

AMERICAN UNIVERSITY OF BEIRUT

DEVELOPING AN INVERSE ANALYSIS FRAMEWORK
FOR QUANTIFYING BLAST LOADS BY INSPECTING THE
CORRESPONDING STRUCTURAL DAMAGES

by
ZAYNAB GHASSAN KANSOUN

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Civil and Environmental Engineering
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

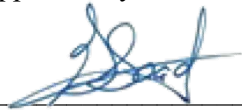


Beirut, Lebanon
April 2022

AMERICAN UNIVERSITY OF BEIRUT

Developing An Inverse Analysis Framework For Quantifying
Blast Loads By Inspecting The Corresponding Structural
Damages

by
ZAYNAB GHASSAN KANSOUN

Approved by:

	Signature
Dr. George Saad, Associate Professor Civil and Environmental Engineering	Advisor
	Signature
Dr. Mounir Mabsout, Professor Civil and Environmental Engineering	Member of Committee
	Signature
Dr. Mayssa Dabaghi, Assistant Professor Civil and Environmental Engineering	Member of Committee

Date of thesis defense: April 01 , 2022

ACKNOWLEDGEMENTS

{ يَرْفَعُ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ { المجادلة ١١

First and foremost, I would like to thank God, the almighty, for granting the wisdom, and determination to overrun all the trials throughout this study. Without his grace and support, this work would not have been possible. The special guidance of Imam Mahdi turned out the tough moments of this journey into a great achievement.

A profound sense of gratitude goes to my supervisor, Dr. George Saad, for his excellent encouragement and assistance. I am extremely grateful for his exceptional knowledge and experience. He is a real inspirational leader.

Thanks for Dr. Mayssa Dabaghi and Dr. Mounir Mabsout, for accepting to be part of my thesis committee. Their valuable revisions and comments have been very important for taking up this work to a more professional level.

This whole success is because I have the greatest family. My father, mother, and two sisters are always there for me. Thanks for my sister, Alaa, for proofreading anytime. I am forever appreciative for their unconditional love which helped surviving all the stress and not letting me give up on the work.

I am indebted to the countless people who believed in me throughout this long process. They were my real source of motivation. My friends never failed to encourage me. Also, I would never forget the inspirational environment my second family at work has offered.

Thanks to the American University of Beirut (AUB) for giving the chance to pursue my Master's degree in such a wonderful environment. AUB provided all the necessary resources to achieve the desired research outcome.

ABSTRACT OF THE THESIS OF

Zaynab Ghassan Kansoun

for

Master of Engineering

Major: Civil Engineering

Title: Developing an Inverse Analysis Framework for Quantifying Blast Loads by Inspecting the Corresponding Structural Damages

The growth in the number of terrorist attacks and accidental explosive events imposed additional demand on post-blast investigation tools. Numerous research addressed the chemical analysis of explosive residuals. Yet, limited work in the literature adhered to quantifying the blast load depending on the existing structural damage. The structural damage assessment is a useful data collection procedure for the post-blast analysis. This research proposes an Inverse approach that can estimate the explosive charge weight based on analyzing post-blast structural damage. The newly proposed approach is based on iterative finite element (FE) analysis-using Abaqus commercial package- and involves digital image processing techniques built-in Matlab software. Furthermore, to effectively identify the size of the blast, the genetic algorithm (GA) is used for optimization. At the end of each iteration, the damage contour plot is exported from Abaqus odb file through a python script. Afterward, the binary version of both the actual damage image and the FE damage plot is investigated and compared with the aim to minimize the difference between the real state and finite element solution. The image analysis is performed by virtue of edge detection operators available in Matlab. Image comparison is done depending on the Complex Wavelet- Structural Similarity Index (CW-SSIM). The concrete behavior under blast loading is represented using the Johnson- Holmquist-2 (Jh-2) damage constitutive material model. Likewise, the blast load effect is described through the Conventional Weapons Effects Blast Loading (CONWEP) model. Finally, the approach is validated using two toy examples starting with a forward FE solution as being the real damage state. The framework was able to predict the detonation mass with an accuracy between 97 to 99%.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	1
ABSTRACT	2
ILLUSTRATIONS	6
TABLES	9
INTRODUCTION.....	10
1.1. Motivation and Background	10
LITERATURE REVIEW.....	14
2.1. Blast Phenomenon	14
2.1.1. Blast Wave:.....	14
2.1.2. Explosion Categories:	15
2.1.3. Blast Wave Physical Interactions:	18
2.1.4. Blast Wave Scaling:.....	22
2.2. Time History of Blast Wave Pressure.....	24
2.3. Prediction of Blast Wave Parameters Using Theoretical and Empirical Methods	25
2.4. Structural Response Under Blast Loading.....	30
2.5. Prediction of Blast Wave Parameters Using Computational Techniques.....	33
2.6. Material Properties Under Dynamic Loading Conditions	35
2.7. Finite Element Modeling and Analysis	39

2.8.Finite Element Modeling of Reinforced Concrete Structures Subjected to Blast Loading Using ABAQUS	42
2.8.1. Element Formulation	42
2.8.2. Material Modeling	44
2.8.3. Load Definition.....	47
2.9.Heuristic Optimization Methods.....	48
2.9.1.Genetic Algorithm Optimization	49
2.10.Image Processing Techniques.....	52
2.10.1.Image Processing Techniques in structural Analysis	53
2.10.2.Edge Detection Techniques in MATLAB	54
2.10.3.Similarity Indices Between Images	55
METHODOLOGY	58
3.1. Finite Element Modeling and Calibration	58
3.1.1. Experiment Description	58
3.1.2. Element Formulation	60
3.1.3.Material Modeling	61
3.1.4.Load Definition.....	64
3.2.Creating Toy Problem for Concept Validation.....	65
3.2.1.Model Description	65
3.2.2. ABAQUS Input File Parameterization	66
3.3 Simulated Damage Output & Odb Scripting	67
3.4. Image Processing Using MATLAB.....	67
3.4.1 .Implementing Edge Detection Algorithm Using Modified Canny Edge Detector.....	67
3.4.2.Implementing Complex Wavelet Structural Similarity Index (CW-SSIM) ..	68
3.5.Implementing Genetic Algorithm Optimization.....	72
4.1.Initial Guess Lower than the Real State Detonation Mass	80

4.2. Initial Guess Higher than the Real State Detonation Mass.....	82
CONCLUSION AND FUTURE WORK.....	85
APPENDIX 1	87
APPENDIX 2	88
APPENDIX 3	90
REFERENCES.....	97

ILLUSTRATIONS

Figure

1. Framework General Approach.....	13
2. Blast Phenomenon	15
3. Free Air Burst Explosion (Shirbhate & Goel, 2020)	16
4. Air Burst Explosion (Shirbhate & Goel, 2020)	17
5. Surface Burst Explosion (Shirbhate & Goel, 2020)	17
6. Fully Vented Explosion (Shirbhate & Goel, 2020)	17
7. Partially Confined Explosion(Shirbhate & Goel, 2020).....	18
8. Totally Confined Explosion (Shirbhate & Goel, 2020).....	18
9. Mach Stem Phenomenon (Shirbhate & Goel, 2020)	19
10. Blast Wave Diffraction (Shirbhate & Goel, 2020)	20
11. Transmission Scenarios	21
12. Scaled distance of blasts with different charge weights and standoff distances (Ne, 1998)	22
13. Pressure-Time History of Blast wave (Defense, 2008)	25
14. Positive phase parameters for spherical free air blast (Defense, 2008).....	28
15. Negative phase parameters for spherical free air blast (Defense, 2008)	29
16. Idealization of Pressure-Time History (Appuhamilage, 2015).....	30
17. Structural Response as a Function of the Pressure Intensity (Defense, 2008)	31
18. Mays and Smith Response Classification.....	31
19. Quasi-static Structural Response (Smith & Hetherington, 1994).....	32
20. Dynamic Structural Response (Smith & Hetherington, 1994)	32
21. Impulsive Structural Response (Smith & Hetherington, 1994).....	33
22. The ranges of strain rates produced by various loading (Zhang et al., 2020).	35
23. DIF of Concrete Due to Viscous and Structural Effects (Appuhamilage, 2015)	36
24. Bischoff and Perry (1991) DIF in compression Scatter.....	37

25. DIF in tension for Concrete	37
26. DIF of Reinforcement Steel According to Malvar and Crawford 1998	38
27. Approaches for Predicting Structural Response	39
28. FE Simulation Process Chart	41
29. Element Formulations in ABAQUS	42
30. Lagrangian vs. Eulerian Formulation (<i>Coupled Eulerian Lagrange (CEL) Analysis with ABAQUS</i> , n.d.).....	42
31. Plasticity Based Material Models for Concrete in Abaqus.....	45
32. Total Pressure Relation in CONWEP	48
33. Total Pressure of Blast Wave.....	48
34. Basic Implementation of GA (Hasanova, 2020).....	49
35. GA Encoding(El Khansa, 2020)	50
36. Crossover	50
37. Some Colors Encoded as RGB (Taylor, 2021).....	52
38. Crack Gauge	53
39. Edge Detection Algorithm in MATLAB	54
40. Gray-scale Image (a), Canny Method (b), Perwitt Method (c) (MathWorks, 2021)	54
41. Classes of Similarity Indices.....	55
42. Testing Apparatus Used in Wang Experiment (W. Wang et al., 2013).....	58
43. Slab Geometry in mm. (W. Wang et al., 2013)	59
44. Support Dimensions (Appuhamilage, 2015)	59
45. Geometric Model of 1/4 of the Slab	61
46. Assigned Sections in the FE Model.....	61
47. Parametric Study for Jh-2 Material Data- Midspan Deflection	62
48. Us-Up Linear Interpolation.....	64
49. Locating the Standoff Distance Reference Point in the FE Model.....	64
50. CONWEP Definition- Blast Load	65
51. ABAQUS Parameterized Input File for First Scenario	66

52. ABAQUS Parameterized Input File for Second Scenario	67
53. FSIM Values and Trend Lines.....	71
54. CW-SSIM Values and Trend Lines.....	71
55. Genetic Algorithm Integration.....	73
56. First Set of Input Data for the Algorithm	77
57. Inputting the Real Damage Image Name and Format	77
58. Cropping the Real Damage Image.....	78
59. GA Input Data.....	78
60. Specification of Minimum Threshold Value to be Displayed	79
61. Sensitivity Study Relating Population Size to Convergence Time.....	81

TABLES

Table

1. Shock Impedance Values (Schwartz, n.d.)	20
2. Heat of detonation & TNT equivalent factor of different explosives (Smith & Hetherington, 1994)	23
3. Waveform Parameter (b) (Appuhamilage, 2015)	24
4. Theoretical and Empirical Equations of Blast Wave Parameters	27
5. Couple & Uncoupled Computational Programs for Blast Prediction (Appuhamilage, 2015)	35
6. Description of Element Formulations Available in ABAQUS	44
7. Intensity Based Similarity Indices	56
8. Material Properties (W. Wang et al., 2013)	60
9. TNT Charge Characteristics	60
10. Experimental vs. Numerical Midspan Displacement Results	63
11. Mesh Size Considered for the Forward Run	65
12. Explosive's Mass in Accordance to the Displacement and Rotational Damage (Abedini & Mutalib, 2020)	72
13. Options Used for GA Implementation	75
14. Input Data	80
15. Variation of Iterated Explosive's Mass Between Generations	81
16. Input Data	82
17. Variation of Iterated Explosive's Mass Across the Generation	83

CHAPTER 1

INTRODUCTION

1.1. Motivation and Background

According to the Action on Armed Violence (AOAV), 29,485 people were killed or injured by explosive weapons in 2019 (Dathan & Overton, 2019). Beirut Port Blast killed more than 220 people, injured more than 6,500, and left around 300,000 people homeless (Facts, 2020). The tremendous loss in human lives has dragged the attention of scientists and engineers worldwide.

In the aim of quantifying blast loads, chemists concentrated on analyzing the explosive's leftover to specify the material's type that detonated (Huri et al., 2017). Physicists and engineers addressed the topic from a different perspective; the blast yield can be estimated depending on analyzing the fire ball size evolution as reported in videos taken at the moment of explosion along with applying distance–time relationships (Aouad et al., 2020; Rigby et al., 2020). Structural engineers in turn; relied on the analysis of post blast structural damage to predict the blast load. The earliest studies considered identifying the charge weight of Oklahoma City Murrah Federal Building explosion by relating the structural damage to the structural capacity and failure mechanism of the designed sections (Mlakar, Sr. et al., 1998; Sozen et al., 1998). Many studies accurately predicted and simulated the real structural damage corresponding to blast loading using finite element analysis (Jahami, 2016; Rao et al., 2018; W. Wang et al., 2013; Zhang et al., 2020). Lately, an advanced approach utilized finite element analysis to identify the amount of explosives that caused Beirut Port Blast

based on matching the number of damaged rows of the silos and the analysis output by means of a trial and error procedure (Temsah et al., 2020).

Each of the damage based approaches found in the literature suffers from its own limitation. The procedure followed in the case of the Murrah Federal Building is tedious since it requires a full capacity analysis of sections. In addition, the results were based on analyzing collapse mechanisms for gravity loading assuming a two dimensional problem with a strength increasing factor. Likewise, the method based on finite element analysis solely is time consuming and limits the iteration process to definite guesses inferred by an experienced blast engineer.

It should therefore be obvious that the proposed methodology will rely on computational intelligence. Basically, computational intelligence methods are worthwhile in reducing human interference (Chou & Ghaboussi, 2001). This research proposes an inverse analysis framework to estimate the blast load by inspecting the corresponding structural damages. The explosive charge weight optimization procedure will be attained by the genetic algorithm; which is the most used tool for heuristic optimization in widely-ranged search space and complicated problems (Basak et al., 2013; Hasanova, 2020).

The purpose of this study is to develop a computationally efficient, automated inverse analysis framework that would accurately quantify the blast load based on the corresponding structural damage.

This would enable the usage of the proposed approach as a forensic tool for post blast investigations. Moreover, it can provide a deeper understanding of the structural behavior of reinforced concrete structures when subjected to blast loading. Thus, help

engineers develop better options for the design of structures in order to avoid the total collapse and save human lives that may be lost.

Starting from the development of a FE model that describes the problem geometry, and depending on the material models available in Abaqus, the behavior of the structure under blast can be simulated.

The approach is validated against a forward FE solution which is calibrated based on existing experiments in the literature. Furthermore, the GA is used to carry out the iterative process needed to quantify the blast load automatically.

Given the versatility of the approach, it will consequently lead to a confident estimate of the magnitude of any explosion by identifying the corresponding structural damage and with the aid of finite element modeling.

The below flowchart briefly explains the general approach implemented in the framework (Figure 1).

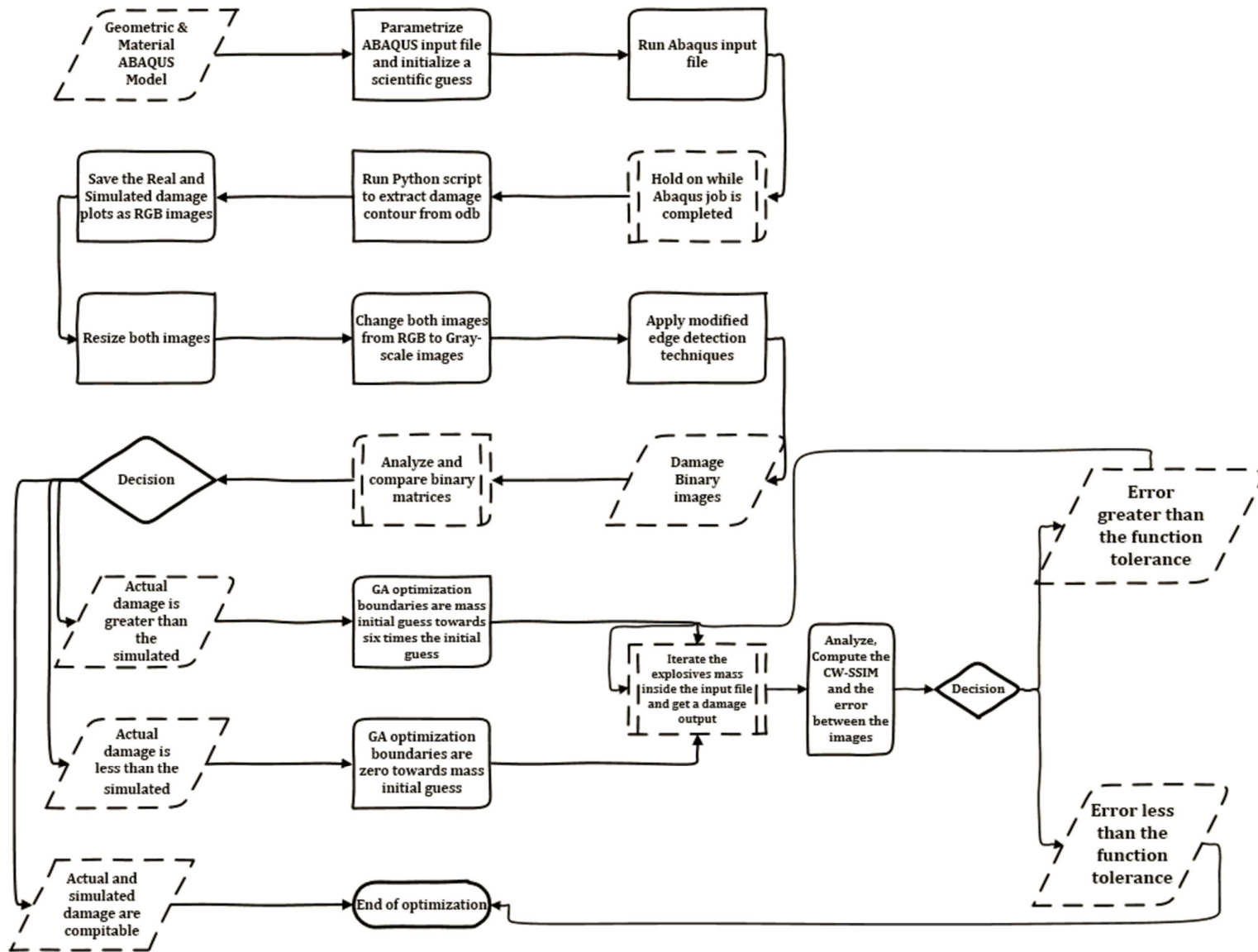


Figure 1: Framework General Approach

CHAPTER 2

LITERATURE REVIEW

2.1. Blast Phenomenon

2.1.1. Blast Wave:

Scientifically, an explosion is characterized by a sudden release of energy to the surrounding environment, and consequently causing a rise in the pressure. The detonation of an explosive material releases energy which is almost totally converted into blast energy. Thus a pressure front associated with the energy release propagates radially in the atmosphere in the form of a strong shock (blast) wave. As the blast wave travels away from the detonation center, the shock velocity diminishes maintaining an exceed of the sonic velocity of the medium. The nature of the energy release as a function of the location of the blast center and the ground specifies the shape of the blast wave (spherical-air burst/ hemispherical-surface burst). When a blast wave hits a solid surface, the wave is reflected and amplified (Castro, 2006; Defense, 2008). Figure 2 summarizes the blast phenomenon.

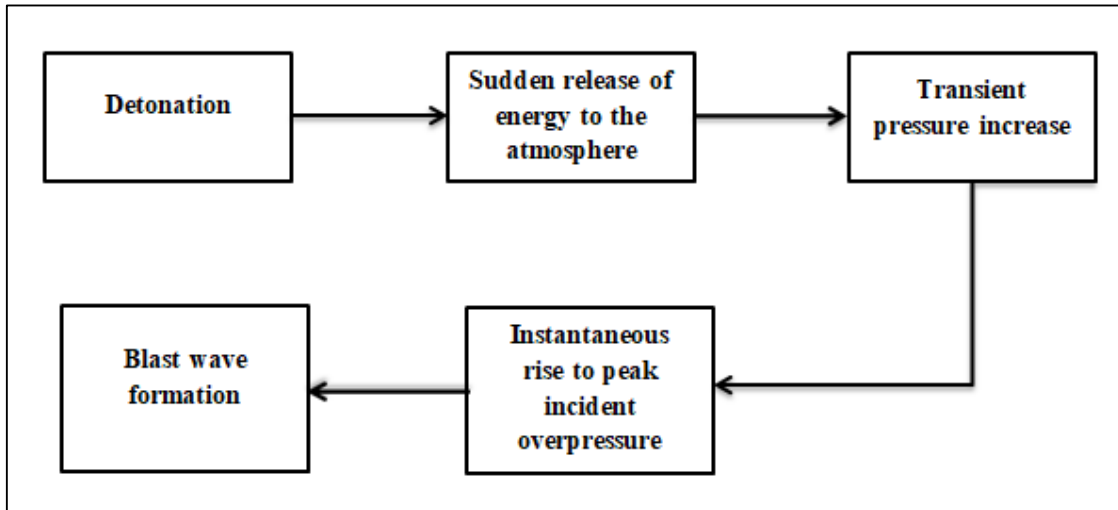


Figure 2: Blast Phenomenon

2.1.2. Explosion Categories:

➤ Unconfined Explosion:

- Free Air Burst Explosion (Figure 3):

The blast takes place in the free air without any amplifications of its wave. The output is the initial shock wave that propagates away from the detonation center (Defense, 2008).

- Air Burst Explosion (Figure 4):

The blast occurs at a distance limited to two to three times of the height of 1-2 story building above a structure in the air. Ground reflections cause wave amplification prior to the blast wave arrival (Defense, 2008).

- Surface Burst Explosion (Figure 5):

The explosive is located close to or on the ground surface permitting amplification to occur at the point of detonation (Defense, 2008).

➤ Confined Explosion:

❖ Fully Vented Explosion (Figure 6):

The explosion takes place within a containment or structure with one or more surfaces open to the atmosphere. Consequently, the shock wave is amplified before leakage (Defense, 2008).

❖ Partially Confined Explosion (Figure 7):

TM5-1300 defines a partially confined explosion as a blast within a structure having limited size openings. Thus the amplified wave is vented to the atmosphere after a period of time. This delay allow for quasi-static pressure accumulation associated with a wave having longer duration when compared to the shock wave (Defense, 2008).

❖ Totally Confined Explosion (Figure 8):

Full confinement leads to internal blast load development and a very long duration gas pressure. Due to this confinement effect, the vented pressure will be very small with approximately null effect on the outside facilities (Defense, 2008).

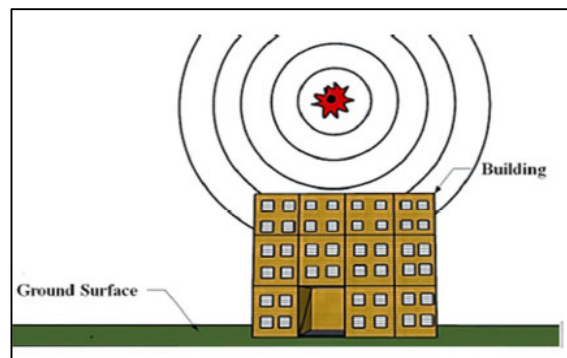


Figure 3: Free Air Burst Explosion (Shirbhate & Goel, 2020)

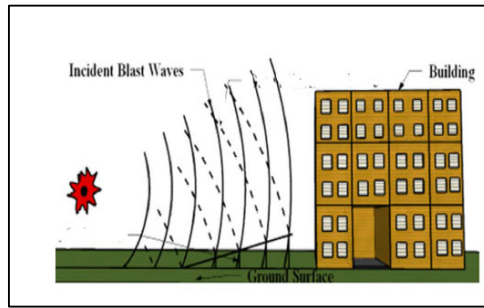


Figure 4: Air Burst Explosion (Shirbhate & Goel, 2020)

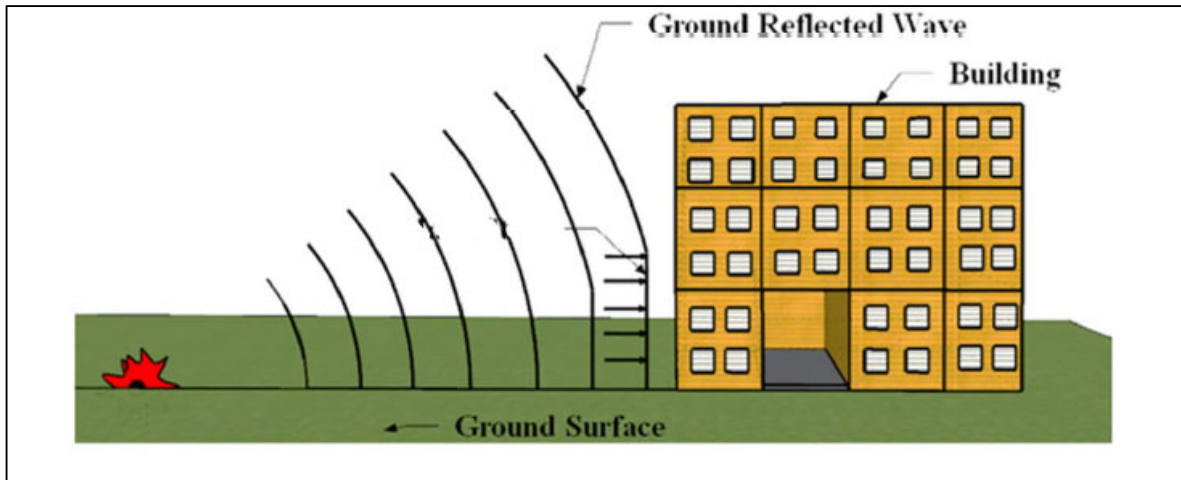


Figure 5: Surface Burst Explosion (Shirbhate & Goel, 2020)

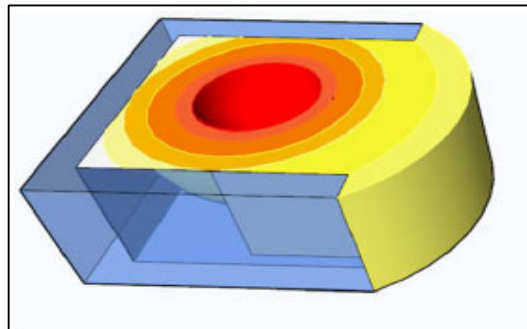


Figure 6: Fully Vented Explosion (Shirbhate & Goel, 2020)

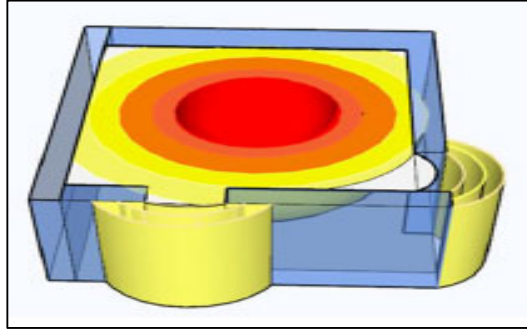


Figure 7: Partially Confined Explosion(Shirbhate & Goel, 2020)

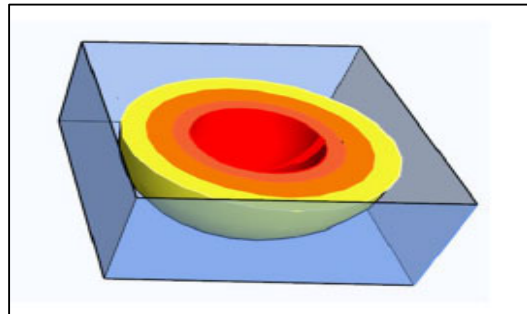


Figure 8: Totally Confined Explosion (Shirbhate & Goel, 2020)

2.1.3. Blast Wave Physical Interactions:

2.1.3.1. Reflection:

When a blast wave hits a denser target, reflection phenomenon takes place. The reflected pressure wave can be higher than that of the incident wave (Castro, 2006). As governed by the incidence angle, reflection can either be regular or Mach. When the angle of incidence is zero, the incident and reflected waves are distinct and they intersect once at the target's reflecting surface. Thus regular reflection takes place. Ernst Mach discovered experimentally that when the angle of incidence

varies between 0° and 90° , a special event occurs: Mach stem (Shirbhate & Goel, 2020). Mach stem is a single wave that merges the incident and reflected pressures (Ne, 1998).

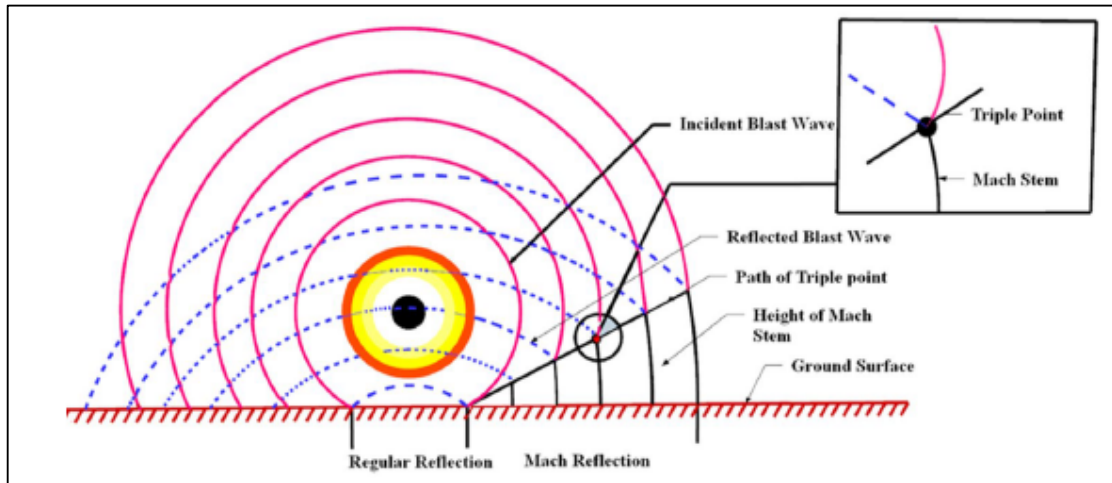


Figure 9: Mach Stem Phenomenon (Shirbhate & Goel, 2020)

The first point at which the incident and reflected wave intersects is known as the triple point. By joining the multiple triple points; the path of the triple point is specified (Figure 9) (Shirbhate & Goel, 2020).

2.1.3.2. Diffraction:

In certain cases, the blast wave may pass over an object and diffract around the target where reduction of intensity occurs (Figure 10) (Shirbhate & Goel, 2020). The height and geometry of the barrier structure influences the diffraction effect. Usually columns, piers, and re-entrant corners of buildings diffract the wave. (Alsubaei, 2015; Baumgart, 2014; Vannucci et al., 2017; W. Wang et al., 2013).

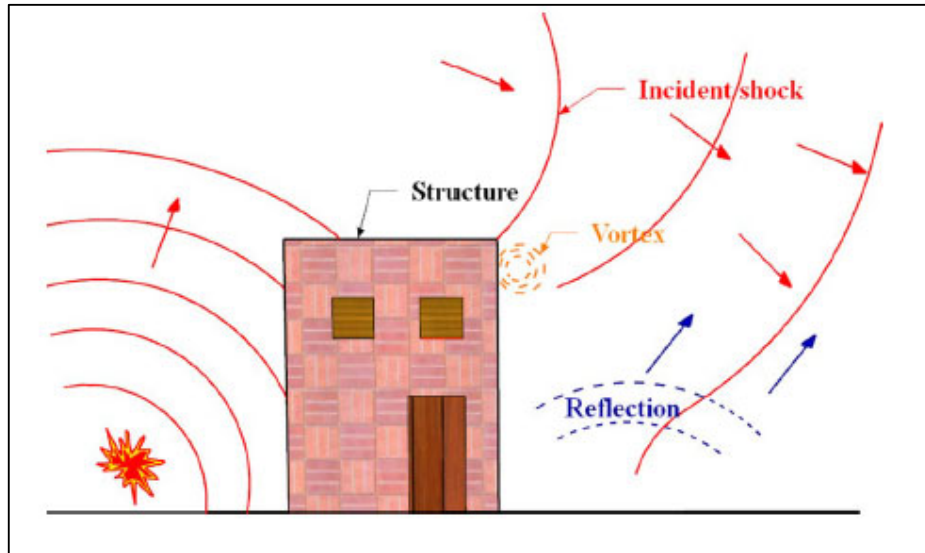


Figure 10: Blast Wave Diffraction (Shirbhate & Goel, 2020)

2.1.3.3. Transmission:

The amount of wave transmitted to an object depends on the shock impedance (z) of the material. Shock impedance is a mechanical property that controls the amount of transmitted and reflected wave. It's defined as the ratio of pressure change ΔP to the velocity change Δu (Jahami, 2016). Table 1 shows impedance values of different mediums, air's impedance is $440 \cdot 10^{-6}$ MPa.s/m (Jahami, 2016).

material	density (kg/m^3)	speed of sound (m/s)	specific impedance (MPa.s/m)
brick	2,200	4,200	9.4
concrete	1,100	3,500	3.8
steel	7,900	6,100	48
water	1,000	1,400	1.4
wood	630	3,600	2.3
rubber	1,100	100	0.11
rock	2,600	6,000	16
diamond	3,500	12,000	42
dirt	1,500	100	0.15

Table 1: Shock Impedance Values (Schwartz, n.d.)

Equations 1 and 2 represent the value of transmitted and reflected pressures respectively as function of the impedance value of mediums a and b and the incident pressure ($P'ia$) (Jahami, 2016). Figure 11 illustrates the transmission scenarios that a blast wave may undergo according to the impedance values of the mediums it's propagating through-highlighting the case of Air to Concrete blast.

$$P'b = P'ia \left[\frac{2zb}{zb+za} \right] \dots\dots\dots (1)$$

$$P'ra = P'ia \left[\frac{zb-za}{zb+za} \right] \dots\dots\dots (2)$$

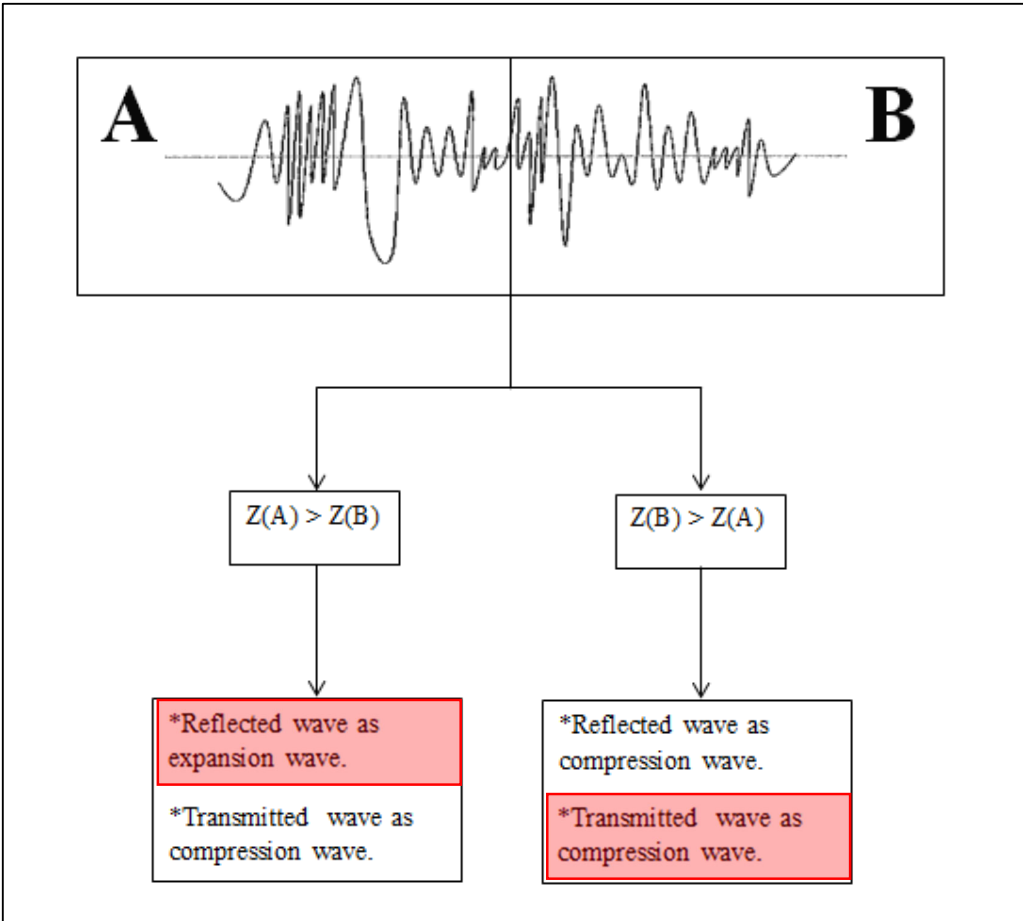


Figure 11: Transmission Scenarios

2.1.4. Blast Wave Scaling:

For the sake of comparing and analyzing different explosive events scaling laws were used. Those events may differ in terms of the explosive type, or the combination of charge weight and standoff distance between the target and detonation center.

❖ Cube-root Scaling Law:

Cube-root scaling law also known as Hopkinson-Cranz scaling law (Ohashi et al., n.d.). It depends on a scaled distance value (Z) to compare blast events in the same atmosphere but of different charge weight or standoff distances (equation 3).

$$Z = \frac{R}{W^{1/3}} \dots\dots\dots (3)$$

Where:

R = Distance between explosive charge and target (Standoff distance) (m)

W = Mass of explosive (kg)

For two explosions with same scaled distance value; the first of small charge weight with small standoff distance will have the same effect of the second explosion with larger charge weight and at a larger distance (Figure 12) (Ne, 1998).

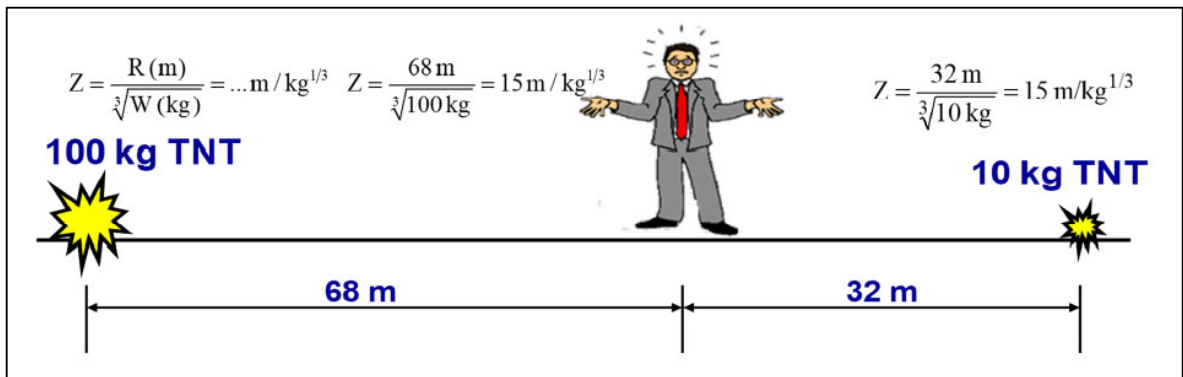


Figure 12: Scaled distance of blasts with different charge weights and standoff distances (Ne, 1998)

❖ **TNT Equivalence:**

A non-dimensional scaling factor is used to compare explosions of different explosives' type. Depending on the explosive's energy output, the factor relates explosive's weight to the equivalent weight of TNT charge (Equation 4). To consider unexpected shock wave reflections and other emergent factors, it is recommended to increase the TNT equivalent weight by 20 percent (Defense, 2008).

$$\text{TNT equivalence factor} = \frac{\text{Heat of Detonation of Explosive (Q}_x\text{)}}{\text{Heat of Detonation of TNT (Q}_{\text{TNT}}\text{)}} \dots\dots\dots (4)$$

Table 2 shows the heat of detonation and TNT equivalent factor of different explosives.

<i>Explosive</i>	<i>Mass Specific energy Q_x(kJ/kg)</i>	<i>TNT Equivalent (Q_x/Q_{TNT})</i>
Amatol 80/20 (80% ammonium nitrate 20% TNT)	2650	0.586
Compound B (60% RDX, 40% TNT)	5190	1.148
RDX (Cyclonite)	5360	1.185
HMX	5680	1.256
Lead azide	1540	0.340
Mercury fulminate	1790	0.395
Nitroglycerin (liquid)	6700	1.481
PETN	5800	1.282
Pentolite 50/50 (50% PETN 50% TNT)	5110	1.129
Tetryl	4520	1.000
TNT	4520	1.000
Torpex (42% RDX, 40% TNT, 18% Aluminium)	7540	1.667
Blasting gelatin (91% nitroglycerin, 7.9% nitrocellulose, 0.9% antacid, 0.2% water)	4520	1.000
60% Nitroglycerin dynamite	2710	0.600

Table 2: Heat of detonation & TNT equivalent factor of different explosives
(Smith & Hetherington, 1994)

2.2. Time History of Blast Wave Pressure

Friedlander (1946) proposed a simple equation that can describe characteristic shape of overpressures at specific locations (Equation 5) (Dewey, 2010) .

$$P_s(t) = P_{s0} e^{-bt/t_d} \left(1 - t/t_d\right) \dots\dots\dots (5)$$

Where P_s is the peak overpressure, t is the arrival time, t_d is the positive duration, and b is waveform parameter depending on the scaled distance (table 3).

Scaled Distance, Z (m/kg ^{1/3})	Waveform parameter, b
0.4	8.50
0.6	8.60
0.8	10.0
1.0	9.00
1.5	3.50
2.0	1.90
5.0	0.65
10.0	0.20
20.0	0.12
50.0	0.24
100.0	0.50

Table 3: Waveform Parameter (b)
(Appuhamilage, 2015)

A typical pressure-time history of a blast wave is described in Figure 13. After the arrival of the blast overpressure (P_{s0}), the positive phase is characterized by a decay of the incident overpressure to the ambient pressure (P_0). The time period of this phase is known as positive phase duration. Following this phase, a further decrease occurs toward a peak negative overpressure (P_{s0^-}): negative phase below ambient pressure. This negative phase is of less importance than the positive phase (Defense, 2008).

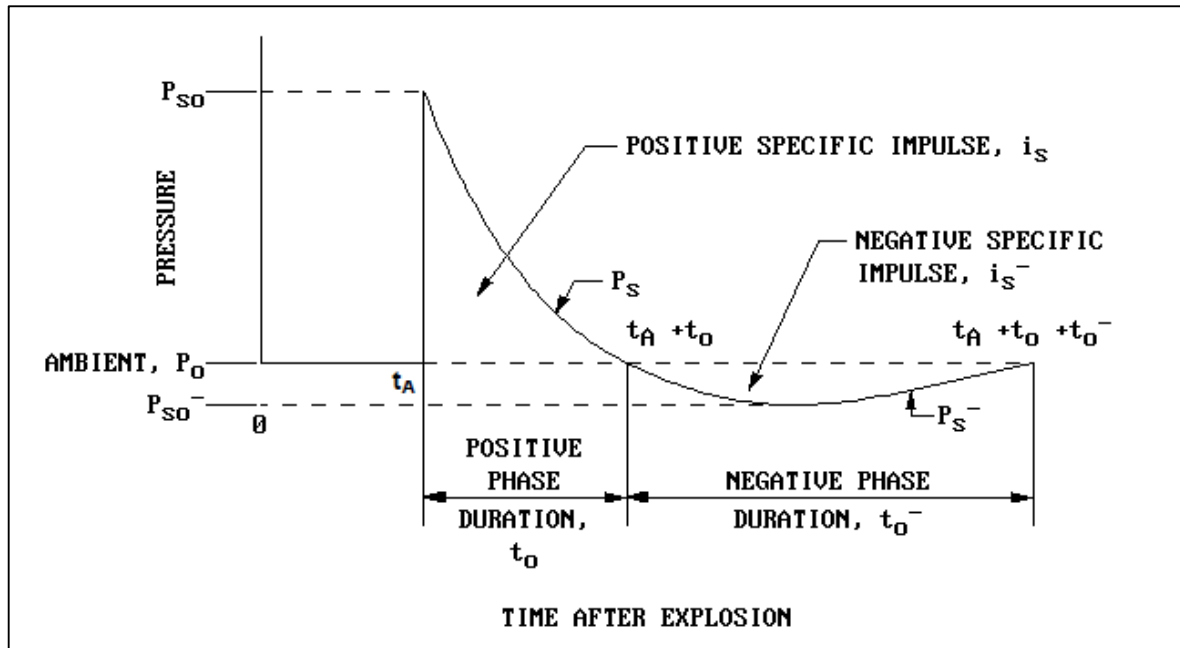


Figure 13: Pressure-Time History of Blast wave (Defense, 2008)

2.3. Prediction of Blast Wave Parameters Using Theoretical and Empirical

Methods

Throughout the literature, many scientists proposed various theoretical and empirical formulas aiming to predict the different parameters of a blast wave. In order to have a complete description of the latter, you must be able to specify each of the following:

- ❖ Positive Peak Incident Overpressure (Pso)
- ❖ Negative Peak Incident Overpressure (Pso-): Although it is not of a great consideration in comparison to the positive incident overpressure, it may cause further damage to already weakened structures (Appuhamilage, 2015).
- ❖ Positive Peak reflected overpressure (Pr): is defined as the reflected pressure measured at the surface of an obstacle in the path of a shock wave. Its value is a function of the incident pressure along with the reflection coefficient

(C_r). This coefficient directly related to the incident angle (α_a)

(Appuhamilage, 2015).

- ❖ Incident (is) and Reflected (ir) Impulse: The impulse describes the momentum change in any system, represented as the area under the pressure-time curve. The incident pressure time history yields the incident impulse while the reflected one results in the reflected impulse (Appuhamilage, 2015).
- ❖ Arrival Time (t_A)
- ❖ Positive Phase Duration(t_o)
- ❖ Negative Phase Duration(t_o-)
- ❖ Shock front velocity (U)
- ❖ Peak Dynamic pressure (q_o)
- ❖ Wavelength (Lw)

Table 4 summarizes the theoretical and empirical equations found in the literature.

Scientist name	Year	Perimeters	Blast Type	Empirical	Theoretical
Brode	1955	Pso	Air Burst		✓
		Pso-			
Henrych &Major	1979	Pso	Air Burst	✓	✓
Newmark &Hansen	1961	Pso	Surface Burst	✓	
Mills	1987	Pso	Surface Burst	✓	
Kingery & Bulmash	1984	Pso	Air & Surface Burst	✓	
		Pr			
		Pso-			

		is & ir			
Swisdak (Simplified Kingery & Bulmash)	1994	Pso	Air & Surface Burst	✓	
		Pso-			
		Pr			
		is & ir			
		t _A & t _o & t _{o-}			
		q _o			

Table 4: Theoretical and Empirical Equations of Blast Wave Parameters

The Unified Facilities Criteria (UFC) code proposed by the US Department of Defense, provided set of charts that easily enable the determination of the blast wave parameters.

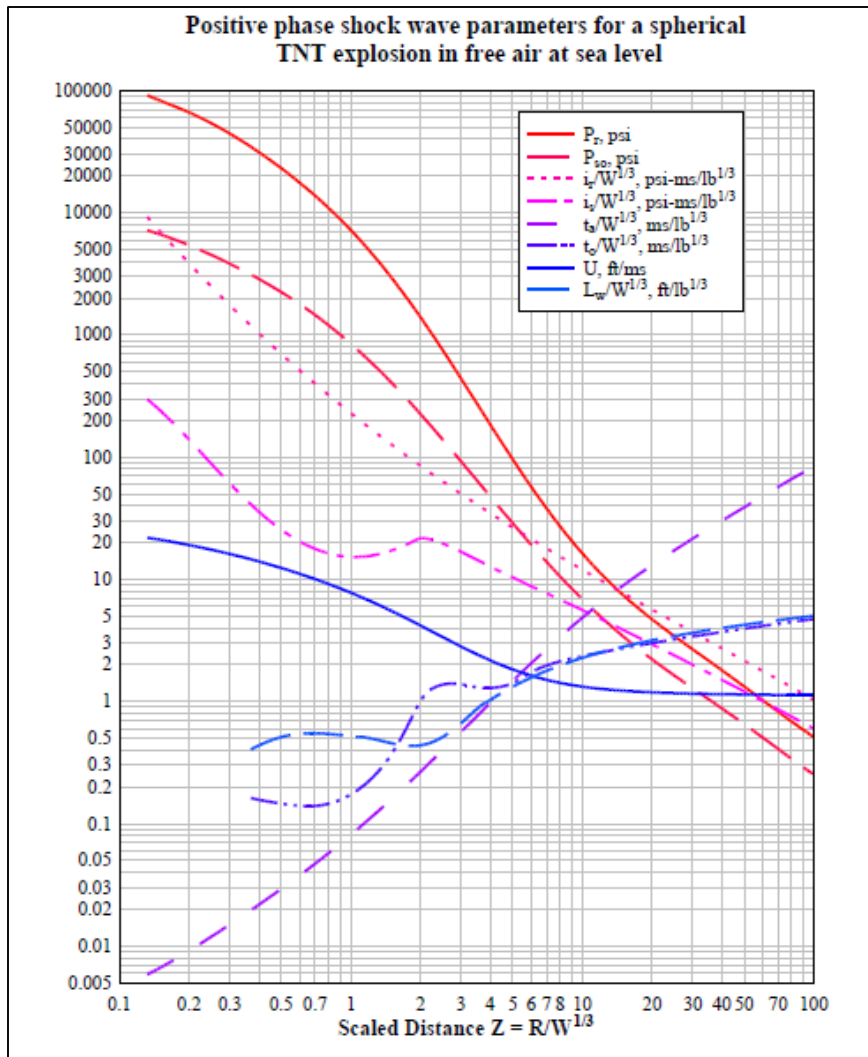


Figure 14: Positive phase parameters for spherical free air blast (Defense, 2008)

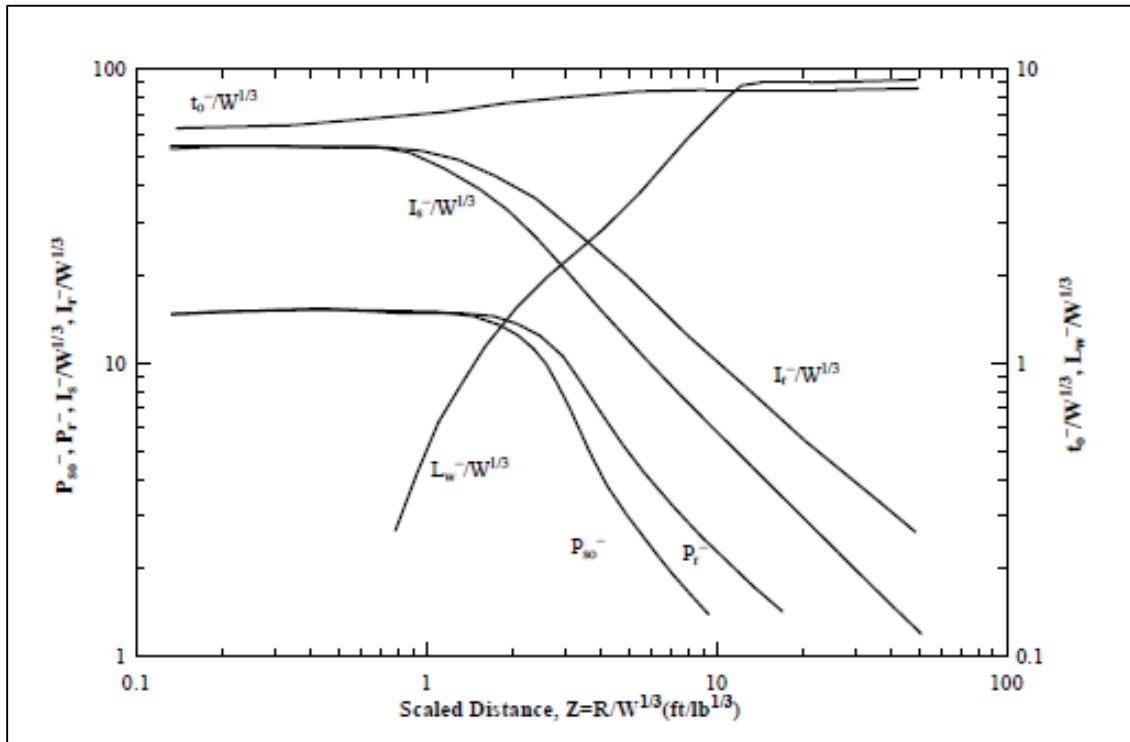


Figure 15: Negative phase parameters for spherical free air blast (Defense, 2008)

Friedlander equation presented in equation 5 represents the typical pressure-time relationship used for the dynamic analysis of structures. However, for simplification, the natural exponential decay can be approximated assuming a linear decay in triangular form (Defense, 2008). This method preserves the maximum value of the overpressure, and estimates the duration as $\frac{2i}{P}$ (impulse obtained from figure 14). A more conservative method proposes using the whole phase duration (t_d) (Figure 16).

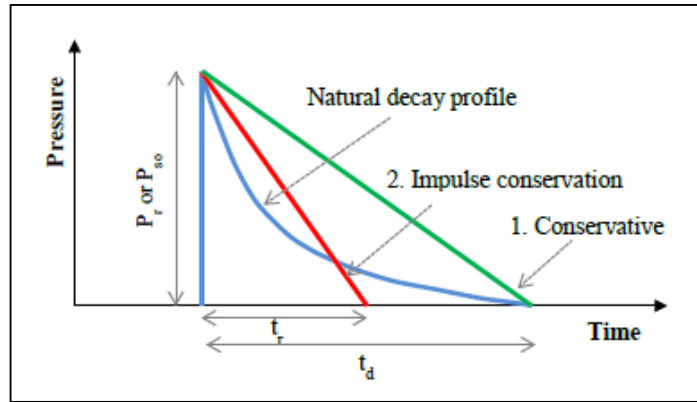


Figure 16: Idealization of Pressure-Time History (Appuhamilage, 2015)

2.4. Structural Response Under Blast Loading

Typically when a structure is subjected to any kind of loading, it deforms and stores energy to accumulate resistance against the load. The type of response against blast loading is dependent on different variables: blast intensity (Figure 17), positive phase duration (t_d), natural period of the structure (T), and the natural circular frequency of vibration (ω).

Mays and Smith (1995) divided the structural response with respect to the blast loading into three categories: Quasi-static, Dynamic, and Impulsive (Figure 18).

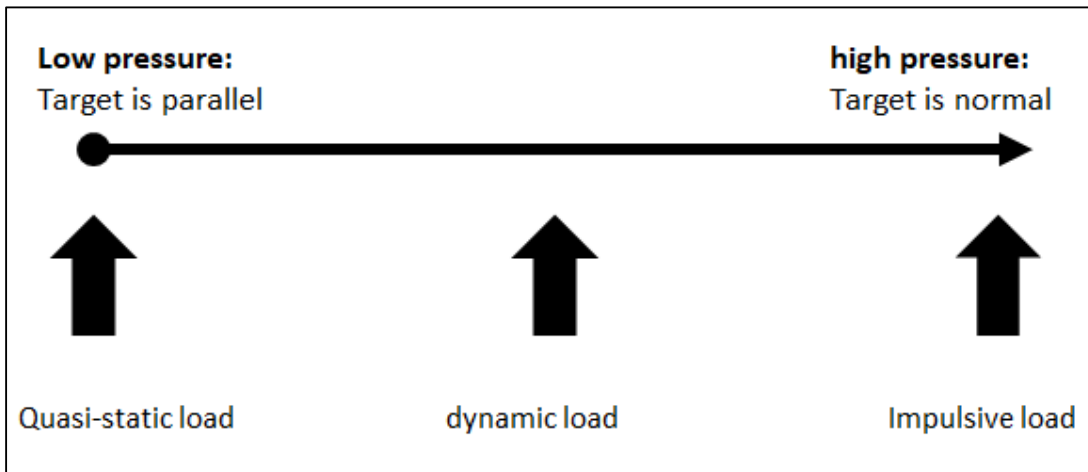


Figure 17: Structural Response as a Function of the Pressure Intensity (Defense, 2008)

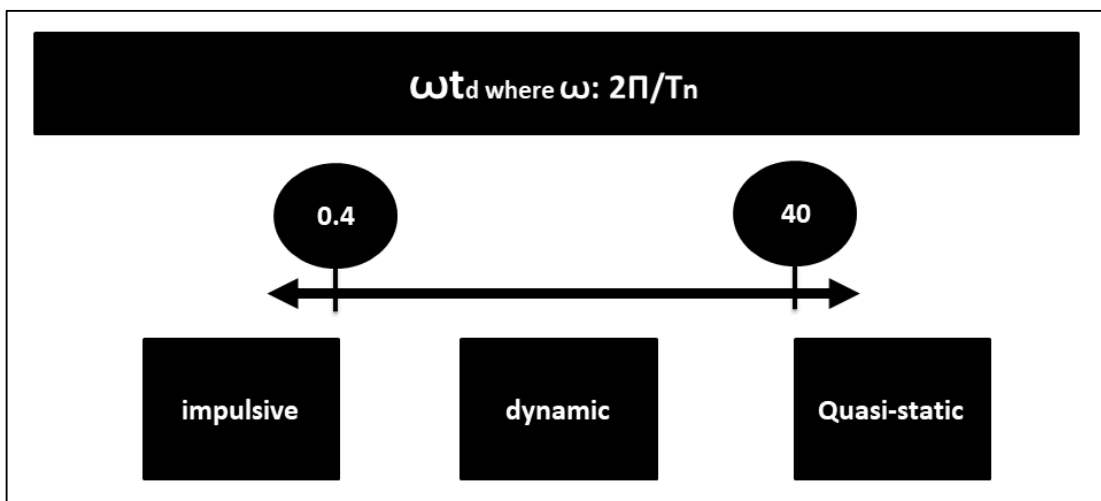


Figure 18: Mays and Smith Response Classification

❖ Quasi- Static Response:

According to the UFC, the design load is considered constant pressure load with a long duration compared to the natural period of the structure. The structure may reach its maximum displacement before the blast load has undergone any decay (Figure 19) (Smith & Hetherington, 1994).

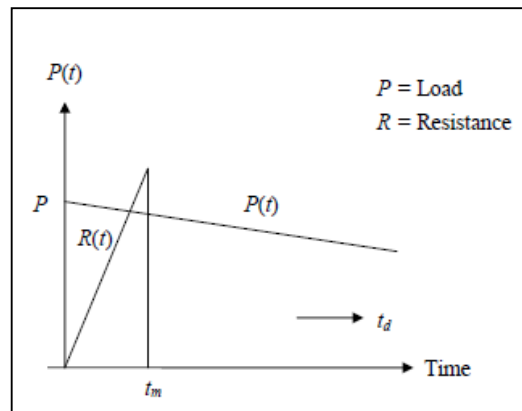


Figure 19: Quasi-static Structural Response (Smith & Hetherington, 1994)

❖ Dynamic Response:

The UFC defines the load as a pressure-time history, where the positive phase duration of the blast \approx the natural period of the target (Figure 20) (Smith & Hetherington, 1994).

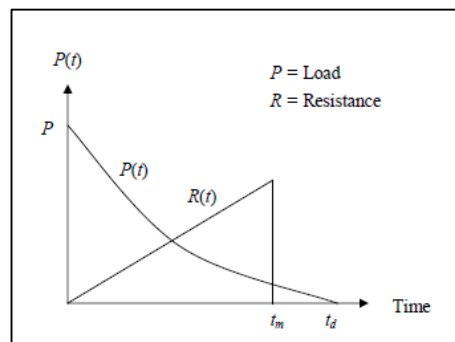


Figure 20: Dynamic Structural Response (Smith & Hetherington, 1994)

❖ Impulsive Response:

In this case the target must be designed for impulse rather than for peak pressure, where the duration of the blast is much smaller than the natural period of the structure. Hence, the deformation occurs after the blast has finished (Figure 21) (Smith & Hetherington, 1994).

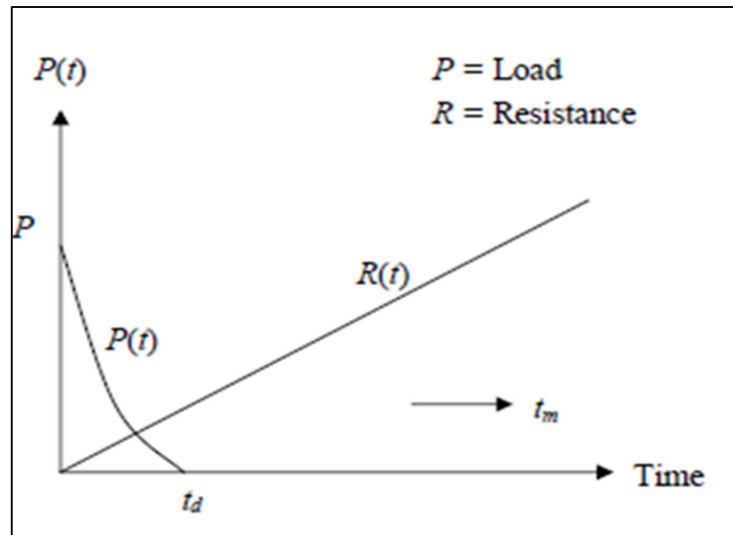


Figure 21: Impulsive Structural Response (Smith & Hetherington, 1994)

2.5. Prediction of Blast Wave Parameters Using Computational Techniques

Computational techniques are reliable and time efficient means for the determination of scientific and engineering problems via the aid of computers and by utilizing iterative and automated processes. Resolving problems using computational manner is incomparably more accurate and time-saving than solving manually (Emetere, 2019).

In blast analysis and design, computational methods are divided into two categories: programs that are capable of predicting either the parameters of the blast wave or the structural response (uncoupled programs), and those that have an additional

module that enables the estimation of both aspects of the blast problem (coupled programs).

Table 5 shows the different computational programs available for blast prediction.

Program	Feature	Type
CONWEP	Blast wave parameters	Uncoupled
BLASTX	Blast wave parameters	Uncoupled
RCBLAST	Blast wave parameters	Uncoupled
CTH	Blast wave parameters	Uncoupled
FEFLO	Blast wave parameters	Uncoupled
FOIL	Blast wave parameters	Uncoupled
HULL	Blast wave parameters	Uncoupled
SHARC	Blast wave parameters	Uncoupled
AIR3D	Blast wave parameters	Uncoupled
DYNA3D	Structural Response	Uncoupled
EPSA-II	Structural Response	Uncoupled
ALEGRA	Blast wave parameters	Coupled
	Structural Response	
ALE3D	Blast wave parameters	Coupled
	Structural Response	
FUSE	Blast wave parameters	Coupled
	Structural Response	
DYTRAN	Blast wave parameters	Coupled
	Structural Response	
MAZe	Blast wave parameters	Coupled

	Structural Response	
LS-DYNA	Blast wave parameters	Coupled
	Structural Response	
AUTODYN	Blast wave parameters	Coupled
	Structural Response	
ABAQUS	Blast wave parameters	Coupled
	Structural Response	

Table 5: Couple & Uncoupled Computational Programs for Blast Prediction

(Appuhamilage, 2015)

2.6. Material Properties Under Dynamic Loading Conditions

Blast waves induce structural loads at very high strain rates up to 10^4 s^{-1} (Figure 22). Those high rates alter the mechanical properties of the materials and consequently the dynamic response of the structure. Studies by Malvar and Ross (1998); showed that reinforced concrete structures exhibit strength increase up to 600% under high strain-

rates from 10 to 1000 s^{-1} . (Appuhamilage, 2015; Castro, 2006; Jahami, 2016; Zhang et al., 2020). More specifically, to study the behaviour of reinforced concrete structures when subjected to blast loading; it's essential to understand the response of concrete and steel under high strain rates.

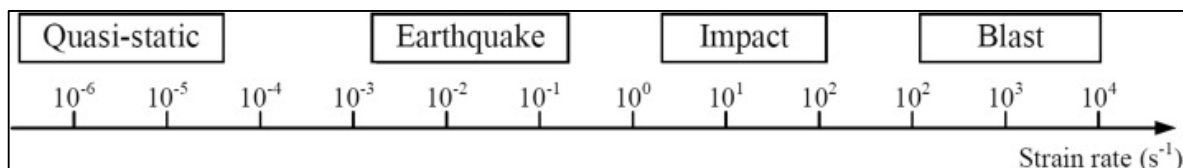


Figure 22: The ranges of strain rates produced by various loading (Zhang et al., 2020)

❖ **Material properties of concrete under dynamic load:**

Brittle material such as concrete demonstrates higher increase in strength when subjected to dynamic loading than ductile material (i.e. steel) (Appuhamilage, 2015). The most common method used to consider the strength enhancement of concrete is the dynamic increase factor (DIF). DIF is referred to as the ratio of dynamic strength to the static one.

Scientifically, many interpretations exist behind the physical reason behind this increase. Viscosity effects and inertia/confinement effects both contribute to rate augmentation (Appuhamilage, 2015).

According to Rossi (1991), the viscous effect is due to the presence of free water between the walls of the micro-pores and capillaries of concrete, which induces a force that opposes the wall separation/displacement. With increasing strain-rate, the velocity of the separation process increases, resulting in a greater opposing force and a higher strength gain. Bischoff and Perry (1991), reported that when a concrete specimen is subjected to a high strain rate, lateral inertial confinement forces are observed. This is justified by the change of the problem from being in plane stress conditions due to quasi-static loading regimes, into plane strain one when subjected to high strain loadings (Schwer, 2009).

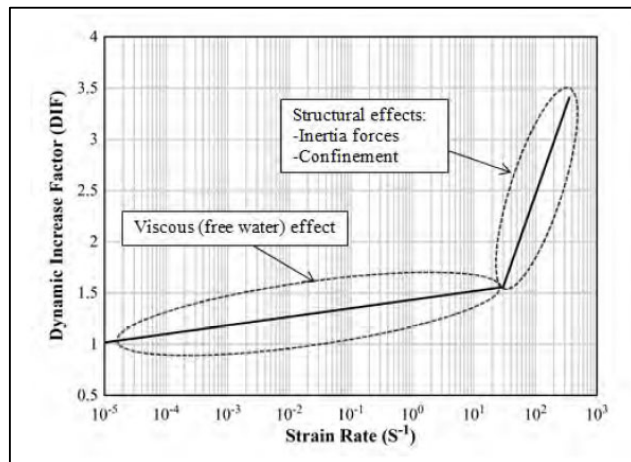


Figure 23: DIF of Concrete Due to Viscous and Structural Effects (Appuhamilage, 2015)

Bischoff and Perry (1991), transformed the results of sixty years of concrete dynamic strength testing into a single data scatter depicting the DIF of concrete under high strain rates (Figure 24). Analytically, the Comité Euro-International du Béton (CEB) proposed the FIP 1990 formulation which is the most widely accepted for the concrete DIF calculation up to 300s^{-1} strain rate.

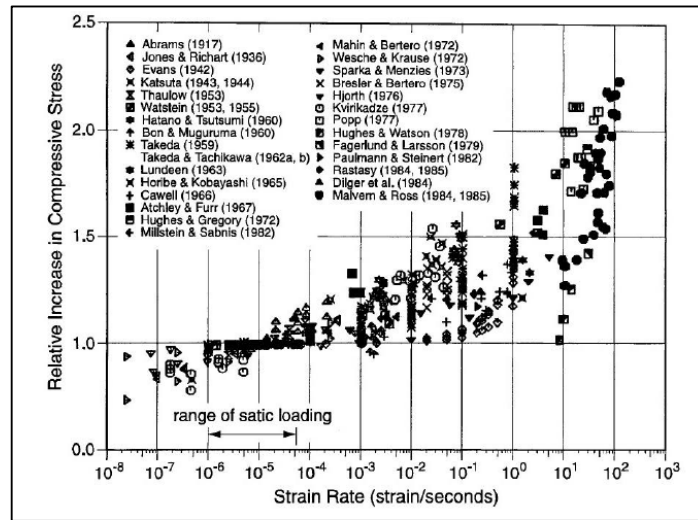


Figure 24: Bischoff and Perry (1991) DIF in compression Scatter

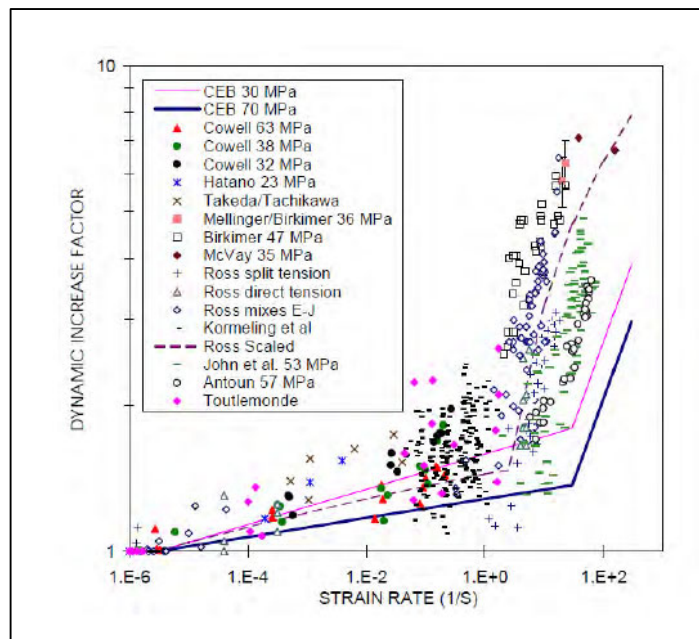


Figure 25: DIF in tension for Concrete

❖ **Material properties of steel under dynamic load:**

Steel is considered to be an isotropic material; where the tensile and compressive characteristics are affected equally. Norris et al. 1959, Keenan and Feldman, 1960, Cowell, 1966, Cowell, 1969, Sozen, 1974, Wakabayashi et al. 1980, Soroushian and Obaseki, 1986, Soroushian and Choi, 1987, and Malvar, 1998, conducted intensive studies on determining and analyzing the dynamic properties of steel. With respect to the reinforcing steel bars, Fu et al. concluded the following (Fu et al., 1991):

1. The modulus of elasticity of rebars is barely affected by the increased strain rates.
2. Both the yield and ultimate strengths exhibit an enhancement when subjected to high strain rates. Similar to concrete studies, lower strength reinforcement experience higher dynamic strength gain.

Malvar and Crawford 1998, derived formulas to predict the DIF for yield strength of reinforcing steel (Figure 26) (Appuhamilage, 2015).

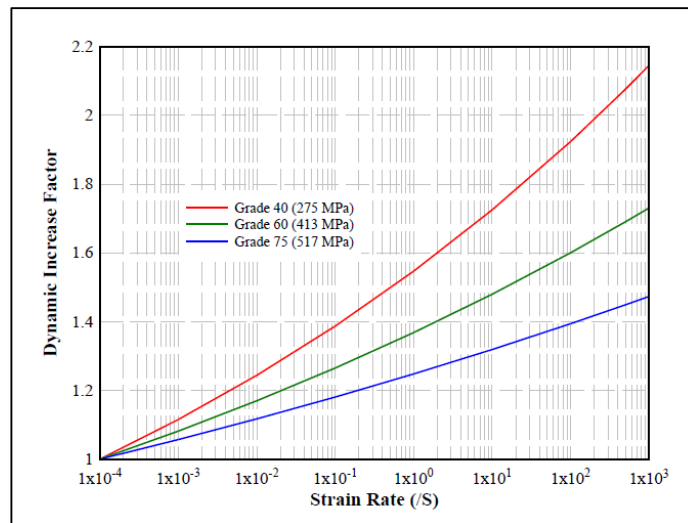


Figure 26: DIF of Reinforcement Steel According to Malvar and Crawford 1998

2.7. Finite Element Modeling and Analysis

It is extremely difficult to accurately anticipate the reaction of structural materials subjected to explosive loading due to the complexity of the loading, rate enhancements, dynamic and inertia effects, and higher modes of vibration (Appuhamilage, 2015). In addition, impact and blast testing on a full scale are generally too complex and expensive to conduct (Castro, 2006). Hence, researches have proposed approaches that properly anticipate the structural response to the effect of explosive loading using acceptable assumptions. Those approaches may be either empirical, semi-empirical, or numerical (Appuhamilage, 2015).

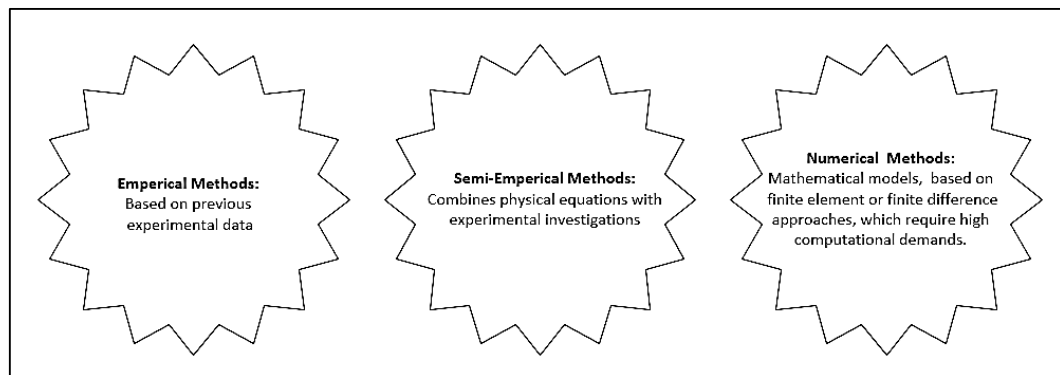


Figure 27: Approaches for Predicting Structural Response

Researchers have studied and analyzed the behavior of structures when subjected to impact and blast loads using numerical techniques such as the FE method (Whirley and Engelmann, 1992, Williams, 1994, Malvar et al., 1997, Thabet and Haldane, 2001, Esper, 2004, Jahami et al., 2019) (Castro, 2006).

Computational models and numerical simulation have become useful tools in the design of structures and the understanding of the physical behavior of structures under dynamic loading. Three-dimensional (3D) numerical simulations are required to more

realistically simulate and predict reinforced concrete response to various charge weights and standoff distances (W. Wang et al., 2013).

Figure 28 illustrates the procedure for simulating the response of reinforced concrete structures to blast loading.

The strain-rate effect causes a material's strength to change over time, as well as its energy-absorption. As a result, it's critical to include the strain-rate effect in the FE model in order to reproduce accurate time-dependent material responses under impact or blast loading (Castro, 2006).

The selection of the finite element software is of great impact on the results' quality of the performed analysis. The general purpose FEM package ABAQUS includes two solvers: standard and explicit. It provides a great deal of power over the simulation. Users can simulate material responses accurately using pre-implemented models. In addition, the meshing and element types utilized in the model may also be highly controlled by the user. Thus, a highly comprehensive model that describes many types of behavior can be created. Furthermore, the program is scriptable and may be run from the command line. ABAQUS is the preferred software package for doing complex simulations when multiple solver methods must be employed and careful control over the mesh and boundary conditions is required (Joel et al., 2005).

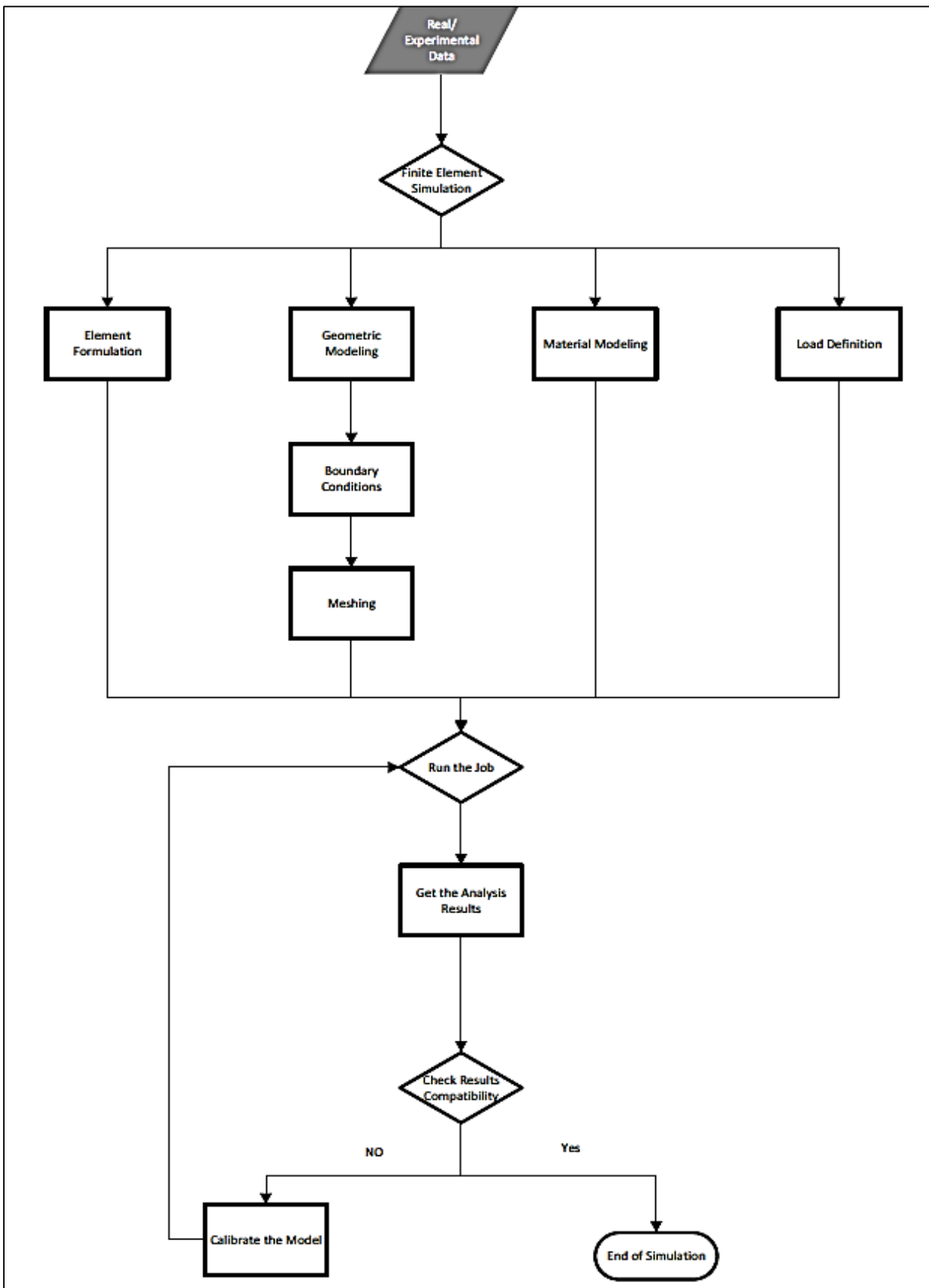


Figure 28: FE Simulation Process Chart

2.8. Finite Element Modeling of Reinforced Concrete Structures Subjected to Blast Loading Using ABAQUS

2.8.1. Element Formulation

ABAQUS supports four element formulations: Lagrangian, Eulerian, Arbitrary Lagrangian Eulerian (ALE), Coupled Eulerian Lagrangian (CEL), and Smoothed Particle Hydrodynamics (SPH) formulation (Figure 29).

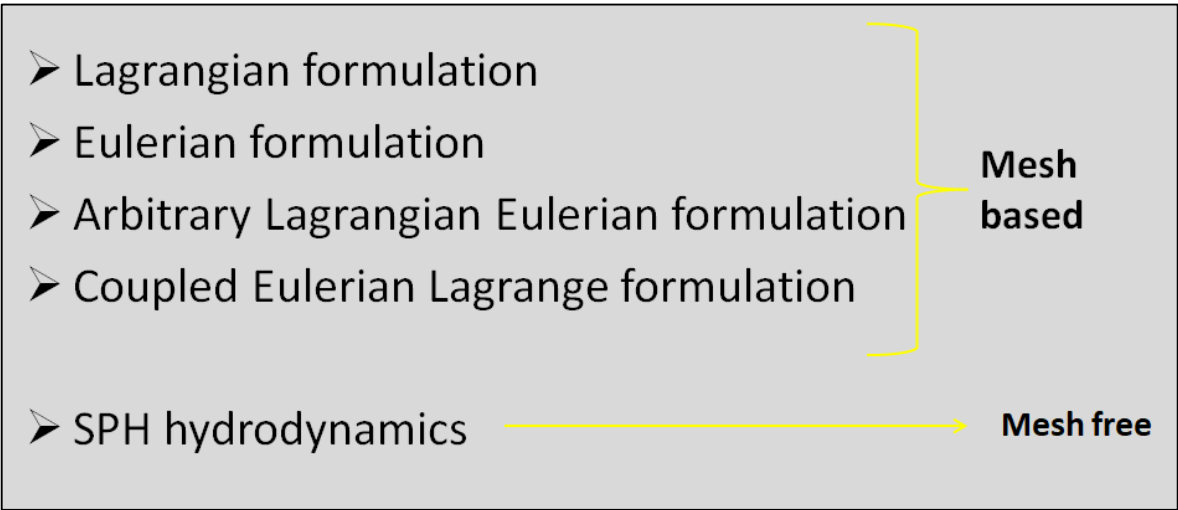


Figure 29: Element Formulations in ABAQUS

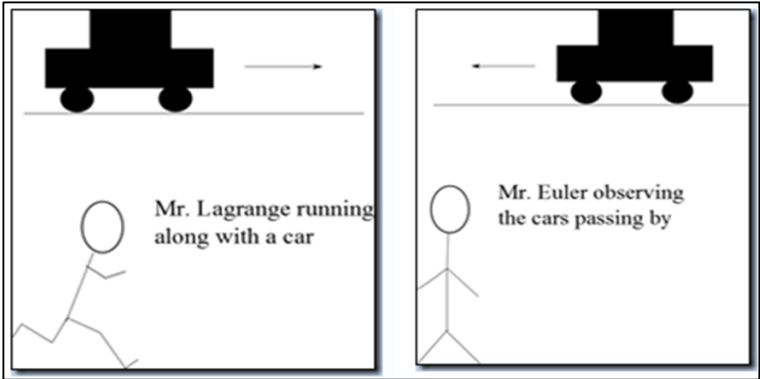


Figure 30: Lagrangian vs. Eulerian Formulation (*Coupled Eulerian Lagrange (CEL) Analysis with ABAQUS, n.d.*)

Table 6 summarizes the difference between each type of the formulations
 (Castro, 2006; *Coupled Eulerian Lagrange (CEL) Analysis with ABAQUS*, n.d.).

Formulation	Criteria	Description
Lagrangian	Mesh Based	Mesh is directly attached to the material
		Mesh undergo distortion and movement with the material flow
		Computational wise: Fastest
		Needs to be accompanied with erosion algorithm
Eulerian	Mesh Based	Mesh is fixed spatially
		Mesh is larger than the original material to allow for its flow
		Can handle some deformation without the need of erosion algorithm
Arbitrary Lagrangian Eulerian (ALE)	Mesh Based	Extension of the Lagrangian: applied to perform automatic rezoning
		Remaps the solution onto new grid
		Rezones by moving the positions of the nodes
		Can't replace Eulerian rezoning for large problems
Coupled Eulerian Lagrangian (CEL)	Mesh Based	Combines two mesh approaches: Lagrangian and Eulerian in the same analysis
		Avoids mesh problems when performing simulations that involve high/extreme deformations

		Bodies undergoing large deformations (Eulerian) while the remaining (Lagrangian). Eulerian (background grid) within which material flows.
		Size of the Eulerian part should be selected in such a way that no material is expected to flow out of it during simulations
Particle Hydrodynamics (SPH)	Mesh Free	Mesh free method
		Schwer (2004) found that the SPH model displayed less resistance as compared to the Lagrangian with erosion and Eulerian models
		According to Schwer (2004), this may be due to the “tensile instability” of the SPH method

Table 6: Description of Element Formulations Available in ABAQUS

2.8.2. Material Modeling

For accurately predicting the structural reaction of reinforced concrete 3D numerical model, selecting the proper constitutive relationship for modeling the concrete and steel reinforcing is critical.

ABAQUS FEM package has number of built-in material models for simulating various materials' behavior under different loading aspects.

- As for **concrete constitutive relationships**; those are generally categorized into either elasticity based models or plasticity based ones (Appuhamilage, 2015).

Figure 31 shows the available plasticity based material models for concrete in Abaqus.

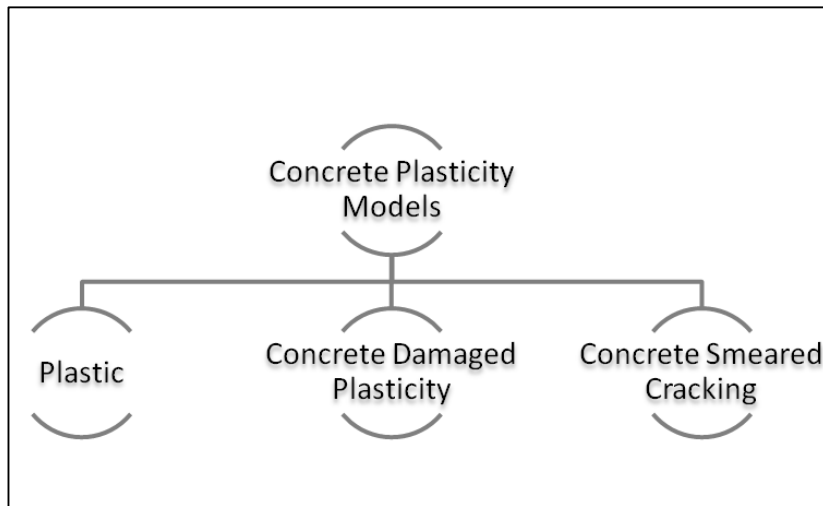


Figure 31: Plasticity Based Material Models for Concrete in Abaqus

In Abaqus material library, in addition to the material models, VUMAT subroutines are also built-in. Those materials can be invoked by using names starting with ABQ_ (SIMULIA, 2013). As well as the yield surface, one of these criteria can be used to determine the failure surface, where the total degradation of the concrete matrix following cracking is defined (Appuhamilage, 2015).

- Plastic Model: The perfect plasticity model ignores the gradual softening of cracked concrete. It assumes that a sudden drop of stress occurs upon failure (Appuhamilage, 2015).

- Concrete Damage Plasticity (CDP) Model: CDP enables the modeling of concrete and other quasi-brittle materials, using the concept of isotropic damaged elasticity and tensile and compressive plasticity. Yet, it needs to be modified to account for rate of straining (SIMULIA, 2013).
 - Concrete Smeared Cracking Model: The model is designed to simulate the behavior of concrete under monotonic loading with low confining pressures (SIMULIA, 2013).
 - Second Version of Johnson-Holmquist (JH-2) Model: JH-2 model consists of three components: strength, damage, and pressure. It accounts for the strain rate effect and considers the gradual softening under increasing plastic strain. In this model, the damage variable increases progressively with plastic deformation (SIMULIA, 2013).
- As for **steel constitutive relationships**; reinforcements in concrete carry axial forces (shear to a minor extent), which could be explained using the uniaxial stress-strain relationship of the material. Similar to concrete, relationships are referred to as elastic or plastic models. Johnson Cook Damage model is the most widely used model for representing steel bars behavior. It's ideal for capturing the behavior under high strain rates, and temperature (Appuhamilage, 2015; W. Wang et al., 2013).
- As for the surrounding air, it can be modeled using the equation of state (EOS)- Shock Hugoniot relation (U_s-U_p). This relation linearly relates the shock and particle velocities.

$U_s = C_0 + S u_p$; where U_s is the shock velocity, U_p is the particle velocity, C_0 is the sound speed in a specific medium, and S ranges from 1.0 to 1.6 (Kerley, 2013).

2.8.3. Load Definition

Generally, Blast load simulation using Abaqus can be conducted using either Jones-Wilkins-Lee (JWL) equation of state or conventional weapons (CONWEP) model built-in Abaqus.

JWL is an Abaqus-implemented physical model that uses the principles of thermodynamics to derive the physics of a chemical blast. It enables a thorough description of a blast phenomenon, involving the propagation of the shock-wave, as well as its reflection on solid surfaces and the expansion of gases. Its application necessitates meshing not only the structure but also the volume of the surrounding air, thus it's computationally heavy (Masi, 2017).

CONWEP model represents the blast phenomenon depending on the scaled distance and relating the mass of the explosive charge to the distance between the detonation center and the target. The total pressure on a surface due to the blast wave is a function of the incident pressure, the reflected pressure, and the angle of incidence (Figure 32& 33) (Jahami, 2016).

$$P(t) = P_{incident}(t)[1 + \cos\theta - 2\cos^2\theta] + P_{reflect}(t)\cos^2\theta \quad \text{for } \cos\theta \geq 0;$$

$$P(t) = P_{incident}(t) \quad \text{for } \cos\theta < 0.$$

Figure 32: Total Pressure Relation in CONWEP

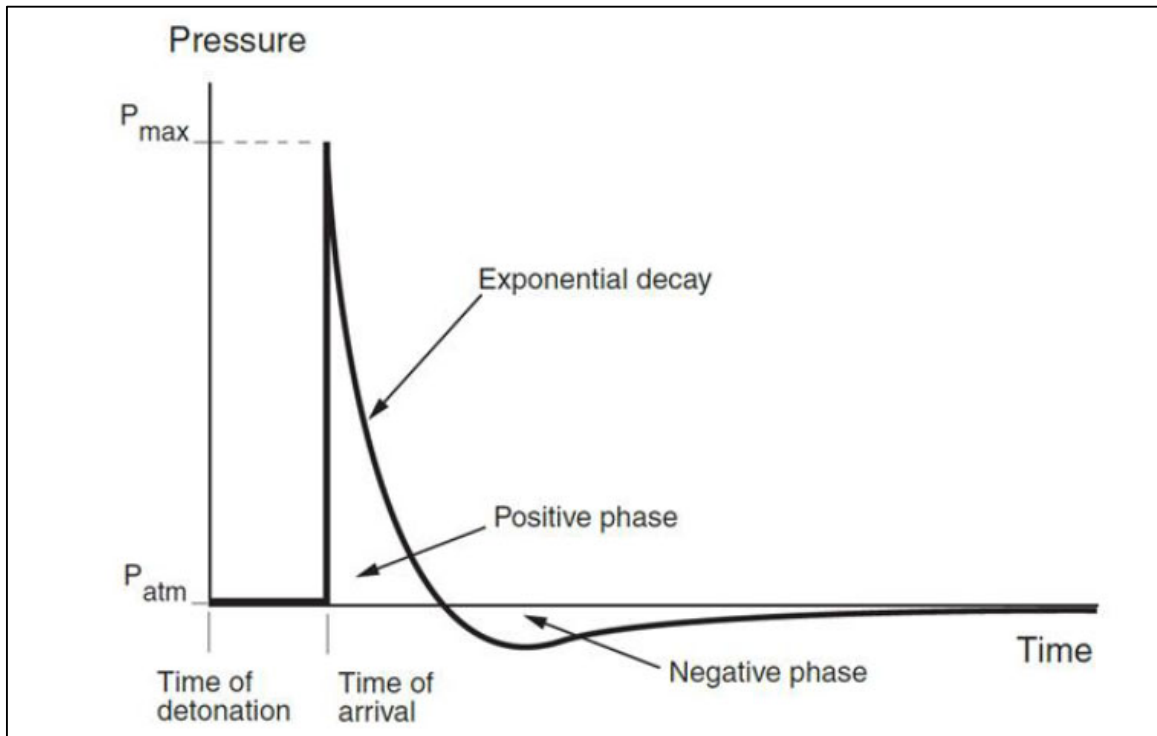


Figure 33: Total Pressure of Blast Wave

2.9. Heuristic Optimization Methods

Heuristic optimization techniques are derivative-free computing procedures for finding an optimal solutions by continuously improving alternatives in relation to a particular measurement. Heuristic optimization methods are iterative based making minimum assumptions regarding the problem being optimized (F.-S. Wang & Chen, 2013). Particle swarm (PSO), Genetic Algorithm (GA), Pattern search, and simulated

annealing algorithm are the most used techniques according to the literature (Basak et al., 2013; El Khansa, 2020; Hasanova, 2020).

2.9.1. Genetic Algorithm Optimization

PSO, GA, and Pattern search are all evolutionary based methods. PSO is continuous in nature, thus it has to be adapted to accommodate discrete problems (Hasanova, 2020). Pattern search optimization is very efficient when the search space is small (Basak et al., 2013). GA can be referred to as the most popular technique among heuristic optimization methods, where the data space is large and the mathematical representation of the objective function is too complicated to be mathematically formulated. It is population based technique where the fittest individuals survive after the continuous evolution of species (Basak et al., 2013; El Khansa, 2020).

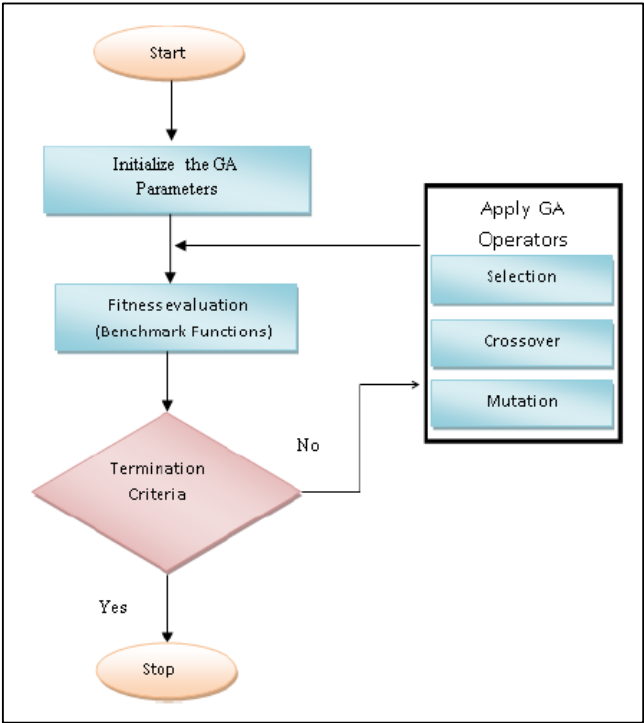


Figure 34: Basic Implementation of GA (Hasanova, 2020)

- The GA is initialized by encoding the design variables of potential solutions as “chromosomes”. The variables can be coded either using binary or real code formats (Figure 35).

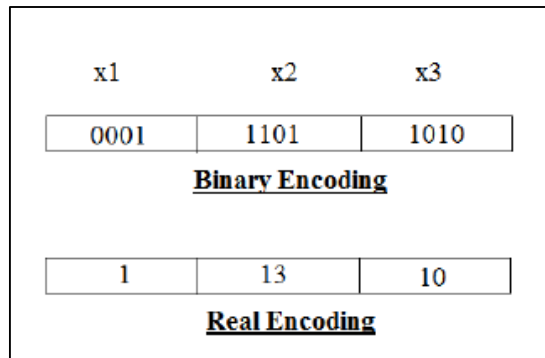


Figure 35: GA Encoding(El Khansa, 2020)

- The fitness evaluation is done based on the performance of the potential solutions according to the objective function.
- Selection operators are used to establish copies of chromosomes that will be used later for crossover and mutation.
- Crossover combines the features of two parents to create a new individual replacing the parents (Figure 36).

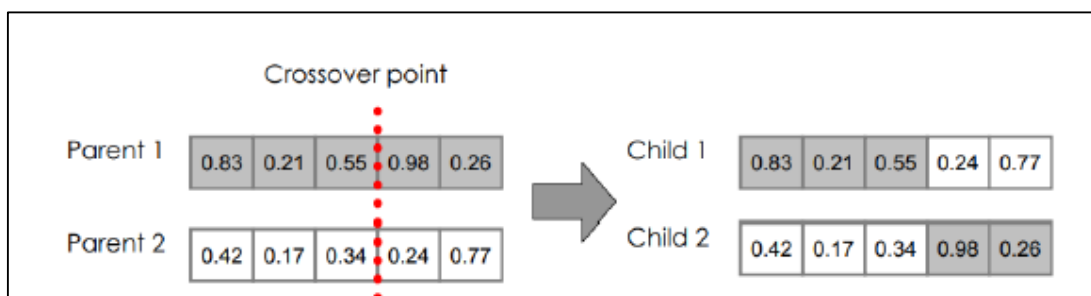


Figure 36: Crossover

- Mutation is randomly altering a gene within the chromosome with probability of getting a better solution
- Stopping criteria: It determines what causes the algorithm to terminate. The criteria can be in terms of maximum number of generations, maximum time, fitness limit, maximum stall generations, maximum stall time, and function tolerance (El Khansa, 2020; MathWorks, 2021; Zhou, 2006).

2.9.1.1.Methods for Maintaining Population Diversity

Maintaining Population diversity and avoiding premature convergence is the most challenging issue related to GA.

- ❖ Tournament Selection: This selection method is efficient in balancing the convergence speed with population diversity. X selected individuals compete where the ones with the highest fitness surpass to the next generation. The number of competing individuals are known as tournament size (El Khansa, 2020) .
- ❖ Rank Selection: According to MATLAB user manual, ranking of chromosomes takes place from the worst to the best. Those ranks are used to specify the probability of selecting the individual (P). It's known to be slower than proportional selection in terms of convergence.

$$P = r_i / \sum r_i$$
- ❖ Proportional Selection: This selection operator is the most popular one, the selection possibility is proportional to the fitness score of the individual. Individuals with the highest fitness values are scaled equally (MathWorks, 2021).

$$P = f_i / \sum f_i$$

- ❖ **Gaussian Mutation:** A random number, taken from a Gaussian distribution centered at zero, is added each vector entry of an individual. It's applied when dealing with unconstrained optimization problem (MathWorks, 2021).
- ❖ **Adaptive Feasible Mutation:** It randomly generates adaptive directions with reference to the last successful/ unsuccessful generation. It's compatible with constrained optimizations (MathWorks, 2021).

2.10. Image Processing Techniques

Any image may be describes as set of square pixels composing as row and column array or matrix. In order to transform the image into a digital form and extract useful information from it, image processing operations are performed.

Mathematically, an image is defined as two real variables with an amplitude such as the brightness at a specific coordinate point (a, b). An RGB image is stored in computers as an m-by-n-by-3 data array representing red, green, and blue color components referring to each pixel. Each color is represented by value between 0 and 255 (maximum brightness) (Taylor, 2021) . Figure 37 shows some colors encoded as red, green, and blue.

Color	Red number	Green number	Blue number
red	255	0	0
purple	255	0	255
yellow	255	255	0
dark yellow	100	100	0
white	255	255	255
black	0	0	0

Figure 37: Some Colors Encoded as RGB (Taylor, 2021)

Images can be converted into gray scale images using either average or weighted method. For average method: $\text{gray-scale} = R/3 + G/3 + B/3$, while the weighted method weighs red, green and blue according to their wavelengths: $\text{gray-scale} = 0.299R + 0.587G + 0.114B$ (Dynamsoft, 2019).

Binarization converts gray-scale images to binary using thresholds. The intensity of each pixel is compared to the specified threshold. If the pixel's threshold is smaller, it is set to zero (black). Otherwise, it is set to one (white) (Dynamsoft, 2019).

Image processing techniques were found to be efficient in the field of medicine, morphology, computer vision, finger print detection, engineering, and remote sensing (Hegadi & Basavaprasad, 2014).

2.10.1. Image Processing Techniques in structural Analysis

Structural Damage of reinforced concrete elements is function of the crack dimensions. Historically, engineers have detected cracks' components visually through measuring their widths by the means of a crack-width card and then transferring that information to drawing sheets (Figure 38) (Rivera et al., 2015). After the technological progress, this data collection was made possible using image processing techniques available in MATLAB. Primarily, Image processing crack detection algorithms were used for pavement cracks detection.

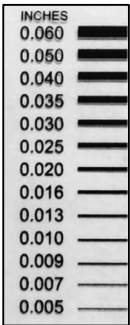


Figure 38: Crack Gauge

Imaging tools were used for identifying blemishes, defects, and stains on surfaces on concrete components (Abdel-Qader et al., 2003; Fujita et al., 2006). Crack widths, lengths, and areal density were detected and analyzed using pixel brightness difference and edge detection methods (Miyamoto et al., 2007; Rivera et al., 2015).

2.10.2. Edge Detection Techniques in MATLAB

According to the MATLAB documentation, it offers via the image processing toolbox a set of edge detection algorithms (Figure 39). Those work on analyzing gray-scale as well as binary images.

Method	Description
'Sobel'	Finds edges at those points where the gradient of the image <i>I</i> is maximum, using the Sobel approximation to the derivative.
'Prewitt'	Finds edges at those points where the gradient of <i>I</i> is maximum, using the Prewitt approximation to the derivative.
'Roberts'	Finds edges at those points where the gradient of <i>I</i> is maximum, using the Roberts approximation to the derivative.
'log'	Finds edges by looking for zero-crossings after filtering <i>I</i> with a Laplacian of Gaussian (LoG) filter.
'zerocross'	Finds edges by looking for zero-crossings after filtering <i>I</i> with a filter that you specify, <i>h</i>
'Canny'	Finds edges by looking for local maxima of the gradient of <i>I</i> . The edge function calculates the gradient using the derivative of a Gaussian filter. This method uses two thresholds to detect strong and weak edges, including weak edges in the output if they are connected to strong edges. By using two thresholds, the Canny method is less likely than the other methods to be fooled by noise, and more likely to detect true weak edges.
'approximate'	Finds edges using an approximate version of the Canny edge detection algorithm that provides faster execution time at the expense of less precise detection. Floating point images are expected to be normalized to the range [0, 1].

Figure 39: Edge Detection Algorithm in MATLAB

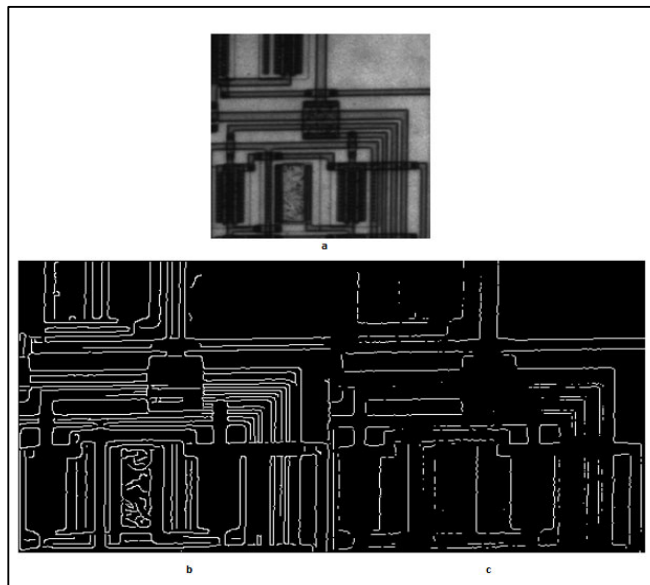


Figure 40: Gray-scale Image (a), Canny Method (b), Prewitt Method (c) (MathWorks, 2021)

2.10.3. Similarity Indices Between Images

One of the most challenging aspects of image processing is image comparison. The evaluation of many image processing and pattern recognition algorithms depends on image similarity indices. These similarity measures were widely utilized in applications like image retrieval, picture registration, and classification (Appana et al., 2021). The similarity score between photos mostly ranges from 0 to 1, depicting the degree of correlation. This value indicates how physically similar images are (Appana et al., 2021). Both intensity variations and geometric distortions should be considered by the similarity index (Figure 41) (Sampat et al., 2009).

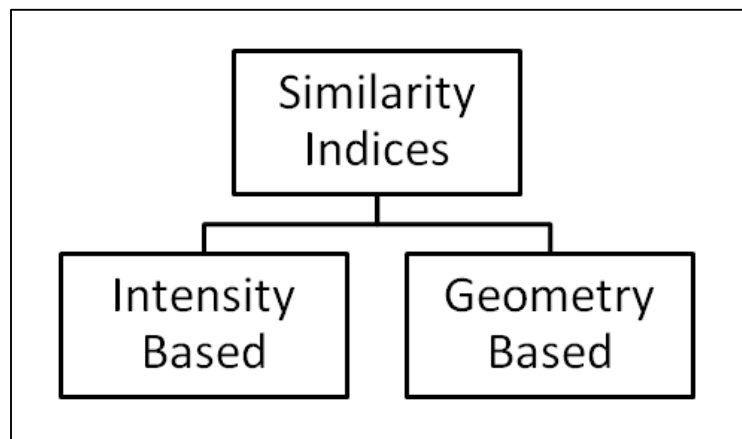


Figure 41: Classes of Similarity Indices

Intensity based indices compute the spatial overlap between pixel intensities of the analyzed images. As a result of their design, these indices penalize images that differ by even a single pixel. Table 7 shows the most common used intensity based indices in processing binary images (Sampat et al., 2009).

Similarity Index, Year	Range
Dice, 1945	0 to 1
Jaccard, 1912	0 to 1
Kulczynski, 1928	0 to inf.
Simpson, 1960	0 to 1
Ochiai, 1957	0 to 1
McConnaughey, 1982	0 to 1
Braun-Blanquet, 1932	-1 to 1
Sokal & Sneath, 1963	0 to 1

Table 7: Intensity Based Similarity Indices

Another extensively used intensity-based indicator is the Mean Square Error (MSE) measuring the disagreement between images. The MSE has no upper bound with a lower bound of zero when images are identical (Sampat et al., 2009).

Geometric based Indicators geometric modifications and transformations of the pixels in question. These include Pixel Correspondence Metric (PCM), Figure of Merit (FOM]), the Partial Hausdorff Distance Metric (PHDM), Closest Distance Metric (CDM), and the point-to-closest-point mean squared error (Sampat et al., 2009).

Wang et al. (2004), proposed a similarity index based on structural information's degradation. The conceptual base is that the human visual system (HVS) principle can be utilized to extract structural features from a visual image. This index relates the statistical mean and the statistical variance of the images. The structural similarity index (SSIM) basic disadvantage is that it is extremely sensitive to image rotation, scaling, and translation (Z. Wang et al., 2004).

Zhang et al. (2011), suggested Feature Similarity Index (FSIM). This index basically uses phase congruency (PC) and gradient magnitude (GM) (Aljanabi et al., 2018).

Sampat et al. (2009), proposed an extension of the SSIM in the complex wavelet domain (CW-SSIM). It provides a measurement that is unaffected by "non-structured" geometric picture deformities, which are often generated by annoyances such relative motion of the image capture equipment rather than genuine changes in the images (Sampat et al., 2009). Scientists discovered that the primary cortex in the HVS acts as focused band-pass oriented filters (wavelets) that breakdown images into many visual channels (Bovik et al., 1990).

Typically in images, phase carries more information about the structure than the magnitude, and translation of image structures results in consistent phase shift. Thus, the CW-SSIM is designed in a way that:

- Measures magnitude and phase distortions separately.
- Is affected by phase distortions than magnitude distortions.
- Is unaffected by constant relative phase shifts.

The maximum value of 1 is only achieved when the compared images are identical (Sampat et al., 2009).

CHAPTER 3

METHODOLOGY

3.1. Finite Element Modeling and Calibration

3.1.1. Experiment Description

A numerical model using ABAQUS/CAE version 6.23 was conducted to simulate the experimental study of a square reinforced concrete slab under close-in explosion by Wang et. al. (W. Wang et al., 2013). Figure 42, 43, and 44 show the testing apparatus as well as the slab and support dimensions used by Wang.



Figure 42: Testing Apparatus Used in Wang Experiment (W. Wang et al., 2013)

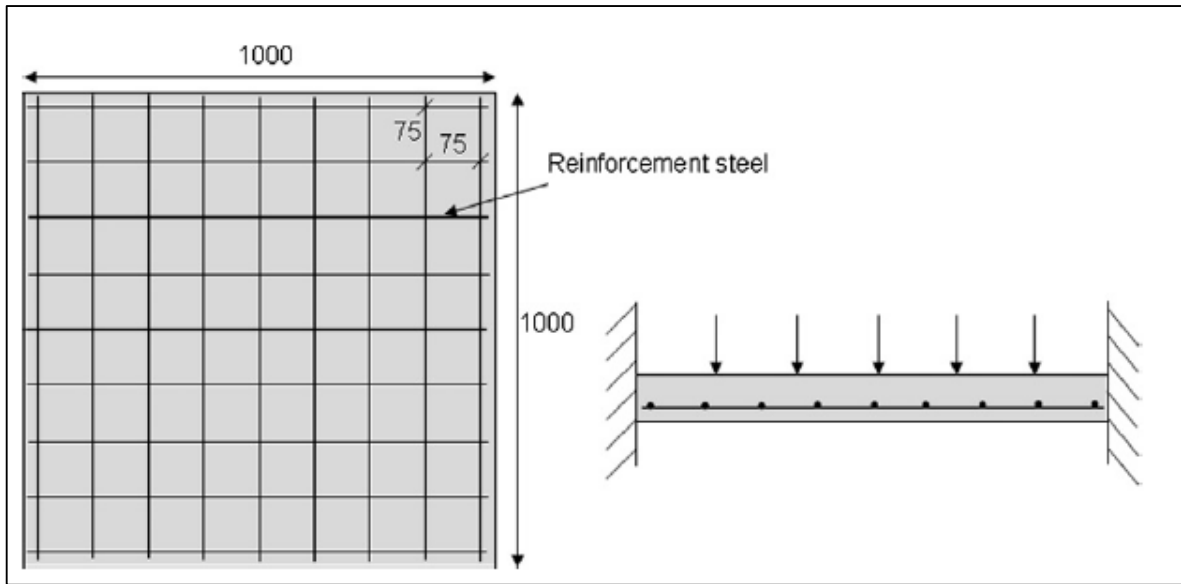


Figure 43: Slab Geometry in mm. (W. Wang et al., 2013)

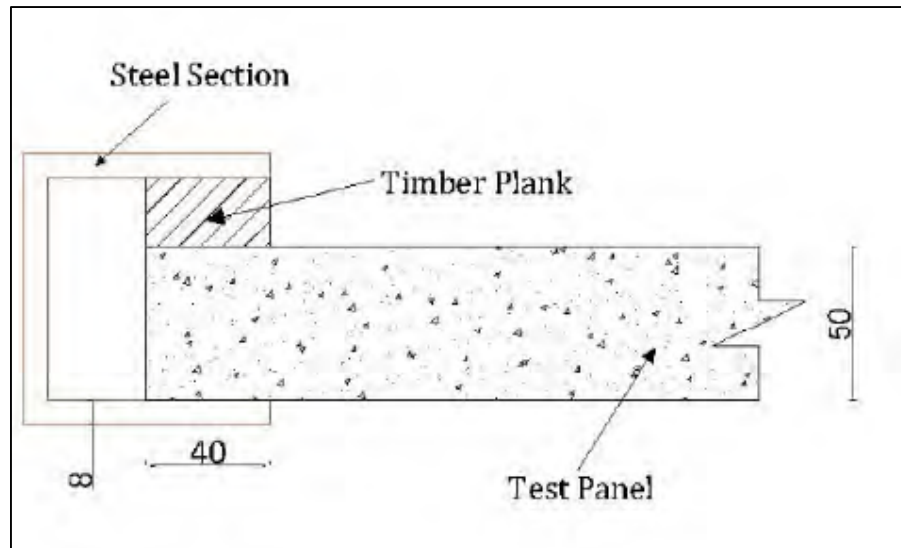


Figure 44: Support Dimensions (Appuhamilage, 2015)

Table 7 summarizes the material properties of concrete and reinforcing steel bars.

Material	Property	Strength (MPa)
Concrete	Compressive Strength (f'_c)	39.5 (31.6 equivalent cylinder strength)
	Elastic modulus (E_c)	28300
	Tensile Strength (f_t)	4.2
Reinforcing Steel	Yield Strength (f_y)	600
	Elastic Modulus (E_s)	200,000

Table 8: Material Properties (W. Wang et al., 2013)

Table 8 shows the cylindrical explosive charge (TNT) characteristics.

Weight (kg)	Standoff Distance (m)	Diameter (mm)	Height (mm)
0.31	0.4	31.4	62.7

Table 9: TNT Charge Characteristics

3.1.2. Element Formulation

In order to avoid mesh problems and high/extreme deformations; CEL modeling technique is used. Where the slab and supporting plates are modeled as 3D deformable solid elements, and the reinforcement bars as beam elements. The whole Lagrangian elements are modeled in a Eulerian background grid where the material flow is expected during the simulation.

Computational wise and due to symmetrical properties, only 1/4 of the slab is modeled (Figure 45).

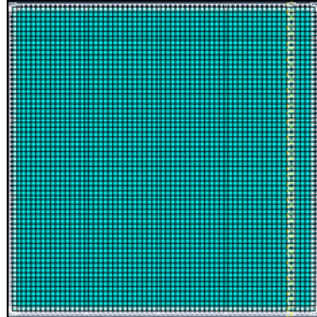


Figure 45: Geometric Model of 1/4 of the Slab

Name	Type
Eulerian	Eulerian
Section-1	Solid, Homogeneous
rebar	Beam, Constant
support	Solid, Homogeneous

Figure 46: Assigned Sections in the FE Model

3.1.3. Material Modeling

- **Concrete Material Model:**

Based on the previously discussed material model capabilities and literature recommendations, the Jh-2 material model is selected to reflect the concrete's behavior under blast load.

The most challenging aspect when dealing with Jh-2 is that the parameters availability is limited to special compressive strengths (mainly 46 Mpa for normal strength concrete).

Consequently, an intensive parametric study is done to reveal the most suitable material data (Figure 47).

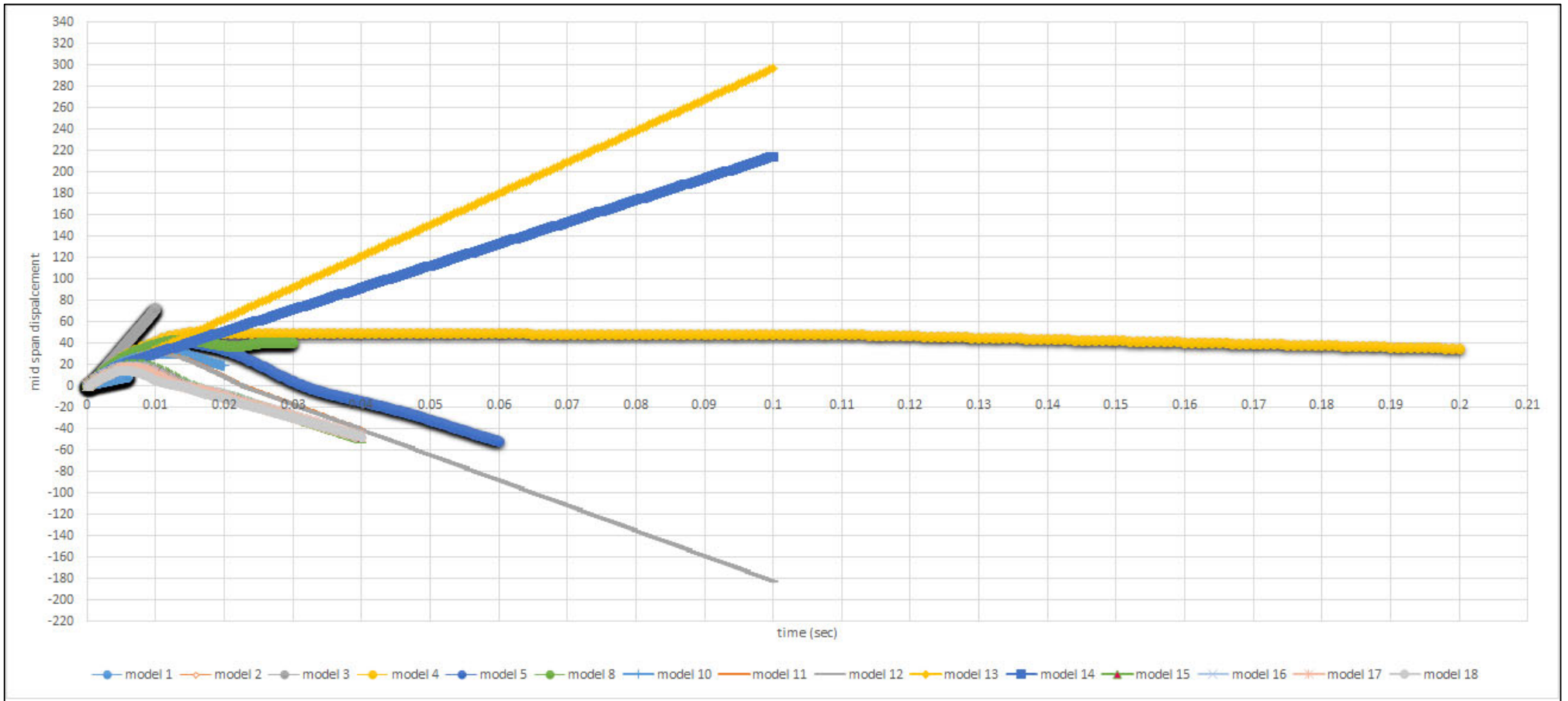


Figure 47: Parametric Study for Jh-2 Material Data- Midspan Deflection

The Johnson Holmquist model requires identifying 23 input data. Appendix A contains the input material models' parameters, mesh size, friction constants, and Eulerian boundary conditions used for the above shown outputs.

Table 9 compares the experimental midspan displacement results to the numerical results for model 18.

Experimental Midspan Displacement	Numerical Midspan Displacement	Percentage Error
15 mm	15.1407 mm	0.938 %

Table 10: Experimental vs. Numerical Midspan Displacement Results

○ ***Reinforcing Steel Material Model:***

Johnson Cook built-in material model is used to predict the rebars behavior.

The material parameters are: reference density, $\rho = 7.83 \text{ g/cm}^3$; bulk modulus, $K = 159 \text{ GPa}$; reference room temperature, $T_{\text{room}} = 300 \text{ K}$; specific heat = 477 J/kg K ; shear modulus, $G = 81.8 \text{ GPa}$; basic yield stress, $A = 792 \text{ MPa}$; hardening constant, $B = 510 \text{ MPa}$; hardening exponent, $n = 0.26$; strain rate constant, $C = 0.014$; thermal softening exponent, $m = 1.03$; and melting temperature, $T_{\text{melt}} = 1793 \text{ K}$ (W. Wang et al., 2013).

○ ***Air Material Model:***

In accordance to Shock Hugoniot relation (U_s-U_p) and using the experimental data of Deal, 1957; where the particle and shock velocity of a shock wave in air were recorded (pressure up to 200bar (2900psi))

(Deal, 1957)- the data required for Us-Up EOS was extracted by virtue of the linear interpolation (Figure 48).

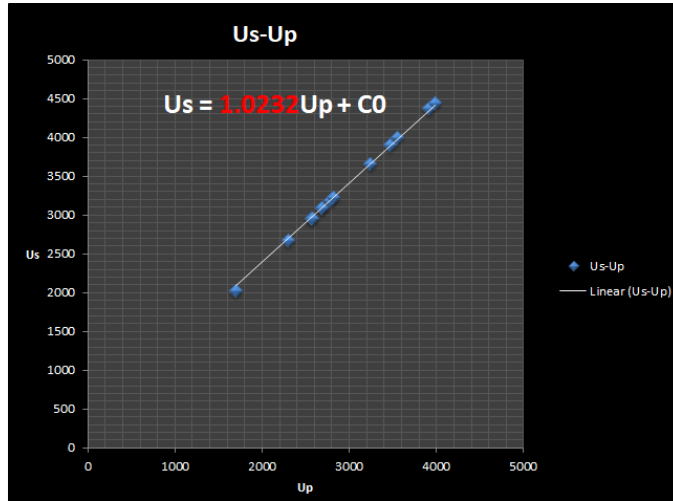


Figure 48: Us-Up Linear Interpolation

3.1.4. Load Definition

The blast load effect is described using CONWEP model as an incident wave by defining the equivalent TNT mass and locating the standoff distance (Figure 49 & 50).

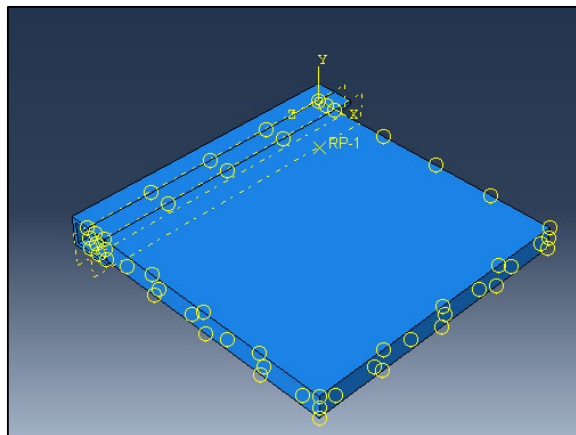


Figure 49: Locating the Standoff Distance Reference Point in the FE Model

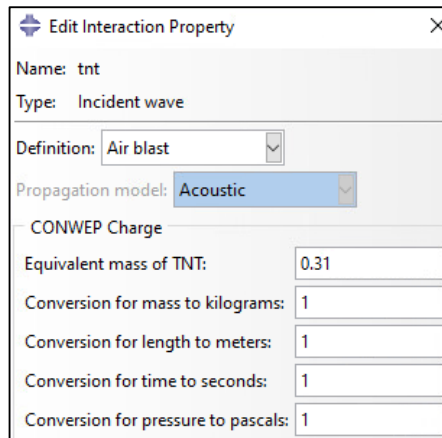


Figure 50: CONWEP Definition- Blast Load

3.2.Creating Toy Problem for Concept Validation

3.2.1.Model Description

After validating the material models for simulation against the experimental results, and for the sake of a faster convergence time; a forward finite element run is performed at a considerably large mesh size (toy run) and with a detonation mass of 0.5 kg of TNT.

Table 11 shows the mesh size considered for the forward run.

Element	Mesh Size (cm)
Eulerian Domain	20
Air	20
Reinforcement	10
Slab	20
Support	1

Table 11: Mesh Size Considered for the Forward Run

The damage output of this run is taken as the real state damage and reference for damage comparison. The toy run takes a computation time of approximately 8 minutes.

3.2.2. ABAQUS Input File Parameterization

To Simulate the real case, where the explosive's mass is an unknown variable, the Abaqus input file is parameterized in terms of TNT mass to be altered and optimized through a MATLAB code (Figure 51 and 52).

The Abaqus input files are text files that have a keyword based format: ASCII (character encoding)- American Standard Code for Information Interchange.

The concept was validated against two scenarios: the first one starting with an initial guess of 0.2 kg < 0.5 kg (Real), and the second one starting with a larger initial guess of 0.9 kg (Figure 51 and 52).

```
**  
** INTERACTION PROPERTIES  
**  
*parameter  
mass=0.2  
*Surface Interaction, name=default  
*Surface Interaction, name=support  
*Friction  
0.6,  
*Surface Behavior, pressure-overclosure=HARD  
*Surface Interaction, name=supportdown  
*Friction  
0.57,  
*Surface Behavior, pressure-overclosure=HARD  
*Incident Wave Interaction Property, name=tnt, type=AIR BLAST  
*CONWEP Charge Property  
<mass>, 1.  
1., 1., 1.  
**
```

Figure 51: ABAQUS Parameterized Input File for First Scenario

```

**
** INTERACTION PROPERTIES
**
*parameter
mass=0.9
*Surface Interaction, name=default
*Surface Interaction, name=support
*Friction
0.6,
*Surface Behavior, pressure-overclosure=HARD
*Surface Interaction, name=supportdown
*Friction
0.57,
*Surface Behavior, pressure-overclosure=HARD
*Incident Wave Interaction Property, name=tnt, type=AIR BLAST
*CONWEP Charge Property
<mass>, 1.
1., 1., 1.
**

```

Figure 52: ABAQUS Parameterized Input File for Second Scenario

3.3 Simulated Damage Output & Odb Scripting

The scripting Language of Abaqus is Python. In order to automatically access the contour Plot of the simulated damage from the Odb Abaqus file after every iteration, a python script is included in a section of the MATLAB code.

Through this script, the user can easily specify the job name, plot type, and face view to be extracted. Subsequently, the Simulated damage output is analyzed through image processing techniques.

Appendix B contains the python script used for damage contour plot extraction.

3.4. Image Processing Using MATLAB

3.4.1. Implementing Edge Detection Algorithm Using Modified Canny Edge Detector

The RGB images of the real and simulated damage are equally resized, and converted into binary images through the command `rgb2gray`.

In order to reduce noise the gray scale images are converted to binary images using a modified canny edge method. At first, the gray image matrix is divided by the maximum pixel intensity to ensure that all threshold values are below 1 and can be displayed on the image. After that, the threshold value to be used for edge detection is specified depending on the main cracks the user chooses to display. The canny edge is applied using a single threshold value rather than a minimum and maximum one.

After the Binarization, to fill the spaces and clean the binary images some morphological operations available in matlab are applied:

- Bridge: Bridges the unconnected pixels in case of the presence of two connected neighbors.
- Close: Applies dilation followed by erosion.
- Clean: Removes the isolated pixels.

In the final image; every one (1) entity inside the binary matrix stands for a cracked location, while un-cracked locations are labeled as zeros (0).

3.4.2.Implementing Complex Wavelet Structural Similarity Index (CW-SSIM)

To decide upon the best similarity index; a comparison was done between the feature similarity index(FSIM), and complex wavelet structural similarity index (CW-SSIM).

The comparison was made between images extracted manually from different finite element runs representing multiple detonation masses.

Figures 53 and 54, show graphs where each line represents a damaged slab corresponding to a specific mass mentioned in the legend.

The graphs illustrate the general trends and similarity values of each of the methods used to compare the damage of the reference slabs to that of other slabs' damage corresponding to different detonation masses (TNT mass in kg).

As shown in Figure 53, although the similarity index values are high, the trend line of the behavior is not reasonably justified. If looking at the red line representing a real state damage for a slab with 0.4 kg TNT explosive mass, it can be noticed that the FSIM indicates 100% similarity when the detonation mass is 0.4 kg. It is supposed that as the detonation mass gets far away of the real detonation value, the FSIM value decreases in linear manner. The Trend line shows an unreasonable fluctuating behavior. Another drawback is that the index is always giving a value higher than 85% similarity despite the difference between the detonation masses. It is obvious that for 0.8 kg guess mass when compared to the 0.4 kg real TNT mass slab's damage (50% error) the similarity reflected through the FSIM is 85% which is a very high inaccurate value. Hence, the index is not accurate and insensitive to the mass variation .

According to Figure 54, although the similarity index isn't that high in values, the trend line of the behavior is reasonably justified as the values of the masses varies away of that of the reference slab's mass. When analyzing the red line representing a real state damage for a slab with 0.4 kg TNT explosive mass, it can be noticed that the CW-SSIM indicates 100% similarity when the detonation mass is 0.4 kg. As the detonation mass gets far away of the real detonation value, the CW-SSIM value decreases in a linear manner. This

accuracy of this index is more verified when inspecting the results of the other reference slabs' graphs. As for the 0.6 kg reference detonation mass (blue curve) , the CW-SSIM (%) strictly increases as the guessed detonation mass (kg) increases towards 0.6 kg (100% similarity – peak value) and then continues to strictly decrease as the guessed detonation mass (kg) increasingly varies away of 0.6 kg. The maximum value of similarity is only achieved when the masses are identical (100%).

Thus, the CW-SSIM is the most reliable index to be used for image comparison.

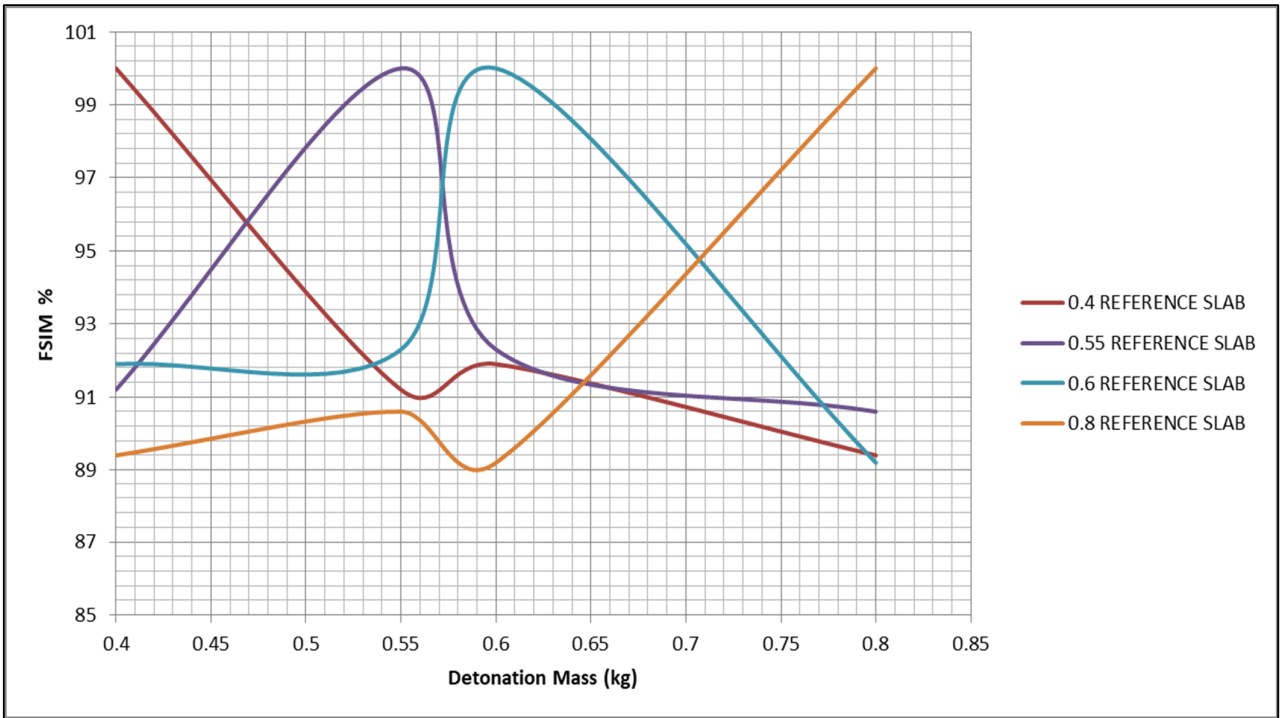


Figure 53: FSIM Values and Trend Lines

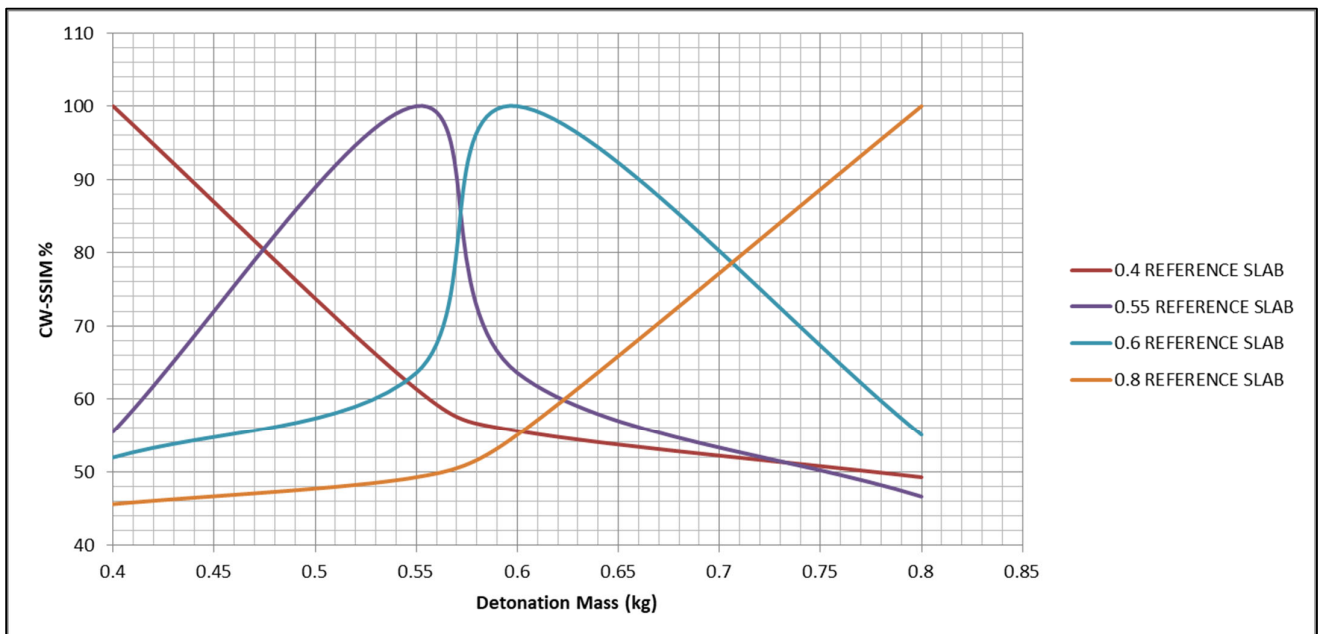


Figure 54: CW-SSIM Values and Trend Lines

To compute the CW-SSIM:

1. Decompose the images using complex steerable pyramid decomposition.
2. Create a local sliding window (7*7) that moves across each wavelet
3. A weighted summation is used to integrate the CW-SSIM index map into a scalar similarity measure. The weighting function is calculated using a Gaussian profile with a standard deviation equals to quarter of the finest decomposed level image.

3.5.Implementing Genetic Algorithm Optimization

To specify the optimization range, the preliminary run is executed based on an initial guess of the explosive mass based on the below reference table (Table 10).

DAMAGE LEVEL	SCALE DISTANCE (Z)(m/Kg ^{1/3})	Rotational DAMAGE (°)	DISPLACEMENT DAMAGE (mm)	MASS I.GUESS (Kg)
LOW DAMAGE	Z>0.68	0<θ<1.7	X _m <15	$m = \left(\frac{R}{0.68}\right)^3$
MODERATE DAMAGE	0.5<Z<0.68	1.7<θ<4.6	15<X _m <40	$m = \left(\frac{R}{0.59}\right)^3$
HIGH DAMAGE	0.35<Z<0.5	4.6<θ<6.8	40<X _m <60	$m = \left(\frac{R}{0.425}\right)^3$
COLLAPSE	Z<0.35	θ>6.8	X _m >60	$m = \left(\frac{R}{0.35}\right)^3$

Table 12: Explosive's Mass in Accordance to the Displacement and Rotational Damage (Abedini & Mutalib, 2020)

*R: is the charge standoff distance.

After analyzing the binary matrices of the preliminary run (initial guess); and followed by the comparison of the number of 1's entities in each matrix: the bounds of the optimization are saved to be:

✚ [initial guess , 6*initial guess] if the ones entities in the real damage matrix are more than those in the simulated damage matrix.

✚ [0, initial guess] if the ones entities in the real damage matrix are less than those in the simulated damage matrix.

The process is iterative until reaching the minimum value of the objective function (error). The error function is $1 - CW-SSIM\%$. Zero error is only reached if the real detonation mass is 100% reached through the process.

Matlab is used to implement the genetic algorithm for detonation mass optimization. Figure 55 summarizes the genetic algorithm integration.

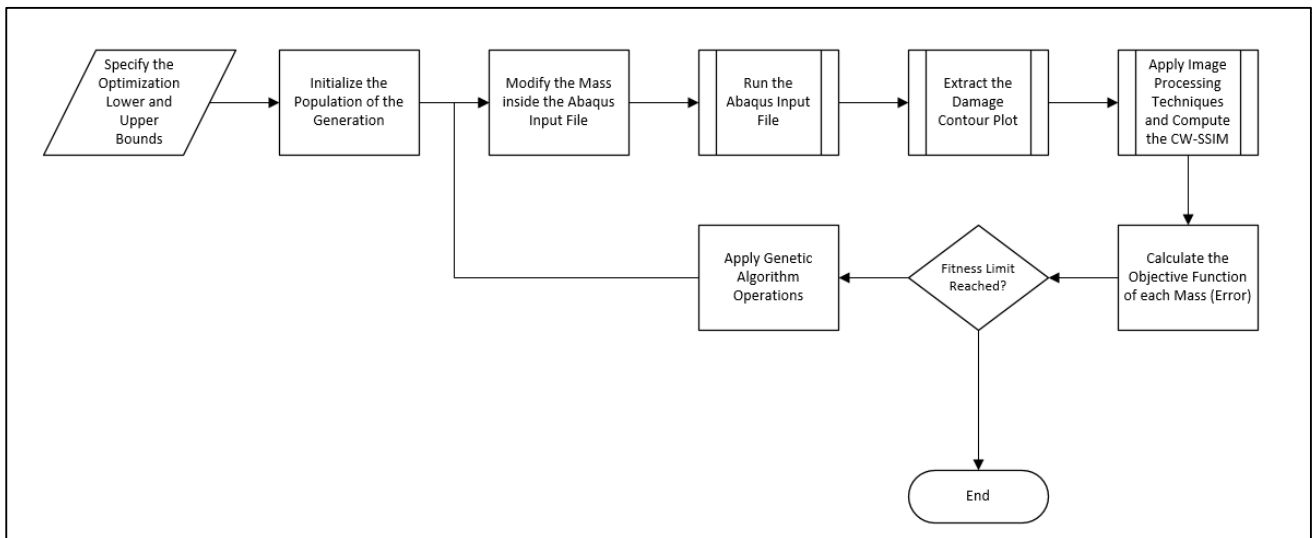


Figure 55: Genetic Algorithm Integration

The population size of the GA is set to be 5. The optimization process ends when reaching a fitness Limit (error value) of 3% or below.

The complete code of the whole algorithm is included in Appendix C.

Table 13 shows the options used for the genetic algorithm.

GA Option	Description
Population Creation Option	Uniform distribution in the initial population range
Fitness Scaling Option	Proportional scaling makes the scaled value of an individual proportional to its raw fitness score.
Selection Option	GA lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on.
Cross-Over Option	Creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child
Mutation Option	Adds a random number taken from a Gaussian distribution with mean 0 to each entry of the parent vector.

Table 13: Options Used for GA Implementation

CHAPTER 4

RESULTS AND DISCUSSION

For the user to safely run the Matlab main code, all the needed files should be saved to the same working directory (i.e. ABAQUS cae model, python script, matlab script, real state damage image).

After running the script, the user has to specify the working directory, the name of the parameterized ABAQUS input file, the value of the detonation mass's initial guess, the name of the ABAQUS explicit job, and the face view of the simulated contour plot to be extracted (Figure 56).

Subsequently, the ABAQUS first job will be automatically submitted based on the user's initial guess. After the job completion, the user has to input the name of the real state damage image specifying its format (Figure 57).

Then, the user can indicate the coordinates needed to crop the real state damage image (Figure 58). This step allows the algorithm to automatically resize the real and simulated contour plot prior to analysis.

Now the optimization process using the genetic algorithm can start. For every iteration, the user has to specify some input data. This data is limited to: the value of the initial guess, the name of the ABAQUS parameterized input file, the name of the job, and the face view of the simulated contour plot to be extracted (Figure 59).

During the iteration, the user has to specify the threshold value to be used for the edge detection depending on the main cracks he/she chooses to display (Figure 60).

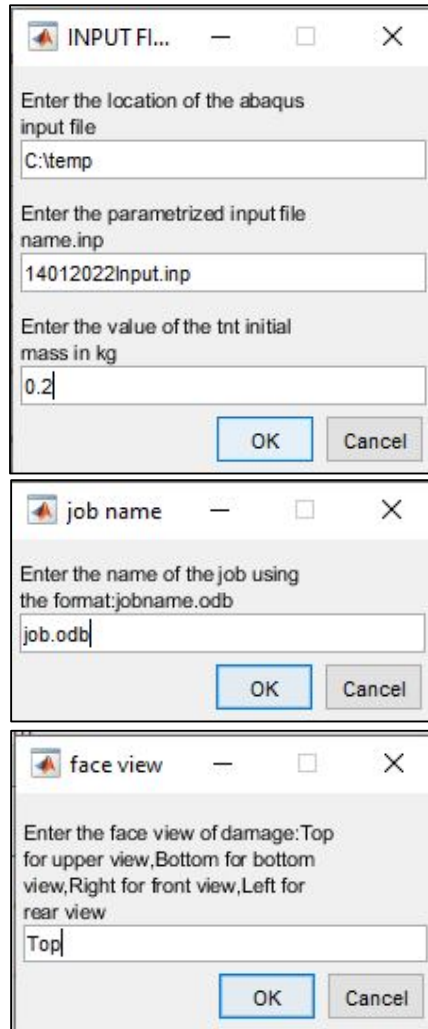


Figure 56: First Set of Input Data for the Algorithm

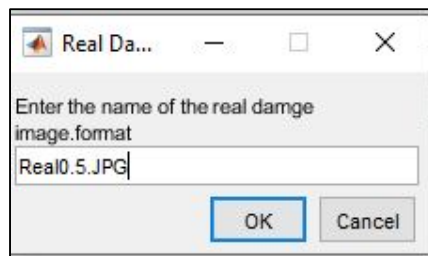


Figure 57: Inputting the Real Damage Image Name and Format

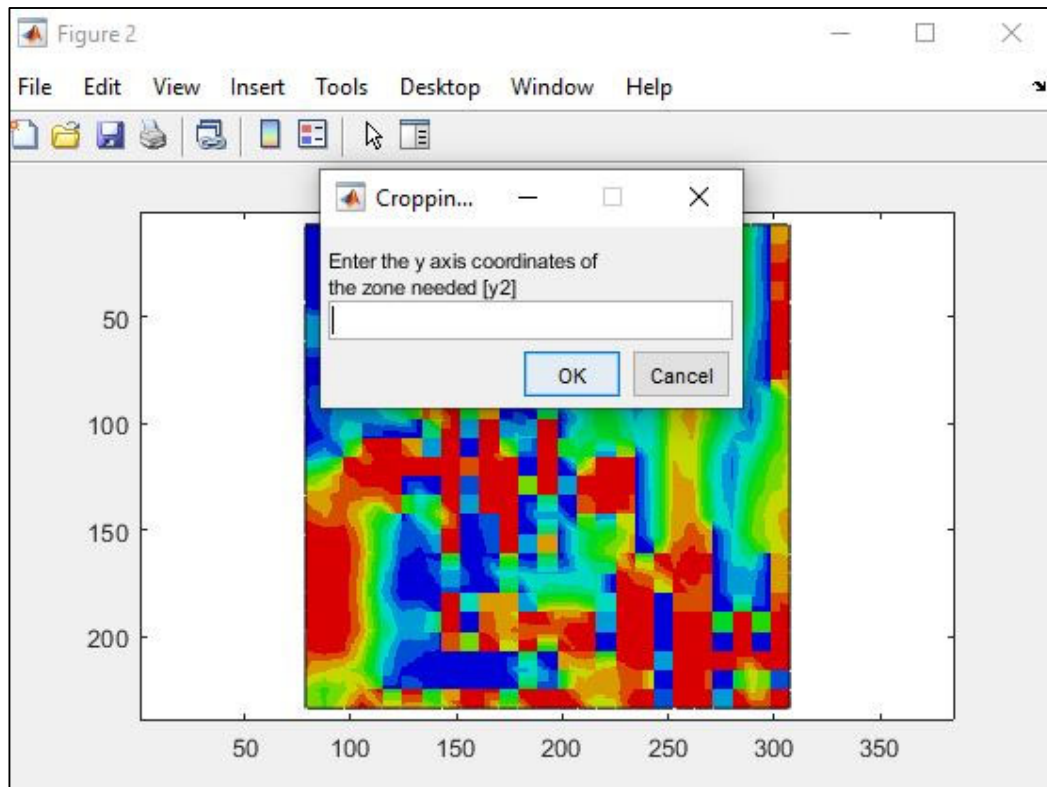


Figure 58: Cropping the Real Damage Image

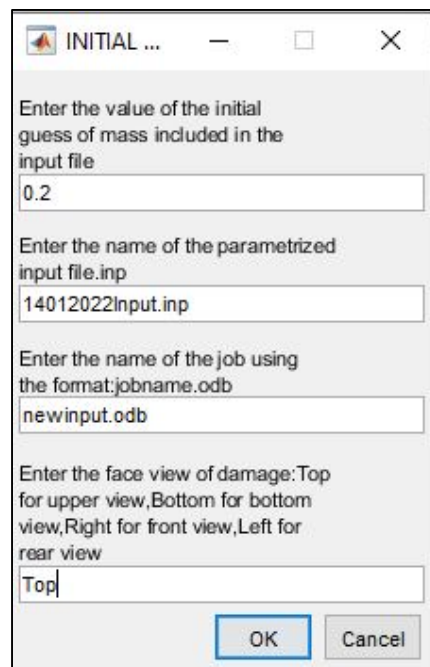


Figure 59: GA Input Data

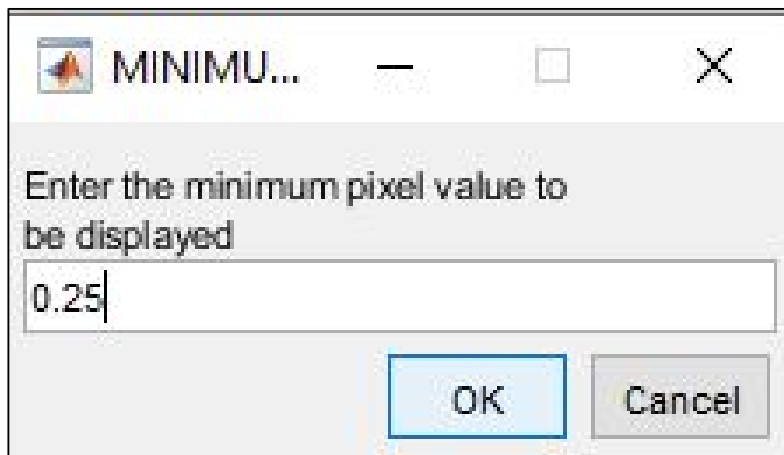
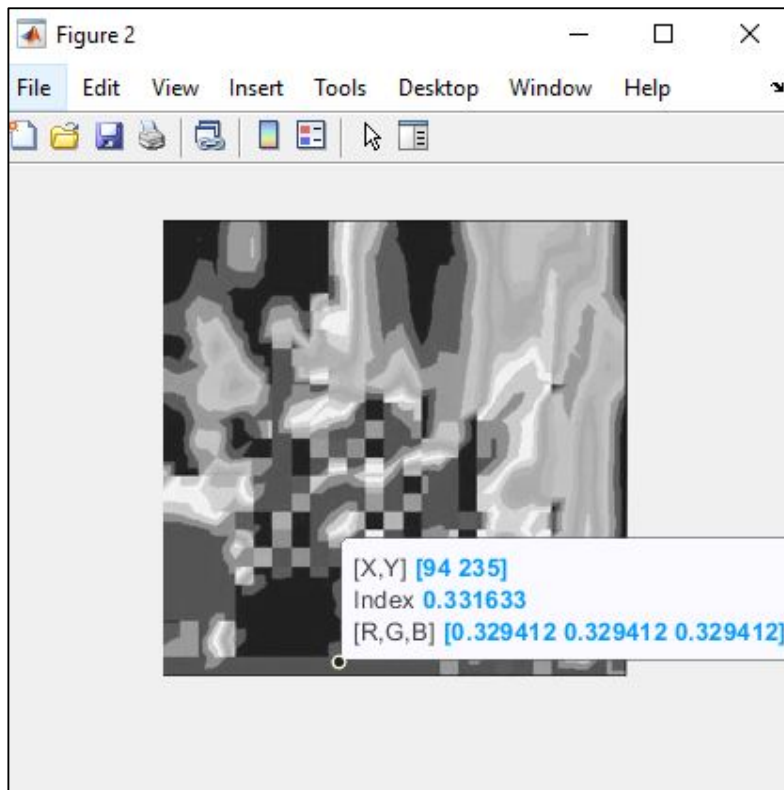


Figure 60: Specification of Minimum Threshold Value to be Displayed

4.1.Initial Guess Lower than the Real State Detonation Mass

The algorithm is tested against a first example starting with an initial guess less than that of the real state damage. Table 14 shows the input data for this example.

INPUT	
Initial Guess	0.2 kg
Real State Damage	0.5 kg
Population Size	5
Fitness Limit	3% = 0.03

Table 14: Input Data

After the preliminary run the boundaries of the optimization were saved to be [0.2,1.2]. The algorithm automatically compares the damage states of both the real and simulated cases. Since the scenario starts with an initial guess lower than the real mass, then the optimization bounds are [lower bound, 6* lower bound].

The population size was specified based on a simple sensitivity study to relate the effect of the population size to the convergence time (Figure 61).

This sensitivity study applies to problems with optimization searching space similar in size to that of this example.

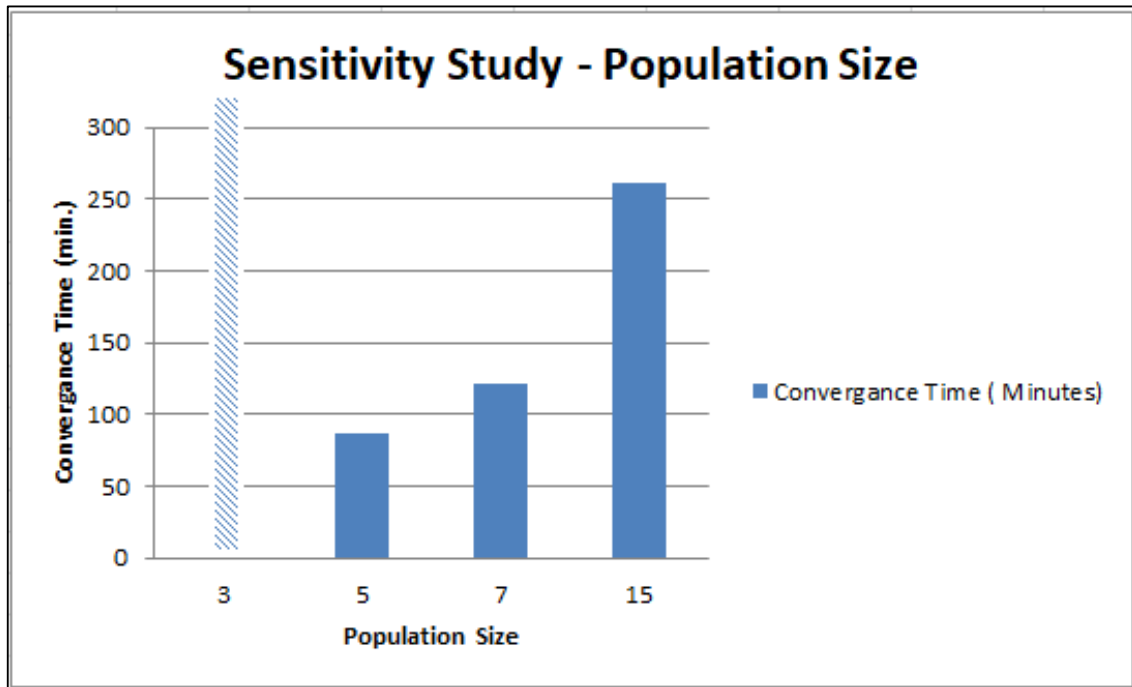


Figure 61: Sensitivity Study Relating Population Size to Convergence Time

As depicted by Figure 61, with population size of 3 no convergence is reached due to lack of diversity between individuals. For greater population size of 5, 7, and 15 individuals. The specified fitness limit (error of 3% or lower) is reached after 2 generations. Consequently, the population size is fixed to 5 individuals per iteration.

The algorithm converged after a total computational time of 87 minutes.

Table 15 shows the variation of the iterated mass (kg) between the generations until convergence.

	Mass (kg)
Generation 0	0.2
	1.15022
	0.234446
	0.581558
	0.638744
Generation 1	0.581558
	0.371187
	0.541846
	0.511524
	0.511524

Table 15: Variation of Iterated Explosive's Mass Between Generations

The optimization starts with a random mass generated along the lower and upper bound (0.2 kg). This step is followed by a random mutation process to generate the individuals of the population. After that, the objective function: $\text{error} = 1 - \text{CW-SSIM}\%$ is computed for every individual to decide upon the fittest which is proportionally scaled according to its raw fitness score. The fittest individual of each generation survives to the successor generation (0.581558 kg).

In the second generation, the steps of the GA are repeated until reaching an individual's fitness limit of a lower value than the pre-specified one (3%).

The algorithm predicted the mass with a total of 2.3048% error corresponding to a detonation mass of 0.511524 kg. This mass is associated with a similarity percentage (CW-SSIM) of 97.6952.

4.2. Initial Guess Higher than the Real State Detonation Mass

The algorithm is also tested against a second example starting with an initial guess greater than that of the real state damage. Table 16 shows the input data for this example.

INPUT	
Initial Guess	0.9 kg
Real State Damage	0.5 kg
Population Size	5
Fitness Limit	3% = 0.03

Table 16: Input Data

After the preliminary run the boundaries of the optimization were saved to be [0,0.9]. The algorithm automatically compares the damage states of both the real and simulated cases. Since the scenario starts with an initial guess higher than the real mass, the optimization bounds are [zero, upper bound].

The search space's size is similar to that of the previous example, thus the same population size is used (5 individuals per generation).

The algorithm converged after the first generation towards a fitness limit less than 3%. The total computational time is 45 minutes. Table 17 shows the variation of the iterated mass (kg) through the generation until convergence.

	Mass (kg)
Generation 0	0.143990
	0.855478
	0.732359
	0.497540
	0.501754

Table 17: Variation of Iterated Explosive's Mass Across the Generation

The optimization starts with a random mass generated along the lower and upper bound (0.143990). This step is followed by a random mutation process to generate the individuals of the population. After that, the objective function: error = 1- CW-SSIM% is computed for every individual to decide upon the fittest which is proportionally scaled according to its raw fitness score. The fittest individual of each generation survives to the successor generation (0.501754 kg).

The algorithm predicted the mass with a total of 0.3508% error corresponding to a detonation mass of 0.501754 kg. This mass is associated with a similarity percentage (CW-SSIM) of 99.6492.

CHAPTER 5

CONCLUSION AND FUTURE WORK

As an outcome of this research, the following can be concluded:

- ❖ The CONWEP model can effectively describe the blast effect on reinforced concrete structures.
- ❖ The Jh-2 and JC material models can adequately simulate the behavior of concrete and reinforcing steel respectively under blast load.
- ❖ CW-SSIM is the most accurate similarity index to be used when comparing images.
- ❖ The proposed algorithm succeeded in predicting the detonation mass with an accuracy between 97 to 99 %.

For advancement and expansion of the proposed algorithm in order to cover a wider area and contribute with better efficiency; the following topics may be considered as future works:

- Using a written subroutine for a better simulation of the material's behavior under blast load.
- Testing the algorithm efficiency for more complex examples and different structure types.
- Extending the work to cover the prediction of a blast of both unknown detonation mass and center's location.
- Applying the algorithm to analyze real blast case study for example Beirut Port Blast can be used as a reference case.

- Making use of the artificial intelligence combined with computer vision in the analysis of images so that complicated damage states can be effectively considered.
- Upgrading the framework with a structural health monitoring feature, through compiling the structural capacities formulas into the algorithm in a way to monitor the damaged elements condition.

APPENDIX 1

JH-2 PARAMETRIC STUDY

CEL modeling Double symmetry		REFERENCE																																				
Parameter		Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12	Model 13	Model 14	Model 15	Model 16	Model 17	Model 18																			
Air	Density = 1.225 kg/m ³	SAME																																				
	DOS Up-ix (D=143, v=1.232, gamma=1.4)																																					
	Specific heat= 20600000																																					
	Viscosity= 8.25e-5																																					
TNT	Castell model 0.31 kg of TNT	SAME																																				
rho0 =	2400																			-	-	-	-	-	-	2350	2350	2350	2400	2400	2400	2400	2400	2400	2400	2400	2400	
G =	1482000000																			-	-	1132000000	1132000000	-	-	1.00E+10	1.00E+10	1.00E+10	1.25E+10	1.25E+10	1.25E+10	1.25E+10	9.20E+11	2E+12	1.30E+12	1.30E+12	1.30E+12	1.00E+12
A (captures response)	1.6																			0.5	-	0.5	0.5	1.3	1.3	1.3	1.6	1.6	7.8304	7.8304	7.8304	7.8304	7.8304	7.8304	7.8304	7.8304	7.8304	7.8304
HD-CONCRETE	N =	0.61	0.63	-	0.63	0.63	-	-	0.88	0.88	0.61	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437																		
	B (As it increases the slip value decreases)	0.7	1.87	-	1.87	1.87	2	2	2	1	0.4101	0.4101	0.4101	0.4101	0.4101	0.4101	0.4101	0.4101	0.4101	0.4101																		
	M =	0.8	0.61	-	0.61	0.61	-	-	0.6	0.6	0.8	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437	0.8437																		
	C =	0.007	-	-	-	-	-	-	-	-	0.007	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006																		
	edot0 =	1	-	-	-	-	-	-	-	-	1	1	1	1	1	1	1	1	1	1																		
	T =	3.54E+06	4.20E+06	-	4.20E+06	4.20E+06	4.20E+06	4.20E+06	4.20E+06	2.30E+06	2.30E+06	2.80E+06	3.40E+06	3.40E+06	3.40E+06	3.40E+06	3400000	3400000	3400000	3400000																		
	sigMax =	8.00E+09	-	-	-	-	-	-	7.00E+09	7.00E+09	7.00E+09	7.00E+09	7.00E+09	7.00E+09	7.00E+09	7000000000	7000000000	7000000000	7000000000																			
	sigTMax =	1.10E+09	-	-	-	-	-	-	-	-	1.10E+09	1.10E+09	1.10E+09	1.10E+09	1.10E+09	1100000000	1100000000	1100000000	1100000000																			
	HEI =	8.00E+07	-	-	-	-	-	-	-	-	1.48E+09	1.48E+09	1.48E+09	1.48E+09	1.48E+09	1481000000	1481000000	1481000000	1481000000																			
	PHI =	4.90E+07	-	-	-	-	-	-	-	-	8.11E+08	8.11E+08	8.11E+08	8.11E+08	8.11E+08	811000000	811000000	811000000	811000000																			
	beta =	1	-	-	-	-	-	-	-	-	1	1	1	1	1	1	1	1	1																			
	D1 =	0.04	-	-	-	-	-	-	-	-	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6																			
	D2 =	1	-	-	-	-	-	-	-	-	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1																			
	alpha =	1	-	-	-	-	-	-	-	-	1	1	1	1	1	1	1	1	1																			
	alpha0 =	0.0008	-	-	-	-	-	0.001	0.001	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01																			
	K1 =	8.50E+10	-	-	-	-	-	-	-	-	1.67E+10	1.67E+10	1.67E+10	1.67E+10	1.67E+10	1667000000	1667000000	1667000000	1667000000																			
	K2 =	-1.71E+11	-	-	-	-	-	-	-	-	7.10E+10	7.10E+10	7.10E+10	7.10E+10	7.10E+10	7100000000	7100000000	7100000000	7100000000																			
	K3 =	2.98E+11	-	-	-	-	-	-	-	-	-2.98E+11	-2.98E+11	-2.98E+11	-2.98E+11	-2.98E+11	-2.98E+11	-2.98E+11	-2.98E+11	-2.98E+11																			
	RS =	0.2	-	-	-	-	-	-	-	-	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2																			
	Damage =	0	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0																			
Steel-C	Density = 7850 kg/m ³	SAME																																				
	E=200e9																																					
	Poisson = 0.28																																					
	Plastic: Johnson-Cook																																					
	A= 792e6																																					
	B=210e6																																					
	n=0.26																																					
	m=1.03																																					
	melting=1793																																					
	transition=900																																					
c=0.014																																						
epslon_dot zero=1																																						
Friction	Support = 0.6 (General Contact); shape of displacement curve at end	-	-	-	-	-	-	0.2	0.6 and 0.57	0.6 and 0.58	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2																		
Slip	Fixed up- slip down	-	-	-	Fixed up- slip	Fixed up- slip down	-	-	Fixed up- slip down	Fixed up- slip down	20e-6	20e-6	20e-6	20e-6	20e-6	20e-6	20e-6	20e-6	20e-6	20e-6																		
Time step	Time step	0.005e-5ms	-	0.01e-10ms	0.2e-20ms	0.06e-6ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms	0.01e-10ms																			
Mesh	mesh size	2 mm	-	20 mm	20 mm	-	-	-	-	8mm	8mm	8mm	8mm	8mm	8mm	8mm	8mm	8mm	8mm																			
Time	run time	9 hours	-	1h 5 min	2h min	-	-	-	-	35 min	35min	35min	35min	35min	3.5 hrs	3.5 hrs	3.5 hrs	3.5 hrs	3.5 hrs																			
Ed. B.C.	Eulerian Boundary Conditions	out non- refl.	-	-	-	-	-	-	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.	out non- in equal.																			

APPENDIX 2

PYTHON SCRIPTING

```
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=45.4553451538086,
    height=59.1360015869141)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup

# to open the job results
#name the file parametrized by find and replace
#jobname(this one to be found then replaced and to be put in line 17 using the format
shown)

o1 = session.openOdb(name='C:/temp/jobname')
session.viewports['Viewport: 1'].setValues(displayedObject=o1)

#in line 55 specify where to save the output
#in line 36 jobname
#in line 48 the view is specified
session.viewports['Viewport: 1'].setValues(displayedObject=o1)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].setValues(
    displayedObject=session.odbs['C:/temp/jobname'])
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    optimizationTasks=OFF, geometricRestrictions=OFF, stopConditions=OFF)
leaf = dgo.LeafFromPartInstance(partInstanceName=("SLAB-1", ))
session.viewports['Viewport: 1'].odbDisplay.displayGroup.replace(leaf=leaf)
session.viewports['Viewport: 1'].view.fitView()
session.viewports['Viewport: 1'].odbDisplay.setPrimaryVariable(
    variableLabel='SDV_DAMAGE', outputPosition=INTEGRATION_POINT, )
session.viewports['Viewport: 1'].odbDisplay.display.setValues(
    plotState=CONTOURS_ON_DEF)
session.viewports['Viewport: 1'].odbDisplay.commonOptions.setValues(
    visibleEdges=FEATURE)
session.viewports['Viewport: 1'].view.setValues(session.views['face'])
```

```
session.viewports['Viewport: 1'].viewportAnnotationOptions.setValues(triad=OFF,  
    legend=OFF, title=OFF, state=OFF, annotations=OFF, compass=OFF)
```

```
session.printOptions.setValues(vpDecorations=OFF, reduceColors=False)  
session.printToFile(fileName='C:/temp/damage-image', format=PNG, canvasObjects=(  
    session.viewports['Viewport: 1'], ))
```

APPENDIX 3

MATLAB CODE

```
%%
%*****COMPLETE
CODE*****
%% THE WHOLE FILES SHOULD BE PLACED IN THE SAME
FOLDER%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%first run to guess the bounds
%for the sake of parametrizing the input file
%under the interaction section
%*parameter
%mass=initial guess
%instead of mass value replace by <mass>
%prompt 1 (work directory and parametrized input file name)
prompt = { 'Enter the location of the abaqus input file','Enter the
parametrized input file name.inp','Enter the value of the tnt initial mass in
kg'};%follow input name by .inp
dlgtitle='INPUT FILE NAME';
dims=[1 35];
answer=inputdlg(prompt,dlgtitle,dims);
%%
%iterating the mass inside the input file
newfolder=answer{1};%new workdirectory
parametrized_input=answer{2};%parametrized abaqus input file name
mass1=answer{3};
cd(newfolder);
%%
%input the job name(job) and view (faceview) to parametrize the python
%script
prompt = { 'Enter the name of the job using the format:jobname.odb'};
dlgtitle='job name';
dims=[1 35];
name=inputdlg(prompt,dlgtitle,dims);
job=name{1};
%second the faceview
prompt = { 'Enter the face view of damage:Top for upper view,Bottom for
bottom view,Right for front view,Left for rear view'};%should be
completed
dlgtitle='face view';
dims=[1 35];
viewing=inputdlg(prompt,dlgtitle,dims);
faceview=viewing{1};
inputfile=fileread(parametrized_input);%read the parametrized input file
finput=fopen('InitialInput.inp','w');%open new input file
fprintf(finput,inputfile);%write the new input file
fclose(finput);%close the new input
%%
%first run in abaqus
%run abaqus
!abaqus job=job input=InitialInput.inp
%pause matlab to wait for abaqus run to end
```

```

pause(10)
while exist('job.lck','file')==2
    pause(0.1)
end
%%
%%result file should be in the same folder and all folders in temp
%extract abaqus damage (save) using python scripting
%python file is parameterized
odbread='odbread.py';
odbfile=fileread(odbread);%read the parametrized odbread.py file
edited_odb_file=regexprep(odbfile,{'jobname','face'},{job,faceview});%change
the mass
fid=fopen('newodbread.py','w');%open new odbread file
fprintf(fid,edited_odb_file);%write the new odbread file
fclose(fid);%close the new odbread
%the below file should be the final modified one (parametrized)
!abaqus cae script=newodbread.py
%%
%real damage image
prompt = { 'Enter the name of the real damage image.format'};
dlgtitle='Real Damage Image';
dims=[1 35];
real=inputdlg(prompt,dlgtitle,dims);
Rimage=real{1};
B=imread(Rimage);
%%
%abaqus damage image
q=imread('damage-image.PNG');% the name of the saved output image %%%
figure
imshow(q);imagesc(q);
%crop the image
prompt1 = { 'Enter the y axis coordinates of the zone needed [y2]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
crop1=inputdlg(prompt1,dlgtitle,dims);
y2=crop1{1};
Y2=str2num(y2);
prompt2 = { 'Enter the y axis coordinates of the zone needed [y1]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
crop2=inputdlg(prompt2,dlgtitle,dims);
y1=crop2{1};
Y1=str2num(y1);
prompt3 = { 'Enter the x axis coordinates of the zone needed [x2]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
crop3=inputdlg(prompt3,dlgtitle,dims);
x2=crop3{1};
X2=str2num(x2);
prompt4 = { 'Enter the x axis coordinates of the zone needed [x1]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
crop4=inputdlg(prompt4,dlgtitle,dims);
x1=crop4{1};
X1=str2num(x1);
y=(Y1)-(Y2);
x=(X2)-(X1);

```

```

targetSize=[y x];%size of the cropped image
r = centerCropWindow2d(size(q),targetSize);
A = imcrop(q,r);%cropped image
imshow(A);
%%
%resize both images to the same size
[v,d,f]=size(A);
R=imresize(B,[v,d]);%real damage modified
%change both images to grayscale & edge detection
Red_gray=rgb2gray(A);
r_gray=rgb2gray(R);
Ae = edge(Red_gray, 'Canny');
Re = edge(r_gray, 'Canny');
% figure
% imshow(Ae);
% figure
% imshow(Re);
%%
%to select the range of optimization
f1=find(Re==1);
f2=find(Ae==1);
if size(f1,1)>size(f2,1)
    lower=mass1;
    LOWER=str2num(lower);
    UPPER=6*LOWER;
elseif size(f1,1)<size(f2,1)
    LOWER=0;
    upper=mass1;
    UPPER=str2num(upper);
end
%%
%GA Optimization Command
nvars=1;%the mass is the variable to be optimized
LB=LOWER;%lower bound
UB=UPPER;%upper bound
options=optimoptions('ga','PopulationSize',5,'FitnessScalingFcn',{@fitscaling
prop},'FitnessLimit',0.03,'PlotFcn','gaplotbestf');%the algorithm will stop
after reaching a similarity of 60% or above

[mass,fval]=ga(@myFunction,nvars,[],[],[],[],LB,UB,[],options);
%%
%the function of damage interms of mass (function nonzeros)
function error = myFunction (mass)
%%
%*****not included in GA initial part*****
%retrieving the mass initial guess and mass of the current iteration
prompt = { 'Enter the value of the initial guess of mass included in the
input file','Enter the name of the parametrized input file.inp','Enter the
name of the job using the format:jobname.odb','Enter the face view of
damage:Top for upper view,Bottom for bottom view,Right for front view,Left
for rear view'};
dlgtitle='INITIAL GUESS OF MASS';
dims=[1 35];
solution=inputdlg(prompt,dlgtitle,dims);
initialguess=solution{1};
parametrized_input=solution{2};
job=solution{3};

```

```

faceview=solution{4};
%%
%*****not included in GA*****
%our loop will start here for each and every mass
%initializing the new iteration input file
inputfile=fileread(parametrized_input);%read the parametrized input file
edited_input_file=regexprep(inputfile,sprintf('mass=%s',initialguess),sprintf
('mass=%g',mass));%%%%change the mass by optimization
fid=fopen('newinput.inp','w');%open new input file
fprintf(fid,edited_input_file);%write the new input file
fclose(fid);%close the new input
%%
%run abaqus
!abaqus job=newinput input=newinput.inp
%pause matlab to wait for abaqus run to end
pause(10)
while exist('newinput.lck','file')==2
    pause(0.1)
end
%%
%%%%result file should be in the same folder and all folders in temp
%extract abaqus damage (save) using python scripting
%python file is parameterized
odbread='odbread.py';
odbfile=fileread(odbread);%read the parametrized odbread.py file
edited_odb_file=regexprep(odbfile,{'jobname','face'},{job,faceview});%change
the mass
fid=fopen('newodbread.py','w');%open new odbread file
fprintf(fid,edited_odb_file);%write the new odbread file
fclose(fid);%close the new odbread
%the below file should be the final modified one (parametrized)
!abaqus cae script=newodbread.py
%%
% w = waitforbuttonpress;
%%
%abaqus damage image
q=imread('damage-image.PNG');% the name of the saved output image
figure
imshow(q);imagesc(q);
promptcrop = { 'Enter the y axis coordinates of the zone needed [y2]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
CROP1=inputdlg(promptcrop,dlgtitle,dims);
y2=CROP1{1};
Y2=str2num(y2);
promptcrop2 = { 'Enter the y axis coordinates of the zone needed [y1]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
CROP2=inputdlg(promptcrop2,dlgtitle,dims);
y1=CROP2{1};
Y1=str2num(y1);
promptcrop3 = { 'Enter the x axis coordinates of the zone needed [x2]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
CROP3=inputdlg(promptcrop3,dlgtitle,dims);
x2=CROP3{1};
X2=str2num(x2);

```

```

promptcrop4 = { 'Enter the x axis coordinates of the zone needed [x1]'};
dlgtitle='Cropping the Damaged Image';
dims=[1 35];
CROP4=inputdlg(promptcrop4,dlgtitle,dims);
x1=CROP4{1};
X1=str2num(x1);
y=(Y1)-(Y2);
x=(X2)-(X1);
targetSize=[y x];%size of the cropped image
r = centerCropWindow2d(size(q),targetSize);
A = imcrop(q,r);%cropped image
%%crop the image
% targetSize=[y x];%size of the cropped image
% r = centerCropWindow2d(size(q),targetSize);
% A = imcrop(q,r);%cropped image
% imshow(A);
%%
%real damage image
prompt = { 'Enter the name of the real damage image.format'};
dlgtitle='Real Damage Image';
dims=[1 35];
real=inputdlg(prompt,dlgtitle,dims);
Rimage=real{1};
B=imread(Rimage);
%resize both images to the same size
[v,d,f]=size(A);
R=imresize(B,[v,d]);%real damage modified
%change both images to grayscale
Red_gray=im2gray(A);
r_gray=im2gray(R);
img = double (r_gray);
MaxThresh=max(max(img));
thresholdpercentage= img./MaxThresh;
figure
imshow(thresholdpercentage);%%%%%%%%%%here we need to pause%%%%%%%%%%
pause(40);
%%
pan=size(thresholdpercentage,1);
leb=size(thresholdpercentage,2);
minimumthresh = { 'Enter the minimum pixel value to be displayed'};
dlgtitle='MINIMUM PIXEL VALUE TO BE DISPLAYED';
dims=[1 35];
minvalueofthresh=inputdlg(minimumthresh,dlgtitle,dims);
t_low=minvalueofthresh{1};
T_Low=str2num(t_low);
T_High = max(max(thresholdpercentage));

T_res = zeros (pan, leb);

for i = 1 : pan
    for j = 1 : leb
        if (thresholdpercentage(i, j) < T_Low)
            T_res(i, j) = 0;
        elseif (thresholdpercentage(i, j) > T_Low)
            T_res(i, j) = 1;
        end;
    end;
end;

```

```

end;
IMBRIDGE= bwmorph(T_res, 'bridge');
IMCLOSE= bwmorph(IMBRIDGE, 'close');
IMCLEAN = bwmorph(IMCLOSE, 'clean');
DoubleRedGray=double(Red_gray);
MaxRed=max(max(DoubleRedGray));
PercentageRed= (DoubleRedGray./MaxRed);
figure
imshow(PercentageRed);%%%%%%%%%%here we need to pause%%%%%%%%
pause(40);
%%
panRED=size(PercentageRed,1);
lebRED=size(PercentageRed,2);
minimumthreshRED = { 'Enter the minimum pixel value to be displayed'};
dlgtitleRED='MINIMUM PIXEL VALUE TO BE DISPLAYED';
dimsRED=[1 35];
minvalueofthreshRED=inputdlg(minimumthreshRED,dlgtitleRED,dimsRED);
t_lowRED=minvalueofthreshRED{1};
T_LowRED=str2num(t_lowRED);
T_HighRED = max(max(PercentageRed));

T_resRED = zeros (panRED, lebRED);

for i = 1 : panRED
    for j = 1 : lebRED
        if (PercentageRed(i, j) < T_LowRED)
            T_resRED(i, j) = 0;
        elseif (PercentageRed(i, j) > T_HighRED)
            T_resRED(i, j) = 1;
        end;
    end;
end;
IMBRIDGERED= bwmorph(T_resRED, 'bridge');
IMCLOSERED= bwmorph(IMBRIDGERED, 'close');
level= 4;
or= 16;
guardb= 0;
K= 0;

[pyr1, pind] = buildSCFpyr(IMCLEAN, level, or-1);%.....% decompose img1
using a complex steerable pyramid decomposition
[pyr2, pind] = buildSCFpyr(IMCLOSERED, level, or-1);%.....% decompose img2
using a complex steerable pyramid decomposition

winsize = 7;
window = ones(7);%.....% The CW-SSIM
indices are computed locally using a sliding
%
%
7-by-7 window that moves across each wavelet subband.

window = window./sum(sum(window));%.....% normalize the window

gb = guardb/(2^(level-1));%.....% the gb
parameter is used to control how much is discarded from the four image
boundaries.

s = pind((level-1)*or+2, :);

```

```

w = fspecial('gaussian', s-winsize+1, s(1)/4);%.....% The CW-SSIM index
map is combined into a scalar similarity measure using a
%
weighted summation.The weighting function is obtained using a Gaussian
%
profile with a standard deviation equaling a quarter of the image size at
%
finest level of pyramid decomposition.

for i=1:or
    bandind = i+(level-1)*or+1;
    band1 = pyrBand(pyr1, pind, bandind);%.....% Access a subband from
a pyramid (see help pyrBand)
    band2 = pyrBand(pyr2, pind, bandind);%.....% Access a subband from
a pyramid (see help pyrBand)
    band1 = band1(gb+1:end-gb, gb+1:end-gb);
    band2 = band2(gb+1:end-gb, gb+1:end-gb);

    corr = band1.*conj(band2);
    varr = abs(band1).^2 + abs(band2).^2;
    corr_band = filter2(window, corr, 'valid');%.....% The CW-SSIM indices
are computed locally using a sliding 7-by-7
    varr_band = filter2(window, varr, 'valid');%.....% window that moves
across each wavelet subband.

    cssim_map = ...
    (2*abs(corr_band) + K)./(varr_band + K);%.....% The purpose of the
small constant K is mainly to improve the robustness of
%
the CW-SSIM measure when the local signal to noise ratios are low.

    band_cssim(i) = sum(sum(cssim_map.*w));%.....% The CW-SSIM index map is
combined into a scalar similarity measure using a
%
weighted summation.
end

cwssimv = mean(band_cssim);%.....% value is 1 for
two identical images.
error=1-cwssimv;
%last section of the loop
%save the fitness of the individual
end

```

REFERENCES

- Abdel-Qader, I., Abudayyeh, O., & Kelly, M. E. (2003). Analysis of Edge-Detection Techniques for Crack Identification in Bridges. *Journal of Computing in Civil Engineering*, 17(4), 255–263. [https://doi.org/10.1061/\(asce\)0887-3801\(2003\)17:4\(255\)](https://doi.org/10.1061/(asce)0887-3801(2003)17:4(255))
- Abedini, M., & Mutalib, A. A. (2020). Investigation into Damage Criterion and Failure Modes of RC Structures When Subjected to Extreme Dynamic Loads. *Archives of Computational Methods in Engineering*, 27(2), 501–515. <https://doi.org/10.1007/s11831-019-09317-z>
- Aljanabi, M. A., Hussain, Z. M., & Lu, S. F. (2018). An Entropy-Histogram Approach for Image. 2018.
- Alsubaei, F. C. F. (2015). Scholarship @ Western Performance of Protective Perimeter Walls Subjected to Explosions in Reducing the Blast Resultants on Buildings.
- Aouad, C., Chemissany, W., Mazzali, P., Temsah, Y., & Jahami, A. (2020). Beirut explosion: Energy yield from the fireball time evolution in the first 230 milliseconds. 1–11. <http://arxiv.org/abs/2010.13537>
- Appana, V., Guttikonda, T. M., Shree, D., Bano, S., & Kurra, H. (2021). Similarity Score of Two Images using Different Measures. *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, 741–746. <https://doi.org/10.1109/ICICT50816.2021.9358789>
- Appuhamilage, G. (2015). Effects of Blast Loading on Reinforced Concrete Facade Systems.
- Basak, R., Sanyal, A., Nath, S. K., & Goswami, R. (2013). Comparative View of Genetic Algorithm and Pattern Search for Global Optimization. *International*

- Journal of Engineering and Science, 3(4), 9–12.
- Baumgart, C. M. (2014). Scholars ' Mine The effects of advanced structural materials to mitigate explosive and impact threats.
- Bovik, A. C., Clark, M., & Geisler, W. S. (1990). Multichannel Texture Analysis Using Localized Spatial Filters. 12(I).
- Castro, P. B. (2006). No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title. Global Shadows: Africa in the Neoliberal World Order, 44(2), 8–10.
- Chou, J. H., & Ghaboussi, J. (2001). Genetic algorithm in structural damage detection. Computers and Structures, 79(14), 1335–1353. [https://doi.org/10.1016/S0045-7949\(01\)00027-X](https://doi.org/10.1016/S0045-7949(01)00027-X)
- Coupled Eulerian Lagrange (CEL) Analysis with ABAQUS. (n.d.). <https://simplifiedfem.wordpress.com/about/coupled-eulerian-lagrange-cel-analysis-with-abaqus/>
- Dathan, J., & Overton, I. (2019). Explosive Violence Monitor 2018. <https://aoav.org.uk/wp-content/uploads/2019/05/Explosive-Violence-Monitor-2018-v5.pdf>
- Deal, W. E. (1957). Shock hugoniot of air. Journal of Applied Physics, 28(7), 782–784. <https://doi.org/10.1063/1.1722854>
- Defense, D. of. (2008). UFC 3-340-02. Structures to resist the effects of the accidental explosions. In Unified Facilities Criteria (Issue December).
- Dewey, J. M. (2010). Dewey Paper #23.
- Dynamsoft. (2019). Color Space Conversion. In Image Processing 101. <https://www.dynamsoft.com/blog/insights/image-processing/image-processing->

101-color-space-conversion/

El Khansa, H. (2020). A COMPUTATIONALLY EFFICIENT FRAMEWORK FOR OPTIMIZING THE DESIGNS OF REINFORCED CONCRETE.

Emetere, M. E. (2019). Studies in Big Data 54 Environmental Modeling Using Satellite Imaging and Dataset Re-processing. Springer, Cham.

https://doi.org/https://doi.org/10.1007/978-3-030-13405-1_2

Facts, F. (2020). Situation update.

Fu, H. C., Erki, M. A., & Seckin, M. (1991). Review of Effects of Loading Rate on Reinforced Concrete. *Journal of Structural Engineering*, 117(12), 3660–3679.

[https://doi.org/10.1061/\(asce\)0733-9445\(1991\)117:12\(3660\)](https://doi.org/10.1061/(asce)0733-9445(1991)117:12(3660))

Fujita, Y., Mitani, Y., & Hamamoto, Y. (2006). A Method for Crack Detection on a Concrete Structure. 18th International Conference on Pattern Recognition

(ICPR'06). <https://doi.org/10.1109/ICPR.2006.98>

Hasanova, N. (2020). A Comparative Study of Particle Swarm Optimization and Genetic Algorithm. *Qubahan Academic Journal*, 1(1), 33–45.

<https://doi.org/10.48161/qaj.v1n1a7>

Hegadi, R., & Basavaprasad, B. (2014). a Study on the Importance of Image Processing and Its Applications. *International Journal of Research in Engineering and*

Technology, 03(15), 155–160. <https://doi.org/10.15623/ijret.2014.0315029>

Huri, M. A. M., Ahmad, U. K., Ibrahim, R., & Omar, M. (2017). Satu ulasan pengesanan residu letupan dari sudut pandangan kimia forensik. *Malaysian Journal*

of Analytical Sciences, 21(2), 267–282. [https://doi.org/10.17576/mjas-2017-2102-](https://doi.org/10.17576/mjas-2017-2102-01)

01

Jahami, A. (2016). *Maters thesis: Characteristics of Shock waves and its effect on*

structural elements.

Joel, D., David, A., & Gardner, J. D. (2005). Comparative Study of Finite Element Simulation Software.

Kerley, G. I. (2013). The Linear US-uP Relation in Shock-Wave Physics. March.
<http://arxiv.org/abs/1306.6916>

Masi, F. (2017). Blast actions from high explosives. Studies on their simulation and effects. <https://doi.org/10.31224/osf.io/ekus8>

MathWorks. (2021). MATLAB Documentation.
<https://www.mathworks.com/help/matlab/>

Miyamoto, A., Konno, M. A., & Bruhwiler, E. (2007). Automatic crack recognition system for concrete structures using image processing approach. *Asian Journal of Information Technology*.

Mlakar, Sr., P. F., Corley, W. G., Sozen, M. A., & Thornton, C. H. (1998). The Oklahoma City Bombing: Analysis of Blast Damage to the Murrah Building. *Journal of Performance of Constructed Facilities*, 12(3), 113–119.
[https://doi.org/10.1061/\(asce\)0887-3828\(1998\)12:3\(113\)](https://doi.org/10.1061/(asce)0887-3828(1998)12:3(113))

Ne, M. (1998). A Visual Model for Blast Waves and Fracture.

Ohashi, K., Kleine, H., & Takayama, K. (n.d.). Characteristics of blast waves generated by milligram charges. 1–7.

Rao, B., Chen, L., Fang, Q., Hong, J., Liu, Z. xian, & Xiang, H. bo. (2018). Dynamic responses of reinforced concrete beams under double-end-initiated close-in explosion. *Defence Technology*, 14(5), 527–539.
<https://doi.org/10.1016/j.dt.2018.07.024>

Rigby, S. E., Lodge, T. J., Alotaibi, S., Barr, A. D., Clarke, S. D., Langdon, G. S., &

- Tyas, A. (2020). Preliminary yield estimation of the 2020 Beirut explosion using video footage from social media. *Shock Waves*, 30(6), 671–675.
<https://doi.org/10.1007/s00193-020-00970-z>
- Rivera, J. P., Josipovic, G., Lejeune, E., Luna, B. N., & Whittaker, A. S. (2015). Automated detection and measurement of cracks in reinforced concrete components. *ACI Structural Journal*, 112(3), 397–405.
<https://doi.org/10.14359/51687424>
- Sampat, M. P., Wang, Z., Gupta, S., Bovik, A. C., & Markey, M. K. (2009). Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18(11), 2385–2401. <https://doi.org/10.1109/TIP.2009.2025923>
- Schwartz, M. (n.d.). Lecture 9 : Reflection , Transmission and Impedance 1 Boundary conditions at a junction. 5, 1–10.
- Schwer, L. E. (2009). Strain Rate Induced Strength Enhancement In Concrete: Much Ado About Nothing? 7th European LS-DYNA Conference, 2, 203–211.
- Shirbhate, P. A., & Goel, M. D. (2020). A Critical Review of Blast Wave Parameters and Approaches for Blast Load Mitigation. May. <https://doi.org/10.1007/s11831-020-09436-y>
- SIMULIA. (2013). Abaqus/CAE User’s Guide.
<http://130.149.89.49:2080/v6.13/books/usi/default.htm>
- Smith, P. O., & Hetherington, J. G. (1994). Blast and Ballistic Loading of Structures.
- Sozen, M. A., Thornton, C. H., Corley, W. G., & Sr., P. F. M. (1998). The Oklahoma City Bombing: Structure and Mechanisms of the Murrah Building. *Journal of Performance of Constructed Facilities*, 12(3), 120–136.
[https://doi.org/10.1061/\(asce\)0887-3828\(1998\)12:3\(120\)](https://doi.org/10.1061/(asce)0887-3828(1998)12:3(120))

- Taylor, A. (2021). CS101-Image-1 Introduction to Digital Images.
<https://web.stanford.edu/class/cs101/image-1-introduction.html>
- Temsah, Y., Jahami, A., Al Timani, B., & Aouad, C. (2020). Beirut Explosion Structural Assessment of the Explosion Magnitude. 1–14.
- Vannucci, P., Masi, F., & Stefanou, I. (2017). A study on the simulation of blast actions on a monument structure.
- Wang, F.-S., & Chen, L.-H. (2013). Heuristic Optimization. In *Encyclopedia of Systems Biology*.
- Wang, W., Zhang, D., Lu, F., Wang, S. chuan, & Tang, F. (2013). Experimental study and numerical simulation of the damage mode of a square reinforced concrete slab under close-in explosion. *Engineering Failure Analysis*, 27, 41–51.
<https://doi.org/10.1016/j.engfailanal.2012.07.010>
- Wang, Z., Bovik, A. C., Sheikh, H. R., Member, S., Simoncelli, E. P., & Member, S. (2004). Image Quality Assessment : From Error Visibility to Structural Similarity. 13(4), 600–612.
- Zhang, C., Gholipour, G., & Mousavi, A. A. (2020). Blast loads induced responses of RC structural members : State-of-the-art review. *Composites Part B*, 195(April), 108066. <https://doi.org/10.1016/j.compositesb.2020.108066>
- Zhou, Y. (2006). Study on genetic algorithm improvement and application. Worcester Polytechnic Institute, 2(May), 1–73.