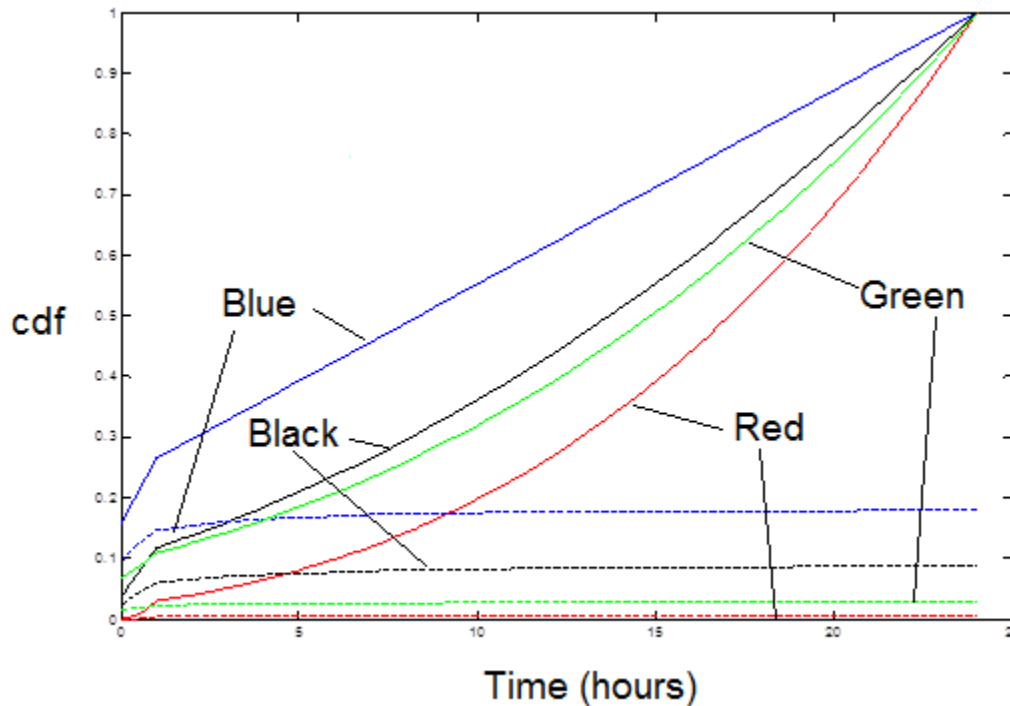


Figure 5.2 – Various cdfs for Time of SBO, Conditional Upon SBO Associated to Specific Types of EDG Failures Occurring During a 24-Hour Mission Time

Finally, this effect of unrecovered SBOs associated to three independent failures to state, was not seen as clearly in the results presented in the work reported in the previous section. This is because of differences in the independent failure data used there, notably a probability of failure to start that was about an order of magnitude larger than that employed in the present note. This failure to start cdf is shown in the green curves in Figure 5.2. The difference is largely accounted for by industry data in Section 5.2, noting one EDG being unavailable (under scheduled maintenance) while actual plant data in the present verification study did not reflect this conclusion, see Appendix C.

Subsection 5.3.c – 2/3 Common Cause Failures

The next step is to incorporate failures consisting of a 2/3 common-cause failures, plus an independent failure of the third EDG. The probabilities and rates for the independent failures (as inferred from the STP PRA model, see Appendix C) already were developed in the preceding section. The additional information needed here is the corresponding probabilities and rates for 2/3 common-cause failures.

These cut sets with descriptions are shown in Table 5.16. It can be known from Table 5.16 that P_{2FTLR} can be written as a function of the common cause doublet failure to load and run over the first hour rate, λ_1 , as shown in Equation (5.24) which can then be solved as shown in Equation (5.25).

$$P_{2FTLR} = (1 - P_{2FTS})(1 - e^{-\lambda_1}) \quad (5.24)$$

$$\lambda_1 = -\ln\left(1 - \frac{P_{2FTLR}}{1 - P_{2FTS}}\right) \quad (5.25)$$

Table 5.16

Important Cut Sets for CC Doublet Failure Based on the Industry Average

Cut set(s)	Description	Variable
34-36	Probability of two-way independent failure of the EDGs to start and an independent failure to run. $= ((P_{2FTS})(P_{IFTR}))$	$P_{1IFTR2FTS}$
N/A	Probability of a 2CC failure to start $= \left(\frac{P_{1IFTR2FTS}}{P_{IFTR}} \right)$	P_{2FTS}
${}^3\sqrt{5}$	Probability of a Single Independent failure to run (contingent upon start occurring which includes the particular EDG being available).	P_{IFTR}
19-21	Probability of two-way failure to run of some pair of EDGs accompanied by independent failure to load and run on the third. $= ((P_{IFTR})(P_{2FTLR}))$	$P_{1IFTR2FTLR}$
N/A	Probability of a 2CC failure to load and run $= \left(\frac{P_{1IFTR2FTLR}}{P_{IFTR}} \right)$	P_{2FTLR}
9-11	Probability of a common-cause doublet failure to run of a pair of EDGs plus an independent failure to run of the third. $= (P_{IFTR})(P_{2FTR})$	$P_{2FTR1IFTR}$
N/A	Probability of a common-cause doublet failure to run of a designated pair of EDGs $= \left(\frac{P_{1IFTR2FTR}}{P_{IFTR}} \right)$	P_{2FTR}

The parameter required as an input for the code is the associated mean for two-way, μ_1 , and can now be solved and shown in Equation (5.26).

$$\mu_1 = 1 / \lambda_1 = - \frac{1}{\ln \left(1 - \frac{P_{2FTLR}}{1 - P_{2FTS}} \right)} \quad (5.26)$$

Note that P_{2FTR} can be written as a function of the failure run over the final 23 hours of operation following a LOOP (considering a 24 hour mission time) rate, λ_2 , as shown in Equation (5.27) which can then be solved as shown in Equation (5.28). Note that λ_2 is contingent upon starting and loading/running for the first hour.

$$P_{2FTR} = (1 - P_{2FTS})(1 - P_{2FTLR})(1 - e^{-\lambda_2}) \quad (5.27)$$

$$\lambda_2 = -\ln \left(1 - \frac{P_{2FTR}}{(1 - P_{2FTS})(1 - P_{2FTLR})} \right) \quad (5.28)$$

The parameter required as an input for the code is the associated mean, μ_2 , and can now be solved and shown in Equation (5.29).

$$\mu_2 = 1 / \lambda_2 = - \frac{1}{\ln \left(1 - \frac{P_{2FTR}}{(1 - P_{2FTS})(1 - P_{2FTLR})} \right)} \quad (5.29)$$

Table 5.17 contains results with the inclusion of common cause doublet events with single independent failures, in addition to the common-cause triplet failures and normal independent failures considered in the earlier in this section. The results largely reflect the trends that were seen in Table 5.13.

Table 5.17

Probabilities of CSBO from Common-cause Triplet, CC Doublet or Independent Failures of EDGs Along with the Total, Given a LOOP Event.

Case	Method	P_{CSBO3}	P_{CSBO2}	P_{CSBO1}	$P_{CSBO} = P_{CSBO3} + P_{CSBO2} + P_{CSBO1}$
A	No credit for recovery (offsite or onsite), from STPQUAD3CCF, with failure probabilities and rates inferred from STP system notebook, see Appendix C.	X	Z	Y	T
B	PRA result, from STP system notebook (relevant minimal cut sets among the 100 most probable), see Appendix C.	$\approx X*1.0$	$=Z*0.99005^a$	$=Y*0.9930^b$	$=T*0.9988^c$
C	Credit for offsite recovery only, computed dynamically via STPQUAD3CCF integration, with partial integrand for nonrecovery integral that is computed en route to full integrand in subroutine INTEGRANDv1pt0 and INTEGRAND3way (Lloyd-Anoba [11])	$=X*0.17976$	$=Z*0.19182$	$=Y*0.03219$	$=T*0.16636$
D	Computed dynamically via STPQUAD3CCF single integration, with integrand for nonrecovery integral that is fully dynamic for offsite recovery but incorporates onsite repair only after full failure of onsite power, per integrand computed via slight modification at last part of relevant INTEGRAND codes (Read-Fleming [14])	$=X*0.12101$	$=Z*0.14733$	$=Y*0.02167$	$=T*0.11294$
E	Full dynamic credit for recovery of offsite and onsite power, single integral evaluated per STPQUAD3CCF	$=X*0.12101$	$=Z*0.13766$	$=Y*0.01476$	$=T*.11178$

^aThe 100 most probable minimal cut sets excludes SBO associated with a common-cause doublet failure to start accompanied by an independent failure to start of the third EDG. These are the 114th through 116th most probable minimal cut sets, in the top 250 list. If their probabilities are added, the resulting probability is $\approx Z*1.0$, which agrees to all five displayed digits with the results from STPQUAD3CCF.

^b If all relevant minimal cut sets are included, as opposed to just those in the top 100, then this becomes $=Y*1.000014$, which agrees with the value from STPQUAD3CCF to within one digit in the fifth place. See subsection for further related discussion.

^c $=T*1.000013$ if the additional minimal cut sets indicated in footnotes a and b are accounted for.

The results for this particular CSBO are shown in the PCSBO2 column of Table 5.17. These results are presented in the green curves of Figure 5.2 in the previous subsection. Qualitatively, they are more similar in nature to the common-cause triplet case than to the three independent failures. This reflects the fact that the distribution in time of the presently considered 2+1 SBOs is more nearly similar to that of the common-cause triplet SBOs than to the three independent-failure SBOs. That is, the shape of the black curves in Figure 5.2 is more nearly similar to that of the blue curves, which are the cdf assuming SBO occurring during a 24-hour mission time absent of recovery for common-cause triplet failures. In comparison, the blue curves are not close in shape to the red curves, which are the cdf, assuming SBO occurs during a 24-hour mission time absent of recovery for independent failures. In other words, the black and blue curves differ primarily in scale, which is not true for the black and red curves.

Subsection 5.3.d– Industrial Average Benchmark

The industrial average benchmarked example uses the technique known as the alpha factor method to convert compiled industry failure data into functional data within a model.

Failure data is collected for the specific components and separated by type of component failure. In the EDG case, these include failure to load and run, failure to start, and failure to run events. These failure distributions as shown in NUREG 6928 [24] as the mean of these distributions were used as our generic value for the EDG of a specific type of failure. Before moving on it is imperative to fully discuss the alpha factor method which is discussed in Subsection 5.3.e.

Subsection 5.3.e– Alpha Factor Method

The alpha factor method, as described here, was adapted from NUREG 4780, Volume I, Section 3.3 [25] and Volume II [26], specifically pages C-4 through C-6 and C-9 through C-11.

A “basic event” is considered to be a failure of some set of k components based on an m -component common-cause group. The symbol, k , is a integer values based on the current component composed inside of the set $1 \leq k \leq m$. The symbol, m , is the total number of components for that particular cut set. In the STPNOC case for looking at EDG failure, m would be equal to three as there are three EDGs. Basic events may also be referred briefly as “ k -way (common-cause) failure”, except that the case $k=1$ shall often be termed as an “independent failure”. The probability (for demand-based failure) or frequency (for time-based failure) of a k -way common-cause failure of some specific set of k components, and only those k components, is designated as $Q_k^{(m)}$. To simplify the language, this section will use probabilities, but the discussion applies equally to frequencies, with slight changes of language.

These probabilities of basic events involving failure of a specified set of components, and only those components, are the basic parameters of any system model, in that system (or subsystem) failure that can be expressed in terms of them. For example, the subsystem of interest in this case is an emergency AC power system comprised of three identical Emergency Diesel Generators. Therefore, the minimal cut sets for a 3-out-of-three, all EDGs have to fail; a success logic gate based of a failure of the system can be seen in Equation (5.30).

$$\begin{aligned}
 \text{IndependentFs} &:= \{A_I, B_I, C_I\} \\
 2/3CCFs &:= \{c_{AB}, C_I\} \{c_{AC}, B_I\} \{c_{BC}, A_I\} \\
 3/3CCFs &:= \{c_{ABC}\}
 \end{aligned} \tag{5.30}$$

In Equation (5.30), the symbols A_I, B_I , and C_I represent independent failures (“*IndependentFs*”) of components (EDGs) one (“A”) through three (“C”), respectively. The variables c_{AB} , c_{AC} and c_{BC} represent a common-cause doublet failure (“*2/3CCFs*”) for component A and B, A and C, and B and C, respectively. The variable c_{ABC} represents a common-cause triplet failure (“*3/3CCFs*”) for component A, B, and C.

Therefore the probability of each type of failure can then be determined which are shown in Equation (5.31).

$$\begin{aligned}
Q_1^{(3)} &= P(A_I) = P(B_I) = P(C_I) \\
Q_2^{(3)} &= P(c_{AB}) = P(c_{AC}) = P(c_{BC}) \\
Q_3^{(3)} &= P(c_{ABC})
\end{aligned}
\tag{5.31}$$

The total probability of the system, $P(s)$, and then be determined based on the probabilities and the cut sets which is shown in Equation (5.32)

$$\begin{aligned}
P(s) &= P(A_I) * P(B_I) * P(C_I) + \\
&P(c_{AB})P(C_I) * P(c_{AC})P(B_I) * P(c_{BC})P(A_I) + \\
&P(c_{ABC})
\end{aligned}
\tag{5.32}$$

Applying Equation (5.31), the system failure probabilities can be altered, resulting in Equation (5.33).

$$Q_S = \left(Q_1^{(3)}\right)^3 + 3Q_2^{(3)}Q_1^{(3)} + Q_3^{(3)}
\tag{5.33}$$

The three terms on the right in Equation (5.33) represent respectively: system failures of each of the three EDG trains independently, failures due to a common-cause doublet failure of one of the (three) pairs of EDGs accompanied by an independent failure of the third, and failures caused by a single common-cause triplet failure.

Other types of failure events are conceivable, but commonly are not considered. An example would be two common-cause doublet failures. This can give rise to inconsistencies in system models. For our purposes, the system model is Equation (5.33). The only types of basic failure events considered are those whose probabilities (or frequencies) appear here, and these basic events are considered mutually exclusive. See page C-2 of NUREG 4780 [26] for further discussion on this matter.

For reference, the need is now to define several additional quantities and express them in terms of probabilities of combinations of basic events. The probability of a k -way failure that leads to some specific component failure is represented by $Q_k^{(m)}$ times the number of ways one can select a set of $k-1$ components from among the $m-1$ remaining

once the required component has been selected. The probability of k -way failure that includes some specific component is shown in Equation (5.34).

$$\binom{m-1}{k-1} Q_k^{(m)} \quad (5.34)$$

Equation (5.34) uses the combination rule of statistics. This rule is shown as the binomial coefficient, which is displayed as Equation (5.35).

$$\binom{m}{k} = \frac{m!}{k!(m-k)!} \quad (5.35)$$

Similarly, the probability of some k -way basic event is shown in Equation (5.36).

$$\binom{m}{k} Q_k^{(m)} \quad (5.36)$$

From Equation (5.34), the quantity defined in Equation (5.37) is the probability of some k -way basic event that includes failure of some specified component.

$$Q_t := \sum_{k=1}^m \binom{m-1}{k-1} Q_k^{(m)} \quad (5.37)$$

Using the previous Equations (5.36) and (5.37), one can determine an “alpha factor” shown in (5.38). This alpha factor is defined as the probability of some k -way basic event occurring over the probability of some basic event occurring; i.e., the probability of a k -way basic event occurs, given that a basic event occurred.

$$\alpha_k^{(m)} := \frac{\binom{m}{k} Q_k^{(m)}}{\sum_{i=1}^m \binom{m}{i} Q_i^{(m)}} \quad (5.38)$$

Equation (5.38) uses Equation (5.36) and the fact that every basic event involves some positive number of component failures. Therefore, it can be noted that Equation (5.38) also represents the probability of some k -way basic event given that some basic event occurs.

Equation (3.39) represents an expected number of component failures given that some basic event occurs designated by α_t .

$$\alpha_t = \sum_{k=1}^m k \alpha_k^{(m)}. \quad (5.39)$$

An algebraic manipulation of Equations (5.39) and (5.37) can be performed to get a better interpretation of α_t . To start this approach, Equation (5.39) can be written in the form of Equation (5.40).

$$\frac{k}{m} \left\{ \sum_{i=k}^m \binom{m}{k} Q_k^{(m)} \right\} \alpha_k^{(m)} = \binom{m-1}{k-1} Q_k^{(m)} \quad (5.40)$$

To understand Equation (5.40) is important to solve for α_t as shown in Equation (5.41). This represents the size of common-cause component group multiplied by the probability of some k -way basic event that includes failure of some specified component over the probability of some basic event occurring.

$$\alpha_t = \frac{\sum_{k=1}^m k \binom{m}{k} Q_i^{(m)}}{\sum_{i=1}^m \binom{m}{k} Q_i^{(m)}} = m \frac{\sum_{k=1}^m \binom{m-1}{k-1} Q_k^{(m)}}{\sum_{k=1}^m \binom{m}{k} Q_k^{(m)}} \quad (5.41)$$

If the sum of each side of Equation (5.41) is taken from $k=1$ to m ($\sum_{k=1}^m k$), Equation (5.40) can lead to Equation (5.42).

$$\frac{1}{m} \left\{ \sum_{k=1}^m \binom{m}{k} Q_k^{(m)} \right\} \sum_{k=1}^m k \alpha_k^{(m)} = \sum_{k=1}^m \binom{m-1}{k-1} Q_k^{(m)} \quad (5.42)$$

Equation (5.42) can then be simplified into Equation (5.43) using the definition shown in Equation (5.39).

$$\sum_{k=1}^m \binom{m}{k} Q_k^{(m)} = \frac{m}{\alpha_t} Q_t \quad (5.43)$$

To better understand, the Equation (5.43) is split into two parts as demonstrated in Equation (5.44) using the definition shown in Equation (5.38).

$$\frac{\alpha_k^{(m)}}{(\alpha_t / m)} Q_t = \binom{m}{k} Q_k^{(m)} \quad (5.44)$$

The right side of Equation (5.44) can be defined as the probability of some k -way basic event. The left side of Equation (5.44) is then defined as the probability of some k -way basic event over the probability of some basic event; multiplied by, the probability of some k -way basic event that includes failure of some specified component; divided by, the probability of some k -way basic event that includes failure of some specified component over the probability of some basic event.

Equation (5.44) can then be solved for the probability $Q_k^{(m)}$. The result is displayed in Equation (5.45).

$$Q_k^{(m)} = \frac{m}{\binom{m}{k}} \frac{\alpha_k^{(m)}}{\alpha_t} Q_t \quad (5.45)$$

In any application of PRA it is the $Q_k^{(m)}$ that are ultimately required, for the insertion into some system failure equation, such as Equation (5.33). In the alpha factor method it is the $\alpha_k^{(m)}$ and Q_i that are supplied to the analyst from industrial averages [18, 19, 24]. These data values then permit the computation of $Q_k^{(m)}$ as seen in Equation (5.45). The result is Equation (C.23) from NUREG-4780 Volume II [26].

Equation (5.45) can be written out in a more simple form which is shown Equation (5.46) as it appears identical to this particular equation shown in Table C-1 of the NUREG 4780 [26].

$$Q_k^{(m)} = \left(\frac{k}{(m-1)!} \right) \left(\frac{\alpha_k}{\alpha_i} \right) Q_i \quad (5.46)$$

Therefore, the example of three EDGs, the specific failure probabilities for independent, common-cause doublet, and common-cause triplet based off the alpha factors, are shown in Equations (5.47) through (5.48), respectively.

$$Q_1^{(3)} = \left(\frac{1}{(3-1)!} \right) \left(\frac{\alpha_1}{\alpha_i} \right) Q_i = \left(\frac{\alpha_1}{\alpha_i} \right) Q_i \quad (5.47)$$

$$Q_2^{(3)} = \left(\frac{2}{(3-1)!} \right) \left(\frac{\alpha_2}{\alpha_i} \right) Q_i = \left(\frac{\alpha_2}{\alpha_i} \right) Q_i \quad (5.48)$$

$$Q_3^{(3)} = \left(\frac{3}{(3-1)!} \right) \left(\frac{\alpha_3}{\alpha_i} \right) Q_i = 3 \left(\frac{\alpha_3}{\alpha_i} \right) Q_i \quad (5.49)$$

In terms of alpha factors for this case, α_i can be represented as shown in (5.50)

$$\alpha_t = \alpha_1 + 2\alpha_2 + 3\alpha_3 \quad (5.50)$$

Plugging in these values found in Equations (5.47)-(5.50) into Equation (5.33) will reveal the formula shown in (5.51).

$$Q_s = \left(\frac{\alpha_1}{\alpha_t}\right)^3 Q_t^3 + 3\left(\frac{\alpha_2}{\alpha_t}\right)\left(\frac{\alpha_1}{\alpha_t}\right) Q_t^2 + 3\left(\frac{\alpha_3}{\alpha_t}\right) Q_t \quad (5.51)$$

Equation (5.51), like in Equation (5.33), has the three terms on the right-hand side. These terms still represent respectively system failures of each of the three EDG trains, failures due to a common-cause doublet failure of one of the (three) pairs of EDGs accompanied by an independent failure of the third, and failures caused by a single common-cause triplet failure.

Subsection 5.3.f – Values Used in Industrial Average Case

The values presented from NUREG 6928 [24] show the industry-average failure probability or rate distribution for each type of component failure. The three component types that are included along with the associated mean failure, description, and designation between probability or rate is shown in Table 5.18.

Table 5.18

NUREG-6929 Component Type Failure Averages [24].

Component Failure Mode	Description	Designation	Mean Value
EDG STBY FTLR	Emergency Diesel Generator (Standby) Fail to Load and Run during the first hour of operation	Probability (Q_{t-FTLR})	2.90E-3
EDG STBY FTR>1H	Emergency Diesel Generator (Standby) Fail to Run after the first hour of operation	Rate (λ_{FTR})	8.48E-4
EDG STBY FTS	Emergency Diesel Generator (Standby) Fail to Start	Probability (Q_{t-FTS})	4.53E-3

The alpha factors are presented in Table 5.19 and acquired from the common-cause component group database [18, 19] based on a common-cause component group of three. It can be noted that the mean was used for the value of the alpha factor in this analysis. The α_i factors based off Equation (5.50) are also displayed within Table 5.19.

Table 5.19

Common-Cause Component Database Alpha Factors (CCCG=3) [18, 19].

Alpha Factors	EDG FLTR	EDG FTR	EDG FTS
α_1	0.989441	0.990122	0.988584
α_2	8.91E-3	6.36E-03	7.40E-03
α_3	1.65E-5	3.52E-03	4.01E-03
α_t	1.012211	1.013402	1.015414

After the values are extracted from [24] and [18, 19], then they can be used within Equation (5.51). Each term in Equation (5.51) can be used with a specific sequence of failures that will make up a cut set. For an example, a two way common-cause failure to load and run of the first two EDGS combined with a independent failure to run of the last EDG will have a certain probability, $Q_{2FTLR-1IFTR}$, associated with it is taken and is calculated using Equations (5.52) through (5.54). Please note that a complex example is used to show how the values and equations relate to each other. Equation (5.52) displays this basic definition of the $Q_{2FTLR-1IFTR}$ example using the variables found in Tables 5.19 and 5.20. The fail to run mean value in [24] displays a rate so it has to be converted to a probability. This exponential relationship is previously presented in Equation (5.21).

$$Q_{2FTLR-1IFTR} = 3 * ((\alpha_{2-FTLR} / \alpha_{t-FTLR})) * Q_{t-FTLR} * (1 - e^{-\lambda_{FTR} * (\alpha_{t-FTR} / \alpha_{t-FTLR}) * t}) \quad (5.52)$$

As one may note from Equation (5.52), the code employed in this verification uses the mutually exclusive version of the non-recovery integral seen in Equation (1.1). This differs from STPQUAD2 and STPQUAD3 verifications. However, these do not change the results significantly, more than the fifth digit.

As one can see the first line of Equation (5.52) relates to the common-cause-failure-doublet-to-load-and-run term, where the second line relates to the independent failure to run term. These probabilities used in the second line of Equation (5.52) can also be solved for using Tables 5.19 and 5.20 data. This turns Equation (5.52) into Equation (5.53).

$$Q_{2FTLR-1IFTR} = 3 * ((\alpha_{2-FTLR} / \alpha_{t-FTLR})) * Q_{t-FTLR} * (1 - e^{-[\lambda_{FTR} * (\alpha_{1-FTR} / \alpha_{t-FTR})]^t}) \quad (5.53)$$

Values can then be substituted into Equation (5.53). It can be solved as shown in Equation (5.54). Note this model assumes a 24 hour mission time so a failure to run after one hour will make $t=23$ in this analysis.

$$\begin{aligned} Q_{2FTLR-1IFTR} &= 3 * (((8.91E - 3) / 1.012211)) \\ &\quad * 2.90E - 3 * (1 - e^{-[0.000848 * (0.990122 / 1.013402)]^{23}}) \\ &= 2.170915E - 07 \end{aligned} \quad (5.54)$$

This is repeated for every cut set within the model. Table 5.20 shows the cut sets used for this verification along with their associated values. The cut sets are grouped together by type of failure that could have occurred. The summation total of the cut sets is the “model’s” CDF. In addition, this total CDF (displayed in yellow, the cell entitled “TOTAL CUT SETS” and the cell directly right of this particular cell) is divided to show the CDF (displayed in green, cells entitled “Total CC Triplet Cut Sets”, “Total CC Doublet Cut Sets”, and “Total Independent Cut Sets” along with the cells directly to the right of each) from independent, common-cause doublet, and common-cause triplet failures. The cut set numbers have no particular significance. The number is simply a naming convention or label system. These values displayed in Table 5.20 also commonly referred to in this document as the Pseudo-PRA model or analysis. The red cells, entitled “CC Triplet Failures”, “CC Doublet Failures”, and “Independent Failures”, refer to a new section of cutsets to differentiate between CC triplets, CC doublets, and independent failures, respectively.

Table 5.20

Cut Set Probabilities and Totals.

Cut set Cases	CC Triplet Failures	Probability
12	Q_{3FTS}	5.3669E-05
2	Q_{3FTR}	2.0322E-04
3	Q_{3FTLR}	1.4182E-05
	Total CC Triplet Cut Sets	=2.7107E-04
	CC Doublet Failures	
4	$Q_{2FTS-1FTR}$	1.8694E-06
5	$Q_{2FTS-1FIS}$	4.3679E-07
6	$Q_{2FTS-1FTLR}$	2.8075E-07
7	$Q_{2FTR-1FTR}$	6.9310E-06
8	$Q_{2FTR-1FTLR}$	1.0409E-06
9	$Q_{2FTR-1FIS}$	1.6194E-06
10	$Q_{2FTLR-1FTR}$	1.4455E-06
11	$Q_{2FTLR-1FTLR}$	2.1709E-07
12	$Q_{2FTLR-1FIS}$	3.3775E-07
	Total CC Doublet Cut Sets	=1.4179E-05
	Independent Failures	
13	$Q_{1FIS-1FTR-1FTLR}$	1.4159E-06
14	Q_{3IFTLR}	2.2780E-08
15	Q_{3IFIS}	8.5784E-08
16	Q_{3IFTR}	6.7251E-06
17	$Q_{2IFTLR-1FIS}$	1.0632E-07
18	$Q_{2IFTLR-1FTR}$	4.5504E-07
19	$Q_{2IFIS-1FTR}$	1.1014E-06
20	$Q_{2IFIS-1FTLR}$	1.6542E-07
21	$Q_{2IFTR-1FIS}$	4.7140E-06
22	$Q_{2IFTR-1FTLR}$	3.0300E-06
	Total Independent Cut Sets	=1.7822E-05
	TOTAL CUT SETS	=3.0307E-04

Using the numbers from this industrial average in the code itself leads to the results as shown in Table 5.21. This table displays the summations found from Table 5.20 along with the difference in results.

Table 5.21

Difference Between Pseudo-PRA and Code Results.

Probability of Critical Station Blackout	Pseudo-PRA Results	Code Results	Difference
Total CC Triplet Cut Sets	=2.7107E-04	=2.7107E-04	=4.0658E-17
Total CC Doublet Cut Sets	=1.4179E-05	=1.4179E-05	=1.0503E-18
Total Independent Cut Sets	=1.7822E-05	=1.7822E-05	=4.6079E-19
TOTAL CUT SETS	=3.0307E-04	=3.0307E-04	=4.0224E-17

The small difference between the Pseudo-PRA and the MATLAB code results indicates that the Pseudo-PRA results are fine for those cut sets. However, there is some deviation which could be due to error tolerance. It could also be round off error from inputs to the code or from the PRA. Therefore, the program was able to benchmark against site-specific numbers and industrial average values. There is some debate suggesting that the Pseudo-PRA values might be underestimating the value or there may be an error in calculations. Therefore, it may be advantageous to add in some conservatism to the final result. In practice, conservatisms are left in the model due to unknown requirements in the future, human error and uncertainties. These variations might be due to uncertainties in instruments used for measurement, lack of failure events or too small a sample size for the distribution, misclassified failure events, or even in the calculation itself.

The code can then be run with these results considering the different types of recovery and the results are shown in Table 5.22. Notice the cases are the same as Table 5.17 with the exception of Case B which is the Pseudo-PRA model results.

Table 5.22

Probabilities of CSBO from the Code Using the Industrial Average and Pseudo-PRA Model.

Case	Method	P_{CSBO3}	P_{CSBO2}	P_{CSBO1}	$P_{CSBO=}$ P_{CSBO3+} P_{CSBO2+} P_{CSBO1}
A	No credit for recovery (offsite or onsite), from STPQUAD3CCF, with failure probabilities and rates inferred from industrial average. ([18, 19, 24])	2.7107E-4	1.4179E-5	1.7822E-5	3.0307E-4
B	Pseudo-PRA Model totals from cutset summations. ([18, 19, 24])	2.7107E-4	1.4179E-5	1.7822E-5	3.0307E-4
C	Credit for offsite recovery only, computed dynamically via STPQUAD3ccf integration, with partial integrand for nonrecovery integral that is computed en route to full integrand in subroutine INTEGRANDv1pt0 and INTEGRAND3way (Lloyd-Anoba [11])	4.8724E-5	2.6495E-6	5.0312E-7	5.1876E-5
D	Computed dynamically via STPQUAD3CCF single integration, with integrand for nonrecovery integral that is fully dynamic for offsite recovery but incorporates onsite repair only after full failure of onsite power, per integrand computed via slight modification at last part of relevant INTEGRAND codes (Read-Fleming [14])	3.2798E-5	2.0350E-6	3.3867E-7	3.5172E-5
E	Full dynamic credit for recovery of offsite and onsite power, single integral evaluated per STPQUAD3CCF	3.2798E-5	1.9154E-6	2.2552E-7	3.4939E-5

The results clearly show that the credit for offsite recovery still vastly decreases the probability of critical station blackout when considering the industrial average (alpha method) case. From the original value to take credit for offsite recovery is about 82.88%. This is where the main reduction comes into play in recovery. If the systems now incorporate onsite repair only after full failure of onsite power, then the system will have an 88.39% reduction in probability of critical station blackout. Finally, if full dynamic recovery of offsite and onsite power is accounted for in the model, then the reduction is about 88.47% of the total. This small change between Case D and Case E does not seem substantial when looking at the total due to it only being an additional reduction of about 0.08%. This is due to CC Triplet failures making up a majority of the total risk and, as previously mentioned earlier in this chapter, a 3CCF shock will not matter if repair starts after all three have failed or after the first one failed. This is because in a 3CCF shock all of the EDGs/Trains fail immediately. This is partially the case in a 2CCF case; however, these results change slightly due to one independent failure that has to go along side of the common-cause doublet failure. Therefore, if the independent happens first, then there is some repair time before the 2CCF shock failure. One would expect independent failures to be affected the most by the time sequencing of repairs. If one looks at the reduction from Case D to Case F only looking at independent failures, then a notable reduction of 33.41% is noted. This could play more of a role in the overall probability of CSBO in two emergency train power plants as those would have a higher probabilistic risk based on independent failures.

This reduction satisfies the requirement of reducing the probability of critical station blackout in the model.

Section 5.4 –Summary of Verified Results

The last column of Table 5.17, Pcsbo, is the fully developed STP-TAMIL code results and is verified throughout this section. The code for each section is attached in Appendix A.3. Case E should be used as the most realistic result if recovery is assumed. Case A is the case most conservative when using the STP-TAMIL code. Case E reflects a

reduction of the total probability of critical station blackout of about 88.822%. This meets expectations and can greatly assist nuclear plants to predict potential outcomes.

For the industrial average case, the numbers were almost exactly as expected for no credit recovery or repair. Additionally, all of the cases went to probability levels that were expected, or decreased from adding recovery very similar to the STP-PRA Case. Case E for this model dropped the initial value by about 88.47%. This was not as large a decrease as in the site specific model but it was very close. This small variation is likely due to how that difference is calculated. The difference is the reduction of PCBO divided by the total PCSBO for no credit recovery (Case A). That means that if the numbers for Case A are different then it makes sense that a percentage change or reduction from those numbers would be slightly different. However, this exact change is not shown due to confidentially agreement with South Texas Project, see Appendix C.

CHAPTER VI IMPLEMENTATION

This chapter will discuss how this research code could be implemented into the PRA models at STPNOC (Section 6.1) as well as other nuclear power plants around the nation. (Section 6.2).

Section 6.1 –STPNOC

South Texas Project Nuclear Operating Company had supplied Texas A&M University with the associated values needed for the input deck of the STP-TAMIL code. The code was run and verified against STP's PRA Fault/Event-Tree model. With the result of the probability of critical station blackout that STPNOC chose, they can include it in their analysis by making it a factor of their CDF, core damage frequency. As the probability of a critical station blackout decreases, so does the CDF. This in turn leads to a lower overall risk for the probability of core damage during a LOOP initiating event. This process could be used to update their model for LOOP recovery with a more accurate way of quantifying the uncertainty and removing some of the unnecessary conservatism improving STPNOC's PRA model.

Section 6.2 –Other Commercial Nuclear Operators

Other commercial nuclear operators could also use this "TAMIL" code. It would not be difficult to update the input deck to make it site specific for a different nuclear plant. A few obvious changes may be number of trains, probabilities and factors used. This would involve optimizing the code for two trains. Once updated, it would be important to check and verify the results against the new plant's PRA Fault/Event-Tree model if possible. Nuclear plants with less than three trains of EDGs can also figure out how much recovery is gained in terms of the probability of critical station blackout by adding a third EDG to determine if it is reasonable and cost-effective to do.

CHAPTER VII

SUMMARY AND CONCLUSIONS

The final chapter will summarize the project (Section 7.1), discuss the significance of the problem (Section 7.2), and give suggestions for improvement (Section 7.3). Section 7.3 will also discuss suggestions for resolution of potential obstacles along with a plan for future action.

Section 7.1 –Summary

A MATLAB^b research code was developed and tested that incorporates and enhances a novel methodology to eliminate excessive conservatism in the evaluation of probability of critical station blackout in a nuclear power plant. The methodology employed involved direct numerical evaluation of multidimensional nonrecovery integrals using efficient computational algorithms. The results of the MATLAB research code was successfully verified against a traditional site-specific PRA analysis to about 0.12% accuracy. Furthermore this analysis confirmed that traditional PRA methods produced a probability of critical station blackout that was overly conservative by just over 88% assuming full recovery of offsite and onsite power.

This thesis provides an algorithm in the computer code “STP-TAMIL” that will compute the probability of critical station blackout following a loss of an offsite power initiating event. The concept was introduced in Chapter I as were differences in current versus new methods of using the non-recovery integral. The theory of forming the non-recovery integral along with what the specific parts represent was shown in Chapter III. The theory involved approaching the non-recovery integral in terms of two asymptotic methods. The first was successful in leading to the non-recovery integral and the other was left to future work. Chapter V provides verifications to prove that the non-recovery integral was representative of the current methods within the industry as well as to show

^b MATLAB is a registered trademark of The MathWorks, Inc.; cf. MathWorks - Trademarks

the improvements. Chapter V also has a verification of the results of STP-TAMIL against an industry average. This verification shows the results in the final form of STP-TAMIL and gives the fully developed probability of critical station blackout based upon industry average numbers and the specific input deck used. Chapter IV displayed how values obtained from STP-TAMIL may differ depending on what inputs were used and what assumptions were made. Chapter II is intended to be a stand-alone user manual for STP-TAMIL operation. The manual is meant to cover installation, creating an input deck, creating new files or distributions, running the code, and reading the output file. Chapter VI describes how commercial nuclear power plant operators can benefit from an improved estimation of core damage risk during a loss of offsite power. It also serves to describe how this program can be implemented at other plants and how to make the code site specific.

Section 7.2 –Significance

This work proves that there is a large reduction in probability of critical station blackout using different types of recovery. For a three EDG model, if one just assumes dynamic offsite power repair is possible, then there is about an 82.88% drop in probability off industry averages. From that assuming that repair of the EDGs can start after all have failed, the probability drops by an additional 5.51% making it a total of a 88.39% drop. Assuming that repair can start on those EDGs when each one fails, the total probability of critical station blackout drops another 0.08%. This makes the total drop in probability of critical station blackout of about 88.47% assuming full recovery is taken into account. In the verification of STPQUAD2 case, the drop from “Case D” to “Case F” (referred to later in Chapter V) was much more significant and was around about 10%. However, this is due to common-cause shock events not being taken into account as different between Case D and Case F, which is correct. However, the difference in independent events between these two cases is about 33.4% so there is a decent reduction, which will have a bigger impact on overall risk in a two emergency train nuclear power plants.

This large reduction of probability of critical station blackout shows that there is a significant over conservatism built into current models that can be adjusted for increased accuracy. This will provide more meaningful results in PRA models and can better prove engineering assumptions of the current fleet of commercial nuclear power plants. The NRC may change the assumed amount of critical time allowable or the mission time for the specific initiating event. This code can be used to plot how changing these times will affect the probability of critical station blackout.

There is a large push in the industry to have more of a production code that can be used throughout the fleet. It is hoped that with more verification and validation, this can be done.

Section 7.3 –Future Work

Additional common-cause data (e.g., site-specific industry data), independent of those extracted from the plant-specific STP PRA estimates, should be used for validation. If the dynamic reliability methodology developed in the present study is to be used going forward, the recommendation is that further development of such data should be a high priority. A more intensive two-EDG model could also be used for the development of the asymptotic approach. There are opportunities to expand the usage of the equations in Chapter III, which could be the focus of further research. The is most notable in the advanced three-EDG Markov model that is mentioned in Chapter III. Future work is also needed in small impacts of error tolerance and in determining expanded integral usage. For the verification problems, no differentiation is made between the various sources of types of loss of offsite power, as ultimately will be necessary. Some of these differences could include the LOOP being caused by weather, the grid, or the switchyard failure [23] which will have to be addressed in future work.

Fully dynamic repair rates based on factors that might affect the specific type of repair of an individual component may be developed through new research. This may lead to improvement in STP-TAMIL. The values presented in Chapter V can be used in other applications that are unforeseen by the author; therefore, a large number of significant figures have been left for use in future calculations. It may also be interesting

to see how dynamic onsite recovery or repair of one EDG while the other remains operating, applies to other nuclear plants. This could be open to future work to see if fully dynamic recovery can be better at other types of plants.

The impact of various repair policies may merit some additional study. To some extent that could be done using the present code; although that would give a “feel” for the impact of different repair policies, but the results may not be accepted by regulatory authorities. Also, a systematic approach to truncation in PRAs could be developed, on the basis of the “Asymptotic Analysis” from Chapter III. The conservatism built into the industrial average will also have to be accounted for and therefore will require the need for future analysis and research.

With continued work, the code may eventually evolve into a production version for commercial nuclear industry applications. This may involve adding new methods for looking at distributions for specific plants. Provisions for expansion have been built into the code as covered in Chapter II. After updates, the code should eventually lead to more versions that fully benchmark to PRA models used at the nuclear sites.

REFERENCES

- [1] S. Rodgers, C. Betancourt, E. Kee, P. Neslon, P. Rodi, "Toward Quantification of the Uncertainty in Estimating Frequency of Critical Station Blackout," Proceedings M&C 2011, International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, May 8-12, 2011, Rio de Janeiro, published in CD format, ISBN 978-85-63688-00-2.
- [2] Blake, E. M. (2010, March). The world list undergoes some changes. Nuclear News, 53(3), 45-73.
- [3] U.S. Nuclear Regulatory Commission Regulations; Title 10, Code of Federal Regulations part 50.71 (h)(1). October 4, 1999. <http://www.nrc.gov/reading-rm/doc-collections/cfr/part050/part050-0071.html>.
- [4] U.S. Nuclear Regulatory Commission. (2011). "Reactor Concepts Manual: Pressurized Water Systems". Retrieved August 18, 2011. <http://www.nrc.gov/reading-rm/basic-ref/teachers/04.pdf>.
- [5] S.A. Eide, C. D. Gentillon, T. E. Wierman and D. M. Rasmuson, "Reevaluation of Station Blackout Risk at Nuclear Power Plants," Report No. NUREG/CR-6890, Vol. 2, December 2005, US Nuclear Regulatory Commission, esp. Section III and Tables 3-1 and 3-2.
- [6] S.A. Eide, C. D. Gentillon, T. E. Wierman and D. M. Rasmuson, "Reevaluation of Station Blackout Risk at Nuclear Power Plants," Report No. NUREG/CR-6890, Vol. 1, December 2005, US Nuclear Regulatory Commission.
- [7] C. Miller, A. Cabbage, D. Dorman, J. Grobe, G. Holahan, et al., "Recommendations for Enhancing Reactor Safety in the 21st Century – The near-term task force review of insights for the Fukushima Dai-ichi accident" July 12, 2011. U.S. Nuclear Regulatory Commission, esp "Recommendation 9".
- [8] S.A. Eide, C. D. Gentillon, T. E. Wierman and D. M. Rasmuson, "Reevaluation of Station Blackout Risk at Nuclear Power Plants," Report No. NUREG/CR-6890, Vol. 2, December 2005, US Nuclear Regulatory Commission, esp. Section III and Table 3-3.
- [9] P. Samanta, I. Kim, S. Uryasev, J. Penoyar, W. Vesely "Emergency Diesel Generator: Maintenance and Failure Unavailability, and Their Risk Impacts," Report No. NUREG/CR-5994, November 1994, US Nuclear Regulatory Commission.
- [10] S.A. Eide, C. D. Gentillon, T. E. Wierman and D. M. Rasmuson, "Reevaluation of Station Blackout Risk at Nuclear Power Plants," Report No. NUREG/CR-6890, Vol. 2, December 2005, US Nuclear Regulatory Commission, esp. Table 4-2.

- [11] M. Lloyd and R. C. Anoba, "A Convolution Approach to Account for Time Dependencies in Severe Accident Cut sets," Procs. ANS International Topical Meeting on the Safety of Thermal Reactors, Portland, Oregon, July 21-25, pp. 29-36 (1991).
- [12] N. Siu, "Risk assessment for dynamic systems: An overview." *Reliab Eng Saf* 1994;43;43-73
- [13] M. Lloyd, "Treatment of Time Interdependencies in Fault Tree Generated Cut set Results," EPRI Report No. 1009187, Palo Alto, CA, 2003.
- [14] J. W. Read and K. N. Fleming, "Electric Power Recovery Models," Procs. PSA '93, Clearwater Beach, FL, January 26-29, Vol. 1, pp. 631-640 (1993).
- [15] S. Rodgers, C. Betancourt, E. Kee, F. Yilmaz, P. Nelson, "Integrated Power Recovery using Markov Modeling," Procs. 18th International Conference on Nuclear Engineering, Xi'an, China, May 17-21 (2010).
- [16] MATLAB R2011b Documentation. Mathworks, Cambridge MA. 2011.
- [17] P. Nelson and S. Wang, "Dynamic Reliability via computational solution of generalized state-transition equations for entry-time processes," *Reliability Engineering & System Safety* Vol. 92, No 9. August 2006. Pp (1281-1293)
- [18] T.E. Wierman, D.M. Rasmuson, A. Mosleh, "Common-Cause Failure Database and Analysis System: Event Data Collection, Classification, and Coding," Report No. NUREG/CR-6268, Rev. 1, September 2007, U.S. Nuclear Regulatory Commission.
- [19] U.S. Nuclear Regulatory Commission, "CCF Parameter Estimations, 2009 Update", <http://nrcoe.inl.gov/results/CCF/ParamEst2009/ccfparamest.htm>, April 2011.
- [20] L.T. Dauer, P. Zanzonico, R.M. Tuttle, D.M. Quinn, H.W. Strauss et al., "The Japanese Tsunami and Resulting Nuclear Emergency at the Fukushima Daiichi Power Facility: Technical, Radiologic, and Response Perspectives," *The Journal of Nuclear Medicine*. Vol. 52, No. 9. September 2011.
- [21] A.C. Kadak, "A future for nuclear energy: pebble bed reactors," *International Journal of Critical Infrastructures*. Vol. 1, No.4. 2005. Pp. 330 – 345

- [22] E. Kee, F. Yilmaz and P. Nelson, "A Comparative Study of Methodologies for Determining Plant Availabilities and Initiator Frequencies," Proceedings of ICONE 17, ASME 2009 International Conference on Nuclear Engineering July 12-16, 2009, Brussels, Belgium.
- [23] C. L. Atwood, D. A. Prawdzik, D. L. Kelly, F. M. Marshall, J. W. Stetkar, "Evaluation of Loss of Offsite Power Events at Nuclear Power Plants: 1980-1996," Report No. NUREG/CR-5496, November 1998, U.S. Nuclear Regulatory Commission.
- [24] S.A. Eide, T.E. Wierman, C.D. Gentillon, D.M. Rasmuson, C.L. Atwood, "Industry-Average Performance for Components and Initiating Events at U.S. Commercial Nuclear Power Plants," Report No. NUREG/CR-6928, February 2007, U.S. Nuclear Regulatory Commission.
- [25] A. Mosleh, K.N. Fleming, G.W. Parry, H.M. Paula, D.H. Worledge, et al., "Procedures for Treating Common Cause Failures in Safety and Reliability Studies: Procedural Framework and Examples," Report No. NUREG/CR-4780 Vol. 1, January 1988, U.S. Nuclear Regulatory Commission.
- [26] A. Mosleh, K.N. Fleming, G.W. Parry, H.M. Paula, D.H. Worledge, et al., "Procedures for Treating Common Cause Failures in Safety and Reliability Studies: Analytical Background and Techniques," Report No. NUREG/CR-4780 Vol. 2, January 1989, U.S. Nuclear Regulatory Commission.

APPENDIX A

STP-TAMIL CODE

The MATLAB code is broken up into a series of M-Files that make it easier for the programmer. These are further broken up to the specific verification study that the code relates too. For verification 1 seen in Section 5.1 for STPQUAD2 suite, the specific M-file, description, and page number are summarized in Table I.1. For verification 2 seen in Section 5.2 for STPQUAD3 suite, the specific M-file, description, and page number are summarized in Table I.2. For the final code or verification for STPQUAD3CCF, seen in Section 5.3, the specific M-file, description, and page number are summarized in Table I.3. Note that the STPQUAD3CCF suite of programs is STP-TAMIL version 1.0.

Table I.1

M-File Names and Descriptions for STPQUAD2 Suite Verification

M-File	Description	Page
STPQUAD2	Main Program, used to run the code, See comments in code for a more thorough description.	144
Lloyd	Code used to Verify Lloyd Approximation [13]	145
INTEGRAND	Used in STPQUAD2 to run the non-recovery integral	146
INTEGRAND1	Used in Lloyd to run the Lloyd approximation to nonrecovery integral	147

Table I.2

M-File Names and Descriptions for STPQUAD3 Verification.

M-File	Description	Page
STPQUAD3	Main Program, used to run the code, See comments in code for a more thorough description.	148
Lloyd3	Code used to Verify the Lloyd Approximation [13]	150
INTEGRANDv1pt0	Used in STPQUAD3 to run the non-recovery integral	152
INTEGRAND3	Used in Lloyd3 to run the Lloyd approximation to nonrecovery integral	153
FTR6890	Custom Distribution Type made up of a cdf and an inverse cdf to calculate the cdf for a specific train for a failure to load and run	154
FTR6890inv	Custom Distribution Type made up of a cdf and an inverse cdf to calculate the inverse cdf for a specific train for a failure to load and run	154
FTR6890pdf	Custom Distribution Type made up of a cdf and an inverse cdf to calculate the inverse cdf for a specific train for a failure to load and run. (Not used)	155

Table I.3

M-File Names and Descriptions for STP-TAMIL* (STPQUAD3CCF Verification).

M-File	Description	Page
STPQUAD3CCF	Main Program, used to run the code, See comments in code for a more thorough description.	155
INTEGRANDv1pt0	Used in STPQUAD3 to run the non-recovery integral	182
INTEGRAND2way	Used in STPQUAD3CCF to run the integrals for common-cause doublet failures as well as repairs	184
INTEGRAND3way	Used in STPQUAD3CCF to run the integrals for common-cause triplet failures as well as repairs	187
FTR6890	Custom Distribution Type made up of a cdf and an inverse cdf to calculate the cdf for a specific train for a failure to load and run	190
FTR6890inv	Custom Distribution Type made up of a cdf and an inverse cdf to calculate the inverse cdf for a specific train for a failure to load and run	191
FTR6890pdf	Custom Distribution Type made up of a cdf and an inverse cdf to calculate the inverse cdf for a specific train for a failure to load and run. (Not used)	193

* For STP-TAMIL to work in this work an associated RunFile and Output File must be created in addition to the m-files as described in Chapter II.

Appendix A.1 – STPQUAD2 Package Code

The MATLAB code title is in bold and everything following is the actual code that can be used in a MATLAB environment. Each specific code is broken up by these bold titles.

“STPQUAD2.m”:

```

%% Begin input
clear all
global F_0 F Finv alphaF betaF gammaF n T_c G alphaG betaG gammaG R tau
alphaR betaR gammaR
n=2; % number of safety trains;
F=cell(1,n);
Finv=cell(1,n);
R=cell(1,n);
T=24; % mission time, in hours
T_c=1; % critical time, in hours
G=@expcdf; % cdf in time (hours) for recovery of offsite power
alphaG=4; betaG=0; gammaG=0; % parameters for G, here mean time to
recovery (hours) for
% expcdf as in ICONE18 paper
%G=@logncdf; % cdf in time (hours) for recovery of offsite power, taken
here as lognormal,
% in consonance with recommendation of Chapter 4 of Vol. 1 of NUREG
6890, 2005
%alphaG=0.793; betaG=1.982; gammaG=0; % parameters for G, here mean
timeto recovery (hours) for
% lognormalcdf as in Table 4-1 of NUREG 6890, ibid.
%
F_0(1)=.0132; % probability of initial failure to start of train i,
from ICONE18 paper
F{1}=@expcdf; % cdf for failure of train 1; here the exponential cdf,
per the ICONE18 paper
Finv{1}=@expinv; % the corresponding inverse cdf for failure of train
1; here again the inverse
% of the exponential cdf,per the ICONE18 paper
alphaF(1)=1/8.35e-4; betaF(1)=0; gammaF(1)=0; % parameters for failure
cdf and its inverse, for
% train 1, here mean time to failure = 1/8.35e-4 hours for expcdf as in
ICONE18 paper
R{1}=@expcdf; % cdf in time (hours) for recovery (repair) of train i
alphaR(1)=12; betaR(1)=0; gammaR(1)=0; % parameters for recovery cdf
for train 1, here the mean time to
% recovery = 12 hours for expcdf as in ICONE18 paper
F_0(2)=.0132; % probability of initial failure to start of train i,
from ICONE18 paper
F{2}=@expcdf; % cdf for failure of train 2; here the exponential cdf,
per the ICONE18 paper
Finv{2}=@expinv; % the corresponding inverse cdf for failure of train
1; hereagain the inverse
% of the exponential cdf,per the ICONE18 paper
alphaF(2)=1/8.35e-4; betaF(2)=0; gammaF(2)=0; % parameters for failure
cdf and its inverse, for
% train 2, here mean time to failure = 1/8.35e-4 hours for expcdf as in
ICONE18 paper
R{2}=@expcdf; % cdf in time (hours) for recovery (repair) of train 2
alphaR(2)=12; betaR(2)=0; gammaR(2)=0; % parameters for recovery cdf
for train 2, here the mean time to
% recovery = 12 hours for expcdf as in ICONE18 paper
%%

```

```

%% Computation of nonrecovery integral
% Convert Riemann-Stieltjes double nonrecovery integral to ordinary
Riemann
% integral
A=0;
B=F_0(1)+(1-F_0(1))*feval(F{1},T,alphaF(1),betaF(1),gammaF(1));
C=0;
D=F_0(2)+(1-F_0(2))*feval(F{2},T,alphaF(2),betaF(2),gammaF(2));
P_CSBO=quad2d(@INTEGRAND,A,B,C,D)
%P_CSBO=quad2d(@INTEGRAND,A,B,C,D,'AbsTol',1.0e-13)
%P_CSBO=quad2d(@INTEGRAND,A,B,C,D,'AbsTol',1.0e-15,'MaxFunEvals',20000)

```

“Lloyd.m”:

```

%% Begin input
clear all
global f F_0 F alphaF betaF gammaF n T_c G alphaG betaG gammaG Rhat
n=2; % number of safety trains;
f=cell(1,n);
F=cell(1,n);
R=cell(1,n);
T=24; % mission time, in hours
T_c=1; % critical time, in hours
G=@expcdf; % cdf in time (hours) for recovery of offsite power
alphaG=4; betaG=0; gammaG=0; % parameters for G, here mean time to
recovery (hours) for
% expcdf as in ICONE18 paper
%
F_0(1)=.0132; % probability of initial failure to start of train i,
from ICONE18 paper
f{1}=@exppdf; % pdf for failure of train 1; here the exponential pdf,
per the ICONE18 paper
F{1}=@expcdf; % cdf for failure of train 1; here the exponential cdf,
per the ICONE18 paper
alphaF(1)=1/8.35e-4; betaF(1)=0; gammaF(1)=0; % parameters for failure
pdf, cdf and its inverse, for
% train 1, here mean time to failure = 1/8.35e-4 hours for expcdf as in
% ICONE18 paper
%Rhat(1)=1.0; % nonrecovery factor for train 1; here no credit taken
for onsite recovery
R{1}=@expcdf;
alphaR(1)=12; betaR(1)=0; gammaR(1)=0; % parameters for recovery cdf
for train 1, here the mean time to
% recovery = 12 hours for expcdf as in ICONE18 paper
Rhat(1)=1-feval(R{1},T_c,alphaR(1),betaR(1),gammaR(1)); %nonrecovery
factor for train 1;
% here as appropriate to exponential cdf in the Read-fleming version of
the
% Lloyd-Anoba approach
F_0(2)=.0132; % probability of initial failure to start of train i,
from ICONE18 paper
f{2}=@exppdf; % pdf for failure of train 2; here the exponential pdf,
per the ICONE18 paper
F{2}=@expcdf; % cdf for failure of train 2; here the exponential cdf,
per the ICONE18 paper

```

```

alphaF(2)=1/8.35e-4; betaF(2)=0; gammaF(2)=0; % parameters for failure
pdf, cdf and its inverse, for
% train 2, here mean time to failure = 1/8.35e-4 hours for expcdf as in
ICONE18 paper
%Rhat(2)=1.0; % nonrecovery factor for train 2; here no credit taken
for onsite recovery
R{2}=@expcdf;
alphaR(2)=12; betaR(2)=0; gammaR(2)=0; % parameters for recovery cdf
for train 1, here the mean time to
% recovery = 12 hours for expcdf as in ICONE18 paper
Rhat(2)=1-feval(R{2},T_c,alphaR(2),betaR(2),gammaR(2)); %nonrecovery
factor for train 2;
% here as appropriate to exponential cdf in the Read-fleming version of
the
% Lloyd-Anoba approach
%%

%% Computation of Lloyd approximation to nonrecovery integral
startpart=(1-G(T_c,alphaG,betaG,gammaG))*F_0(1)*F_0(2);
runpart=quad(@INTEGRAND1,0,T,1.0e-16);
%runpart=quad(@INTEGRAND1,0,T)
% Following assumes identical EDGs
P_CSBO=Rhat(1)*Rhat(2)*(startpart+runpart)

```

“INTEGRAND.m”:

```

function H=INTEGRAND(F1,F2)
global F_0 F Finv alphaF betaF gammaF n G T_c G alphaG betaG gammaG R
tau alphaR betaR gammaR
tau=cell(1,n);

%%
% Identify corresponding times, tau1, tau2
C1=F1<F_0(1);
[m1 n1]=size(F1);
for i=1:m1
    for j=1:n1
        if C1(i,j)==1
            tau{1}(i,j)=0;
        else
            tau{1}(i,j)=feval(Finv{1,1},((F1(i,j)-F_0(1))./(1-
F_0(1))),alphaF(1),betaF(1),gammaF(1));
        end
    end
end
C2=F2<F_0(2);
[m2 n2]=size(F2);
for i=1:m2
    for j=1:n2
        if C2(i,j)==1
            tau{2}(i,j)=0;
        else
            tau{2}(i,j)=feval(Finv{1,2},((F2(i,j)-F_0(2))./(1-
F_0(2))),alphaF(2),betaF(2),gammaF(2));
        end
    end
end

```

```

        end
    end
end
H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%
% Determine taumax
taumax=tau{1};
for i=2:n
    taumax=max(taumax,tau{i});
end
% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio paper
H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG));
% Now take credit for recovery of emergency power trains
for i=1:n
% Take full credit for recovery of emergency power trains
    H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i)));
% Alternately take only credit for recovery of emergency power trains
that
% comes from repair work following full failure of onsite power (per
% Lloyd-Anoba, for the case coincident with Read-Fleming)
    %H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i)));
end

```

“INTEGRAND1.m”:

```

function H=INTEGRAND1(taumax)
global f F_0 F alphaF betaF gammaF n G T_c G alphaG betaG gammaG Rhat

[m1 n1]=size(taumax);
H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper.

% Credit for offsite recovery only following SBO
H=H.*(1-feval(G,T_c,alphaG,betaG,gammaG));
% Now take credit for dynamic recovery of offsite power, as in Section
3.4 of Rio paper
%H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG));

% Now probability others failed previously
%Hsum=feval(f{1},taumax,alphaF(1),betaF(1),gammaF(1)).*feval(F{2},tauma
x,alphaF(2),betaF(2),gammaF(2));
Hsum1=(1-F_0(1)).*feval(f{1},taumax,alphaF(1));

```



```

Hsum1=Hsum1.*(F_0(2).*ones(m1,n1)+(1-
F_0(2)).*feval(F{2},taumax,alphaF(2),betaF(2),gammaF(2)));
%Hsum=Hsum+feval(f{2},taumax,alphaF(2),betaF(2),gammaF(2)).*feval(F{1},
taumax,alphaF(1),betaF(1),gammaF(1));
Hsum2=(1-F_0(2)).*feval(f{2},taumax,alphaF(2));
Hsum2=Hsum2.*(F_0(1).*ones(m1,n1)+(1-
F_0(1)).*feval(F{1},taumax,alphaF(1),betaF(1),gammaF(1)));
Hsum=Hsum1+Hsum2;
H=H.*Hsum;

```

Appendix A.2 – STPQUAD3 Package Code

The MATLAB code title is in bold and everything following is the actual code that can be used in a MATLAB environment. Each specific code is broken up by these bold titles.

“STPQUAD3.m”:

```

%% Begin input
clear all
global F_0 F Finv alphaF betaF gammaF n T_c G alphaG betaG gammaG R tau
alphaR betaR gammaR
n=3; % number of safety trains;
F=cell(1,n);
Finv=cell(1,n);
R=cell(1,n);
T=24; % mission time, in hours
T_c=1; % critical time, in hours

G=@logncdf; % cdf in time (hours) for recovery of offsite power, here
taken as lognormal cdf
% per Word document "Verification 2"
alphaG=0.300; betaG=1.064; gammaG=0; % parameters for G, here
respectively mean time to
% recovery (hours) and standard deviation for the associated normal
distribution, as in
% the Word document "Verification 2"

F_0(1)=.014; % probability of initial failure to start of train i, per
"Verification 2".
F{1}=@FTR6890; % cdf for failure of train 1 to start and load; here a
piecewise
% exponential cdf, per "Verification 2"
Finv{1}=@FTR6890inv; % the corresponding inverse cdf for failure of
train 1 to start and load; here
% the inverse of the piecewise exponential function coded into FTR6890,
per
% "Verification 2"

```

```

alphaF(1)=400; betaF(1)=1; gammaF(1)=1250; % parameters for cdf for
failure to run, here
% alphaF = mu1 = exponential mean up to time betaF = cutoff time, and
gammaF = subsequent
% exponential mean conditional on reaching the cutoff time, numerical
% values per "Verification 2"
R{1}=@wblcdf; % cdf in time (hours) for recovery (repair) of train 1.
Here the Weibull
% cdf, per "Verification 2"
alphaR(1)=15.5; betaR(1)=0.739; gammaR(1)=0; % parameters for recovery
cdf for train 1, here
% the respective values of the scale and shape parameters for a Weibull
% distribution, as detailed in "Verification 2" (alphaR in hours)

F_0(2)=.014; % probability of initial failure to start of train 2, per
"Verification 2".
F{2}=@FTR6890; % cdf for failure of train 2 to start and load; here a
piecewise
% exponential cdf, per "Verification 2"
Finv{2}=@FTR6890inv; % the corresponding inverse cdf for failure of
train 2 to start and load; here
% the inverse of the piecewise exponential function coded into FTR6890,
per
% "Verification 2"
alphaF(2)=400; betaF(2)=1; gammaF(2)=1250; % parameters for cdf for
failure to run, here
% alphaF = mu1 = exponential mean up to time betaF = cutoff time, and
gammaF = subsequent
% exponential mean conditional on reaching the cutoff time, numerical
% values per "Verification 2"
R{2}=@wblcdf; % cdf in time (hours) for recovery (repair) of train 2.
Here the Weibull
% cdf, per "Verification 2"
alphaR(2)=15.5; betaR(2)=0.739; gammaR(2)=0; % parameters for recovery
cdf for train 2, here
% the respective values of the scale and shape parameters for a Weibull
% distribution, as detailed in "Verification 2" (alphaR in hours)

F_0(3)=.014; % probability of initial failure to start of train 3, per
"Verification 2".
F{3}=@FTR6890; % cdf for failure of train 3 to start and load; here a
piecewise
% exponential cdf, per "Verification 2"
Finv{3}=@FTR6890inv; % the corresponding inverse cdf for failure of
train 3 to start and load; here
% the inverse of the piecewise exponential function coded into FTR6890,
per
% "Verification 2"
alphaF(3)=400; betaF(3)=1; gammaF(3)=1250; % parameters for cdf for
failure to run, here
% alphaF = mu1 = exponential mean up to time betaF = cutoff time, and
gammaF = subsequent
% exponential mean conditional on reaching the cutoff time, numerical
% values per "Verification 2"
R{3}=@wblcdf; % cdf in time (hours) for recovery (repair) of train 3.
Here the Weibull

```

```

% cdf, per "Verification 2"
alphaR(3)=15.5; betaR(3)=0.739; gammaR(3)=0; % parameters for recovery
cdf for train 3, here
% the respective values of the scale and shape parameters for a Weibull
% distribution, as detailed in "Verification 2" (alphaR in hours)
%%

%
%% Computation of nonrecovery integral
% Convert Riemann-Stieltjes triple nonrecovery integral to ordinary
Riemann
% integral
A=0;
B=F_0(1)+(1-F_0(1))*feval(F{1},T,alphaF(1),betaF(1),gammaF(1));
C=0;
D=F_0(2)+(1-F_0(2))*feval(F{2},T,alphaF(2),betaF(2),gammaF(2));
E=0;
F=F_0(3)+(1-F_0(3))*feval(F{3},T,alphaF(3),betaF(3),gammaF(3));
P_CSBO=triplequad(@INTEGRANDv1pt0,A,B,C,D,E,F,1.0e-10) % Recommended
absolute error tolerance
% is 1.0e-10. See "Verification 2" for the basis for this
recommendation.

```

“Lloyd3.m”:

```

%% Begin input
clear all
global f F_0 F alphaF betaF gammaF n T_c G alphaG betaG gammaG Rhat
n=2; % number of safety trains;
f=cell(1,n);
F=cell(1,n);
R=cell(1,n);
T=24; % mission time, in hours
T_c=1; % critical time, in hours
%G=@Gzero; % Set G=0; no credit for offsite recovery; aid in debugging
G=@logncdf; % cdf in time (hours) for recovery of offsite power
alphaG=0.300; betaG=1.064; gammaG=0; % parameters for G, here mean and
standard deviation
% (natural log hours) of associated normal distribution
%

F_0(1)=.014; % probability of initial failure to start of train i, from
% Table 4.2 (p. 24) of Vol. 2 of NUREG/CR-6890
%%
f{1}=@FTR6890pdf; % pdf for failure of train 1; here the two-regime
% exponential pdf, per "Verification 2"
F{1}=@FTR6890; % cdf for failure of train 1; here the two-regime
% exponential pdf, per "Verification 2"
alphaF(1)=1/2.5e-3; betaF(1)=1; gammaF(1)=1/8.0e-4; % parameters for
failure pdf, cdf and its inverse, for
% train 1, here mean time to failure prior to cutoff, cutoff (hours)
and mean time to failure
% after cutoff, as in "Verification 2," all times in hours
%%
%Rhat(1)=1.0; % nonrecovery factor for train 1; here no credit taken
for onsite recovery

```

```

R{1}=@wblcdf; % cdf in time (hours) for recovery (repair) of train 1.
Here the Weibull
% cdf, per "Verification 2"
alphaR(1)=15.5; betaR(1)=0.739; gammaR(1)=0;
% parameters for recovery cdf for train 1, here
% the respective values of the scale and shape parameters for a Weibull
% distribution, as detailed in "Verification 2" (alphaR in hours)
Rhat(1)=1-feval(R{1},T_c,alphaR(1),betaR(1),gammaR(1)); %nonrecovery
factor for train 1;
% here as appropriate to Weibull cdf in the Read-Fleming version of the
% Lloyd-Anoba approach
%%
F_0(2)=.014; % probability of initial failure to start of train i, from
% Table 4.2 (p. 24) of Vol. 2 of NUREG/CR-6890
f{2}=@FTR6890pdf; % pdf for failure of train 1; here the two-regime
% exponential pdf, per "Verification 2"
F{2}=@FTR6890; % cdf for failure of train 1; here the two-regime
% exponential pdf, per "Verification 2"
alphaF(2)=1/2.5e-3; betaF(2)=1; gammaF(2)=1/8.0e-4; % parameters for
failure pdf, cdf and its inverse, for
% train 1, here mean time to failure prior to cutoff, cutoff (hours)
and mean time to failure
% after cutoff, as in "Verification 2," all times in hours
%%
Rhat(2)=1.0; % nonrecovery factor for train 1; here no credit taken
for onsite recovery
R{2}=@wblcdf; % cdf in time (hours) for recovery (repair) of train 1.
Here the Weibull
% cdf, per "Verification 2"
alphaR(2)=15.5; betaR(2)=0.739; gammaR(2)=0;
% parameters for recovery cdf for train 2, here
% the respective values of the scale and shape parameters for a Weibull
% distribution, as detailed in "Verification 2" (alphaR in hours)
Rhat(2)=1-feval(R{2},T_c,alphaR(2),betaR(2),gammaR(2)); %nonrecovery
factor for train 1;
% here as appropriate to Weibull cdf in the Read-Fleming version of the
% Lloyd-Anoba approach
%%
F_0(3)=.014; % probability of initial failure to start of train i, from
% Table 4.2 (p. 24) of Vol. 2 of NUREG/CR-6890
f{3}=@FTR6890pdf; % pdf for failure of train 1; here the two-regime
% exponential pdf, per "Verification 2"
F{3}=@FTR6890; % cdf for failure of train 1; here the two-regime
% exponential pdf, per "Verification 2"
alphaF(3)=1/2.5e-3; betaF(3)=1; gammaF(3)=1/8.0e-4; % parameters for
failure pdf, cdf and its inverse, for
% train 1, here mean time to failure prior to cutoff, cutoff (hours)
and mean time to failure
% after cutoff, as in "Verification 2," all times in hours
% HERE; must change Rhat as below but with new recovery in order to
take
% full credit for on-site recovery
Rhat(3)=1.0; % nonrecovery factor for train 3; here no credit taken
for onsite recovery
R{3}=@wblcdf; % cdf in time (hours) for recovery (repair) of train 1.
Here the Weibull
% cdf, per "Verification 2"

```

```

alphaR(3)=15.5; betaR(3)=0.739; gammaR(3)=0;
% parameters for recovery cdf for train 2, here
% the respective values of the scale and shape parameters for a Weibull
% distribution, as detailed in "Verification 2" (alphaR in hours)
Rhat(3)=1-feval(R{3},T_c,alphaR(3),betaR(3),gammaR(3)); %nonrecovery
factor for train 1;
% here as appropriate to Weibull cdf in the Read-Fleming version of the
% Lloyd-Anoba approach
%%
% Computation of Lloyd approximation to nonrecovery integral
startpart=(1-G(T_c,alphaG,betaG,gammaG))*F_0(1)*F_0(2)*F_0(3);
%runpart=quadl(@INTEGRAND3,0,T);
runpart=quadl(@INTEGRAND3,0,T,1.0e-10); % tolerance of 1.0e-10
%recommended
%runpart=quad(@INTEGRAND3,0,T,1.0e-14);
%runpart=quad(@INTEGRAND3,0,T);
% Following assumes identical EDGs ??Why
P_CSBO=Rhat(1)*Rhat(2)*Rhat(3)*(startpart+runpart)

```

“INTEGRANDv1pt0.m”:

```

function H=INTEGRANDv1pt0(F1,F2,F3)
global F_0 F Finv alphaF betaF gammaF n G T_c G alphaG betaG gammaG R
tau alphaR betaR gammaR
tau=cell(1,n);

%%
% Identify corresponding times, tau1, tau2, tau3
C1=F1<F_0(1);
[m1 n1]=size(F1);
for i=1:m1
    for j=1:n1
        if C1(i,j)==1
            tau{1}(i,j)=0;
        else
            tau{1}(i,j)=feval(Finv{1,1},((F1(i,j)-F_0(1))./(1-
F_0(1))),alphaF(1),betaF(1),gammaF(1));
        end
    end
end
C2=F2<F_0(2);
[m2 n2]=size(F2);
for i=1:m2
    for j=1:n2
        if C2(i,j)==1
            tau{2}(i,j)=0;
        else
            tau{2}(i,j)=feval(Finv{1,2},((F2(i,j)-F_0(2))./(1-
F_0(2))),alphaF(2),betaF(2),gammaF(2));
        end
    end
end
C3=F3<F_0(3);
[m3 n3]=size(F3);
for i=1:m3

```

```

    for j=1:n3
        if C3(i,j)==1
            tau{3}(i,j)=0;
        else
            tau{3}(i,j)=feval(Finv{1,3},((F3(i,j)-F_0(2))./(1-
F_0(3))),alphaF(2),betaF(2),gammaF(2));
        end
    end
end

H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the "multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%
% Determine taumax
taumax=tau{1};
for i=2:n
    taumax=max(taumax,tau{i});
end
% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio paper
H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG));
% Now take credit for recovery of emergency power trains
for i=1:n
    H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming)
    %H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i)));
end

```

“INTEGRAND3.m”:

```

function H=INTEGRAND3(taumax)
global f F_0 F alphaF betaF gammaF n G T_c G alphaG betaG gammaG Rhat

[m1 n1]=size(taumax);
H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the "multiple trains, no recovery" section
of
% the Rio paper.

% Credit for offsite recovery only following SBO
% H=H.*(1-feval(G,T_c,alphaG,betaG,gammaG));
% Now take credit for dynamic recovery of offsite power, as in Section
3.4 of Rio paper

```

```

H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG));

% Now probability others failed previously
Hsum1=(1-F_0(1)).*feval(f{1},taumax,alphaF(1),betaF(1),gammaF(1)); %
EDG 1 is last to fail (to run)
Hsum1=Hsum1.*(F_0(2).*ones(m1,n1)+(1-
F_0(2)).*feval(F{2},taumax,alphaF(2),betaF(2),gammaF(2)));
% EDG 2 fails earlier, either to start or to run
Hsum1=Hsum1.*(F_0(3).*ones(m1,n1)+(1-
F_0(3)).*feval(F{3},taumax,alphaF(3),betaF(3),gammaF(3)));
% EDG 3 fails earlier, either to start or to run
Hsum2=(1-F_0(2)).*feval(f{2},taumax,alphaF(2),betaF(2),gammaF(2)); %
EDG 2 is last to fail (to run)
Hsum2=Hsum2.*(F_0(1).*ones(m1,n1)+(1-
F_0(1)).*feval(F{1},taumax,alphaF(1),betaF(1),gammaF(1)));
% EDG 1 fails earlier, either to start or to run
Hsum2=Hsum2.*(F_0(3).*ones(m1,n1)+(1-
F_0(3)).*feval(F{3},taumax,alphaF(3),betaF(3),gammaF(3)));
% EDG 3 fails earlier, either to start or to run
Hsum3=(1-F_0(3)).*feval(f{3},taumax,alphaF(3),betaF(3),gammaF(3)); %
EDG 3 is last to fail (to run)
Hsum3=Hsum3.*(F_0(1).*ones(m1,n1)+(1-
F_0(1)).*feval(F{1},taumax,alphaF(1),betaF(1),gammaF(1)));
% EDG 1 fails earlier, either to start or to run
Hsum3=Hsum3.*(F_0(2).*ones(m1,n1)+(1-
F_0(2)).*feval(F{2},taumax,alphaF(2),betaF(2),gammaF(2)));
% EDG 2 fails earlier, either to start or to run
Hsum=Hsum1+Hsum2+Hsum3;
H=H.*Hsum;

```

“FTR6890.m”:

```

function P=FTR6890(t,mu1,cutpoint,mu2)
%cdf for a rv distributed exponentially with mean mu1 for times t <=
%cutpoint, and distributed afterward, conditionally upon exceeding
cutpoint,
%exponentially with mean mu2

if t<= cutpoint
    P=expcdf(t,mu1);
else
    P1=expcdf(cutpoint,mu1);
    P=P1+(1-P1)*expcdf(t-cutpoint,mu2);
end

```

“FTR6890inv.m”:

```

function t=FTR6890inv(P,mu1,cutpoint,mu2)
%CDF for a rv distributed exponentially with mean mu1 for times t <=
%cutpoint, and distributed afterward, conditionally upon exceeding
cutpoint,
%exponentially with mean mu2

```

```

Pcut=expcdf(cutpoint,mu1);
if P<=Pcut
    t=expinv(P,mu1);
else
    Pcut=expcdf(cutpoint,mu1);
    Pexcess=(P-Pcut)/(1-Pcut);
    t=cutpoint+expinv(Pexcess,mu2);
end

```

“FTR6890pdf.m”:

```

function P=FTR6890pdf(t,mu1,cutpoint,mu2)
%pdf for a rv distributed exponentially with mean mu1 for times t <=
%cutpoint, and distributed afterward, conditionally upon exceeding
%cutpoint,
%exponentially with mean mu2

if t<= cutpoint
    P=expcdf(t,mu1);
else
    P1=expcdf(cutpoint,mu1);
    P=(1-P1)*expcdf(t-cutpoint,mu2);
end

```

Appendix A.3 – STP-TAMIL (STPQUAD3CCF) Package Code

The MATLAB code title is in bold and everything following is the actual code that can be used in a MATLAB environment. Each specific code is broken up by these bold titles. It also has to be noted that this file also needs the “RunFile.xls” file to work.

“STPQUAD3CCF.m”:

```

%%***** STP-TAMIL VERSION 1.0 *****
%%Begin input
clear all
clc
global outputP F_0 F Finv alphaF betaF gammaF deltaF n T_c G alphaG
betaG gammaG deltaG R tau alphaR betaR gammaR deltaR
global cFTSLDisType1 cFTSLDisType2 cFTSLDisType3 invFTSLDisType1
invFTSLDisType2 invFTSLDisType3 pdfFTSLDisType1 pdfFTSLDisType2
pdfFTSLDisType3
global traincounter
global FTS3way FTR3way FTR3wayinv alpha3way beta3way gamma3way
delta3way
global FTS2way FTR2way FTR2wayinv alpha2way beta2way gamma2way
delta2way
global RecoveryType RecoveryType2 RecoveryType3

```



```

[numAMFinvcdf3, textAMFinvcdf3, AMFinvcdf3] = xlsread('RunFile.xlsx',
1, sprintf('AW%d', i));
%pdf File (not used left for exapandability)
%[numAMFpdf3, textAMFpdf3, AMFpdf3] = xlsread('RunFile.xlsx', 1,
sprintf('BH%d', i));

%2way
%Cdf File
[numAMFcdf2, textAMFcdf2, AMFcdf2] = xlsread('RunFile.xlsx', 1,
sprintf('AX%d', i));
%invCdf File
[numAMFinvcdf2, textAMFinvcdf2, AMFinvcdf2] = xlsread('RunFile.xlsx',
1, sprintf('AY%d', i));
%pdf File (not used left for exapandability)
%[numAMFpdf2, textAMFpdf2, AMFpdf2] = xlsread('RunFile.xlsx', 1,
sprintf('BK%d', i));

%Independent
%Cdf File
[numAMFcdf, textAMFcdf, AMFcdf] = xlsread('RunFile.xlsx', 1,
sprintf('AZ%d', i));
%invCdf File
[numAMFinvcdf, textAMFinvcdf, AMFinvcdf] = xlsread('RunFile.xlsx', 1,
sprintf('BA%d', i));
%pdf File (not used left for exapandability)
%[numAMFpdf, textAMFpdf, AMFpdf] = xlsread('RunFile.xlsx', 1,
sprintf('BN%d', i));

%-----New MATLAB-Files Failure of ith Train Files(added M
File Factors)-----
% This section is for if there is a different distribution files wanted
for
% each train
%Failure Train 1
%Cdf File
[numAMFT, textAMFT, AMFT] = xlsread('RunFile.xlsx', 1, sprintf('AZ%d',
i));
%invCdf File
[numAMFTinv, textAMFTinv, AMFTinv] = xlsread('RunFile.xlsx', 1,
sprintf('BA%d', i));
%pdf File (not used left for exapandability)
%[numAMFTpdf, textAMFTpdf, AMFTpdf] = xlsread('RunFile.xlsx', 1,
sprintf('BQ%d', i));

%Failure Train 2
%Cdf File
[numAMFT2, textAMFT2, AMFT2] = xlsread('RunFile.xlsx', 1,
sprintf('AZ%d', i));
%invCdf File
[numAMFT2inv, textAMFT2inv, AMFT2inv] = xlsread('RunFile.xlsx', 1,
sprintf('BA%d', i));
%pdf File (not used left for exapandability)
%[numAMFT2pdf, textAMFT2pdf, AMFT2pdf] = xlsread('RunFile.xlsx', 1,
sprintf('BT%d', i));

%Failure Train 3
%Cdf File

```

```

[numAMFT3, textAMFT3, AMFT3] = xlsread('RunFile.xlsx', 1,
sprintf('AZ%d', i));
%invCdf File
[numAMFT3inv, textAMFT3inv, AMFT3inv] = xlsread('RunFile.xlsx', 1,
sprintf('BA%d', i));
%pdf File (not used left for exapandability)
%[numAMFT3pdf, textAMFT3pdf, AMFT3pdf] = xlsread('RunFile.xlsx', 1,
sprintf('BW%d', i));

%-----New MATLAB-Files Factors (added M File Factors)-----
-----
% This section is for if there is a different distribution wanted for
each
% way (3/3, 2/3, 1/3) of failure
%3way
%Alpha
%[nummalph3, textmalph3, malph3]= xlsread('RunFile.xlsx', 1,
sprintf('L%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)
%Gamma
%[nummgamma3, textmgamma3, mgamma3]= xlsread('RunFile.xlsx', 1,
sprintf('N%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)

%2way
%Alpha
%[nummalph2, textmalph2, malph2]= xlsread('RunFile.xlsx', 1,
sprintf('S%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)
%Gamma
%[nummgamma2, textmgamma2, mgamma2]= xlsread('RunFile.xlsx', 1,
sprintf('U%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)

%Independent
%Alpha
%[nummalph, textmalph, malph]= xlsread('RunFile.xlsx', 1,
sprintf('Z%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)
%Gamma
%[nummgamma, textmgamma, mgamma]= xlsread('RunFile.xlsx', 1,
sprintf('AB%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)

%-----Distribution Types-----
-----
%cdf for recovery of Train i type (R)
[numRDT1, textRDT1, RecDisType1]= xlsread('RunFile.xlsx', 1,
sprintf('AT%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)
[numRDT2, textRDT2, RecDisType2]= xlsread('RunFile.xlsx', 1,
sprintf('AT%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)
[numRDT3, textRDT3, RecDisType3]= xlsread('RunFile.xlsx', 1,
sprintf('AT%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)

```

```

%cdf for FTSL of Train i type
[numcFTSL1, textcFTSL1, cFTSLDisType1]= xlsread('RunFile.xlsx', 1,
sprintf('AU%d', i)); %Type of Distribution for Cdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)
[numcFTSL2, textcFTSL2, cFTSLDisType2]= xlsread('RunFile.xlsx', 1,
sprintf('AU%d', i)); %Type of Distribution for Cdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)
[numcFTSL3, textcFTSL3, cFTSLDisType3]= xlsread('RunFile.xlsx', 1,
sprintf('AU%d', i)); %Type of Distribution for Cdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)

%invcdf for FTSL of Train i type (Points to same as cdf left in for
%expandability)
[numinvFTSL1, textinvFTSL1, invFTSLDisType1]= xlsread('RunFile.xlsx',
1, sprintf('AU%d', i)); %Type of Distribution for invCdf of FTSL of
train i (1-Weibull, 2-Lognormal, 3-Exponential)
[numinvFTSL2, textinvFTSL2, invFTSLDisType2]= xlsread('RunFile.xlsx',
1, sprintf('AU%d', i)); %Type of Distribution for invCdf of FTSL of
train i (1-Weibull, 2-Lognormal, 3-Exponential)
[numinvFTSL3, textinvFTSL3, invFTSLDisType3]= xlsread('RunFile.xlsx',
1, sprintf('AU%d', i)); %Type of Distribution for invCdf of FTSL of
train i (1-Weibull, 2-Lognormal, 3-Exponential)
%Deleted cdf from input and added inv so FTR6890inv would still work
%properly
if (isempty(numinvFTSL1)) else
%Removes cdf and adds inv due to Custom Distribution Code for Train 1
invFTSLDisType1= strtok(invFTSLDisType1, 'cdf');
invFTSLDisType1= strcat(invFTSLDisType1, 'inv');
end
if (isempty(numinvFTSL2)) else
%Removes cdf and adds inv due to Custom Distribution Code for Train 2
invFTSLDisType2= strtok(invFTSLDisType2, 'cdf');
invFTSLDisType2= strcat(invFTSLDisType2, 'inv');
end
if (isempty(numinvFTSL3)) else
%Removes cdf and adds inv due to Custom Distribution Code for Train 3
invFTSLDisType3= strtok(invFTSLDisType3, 'cdf');
invFTSLDisType3= strcat(invFTSLDisType3, 'inv');
end

%pdf for FTSL of Train i type (not used left for expandability)
%[numpFTSL1, textpFTSL1, pdfFTSLDisType1]= xlsread('RunFile.xlsx', 1,
sprintf('BC%d', i)); %Type of Distribution for pdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)
%[numpFTSL2, textpFTSL2, pdfFTSLDisType2]= xlsread('RunFile.xlsx', 1,
sprintf('BD%d', i)); %Type of Distribution for pdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)
%[numpFTSL3, textpFTSL3, pdfFTSLDisType3]= xlsread('RunFile.xlsx', 1,
sprintf('BE%d', i)); %Type of Distribution for pdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)

%Type of Distribution for Cdf of offsite recovery (G)
[numODT, textODT, rawdataODT] = xlsread('RunFile.xlsx', 1,
sprintf('AS%d', i));
%-----
-----

```

```

%-----Recovery Types-----
-
%1 for No Credit (offsite/onsite),
%2 for Offsite Credit Only,
%3 for Offsite + Onsite Repair only after Full Failure,
%4 for Full Recovery (NORMALLY/DEFAULT)

%-----For 3-way
RecoveryType3= xlsread('RunFile.xlsx', 1, sprintf('AP%d', i));
%-----For 2-way
RecoveryType2= xlsread('RunFile.xlsx', 1, sprintf('AQ%d', i));
%-----For Ind
RecoveryType= xlsread('RunFile.xlsx', 1, sprintf('AR%d', i));

%-----
----

% Mission Time, in hours
T=xlsread('RunFile.xlsx', 1, sprintf('C%d', i));

% Critical Time, in hours
T_c=xlsread('RunFile.xlsx', 1, sprintf('D%d', i));

alphaG=xlsread('RunFile.xlsx', 1, sprintf('E%d', i));
betaG=xlsread('RunFile.xlsx', 1, sprintf('F%d', i));
gammaG=xlsread('RunFile.xlsx', 1, sprintf('G%d', i));
deltaG=xlsread('RunFile.xlsx', 1, sprintf('H%d', i));
% parameters for G, here respectively mean time to
% recovery (hours) and standard deviation for the associated normal
distribution, as in
% the Word document "Verification 2"
% STP System Notebook as forwarded by Shawn Rodgers on 9/22/11

%-----3-WAY-----
--

if (n==3)

%% Input data for three-way common-cause failures
%FTS3way=xlsread('RunFile.xlsx', 1, sprintf('I%d', i));
%% Probability of common-cause failure of all three EDGs to start, here
% from second and fourth minimal cutsets in STP System Notebook (fan
failures
% assumed all to be failures to start)
FTS3way=xlsread('RunFile.xlsx', 1, sprintf('I%d', i));
P3FTLR = xlsread('RunFile.xlsx', 1, sprintf('J%d', i));
ETol3way =xlsread('RunFile.xlsx', 1, sprintf('K%d', i)); %Error
Tolerance of Single Integral
%P3FTLR=xlsread('RunFile.xlsx', 1, sprintf('J%d', i));
% Probability of 3-way Failure to Load and Run for one hour, per third
% minimal cutset in system notebook

%P3FTR =xlsread('RunFile.xlsx', 1, sprintf('K%d', i));
% 3-way Failure to Run between hours 1 and 24, per first minimal cutset
% in system notebook

% % This for 1-exp(-time*rate)

```

```

% ~= time*rate, or 1-exp(-lambda*t) ~= lambda*t
% Next without that approximation
beta3way=xlsread('RunFile.xlsx', 1, sprintf('M%d', i));
delta3way=xlsread('RunFile.xlsx', 1, sprintf('O%d', i));

%if (malpha3 == 'CodeRun')
    % alpha3way = -1/(log(1-P3FTLR/(1-FTS3way)));
    % alpha3way = -1/(log(1-P3FTLR));
    %Uses alpha defined above
%else
    %otherwise uses new alpha
    alpha3way =xlsread('RunFile.xlsx', 1, sprintf('L%d', i));
%end
%if (mgamma3 == 'CodeRun')
    % gamma3way = -23/(log(1-P3FTR/((1-FTS3way)*(1-P3FTLR))));
    % gamma3way = 1/(3*(log(1-P3FTR/((1-FTS3way)*(1-P3FTLR))));
    %Uses gamma defined above
%else
    %otherwise uses new gamma
    gamma3way =xlsread('RunFile.xlsx', 1, sprintf('N%d', i));
%end
% parameters for cdf for
% 3-way CC failure to run, here alpha3way = mu1= exponential mean up
to time beta3way = cutoff time,
% and gamma3way = subsequent exponential mean conditional on reaching
the cutoff time, numerical
% values per first and third top minimal cutsets in STP system notebook
end
%-----INDEPENDENT-----
----

%% Input data for independent failures
%P3IFTLR=xlsread('RunFile.xlsx', 1, sprintf('W%d', i)); % From 52nd
minimal cutset

%PIFTLR=P3IFTLR^(1/3); % Independent failure to load and run

%P1IFTS2IFTLR=xlsread('RunFile.xlsx', 1, sprintf('X%d', i)); %
Independent failures of one specified EDG to start and other two to
load
% and run, per 81st-83rd minimal cutsets in system notebook

%PIFTS=P1IFTS2IFTLR/(PIFTLR)^2; % Independent failure to start
%lambda1=-log(1-PIFTLR/(1-PIFTS)); % Independent failure rate during
first hour

%P3IFTR=xlsread('RunFile.xlsx', 1, sprintf('Y%d', i));
% From fifth most important cutset in the system notebook

%PIFTR=(P3IFTR)^(1/3);
%lambda2=-log(1-PIFTR/((1-PIFTS)*(1-PIFTLR)))/23;

PIFTS=xlsread('RunFile.xlsx', 1, sprintf('W%d', i)); % Independent
failure to start
PIFTLR=xlsread('RunFile.xlsx', 1, sprintf('X%d', i));% Probability of
independent failure to load and run

```

```

EtolInd=xlsread('RunFile.xlsx', 1, sprintf('Y%d', i));% Error Tolerance
of Triple Integral

betaInd=xlsread('RunFile.xlsx', 1, sprintf('AA%d', i));
deltaInd=xlsread('RunFile.xlsx', 1, sprintf('AC%d', i));

%if (malpha == 'CodeRun')
%  alphaInd=1/lambda1; % Mean time to independent failure, during first
hour
%Uses alpha defined above
%else
%otherwise uses new alpha
alphaInd=xlsread('RunFile.xlsx', 1, sprintf('Z%d', i));
%end
%if (mgamma == 'CodeRun')
%  gammaInd = 1/lambda2;
%Uses gamma defined above
%else
%otherwise uses new gamma
gammaInd =xlsread('RunFile.xlsx', 1, sprintf('AB%d', i));
%end
% parameters for cdf for
% failure to run, here alphaInd = mu1 = exponential mean up to time
betaInd = cutoff time (hours),
% and gammaInd = subsequent exponential mean conditional on reaching
the
% cutoff time

%-----2-WAY-----
--
if (n==2) || (n==3)
%% Input data for two-way common-cause failures
%P1IFTR2FTR=xlsread('RunFile.xlsx', 1, sprintf('P%d', i));
% Probability of common-cause failure of 2/3 EDGs to start,
% and third to fail to run, here from 34th-36th minimal cutsets

%P1FTR2FTLR=xlsread('RunFile.xlsx', 1, sprintf('Q%d', i));
%Probability of 2-way common-cause failure of two designated EDGs
% to load and run, and of the third to fail independently to run, per
minimal cutsets
% 19-21

%P1FTR2FTR=xlsread('RunFile.xlsx', 1, sprintf('R%d', i));
%% Probability of 2-way common-cause failure of two designated EDGs
% to run, and of the third to fail independently to run, per minimal
cutsets
% 9-11

%P2FTR=P1IFTR2FTR/PIFTR;
%P2FTLR=P1FTR2FTLR/PIFTR;

%lambda1=-log(1-P2FTLR/(1-P2FTR)); % Independent failure rate during
first
%hour

%P2FTR=P1FTR2FTR/PIFTR;

```

```

%lambda2=-log(1-P2FTR/((1-P2FTS)*(1-P2FTLR)))/23;

P2FTS=xlsread('RunFile.xlsx', 1, sprintf('P%d', i));
P2FTLR=xlsread('RunFile.xlsx', 1, sprintf('Q%d', i));
Etol2way =xlsread('RunFile.xlsx', 1, sprintf('R%d', i));
beta2way=xlsread('RunFile.xlsx', 1, sprintf('T%d', i));
delta2way=xlsread('RunFile.xlsx', 1, sprintf('V%d', i));
%if (malpha2 == 'CodeRun')
%   alpha2way=1/lambda1; % Mean time to independent failure, during
first hour
    %Uses alpha defined above
%else
    %otherwise uses new alpha
    alpha2way=xlsread('RunFile.xlsx', 1, sprintf('S%d', i));
%end
%if (mgamma2 == 'CodeRun')
%gamma2way=1/lambda2;
    %Uses gamma defined above
%else
    %otherwise uses new gamma
    gamma2way =xlsread('RunFile.xlsx', 1, sprintf('U%d', i));
%end

% parameters for cdf for
% failure to run, here alphaind = mu1 = exponential mean up to time
betaind = cutoff time (hours),
% and gammaind = subsequent exponential mean conditional on reaching
the
% cutoff time
end

% Below are place holder Variables for alphaR(1),betaR(1),gammaR(1),and
deltaR(1)
% used to make sure Default Values would work if needed
ph1=xlsread('RunFile.xlsx', 1, sprintf('AD%d', i));
ph2=xlsread('RunFile.xlsx', 1, sprintf('AE%d', i));
ph3=xlsread('RunFile.xlsx', 1, sprintf('AF%d', i));
ph4=xlsread('RunFile.xlsx', 1, sprintf('AG%d', i));
alphaR(1)=ph1;
betaR(1)=ph2;
gammaR(1)=ph3;
deltaR(1)=ph4;
% parameters for recovery cdf for train 1, here
% the respective values of the scale and shape parameters for a Weibull
% distribution, as detailed in "Verification 2" (alphaR in hours)

if (n==2) || (n==3)

% Below are place holder Variables for alphaR(2),betaR(2),gammaR(2),and
deltaR(2)
% used to make sure Default Values would work if needed
ph5=xlsread('RunFile.xlsx', 1, sprintf('AH%d', i));
ph6=xlsread('RunFile.xlsx', 1, sprintf('AI%d', i));
ph7=xlsread('RunFile.xlsx', 1, sprintf('AJ%d', i));
ph8=xlsread('RunFile.xlsx', 1, sprintf('AK%d', i));
alphaR(2)=ph5;

```



```

if (isempty(T_c))
    %If true, then write a statement to exit the loop on next through
    xlswrite('RunFile.xlsx', 1, 1, sprintf('D%d', i));
    T_c=xlsread('RunFile.xlsx', 1, sprintf('D%d', i));
    strn=strn+1;
        dvalstr{strn} = fprintf('D%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(alphaG))
    %If true, then write a statement to exit the loop on next through
    alphaG=0.3;
        xlswrite('RunFile.xlsx', alphaG, 1, sprintf('E%d',
i));
    alphaG=xlsread('RunFile.xlsx', 1, sprintf('E%d', i));
    strn=strn+1;
        dvalstr{strn} = fprintf('E%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(betaG))
    %If true, then write a statement to exit the loop on next through
    betaG=1.064;
        xlswrite('RunFile.xlsx', betaG, 1, sprintf('F%d', i));
betaG=xlsread('RunFile.xlsx', 1, sprintf('F%d', i));
    strn=strn+1;
        dvalstr{strn} = fprintf('F%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(gammaG))
    %If true, then write a statement to exit the loop on next through
    gammaG=0;
        xlswrite('RunFile.xlsx', gammaG, 1, sprintf('G%d', i));
    gammaG=xlsread('RunFile.xlsx', 1, sprintf('G%d', i));
    strn=strn+1;
        dvalstr{strn} = fprintf('G%d', i);
dvalstr{strn} = fprintf('\n');

end

if (isempty(deltaG))
    %If true, then write a statement to exit the loop on next through
    deltaG=0;
        xlswrite('RunFile.xlsx', deltaG, 1, sprintf('H%d', i));
    deltaG=xlsread('RunFile.xlsx', 1, sprintf('H%d', i));
    strn=strn+1;
        dvalstr{strn} = fprintf('H%d', i);
dvalstr{strn} = fprintf('\n');
end

if (n==3)
if (isempty(FTS3way))
    %If true, then write a statement to exit the loop on next through
    FTS3way=0.0000536686514072092;
        xlswrite('RunFile.xlsx', FTS3way, 1, sprintf('I%d', i));

```

```

        FTS3way=xlsread('RunFile.xlsx', 1, sprintf('I%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('I%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(P3FTLR))
    %If true, then write a statement to exit the loop on next through
    P3FTLR=0.000014181825726059;
    xlswrite('RunFile.xlsx', P3FTLR, 1, sprintf('J%d', i));
    P3FTLR = xlsread('RunFile.xlsx', 1, sprintf('J%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('J%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(ETol3way))
    %If true, then write a statement to exit the loop on next through
    ETol3way=0.0000000001;
    xlswrite('RunFile.xlsx', ETol3way, 1, sprintf('K%d', i));
    ETol3way =xlsread('RunFile.xlsx', 1, sprintf('K%d', i)); %Error
Tolerance of Single Integral
strn=strn+1;
    dvalstr{strn} = fprintf('K%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(alpha3way))
    %If true, then write a statement to exit the loop on next through
    alpha3way=70512.2830010499;
    xlswrite('RunFile.xlsx', alpha3way, 1, sprintf('L%d', i));
    alpha3way =xlsread('RunFile.xlsx', 1, sprintf('L%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('L%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(beta3way))
    %If true, then write a statement to exit the loop on next through
    beta3way=1;
    xlswrite('RunFile.xlsx', beta3way, 1, sprintf('M%d', i));
    beta3way=xlsread('RunFile.xlsx', 1, sprintf('M%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('M%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(gamma3way))
    %If true, then write a statement to exit the loop on next through
    gamma3way=113167.568967981;
    xlswrite('RunFile.xlsx', gamma3way, 1, sprintf('N%d', i));
    gamma3way =xlsread('RunFile.xlsx', 1, sprintf('N%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('N%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(delta3way))
    %If true, then write a statement to exit the loop on next through
    delta3way=0;
    xlswrite('RunFile.xlsx', delta3way, 1, sprintf('O%d', i));
    delta3way=xlsread('RunFile.xlsx', 1, sprintf('O%d', i));
strn=strn+1;

```

```

        dvalstr{strn} = fprintf('O%d', i);
dvalstr{strn} = fprintf('\n');
end
end
if (n==3) || (n==2)
if (isempty(P2FTS))
    %If true, then write a statement to exit the loop on next through
    P2FTS=0.0000330131355289566;
    xlswrite('RunFile.xlsx', P2FTS, 1, sprintf('P%d', i));
    P2FTS=xlsread('RunFile.xlsx', 1, sprintf('P%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('P%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(P2FTLR))
    %If true, then write a statement to exit the loop on next through
    P2FTLR=0.0000255272863069063;
    xlswrite('RunFile.xlsx', P2FTLR, 1, sprintf('Q%d', i));
    P2FTLR=xlsread('RunFile.xlsx', 1, sprintf('Q%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('Q%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(Etol2way))
    %If true, then write a statement to exit the loop on next through
    Etol2way=0.0000000001;
    xlswrite('RunFile.xlsx', Etol2way, 1, sprintf('R%d', i));
    Etol2way=xlsread('RunFile.xlsx', 1, sprintf('R%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('R%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(alpha2way))
    %If true, then write a statement to exit the loop on next through
    alpha2way=39173.2683325821;
    xlswrite('RunFile.xlsx', alpha2way, 1, sprintf('S%d', i));
    alpha2way=xlsread('RunFile.xlsx', 1, sprintf('S%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('S%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(beta2way))
    %If true, then write a statement to exit the loop on next through
    beta2way=1;
    xlswrite('RunFile.xlsx', beta2way, 1, sprintf('T%d', i));
    beta2way=xlsread('RunFile.xlsx', 1, sprintf('T%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('T%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(gamma2way))
    %If true, then write a statement to exit the loop on next through
    gamma2way=187900.869229856;
    xlswrite('RunFile.xlsx', gamma2way, 1, sprintf('U%d', i));
    gamma2way=xlsread('RunFile.xlsx', 1, sprintf('U%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('U%d', i);

```

```

dvalstr{strn} = fprintf('\n');
end
if (isempty(delta2way))
    %If true, then write a statement to exit the loop on next through
    delta2way=0;
    xlswrite('RunFile.xlsx', delta2way, 1, sprintf('V%d', i));
    delta2way=xlsread('RunFile.xlsx', 1, sprintf('V%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('V%d', i);
dvalstr{strn} = fprintf('\n');
end
end

if (isempty(PIFTS))
    %If true, then write a statement to exit the loop on next through
    PIFTS=0.004410305077535;
    xlswrite('RunFile.xlsx', PIFTS, 1, sprintf('W%d', i));
    PIFTS=xlsread('RunFile.xlsx', 1, sprintf('W%d', i)); %
Independent failure to start
strn=strn+1;
    dvalstr{strn} = fprintf('W%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(PIFTLR))
    %If true, then write a statement to exit the loop on next through
    PIFTLR=0.000014181825726059;
    xlswrite('RunFile.xlsx', PIFTLR, 1, sprintf('X%d', i));
    PIFTLR=xlsread('RunFile.xlsx', 1, sprintf('X%d', i));%
Probability of independent failure to load and run
strn=strn+1;
    dvalstr{strn} = fprintf('X%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(EtolInd))
    %If true, then write a statement to exit the loop on next through
    EtolInd=0.0000000001;
    xlswrite('RunFile.xlsx', EtolInd, 1, sprintf('Y%d', i));
    EtolInd=xlsread('RunFile.xlsx', 1, sprintf('Y%d', i));% Error
Tolerance of Triple Integral
strn=strn+1;
    dvalstr{strn} = fprintf('Y%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(alphaInd))
    %If true, then write a statement to exit the loop on next through
    alphaInd=352.262864884551;
    xlswrite('RunFile.xlsx', alphaInd, 1, sprintf('Z%d', i));
    alphaInd=xlsread('RunFile.xlsx', 1, sprintf('Z%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('Z%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(betaInd))
    %If true, then write a statement to exit the loop on next through
    betaInd=1;
    xlswrite('RunFile.xlsx', betaInd, 1, sprintf('AA%d', i));
    betaInd=xlsread('RunFile.xlsx', 1, sprintf('AA%d', i));

```

```

strn=strn+1;
    dvalstr{strn} = fprintf('AA%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(gammaind))
    %If true, then write a statement to exit the loop on next through
    gammaind=1206.97199769511;
    xlswrite('RunFile.xlsx', gammaind, 1, sprintf('AB%d', i));
    gammaind =xlsread('RunFile.xlsx', 1, sprintf('AB%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AB%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(deltaind))
    %If true, then write a statement to exit the loop on next through
    deltaind=0;
    xlswrite('RunFile.xlsx', deltaind, 1, sprintf('AC%d', i));
    deltaind=xlsread('RunFile.xlsx', 1, sprintf('AC%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AC%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(ph1))
    %If true, then write a statement to exit the loop on next through
    ph1=15.5;
    xlswrite('RunFile.xlsx', ph1, 1, sprintf('AD%d', i));
    alphaR(1)=xlsread('RunFile.xlsx', 1, sprintf('AD%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AD%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(ph2))
    %If true, then write a statement to exit the loop on next through
    ph2=0.739;
    xlswrite('RunFile.xlsx', ph2, 1, sprintf('AE%d', i));

betaR(1)=xlsread('RunFile.xlsx', 1, sprintf('AE%d', i));
    strn=strn+1;
    dvalstr{strn} = fprintf('AE%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(ph3))
    %If true, then write a statement to exit the loop on next through
    ph3=0;
    xlswrite('RunFile.xlsx', ph3, 1, sprintf('AF%d', i));
    gammaR(1)=xlsread('RunFile.xlsx', 1, sprintf('AF%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AF%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(ph4))
    %If true, then write a statement to exit the loop on next through
    ph4=0;
    xlswrite('RunFile.xlsx', ph4, 1, sprintf('AG%d', i));
    deltaR(1)=xlsread('RunFile.xlsx', 1, sprintf('AG%d', i));
strn=strn+1;

```

```

        dvalstr{strn} = fprintf('AG%d', i);
dvalstr{strn} = fprintf('\n');
end

if (n==2) || (n==3)

if (isempty(ph5))
    %If true, then write a statement to exit the loop on next through
    ph5=15.5;
    xlswrite('RunFile.xlsx', ph5, 1, sprintf('AH%d', i));
    alphaR(2)=xlsread('RunFile.xlsx', 1, sprintf('AH%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AH%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(ph6))
    %If true, then write a statement to exit the loop on next through
    ph6=0.739;
    xlswrite('RunFile.xlsx', ph6, 1, sprintf('AI%d', i));
betaR(2)=xlsread('RunFile.xlsx', 1, sprintf('AI%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AI%d', i);
dvalstr{strn} = fprintf('\n');
end
if (isempty(ph7))
    %If true, then write a statement to exit the loop on next through
    ph7=0;
    xlswrite('RunFile.xlsx', ph7, 1, sprintf('AJ%d', i));
gammaR(2)=xlsread('RunFile.xlsx', 1, sprintf('AJ%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AJ%d', i);
    dvalstr{strn} = fprintf('\n');

end
if (isempty(ph8))
    %If true, then write a statement to exit the loop on next through
    ph8=0;
    xlswrite('RunFile.xlsx', ph8, 1, sprintf('AK%d', i));
deltaR(2)=xlsread('RunFile.xlsx', 1, sprintf('AK%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AK%d', i);
    dvalstr{strn} = fprintf('\n');

end
end

if (n==3)
if (isempty(ph9))
    %If true, then write a statement to exit the loop on next through
    ph9=15.5;
    xlswrite('RunFile.xlsx', ph9, 1, sprintf('AL%d', i));
    alphaR(3)=xlsread('RunFile.xlsx', 1, sprintf('AL%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AL%d', i);
    dvalstr{strn} = fprintf('\n');

end
end

```

```

if (isempty(ph10))
    %If true, then write a statement to exit the loop on next through
    ph10=0.739;
    xlswrite('RunFile.xlsx', ph10, 1, sprintf('AM%d', i));
betaR(3)=xlsread('RunFile.xlsx', 1, sprintf('AM%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AM%d', i);
    dvalstr{strn} = fprintf('\n');

end
if (isempty(ph11))
    %If true, then write a statement to exit the loop on next through
    ph11=0;
    xlswrite('RunFile.xlsx', ph11, 1, sprintf('AN%d', i));
gammaR(3)=xlsread('RunFile.xlsx', 1, sprintf('AN%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AN%d', i);
    dvalstr{strn} = fprintf('\n');

end
if (isempty(ph12))
    %If true, then write a statement to exit the loop on next through
    ph12=0;
    xlswrite('RunFile.xlsx', ph12, 1, sprintf('AO%d', i));
deltaR(3)=xlsread('RunFile.xlsx', 1, sprintf('AO%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AO%d', i);
    dvalstr{strn} = fprintf('\n');

end

if (isempty(RecoveryType3))
    %If true, then write a statement to exit the loop on next through
    RecoveryType3=1;
    xlswrite('RunFile.xlsx', RecoveryType3, 1, sprintf('AP%d',
i));
RecoveryType3= xlsread('RunFile.xlsx', 1, sprintf('AP%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AP%d', i);
    dvalstr{strn} = fprintf('\n');

end
end
if (n==2) || (n==3)

if (isempty(RecoveryType2))
    %If true, then write a statement to exit the loop on next through
    RecoveryType2=1;
    xlswrite('RunFile.xlsx', RecoveryType2, 1, sprintf('AQ%d', i));
    RecoveryType2= xlsread('RunFile.xlsx', 1, sprintf('AQ%d', i));
strn=strn+1;
    dvalstr{strn} = fprintf('AQ%d', i);
    dvalstr{strn} = fprintf('\n');
end
end
if (isempty(RecoveryType))
    %If true, then write a statement to exit the loop on next through

```



```

        RecoveryType=1;
        xlswrite('RunFile.xlsx', RecoveryType, 1, sprintf('AR%d', i));
        RecoveryType= xlsread('RunFile.xlsx', 1, sprintf('AR%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('AR%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(textODT))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'logncdf'}, 1, sprintf('AS%d', i));
        [numODT, textODT, rawdataODT] = xlsread('RunFile.xlsx', 1,
sprintf('AS%d', i));

strn=strn+1;
        dvalstr{strn} = fprintf('AS%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(textRDT1))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'wblcdf'}, 1, sprintf('AT%d', i));
[numRDT1, textRDT1, RecDisType1]= xlsread('RunFile.xlsx', 1,
sprintf('AT%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)
strn=strn+1;
        dvalstr{strn} = fprintf('AT%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(textRDT2))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'wblcdf'}, 1, sprintf('AT%d', i));
[numRDT2, textRDT2, RecDisType2]= xlsread('RunFile.xlsx', 1,
sprintf('AT%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)

end

if (isempty(textRDT3))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'wblcdf'}, 1, sprintf('AT%d', i));
[numRDT3, textRDT3, RecDisType3]= xlsread('RunFile.xlsx', 1,
sprintf('AT%d', i)); %Type of Distribution for Cdf of Recovery of train
i (1-Weibull, 2-Lognormal, 3-Exponential)

end

if (isempty(textcFTSL1))
    %If true, then write a statement to exit the loop on next through

```

```

        xlswrite('RunFile.xlsx', {'expcdf'}, 1, sprintf('AU%d', i));
[numcFTSL1, textcFTSL1, cFTSLDisType1]= xlsread('RunFile.xlsx', 1,
sprintf('AU%d', i)); %Type of Distribution for Cdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)
strn=strn+1;
        dvalstr{strn} = fprintf('AU%d', i);
dvalstr{strn} = fprintf('\n');
end
if (n==2)|| (n==3)
if (isempty(textcFTSL2))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'expcdf'}, 1, sprintf('AU%d', i));
[numcFTSL2, textcFTSL2, cFTSLDisType2]= xlsread('RunFile.xlsx', 1,
sprintf('AU%d', i)); %Type of Distribution for Cdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)

end
if (n==3)
if (isempty(textcFTSL3))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'expcdf'}, 1, sprintf('AU%d', i));
[numcFTSL3, textcFTSL3, cFTSLDisType3]= xlsread('RunFile.xlsx', 1,
sprintf('AU%d', i)); %Type of Distribution for Cdf of FTSL of train i
(1-Weibull, 2-Lognormal, 3-Exponential)

end
end
end
if (isempty(textinvFTSL1))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'expcdf'}, 1, sprintf('AU%d', i));
[numinvFTSL1, textinvFTSL1, invFTSLDisType1]= xlsread('RunFile.xlsx',
1, sprintf('AU%d', i)); %Type of Distribution for invCdf of FTSL of
train i (1-Weibull, 2-Lognormal, 3-Exponential)
invFTSLDisType1= strtok(invFTSLDisType1, 'cdf');
invFTSLDisType1= strcat(invFTSLDisType1, 'inv');

else
invFTSLDisType1= strtok(invFTSLDisType1, 'cdf');
invFTSLDisType1= strcat(invFTSLDisType1, 'inv');
end
if (n==2)|| (n==3)

if (isempty(textinvFTSL2))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'expcdf'}, 1, sprintf('AU%d', i));
[numinvFTSL2, textinvFTSL2, invFTSLDisType2]= xlsread('RunFile.xlsx',
1, sprintf('AU%d', i)); %Type of Distribution for invCdf of FTSL of
train i (1-Weibull, 2-Lognormal, 3-Exponential)
invFTSLDisType2= strtok(invFTSLDisType2, 'cdf');
invFTSLDisType2= strcat(invFTSLDisType2, 'inv');

else

```

```

invFTSLDisType2= strtok(invFTSLDisType2, 'cdf');
invFTSLDisType2= strcat(invFTSLDisType2, 'inv');
end
if (n==3)
if (isempty(textinvFTSL3))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'expcdf'}, 1, sprintf('AU%d', i));
[numinvFTSL3, textinvFTSL3, invFTSLDisType3]= xlsread('RunFile.xlsx',
1, sprintf('AU%d', i)); %Type of Distribution for invCdf of FTSL of
train i (1-Weibull, 2-Lognormal, 3-Exponential)
        invFTSLDisType3= strtok(invFTSLDisType3, 'cdf');
invFTSLDisType3= strcat(invFTSLDisType3, 'inv');
strn=strn+1;

else
        invFTSLDisType3= strtok(invFTSLDisType3, 'cdf');
invFTSLDisType3= strcat(invFTSLDisType3, 'inv');
end

if (isempty(textAMFcdf3))
    %If true, then write a statement to exit the loop on next through
        xlswrite('RunFile.xlsx', {'FTR6890'}, 1, sprintf('AV%d', i));
[numAMFcdf3, textAMFcdf3, AMFcdf3] = xlsread('RunFile.xlsx', 1,
sprintf('AV%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('AV%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(textAMFinvcdf3))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'FTR6890inv'}, 1, sprintf('AW%d', i));
[numAMFinvcdf3, textAMFinvcdf3, AMFinvcdf3] =
xlsread('RunFile.xlsx', 1, sprintf('AW%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('AW%d', i);
dvalstr{strn} = fprintf('\n');
end
end
if (isempty(textAMFcdf2))
    %If true, then write a statement to exit the loop on next through
        xlswrite('RunFile.xlsx', {'FTR6890'}, 1, sprintf('AX%d',
i));
[numAMFcdf2, textAMFcdf2, AMFcdf2] = xlsread('RunFile.xlsx', 1,
sprintf('AX%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('AX%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(textAMFinvcdf2))
    %If true, then write a statement to exit the loop on next through

```

```

        xlswrite('RunFile.xlsx', {'FTR6890inv'}, 1, sprintf('AY%d', i));
        [numAMFinvcdf2, textAMFinvcdf2, AMFinvcdf2] =
xlsread('RunFile.xlsx', 1, sprintf('AY%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('AY%d', i);
dvalstr{strn} = fprintf('\n');
end
end
if (isempty(textAMFcdf))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'FTR6890'}, 1,
sprintf('AZ%d', i));
    [numAMFcdf, textAMFcdf, AMFcdf] = xlsread('RunFile.xlsx', 1,
sprintf('AZ%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('AZ%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(textAMFinvcdf))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'FTR6890inv'}, 1, sprintf('BA%d', i));
        [numAMFinvcdf, textAMFinvcdf, AMFinvcdf] =
xlsread('RunFile.xlsx', 1, sprintf('BA%d', i));
strn=strn+1;
        dvalstr{strn} = fprintf('BA%d', i);
dvalstr{strn} = fprintf('\n');
end

if (isempty(textAMFT))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'FTR6890'}, 1,
sprintf('AZ%d', i));
    [numAMFT, textAMFT, AMFT] = xlsread('RunFile.xlsx', 1,
sprintf('AZ%d', i));

end

if (isempty(textAMFTinv))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'FTR6890inv'}, 1, sprintf('BA%d', i));
        [numAMFTinv, textAMFTinv, AMFTinv] = xlsread('RunFile.xlsx', 1,
sprintf('BA%d', i));

end
if (n==2)|| (n==3)

if (isempty(textAMFT2))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'FTR6890'}, 1,
sprintf('AZ%d', i));

```

```

[numAMFT2, textAMFT2, AMFT2] = xlsread('RunFile.xlsx', 1,
sprintf('AZ%d', i));

end

if (isempty(textAMFT2inv))
    %If true, then write a statement to exit the loop on next through

    xlswrite('RunFile.xlsx', {'FTR6890inv'}, 1, sprintf('BA%d', i));
    [numAMFT2inv, textAMFT2inv, AMFT2inv] = xlsread('RunFile.xlsx',
1, sprintf('BA%d', i));

end
if (n==3)
if (isempty(textAMFT3))
    %If true, then write a statement to exit the loop on next through

        xlswrite('RunFile.xlsx', {'FTR6890'}, 1,
sprintf('AZ%d', i));
[numAMFT3, textAMFT3, AMFT3] = xlsread('RunFile.xlsx', 1,
sprintf('AZ%d', i));

end

if (isempty(textAMFT3inv))
    %If true, then write a statement to exit the loop on next through

    xlswrite('RunFile.xlsx', {'FTR6890inv'}, 1, sprintf('BA%d', i));
    [numAMFT3inv, textAMFT3inv, AMFT3inv] = xlsread('RunFile.xlsx',
1, sprintf('BA%d', i));

end

end
end
strn=strn+1;
dvalstr{strn} = fprintf('\n');
%i=i+1;
strn=strn+1;
if (strn<5)
    dvalstr{strn} = fprintf('No cells were left blank.');
```

```

else
    dvalstr{strn} = fprintf('Note: Default values were added to
"RunFile.xlsx"');
```

```

end
strn=strn+1;
dvalstr{strn} = fprintf('\n');
dvalstr{strn} = fprintf('\n');

strn=strn+1;
dvalstr{strn} = fprintf('The Code calculated:');
strn=strn+1;
dvalstr{strn} = fprintf('\n');
```

```

if ((n==3)|| (n==2))
```



```

% of the three EDGs; here the inverse of the piecewise exponential
function coded into FTR6890, per
% "Verification 2"

if ((n==3)|| (n==2))
FTR2way=str2func(AMFcdf2);
%FTR2way=@FTR6890; % cdf for 2-way common-cause failure of EDGs to load
and run,
% conditional on starting; here a two-regime exponential cdf, per
"Verification 2"
FTR2wayinv=str2func(AMFinvcdf2);
%FTR2wayinv=@FTR6890inv; % the corresponding inverse cdf for 2-way
common-cause failures to run
% of the three EDGs; here the inverse of the piecewise exponential
function coded into FTR6890, per
% "Verification 2"
end
% probability of initial failure to start of train 1, per
%"Verification 2".
F_0(1)=PIFTS; % Same, for same initial failure to start for all trains

F{1}=str2func(AMFT);
%F{1}=@FTR6890; % cdf for failure of train 1 to start and load; here a
piecewise
% exponential cdf, per "Verification 2"
Finv{1}=str2func(AMFTinv);
%Finv{1}=@FTR6890inv; % the corresponding inverse cdf for failure of
train 1 to start and load; here
% the inverse of the piecewise exponential function coded into FTR6890,
per
% "Verification 2"

% parameters for cdf for failure to run, here
% alphaF = mu1 = exponential mean up to time betaF = cutoff time, and
gammaF = subsequent
% exponential mean conditional on reaching the cutoff time, numerical
% values per "Verification 2"
alphaF(1)=alphaind;
betaF(1)=betaind;
gammaF(1)=gammaind;
deltaF(1)=deltaind;

R{1} = str2func(RecDisType1);
% cdf in time (hours) for recovery (repair) of train 1. Here the
Weibull
% cdf, per "Verification 2"

if ((n==3)|| (n==2))
% probability of initial failure to start of train 2, per "Verification
2".
F_0(2)=PIFTS; % Same, for same initial failure to start for all trains

F{2}=str2func(AMFT2);
%F{2}=@FTR6890; % cdf for failure of train 2 to start and load; here a
piecewise
% exponential cdf, per "Verification 2"

```

```

Finv{2}=str2func(AMFT2inv);
%Finv{2}=@FTR6890inv; % the corresponding inverse cdf for failure of
train 2 to start and load; here
% the inverse of the piecewise exponential function coded into FTR6890,
per
% "Verification 2"

% parameters for cdf for failure to run, here
% alphaF = mul = exponential mean up to time betaF = cutoff time, and
gammaF = subsequent
% exponential mean conditional on reaching the cutoff time, numerical
% values per "Verification 2"
alphaF(2)=alphaind;
betaF(2)=betaind;
gammaF(2)=gammaind;
deltaF(2)=deltaind;

R{2} = str2func(RecDisType2);
%R{2}=@wblcdf; % cdf in time (hours) for recovery (repair) of train 2.
Here the Weibull
% cdf, per "Verification 2"

if (n==3)
% probability of initial failure to start of train 3, per "Verification
2".
F_0(3)=PIFTS; % Same, for same initial failure to start for all trains

F{3}=str2func(AMFT3);
%F{3}=@FTR6890; % cdf for failure of train 3 to start and load; here a
piecewise
% exponential cdf, per "Verification 2"

Finv{3}=str2func(AMFT3inv);
%Finv{3}=@FTR6890inv; % the corresponding inverse cdf for failure of
train 3 to start and load; here
% the inverse of the piecewise exponential function coded into FTR6890,
per
% "Verification 2"

% parameters for cdf for failure to run, here
% alphaF = mul = exponential mean up to time betaF = cutoff time, and
gammaF = subsequent
% exponential mean conditional on reaching the cutoff time, numerical
% values per "Verification 2"
alphaF(3)=alphaind;
betaF(3)=betaind;
gammaF(3)=gammaind;
deltaF(3)=deltaind;

R{3} = str2func(RecDisType3); % cdf in time (hours) for recovery
(repair) of train 3. Here the Weibull
% cdf, per "Verification 2"

end
end

```



```

%%

%*****Calculation*****
% Convert Riemann-Stieltjes triple nonrecovery integral to ordinary
Riemann
% integral
A=0;
traincounter=1;
%B=F_0(1)+(1-F_0(1))*feval(F{1},T,alphaF(1),betaF(1),gammaF(1)); %
Basic events independent
B=F_0(1)+feval(F{1},T,alphaF(1),betaF(1),gammaF(1)); % Basic events
mutually exclusive
C=0;
if (n==2) || (n==3)
traincounter=2;
% D=F_0(2)+(1-F_0(2))*feval(F{2},T,alphaF(2),betaF(2),gammaF(2)); %
Independent
D=F_0(2)+feval(F{2},T,alphaF(2),betaF(2),gammaF(2)); % Exclusive
E=0;
if (n==3)
traincounter=3;
% Fupper=F_0(3)+(1-F_0(3))*feval(F{3},T,alphaF(3),betaF(3),gammaF(3));
% Independent
PIFTR2=feval(F{3},T,alphaF(3),betaF(3),gammaF(3))-
feval(F{3},betaF(3),alphaF(3),betaF(3),gammaF(3));
Fupper=F_0(3)+feval(F{3},T,alphaF(3),betaF(3),gammaF(3)); % Exclusive
% P_CSBO1=triplequad(@INTEGRANDv1pt0,A,B,C,D,E,F) % Default absolute
error
%tolerance of 1.0e-6
else
    PIFTR2=0
    Fupper=0
end
else
    D=0
    E=0
    PIFTR2=0
    Fupper=0
end
    P_CSBO1=triplequad(@INTEGRANDv1pt0,A,B,C,D,E,Fupper,EtolInd) %
Recommended absolute error tolerance
% is 1.0e-10. See "Verification 2" for the basis for this
recommendation.

if (n==2) || (n==3)
%% Next, 2-way common-cause failure plus an independent failure
% P_CSBO2=0 % Stub
% Convert Riemann-Stieltjes double nonrecovery integral to ordinary
Riemann
% integral
A=0;
% B=F_0(1)+(1-F_0(1))*feval(F{1},T,alphaF(1),betaF(1),gammaF(1)); %BEs
Independent
B=F_0(1)+feval(F{1},T,alphaF(1),betaF(1),gammaF(1)); %BEs Exclusive
C=0;
% D=P2FTS+(1-P2FTS)*feval(FTR2way,T,alpha2way,beta2way,gamma2way); %
Independent

```

```

D=P2FTS+feval(FTR2way,T,alpha2way,beta2way,gamma2way); % Exclusive
P_CSBO2=3.0*dblquad(@INTEGRAND2way,A,B,C,D,ETol2way) % Recommended
absolute error tolerance
% is 1.0e-10. See "Verification 2" for the basis for this
recommendation.
% Three for multiplicity
end
%% Finally, with 3-way Common-Cause Failures

if (n==3)
% 3-Way failure to start and to run. In the terminology of the
asymptotic development
% this includes the S3 and R3 classes of failure.
A=0;
% B=FTS3way+(1-FTS3way)*feval(FTR3way,T,alpha3way,beta3way,gamma3way);
%Independent
B=FTS3way+feval(FTR3way,T,alpha3way,beta3way,gamma3way); %Mutually
exclusive
P_CSBO3=quad(@INTEGRAND3way,A,B,ETol3way)
end

if (n==1)
P_CSBO2=0;
P_CSBO3=0;
else if (n==2)
P_CSBO3=0
else if(n>3)
ERROR_N = fprintf('Too many Trains to compute with this code, the
Maximum is 3');
end
end
end
%% Total
P_CSBO=P_CSBO1 + P_CSBO2 + P_CSBO3

%*****
****

%Creates an Array of P_CSBO values for the output file
outputP(i)= P_CSBO;
outputP1(i)= P_CSBO1;
outputP2(i)= P_CSBO2;
outputP3(i)= P_CSBO3;

%Variable to go to display the correct data row in the Output File
i2=i-1;
%Array that defines the Title Cells for the Output File
stringout = {'Data Row', 'P_CSBO', 'P_CSBO1', 'P_CSBO2','P_CSBO3'};
%Fills the Title Cells in the Output File
xlswrite('OutputFile.xlsx', stringout, 1, 'A1');
%Fills the Data Row column with the specific data row number
xlswrite('OutputFile.xlsx', i2, 1, sprintf('A%d', i));
%Fills the P_CSBO Column with values in the output file
xlswrite('OutputFile.xlsx', outputP(i), 1, sprintf('B%d', i));
%Fills the P_CSBO1 Column with values in the output file
xlswrite('OutputFile.xlsx', outputP1(i), 1, sprintf('C%d', i));
%Fills the P_CSBO2 Column with values in the output file

```

```

xlswrite('OutputFile.xlsx', outputP2(i), 1, sprintf('D%d', i));
%Fills the P_CSBO3 Column with values in the output file
xlswrite('OutputFile.xlsx', outputP3(i), 1, sprintf('E%d', i));

end
i = i+1;
end
% End of Program Message
disp('END OF PROGRAM      WHOOP!');

```

“INTEGRANDv1pt0.m”:

```

function H=INTEGRANDv1pt0(F1,F2,F3)
global F_0 F Finv alphaF betaF gammaF deltaF n G T_c G alphaG betaG
gammaG deltaG R tau alphaR betaR gammaR deltaR
global RecoveryType

tau=cell(1,n);

%% Line 5
% Identify corresponding times, tau1, tau2, tau3
C1=F1<F_0(1);
[m1 n1]=size(F1);
for i=1:m1
    for j=1:n1
        if C1(i,j)==1
            tau{1}(i,j)=0;
        else
            tau{1}(i,j)=feval(Finv{1,1},((F1(i,j)-F_0(1))./(1-
F_0(1))),alphaF(1),betaF(1),gammaF(1),deltaF(1));
        end
    end
end
C2=F2<F_0(2);
[m2 n2]=size(F2);
for i=1:m2
    for j=1:n2
        if C2(i,j)==1
            tau{2}(i,j)=0;
        else
            tau{2}(i,j)=feval(Finv{1,2},((F2(i,j)-F_0(2))./(1-
F_0(2))),alphaF(2),betaF(2),gammaF(2),deltaF(2));
        end
    end
end
C3=F3<F_0(3);
[m3 n3]=size(F3);
for i=1:m3
    for j=1:n3
        if C3(i,j)==1
            tau{3}(i,j)=0;
        else
            tau{3}(i,j)=feval(Finv{1,3},((F3(i,j)-F_0(2))./(1-
F_0(3))),alphaF(3),betaF(3),gammaF(3),deltaF(3));
        end
    end
end

```

```

    end
end

H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

if (RecoveryType==1)

%%
% Determine taumax
taumax=tau{1};
for i=2:n
    taumax=max(taumax,tau{i});
end
% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio
% paper
H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:n
    % H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
end
else if (RecoveryType==2)

%%
% Determine taumax
taumax=tau{1};
for i=2:n
    taumax=max(taumax,tau{i});
end
% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio
% paper
H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:n
    % H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));

```

```

end
    else if (RecoveryType==3)
    % Determine taumax
    taumax=tau{1};
    for i=2:n
        taumax=max(taumax,tau{i});
    end
    % Now take credit for recovery of offsite power, as in Section 3.4 of
    Rio
    % paper
    H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
    % Now take credit for recovery of emergency power trains
    for i=1:n
        % H=H.*(1-feval(R{i},taumax+T_c-
    tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
        % Alternately take only credit for recovery of emergency power
    trains that
        % comes from repair work following full failure of onsite power
    (per
        % Lloyd-Anoba, for the case coincident with Read-Fleming)
        H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    end
        else if (RecoveryType==4)

%%
    % Determine taumax
    taumax=tau{1};
    for i=2:n
        taumax=max(taumax,tau{i});
    end
    % Now take credit for recovery of offsite power, as in Section 3.4 of
    Rio
    % paper
    H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
    % Now take credit for recovery of emergency power trains
    for i=1:n
        H=H.*(1-feval(R{i},taumax+T_c-
    tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
        % Alternately take only credit for recovery of emergency power
    trains that
        % comes from repair work following full failure of onsite power
    (per
        % Lloyd-Anoba, for the case coincident with Read-Fleming)
        % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    end
    end
    end
    end
end

```

“INTEGRAND2way.m”:

```

function H=INTEGRAND2way(F1,F2)
global outputP F_0 F Finv alphaF betaF gammaF deltaF n T_c G alphaG
betaG gammaG deltaG R tau alphaR betaR gammaR deltaR

```

```

global RecoveryType2

tau=cell(1,n);

%%
% Identify corresponding times, tau1, tau2
C1=F1<F_0(1);
[m1 n1]=size(F1);
for i=1:m1
    for j=1:n1
        if C1(i,j)==1
            tau{1}(i,j)=0;
        else
            tau{1}(i,j)=feval(Finv{1,1},((F1(i,j)-F_0(1))./(1-
F_0(1))),alphaF(1),betaF(1),gammaF(1),deltaF(1));
        end
    end
end
C2=F2<F_0(2);
[m2 n2]=size(F2);
for i=1:m2
    for j=1:n2
        if C2(i,j)==1
            tau{2}(i,j)=0;
        else
            tau{2}(i,j)=feval(Finv{1,2},((F2(i,j)-F_0(2))./(1-
F_0(2))),alphaF(2),betaF(2),gammaF(2),deltaF(2));
        end
    end
end

%*****CREDIT FOR
RECOVERY*****

if (RecoveryType2==1)

H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%
% Determine taumax
taumax=tau{1};
taumax=max(taumax,tau{i});

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio
% paper
%H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
%for i=1:2

```

```

    % H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
%end
else if (RecoveryType2==2)

H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%
% Determine taumax
taumax=tau{1};
taumax=max(taumax,tau{i});

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio
% paper
H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:2
    % H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
end

else if (RecoveryType2==3)

H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%
% Determine taumax
taumax=tau{1};
taumax=max(taumax,tau{i});

```

```

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio
% paper
H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:2
    % H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming)
    H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
end
else if (RecoveryType2==4)

H=ones(m1,n1); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%
% Determine taumax
taumax=tau{1};
taumax=max(taumax,tau{i});

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio
% paper
H=H.*(1-feval(G,taumax+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:2
    H=H.*(1-feval(R{i},taumax+T_c-
tau{i},alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
end
    end
    end
end
end
end

```

“INTEGRAND3way.m”:

```

function H=INTEGRAND3way(F3way)
global outputP F_0 F Finv alphaF betaF gammaF deltaF n T_c G alphaG
betaG gammaG deltaG R tau alphaR betaR gammaR deltaR
global FTS3way FTR3way FTR3wayinv alpha3way beta3way gamma3way
delta3way

```



```

global RecoveryType3

tau=cell(1,n); %??

%%
% Identify corresponding times, tau3way
C3way=F3way<FTS3way;
[m3way n3way]=size(F3way);
tau3way=zeros(m3way,n3way);
for j=1:n3way
    if C3way(1,j)==1
        tau3way(1,j)=0;
    else
        tau3way(1,j)=feval(FTR3wayinv,(F3way(1,j)-FTS3way)/(1-
FTS3way),alpha3way,beta3way,gamma3way,delta3way);
        % HERE Above assignment is giving empty matrix rather than a
scalar
        % value
    end
end

if (RecoveryType3==1)

H=ones(m3way,n3way); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio paper
%H=H.*(1-feval(G,tau3way+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:n
    % Recovery function
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming; for the
present 3-way CCF
    % case this Lloyd-Anoba treatment is no different from full credit
for repair, because
    % all (30 trains fail at the same time)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i)));
end
else if (RecoveryType3==2)

H=ones(m3way,n3way); % Correct if credit taken only for nonfailure of
emergency

```

```

% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio paper
H=H.*(1-feval(G,tau3way+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:n
    % Recovery function
    %H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming; for the
present 3-way CCF
    % case this Lloyd-Anoba treatment is no different from full credit
for repair, because
    % all (30 trains fail at the same time)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i)));
end
    else if (RecoveryType3==3)

H=ones(m3way,n3way); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the 'multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio paper
H=H.*(1-feval(G,tau3way+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:n
    % Recovery function
    %H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming; for the
present 3-way CCF
    % case this Lloyd-Anoba treatment is no different from full credit
for repair, because
    % all (30 trains fail at the same time)
    H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i)));
end
    else if (RecoveryType3==4)

```

```

H=ones(m3way,n3way); % Correct if credit taken only for nonfailure of
emergency
% trains, as in approach of the "multiple trains, no recovery" section
of
% the Rio paper. Looks o.k. for F = exponential integral, but need to
% compare to ICONE18 paper and to Rio paper

%%

% Now take credit for recovery of offsite power, as in Section 3.4 of
Rio paper
H=H.*(1-feval(G,tau3way+T_c,alphaG,betaG,gammaG,deltaG));
% Now take credit for recovery of emergency power trains
for i=1:n
    % Recovery function
    H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i),deltaR(i)));
    % Alternately take only credit for recovery of emergency power
trains that
    % comes from repair work following full failure of onsite power
(per
    % Lloyd-Anoba, for the case coincident with Read-Fleming; for the
present 3-way CCF
    % case this Lloyd-Anoba treatment is no different from full credit
for repair, because
    % all (30 trains fail at the same time)
    % H=H.*(1-feval(R{i},T_c,alphaR(i),betaR(i),gammaR(i)));
end
        end
    end
end
end

```

“FTR6890.m”:

```

function P=FTR6890(t,mu1,cutpoint,mu2,sigma)
%cdf for a rv distributed exponentially with mean mu1 for times t <=
%cutpoint, and distributed afterward, conditionally upon exceeding
cutpoint,
%exponentially with mean mu2
global cFTSLDisType1 cFTSLDisType2 cFTSLDisType3
global traincounter

%For Train 1's Distribution Type
if (traincounter == 1)
    %If time is less than or equal to the defined cutoff point
    if t<= cutpoint
        %converts the input string (From runfile) into a function
Pfun=str2func(cFTSLDisType1);
%Calls the distribution Type that is defined by the function
P=feval(Pfun,t,mu1);
    else
        %converts the input string (From runfile) into a function

        P1fun=str2func(cFTSLDisType1);

```

```

        %Calls the distribution Type that is defined by the function
using
        %cutoff point

P1=feval (P1fun, cutpoint, mu1);

        %Add the previous one minus the previous times to the cdf that
        % uses the distribution Type that is defined by the function
        % using time minus the cutoff point
P=P1+feval (P1fun, t-cutpoint, mu2);
end

else
        %For Train 2 (Same comments as above but for Train 2)
if (traincounter == 2)
        if t<= cutpoint
                Pfun=str2func (cFTSLDisType3);
                P=feval (Pfun, t, mu1);
        else
                P1fun=str2func (cFTSLDisType3);
                P1=feval (P1fun, cutpoint, mu1);
                P=P1+feval (P1fun, t-cutpoint, mu2);
        end
end
else
        %For Train 3 (Same comments as above but for Train 3)
if t<= cutpoint
        Pfun=str2func (cFTSLDisType3);
        P=feval (Pfun, t, mu1);
else
        P1fun=str2func (cFTSLDisType3);
        P1=feval (P1fun, cutpoint, mu1);
        P=P1+feval (P1fun, t-cutpoint, mu2);
end
end
end
end

```

“FTR6890inv.m”:

```

function t=FTR6890inv (P, mu1, cutpoint, mu2, sigma)
%CDF for a rv distributed exponentially with mean mu1 for times t <=
%cutpoint, and distributed afterward, conditionally upon exceeding
cutpoint,
%exponentially with mean mu2
%
global invFTSLDisType1 invFTSLDisType2 invFTSLDisType3 cFTSLDisType1
cFTSLDisType2 cFTSLDisType3
global traincounter

%-----
%This is so one can enter only one distribution type for this entire
code
%For example, If its logninv then logncdf is also used within this
script

```

```

%Removes Inverse and adds Cdf due to Custom Distribution Code for Train
1
rinv1= strtok(invFTSLDisType1, 'inv');
adcdf1= strcat(rinv1, 'cdf');

%Removes Inverse and adds Cdf due to Custom Distribution Code for Train
2
rinv2= strtok(invFTSLDisType2, 'inv');
adcdf2= strcat(rinv2, 'cdf');

%Removes Inverse and adds Cdf due to Custom Distribution Code for Train
3
rinv3= strtok(invFTSLDisType3, 'inv');
adcdf3= strcat(rinv3, 'cdf');
%-----

%For Train 1
if (traincounter == 1)
    %converts the input string (From runfile) into a
function
    Pcutfun=str2func(adcdf1);
    %Calls the distribution Type that is defined by the function
using
    %cutoff point and creates a new P cutoff
    Pcut=feval(Pcutfun,cutpoint,mu1);
    %If Input P is less than P cut off
if P<=Pcut
    %converts the input string (From runfile) into a function
    tfun = str2func(invFTSLDisType1);
    %Calls the distribution Type that is defined by the function using
    %the Probability parameter point
    t=feval(tfun,P,mu1);
else
    %converts the input string (From runfile) into a function
    Pcutfun=str2func(adcdf1);
    %Pcut off distribution run again
    Pcut=feval(Pcutfun,cutpoint,mu1);
    %Creates a new P based off the difference and 1-Pcut
    Pexcess=(P-Pcut);
    %converts the input string (From runfile) into a function
    tfun = str2func(invFTSLDisType1);
    %takes cutpoint and adds the tfun distribution type with mu2
    %parameter
    t=cutpoint+feval(tfun,Pexcess,mu2);
end

else if (traincounter == 2)          %For Train 2 (Same comments as above
but for Train 2)
    Pcutfun=str2func(adcdf2);
    Pcut=feval(Pcutfun,cutpoint,mu1);
if P<=Pcut
    tfun = str2func(invFTSLDisType2);
    t=feval(tfun,P,mu1);

else

```

```

Pcutfun=str2func(adcdf2);
Pcut=feval(Pcutfun,cutpoint,mu1);
Pexcess=(P-Pcut);
    tfun = str2func(invFTSLDisType2);
    t=cutpoint+feval(tfun,Pexcess,mu2);
end
    else %For Train 3 (Same comments as above but for Train 3)
    Pcutfun=str2func(adcdf3);
    Pcut=feval(Pcutfun,cutpoint,mu1);
if P<=Pcut
    tfun = str2func(invFTSLDisType3);
    t=feval(tfun,P,mu1);
else
    Pcutfun=str2func(adcdf3);
    Pcut=feval(Pcutfun,cutpoint,mu1);
    Pexcess=(P-Pcut);
    tfun = str2func(invFTSLDisType3);
    t=cutpoint+feval(tfun,Pexcess,mu2);
end
    end
end
end

```

“FTR6890pdf.m”:

```

function P=FTR6890pdf(t,mu1,cutpoint,mu2,mu3)
%pdf for a rv distributed exponentially with mean mu1 for times t <=
%cutpoint, and distributed afterward, conditionally upon exceeding
%cutpoint,
%exponentially with mean mu2
global pdfFTSLDisType1 pdfFTSLDisType2 pdfFTSLDisType3
global traincounter

%This code is not really used and is left as an example of what one can
do
%with it.

%For Train 1's Distribution Type
if (traincounter == 1)
    %If time is less than or equal to the defined cutoff point
if t<= cutpoint
    %converts the input string (From runfile) into a function
    Pfun=str2func(pdfFTSLDisType1);
    %Calls the distribution Type that is defined by the function
    P=feval(Pfun,t,mu1);
else
    %converts the input string (From runfile) into a function
    P1fun=str2func(pdfFTSLDisType1)
    %Calls the distribution Type that is defined by the function using
    %cutoff point
    P1=feval(P1fun,cutpoint,mu1);
    %1 minus the previous times the cdf that uses the distribution Type
that is
    %defined by the function using time minus the cutoff point
    P=(1-P1)*feval(P1fun,t-cutpoint,mu2);

```

```
end
    %For Train 2 (Same comments as above but for Train 2)
else if (traincounter == 2)
if t<= cutpoint
    Pfun2=str2func(pdfFTSLDisType2);
    P=feval(Pfun2,t,mu1);
else

    P2fun=str2func(pdfFTSLDisType2)
    P2=feval(P2fun,cutpoint,mu1);
    P=(1-P2)*feval(P2fun,t-cutpoint,mu2);
end
else %For Train 3 (Same comments as above but for Train 3)
if t<= cutpoint
    Pfun3=str2func(pdfFTSLDisType3);
    P=feval(Pfun3,t,mu1);
else

    P3fun=str2func(pdfFTSLDisType3)
    P3=feval(P3fun,cutpoint,mu1);
    P=(1-P3)*feval(P3fun,t-cutpoint,mu2);
end

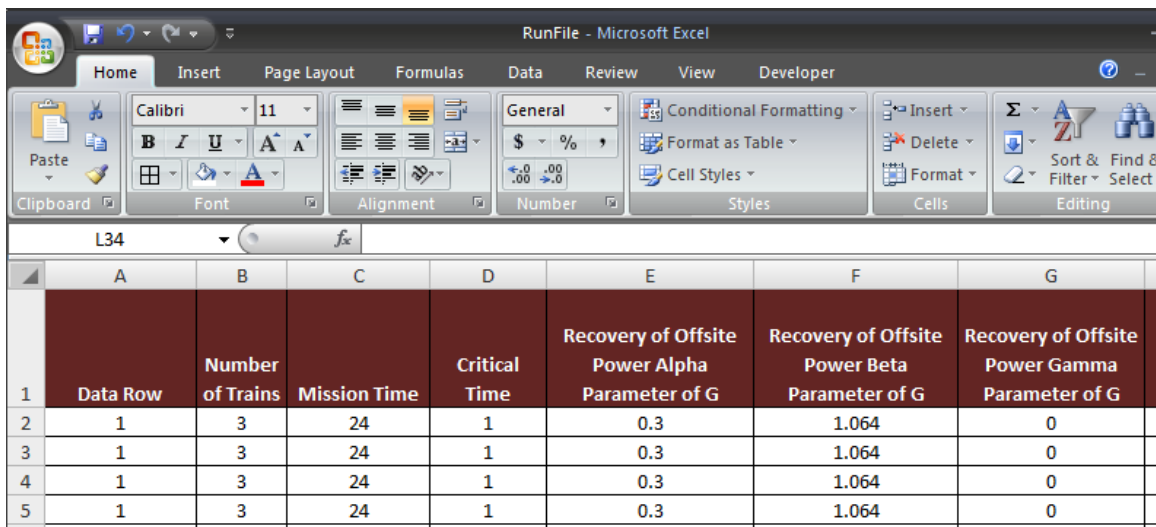
end
end
end
```

APPENDIX B

RUN-FILE EXAMPLE

Appendix B will display a full example of the Microsoft Excel file “RunFile.xlsx” that can be built in Excel that will be used as the input deck of STP-TAMIL. It is shown in Figures B.1-B.13. This is one of the examples used to determine the industrial average for three-EDGs. Note that the three cases change by the type of recovery:

- i. Data Row 1 refers to No Credit,
- ii. Data Row 2 refers to offsite only,
- iii. Data Row 3 refers to offsite plus onsite after full failure, and
- iv. Data Row 4 refers to full dynamic recovery of onsite and offsite power.



	A	B	C	D	E	F	G
1	Data Row	Number of Trains	Mission Time	Critical Time	Recovery of Offsite Power Alpha Parameter of G	Recovery of Offsite Power Beta Parameter of G	Recovery of Offsite Power Gamma Parameter of G
2	1	3	24	1	0.3	1.064	0
3	1	3	24	1	0.3	1.064	0
4	1	3	24	1	0.3	1.064	0
5	1	3	24	1	0.3	1.064	0

Figure B.1 - First Part of RunFile example

	G	H	I	J	K
	Recovery of Offsite Power Gamma Parameter of G	Recovery of Offsite Power Delta Parameter of G	Probability of Failure to Start (3CCF)	Probability of Failure to Load and Run (3CCF)	Error Tolerance for single integral to compute CC Triplet failure probability
1	0	0	5.36687E-05	1.41818E-05	1.00E-14
2	0	0	5.36687E-05	1.41818E-05	1.00E-14
3	0	0	5.36687E-05	1.41818E-05	1.00E-14
4	0	0	5.36687E-05	1.41818E-05	1.00E-14
5	0	0	5.36687E-05	1.41818E-05	1.00E-14

Figure B.2 - Second Part of RunFile example

	L	M	N	O	P
	Alpha Parameter for CC Triplet	Beta Parameter for CC Triplet	Gamma Parameter for CC Triplet	Delta Parameter for CC Triplet	Probability CC Doublet CC Failure to Start (2CCF)
1	7.05E+04	1	1.13E+05	0	3.301314E-05
2	7.05E+04	1	1.13E+05	0	3.301314E-05
3	7.05E+04	1	1.13E+05	0	3.301314E-05
4	7.05E+04	1	1.13E+05	0	3.301314E-05
5	7.05E+04	1	1.13E+05	0	3.301314E-05

Figure B.3 - Third Part of RunFile example

	Q	R	S	T
1	Probability of CC Doublet CC Failure to Load and Run (ZCCF)	Error Tolerance for double integral to compute CC Doublet failure probability	Alpha Parameter for CC Doublet	Beta Parameter for CC Doublet
2	2.552729E-05	1.00E-12	3.92E+04	1
3	2.552729E-05	1.00E-12	3.92E+04	1
4	2.552729E-05	1.00E-12	3.92E+04	1
5	2.552729E-05	1.00E-12	3.92E+04	1

Figure B.4 - Fourth Part of RunFile example

	U	V	W	X
1	Gamma Parameter for CC Doublet	Delta Parameter for CC Doublet	Probability of Three Independent Failures to Start (Ind.)	Probability of Three Independent Failures to Load and Run (Ind.)
2	1.88E+05	0	4.410305E-03	1.42E-05
3	1.88E+05	0	4.410305E-03	1.42E-05
4	1.88E+05	0	4.410305E-03	1.42E-05
5	1.88E+05	0	4.410305E-03	1.42E-05

Figure B.5 - Fifth Part of RunFile example

	Y	Z	AA	AB	AC
1	Error Tolerance for Triple Integral to compute Independent failure probability	Alpha Parameter for Independent	Beta Parameter for Independent	Gamma Parameter for Independent	Delta Parameter for Independent
2	1.00E-10	3.52E+02	1	1.21E+03	0
3	1.00E-10	3.52E+02	1	1.21E+03	0
4	1.00E-10	3.52E+02	1	1.21E+03	0
5	1.00E-10	3.52E+02	1	1.21E+03	0

Figure B.6 - Sixth Part of RunFile example

	AD	AE	AF	AG
1	Recovery of Train 1 Alpha Parameter for R{1}	Recovery of Train 1 Beta Parameter for R{1}	Recovery of Train 1 Gamma Parameter for R{1}	Recovery of Train 1 Delta Parameter for R{1}
2	15.5	0.739	0	0
3	15.5	0.739	0	0
4	15.5	0.739	0	0
5	15.5	0.739	0	0

Figure B.7 - Seventh Part of RunFile example

	AH	AI	AJ	AK
1	Recovery of Train 2 Alpha Parameter for R{2}	Recovery of Train 2 Beta Parameter for R{2}	Recovery of Train 2 Gamma Parameter for R{2}	Recovery of Train 2 Delta Parameter for R{2}
2	15.5	0.739	0	0
3	15.5	0.739	0	0
4	15.5	0.739	0	0
5	15.5	0.739	0	0

Figure B.8 - Eighth Part of RunFile example

	AL	AM	AN	AO
1	Recovery of Train 3 Alpha Parameter for R{3}	Recovery of Train 3 Beta Parameter for R{3}	Recovery of Train 3 Gamma Parameter for R{3}	Recovery of Train 3 Delta Parameter for R{3}
2	15.5	0.739	0	0
3	15.5	0.739	0	0
4	15.5	0.739	0	0
5	15.5	0.739	0	0
6				

Figure B.9 - Ninth Part of RunFile example

	AP	AO	AA
1	Type of Recovery for 3CCF: Enter a number 1-4 (1 for No Credit [offsite/onsite], 2 for Offsite Credit Only, 3 for Offsite + Onsite Repair only after Full Failure, 4 for Full Recovery)	Type of Recovery for 2CCF: Enter a number 1-4 (1 for No Credit [offsite/onsite], 2 for Offsite Credit Only, 3 for Offsite + Onsite Repair only after Full Failure, 4 for Full Recovery)	Type of Recovery for Independent Failure: Enter a number 1-4 (1 for No Credit [offsite/onsite], 2 for Offsite Credit Only, 3 for Offsite + Onsite Repair only after Full Failure, 4 for Full Recovery)
2	1	1	1
3	2	2	2
4	3	3	3
5	4	4	4

Figure B.10 - Tenth Part of RunFile example

	AS	AT	AU	AV
1	Type of Distribution for G, cdf in time for recovery of offsite power	Type of Distribution for R, cdf in time for repair of ith train	Type of Distribution for Failure to Load and Run cdf in time for ith train	File Generated for CC Triplet Failure to Load and Run cdf
2	logncdf	wblcdf	expcdf	FTR6890
3	logncdf	wblcdf	expcdf	FTR6890
4	logncdf	wblcdf	expcdf	FTR6890
5	logncdf	wblcdf	expcdf	FTR6890

Figure B.11 - Eleventh Part of RunFile example

The screenshot shows the Microsoft Excel interface with the 'Home' tab selected. The active cell is AV14. The table below is displayed in the worksheet.

	AW	AX	AY
1	File Generated for CC Triplet Failure to Load and Run inverse cdf	File Generated for CC Doublet Failure to Load and Run cdf	File Generated for CC Doublet Failure to Load and Run inverse cdf
2	FTR6890inv	FTR6890	FTR6890inv
3	FTR6890inv	FTR6890	FTR6890inv
4	FTR6890inv	FTR6890	FTR6890inv
5	FTR6890inv	FTR6890	FTR6890inv

Figure B.12 - Twelfth Part of RunFile example

The screenshot shows the Microsoft Excel interface with the 'Home' tab selected. The active cell is BB9. The table below is displayed in the worksheet.

	AZ	BA
1	File Used for Independent Failure to Load and Run cdf	File Used for Independent Failure to Load and Run inverse cdf
2	FTR6890	FTR6890inv
3	FTR6890	FTR6890inv
4	FTR6890	FTR6890inv
5	FTR6890	FTR6890inv

Figure B.13 - Thirteenth Part of RunFile example

APPENDIX C
LETTER FROM STPNOC

Appendix C is a letter from South Texas Project Nuclear Operating Company describing the confidentiality notice. It also shows that STP verified that the results obtained from the code were based off STP data and that the code verified correctly. The rest of this page is intentionally left blank as the letter appears on the next page.



South Texas Project Electric Generating Station P.O. Box 289 Wadsworth, Texas 77483

"Confidentiality Notice for STP-TAMIL Values and Authorization of Validation"

This document is written in concurrence with the thesis entitled "Algorithms for Incorporation of Dynamic Recovery in Estimating Frequency of Critical Station Blackout".

South Texas Project Nuclear Operating Company (STP or STPNOC) sent results from the probabilistic risk assessment (PRA) model developed for a loss of offsite power (LOOP) initiating event to the Texas A&M University's research team consisting of, but not limited to, Paul Rodi and Dr. Paul Nelson. These results were used to validate the MATLAB code developed by Paul Rodi and Dr. Paul Nelson entitled "STP-TAMIL", which stands for "South Texas Project – Texas A&M Improved LOOP" code.

Due to confidentiality, the exact values are withheld. South Texas Project agrees that the validation was completed correctly using the STP PRA System Notebook numbers. The System Notebook refers to numbers generated by the PRA Program that correctly represents STPNOC commercial nuclear power units.

In the validation found within the thesis, values for STP probabilities and the specific cut sets for how those probabilities lead to core damage during a LOOP initiating event are used. The cut sets used to create a total cut set failure probability match what is expected from the STP System Notebook with the deviations discussed in the thesis. STPNOC is in consensus with the top four cut sets involving a 3/3 common cause event. These cut sets involve the emergency diesel generators and the fans used to provide cooling to those generators. We, STPNOC, also agree that the estimations used are representative of the top 100 minimal cut set list of independent, 3/3, 2/3 common-cause and total failures. Total failures are defined as the addition of probabilities for 3/3 and 2/3 common-cause failure events in addition to independent failure cut sets probabilities.

It is also agreed upon that the values used for variables X, Y, W, Z, T correspond with their specific usage in relation to the STP System Notebook results. These are defined as

X = 3/3 CC contribution

Z = 2/3 CC contribution

Y = Independent contribution

W = Total Contribution of 3/3 CC and Independent

T = Total Contribution (3/3 CC, 2/3 CC and Independent)

Printed Name SHAWN RODI Signature Shawn Rodi Date 3-6-2012

VITA

Name: Paul J. Rodi

Address: 337 Zachry Engineering Building
3133 TAMU
College Station, TX 77843-3133

Email Address: RODIPJ@TAMU.EDU

Education: B.S., Mechanical Engineering, Virginia Commonwealth University, 2010
M.S., Nuclear Engineering, Texas A&M University, 2012