

COMPUTATIONAL MECHANICS FOR SHAPE OPTIMIZATION AND
AIRPLANE BOMBING BASED ON OPENFOAM AND LS-DYNA SOFTWARE

A Dissertation

by

JEAN YEHL

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Chair of Committee,	Goong Chen
Committee Members,	Prabir Daripa
	Stephen A. Fulling
	Siu Ah Chin
Head of Department,	Emil J. Straube

December 2018

Major Subject: Mathematics

Copyright 2018 Jean Yeh

ABSTRACT

In this dissertation, two problems are studied by computational mechanics. The first problem is the development and application of open-source software OpenFOAM to solve a coupled system from shape optimization on a time-dependent changing domain to minimize a cost functional. Previously, researcher have treated this problem in 2D space, but here the author will extend the computation to 3D space. She will use the finite volume methods and OpenFOAM to build time-varying domains for optimization.

The second problem is about computational forensics. The author will discuss some photographic evidence, introduce mathematical modeling and do computational modeling regarding this particular Daallo Airlines bombing case, in which only a small amount of explosives was used. Mathematical and computer modeling of viscoplasticity, fracture and explosion by LS-DYNA will be used to study this problem and we computed on the supercomputer at Texas A&M University's High Performance Research Computing Center. All of the numerical results of airplane bombing can be seen in the many video animations we made from the post-processed supercomputer results. The associated physical phenomena will also be interpreted.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my deepest gratitude to Dr. Goong Chen. It has been an honor to be his Ph.D. student. He has always been patiently training me and making me a better researcher. I really appreciate his valuable suggestions, expert guidance and continuous inspiration. He not only showed me the ways of scientific research, but also encouraged me through helpful discussions. I also appreciate the financial support of research assistantships from the Math Department at Texas A&M University.

I would also like to thank all the people in my research group. I enjoyed having discussions with you all. I especially thank Dr. Cong Gu for helping me learn LS-DYNA. In addition, I want to thank the High- Performance Computing Center and their help at Texas A&M University. I thank them for allowing me access of all the software and providing me with invaluable technical support.

My special thanks goes to my dearest friends for all the life experiences we have shared together. Last but not least, I would like to thank my parents, my sibling, my cousins, my aunt and my uncle for their emotional support.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Goong Chen, Prabir Daripa and Stephen A. Fulling of the Department of Mathematics and Professor Siu Ah Chin of the Department of Physics.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University and a dissertation research fellowship has no outside funding was received for the research and compilation of this document.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGMENTS	iii
CONTRIBUTORS AND FUNDING SOURCES	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES.....	ix
1. INTRODUCTION.....	1
2. NUMERICAL MODELING AND COMPUTATION OF 3D SHAPE OPTIMIZATION PROBLEMS BY OPENFOAM.....	3
2.1 Introduction.....	3
2.2 The Computational Method	6
2.2.1 Software OpenFOAM	6
2.2.2 Modeling.....	7
2.3 Example in 2D	11
2.4 Example in 3D	17
2.5 Conclusions.....	20
3. TOWARD COMPUTATIONAL FORENSICS FOR THE LAPTOP BOMBING TER- RORISM CASE OF DAALLO AIRLINES FLIGHT 159, FEBRUARY 2, 2016.....	21
3.1 Introduction.....	21
3.1.1 Airplane bombing experiments by MythBusters.....	24
3.2 Computer modeling for the airplane, explosion and validation	26
3.2.1 Software LS-DYNA	26
3.2.2 Mechanical models and simplification	28
3.2.2.1 Whole Plane Simulation	30
3.2.2.2 Aircraft Windows.....	32
3.2.3 Modeling the explosive.....	32
3.2.4 Cabin Pressurization.....	34
3.2.5 Cracks and fracture criteria	36
3.2.6 Validation of SPH Blast Simulation	37
3.3 Computer simulation of the Daallo Airline Flight 159 bombing	43

3.4 Conclusions.....	50
4. SUMMARY	52
REFERENCES	53
APPENDIX A. PART OF THE K-FILE.....	56

LIST OF FIGURES

FIGURE	Page
2.1	The graphics of Z_d , the desired target shape. 12
2.2	The mesh where the number of the boundary points selected is 64. 12
2.3	The graph of \hat{u}_h^t as a function of θ , at time $t=0.84$, where the number of the boundary points selected is 64, 128 and 256. 13
2.4	The graphics of p_h^t on $\Omega^{\hat{t}_h, h}$ where the number of the boundary points selected is 128. 14
2.5	The graphics of w_h^t on $\Omega^{\hat{t}_h, h}$ where the number of the boundary points selected is 128. 15
2.6	The graph of J_h^t as a function of t , where the number of the boundary points selected is 128. 16
2.7	The graph of J_h^t as a function of time t , where the number of the boundary points selected is 64, 128 and 256. 16
2.8	The mesh when a $13*13*13$ cube is projected to a sphere. The right one is the cross-sectional surface. 17
2.9	The graph of J_h^t as a function of time t , where z_d in (2.13) is used in (2.3) 18
2.10	The left graphics shows $p_h^{\hat{t}_h}$ on $\Omega^{\hat{t}_h}$. The right graphics is the semi sphere with $(1, 1, 0)$ as normal vector on the cross-sectional surface. 18
2.11	The graphics shows $w_h^{\hat{t}_h}$ on $\Omega^{\hat{t}_h}$ 19
2.12	The graph of \hat{u}_h^t as a function of θ on with the intersection of $x+y=0$ and the sphere, at time $t=0.47$ 19
3.1	Flight origin: MGQ (Aden Adde International Airport). Destination: JIB (DjiboutiAmbouli International Airport). The location where a burnt body of the terrorist was found: Balad. 22
3.2	The photos of DAO159 taken from the Internet [1, 2, 3]. 23
3.3	The result by a special explosive charge on the circumference of the window by a MythBusters' experiment. For video, see https://youtu.be/Fi1_117M8FA?t=2m30s .. 25

3.4	Explosion by a large explosive charge by a MythBusters' experiment. For video, see https://youtu.be/Fi1_1I7M8FA?t=4m17s	26
3.5	Mesh refinements.....	31
3.6	The left figure shows the set-up of the inner and outer air particles using SPH. The right figure shows the air particles have been disturbed without flow after the fuselage has been breached.	35
3.7	The arrows indicate how the LOAD works on the surface.	36
3.8	Explosive set-up geometry.	38
3.9	Snapshots of simulation for the Shirey's explosion test.	39
3.10	Threshold causes for 6 mm steel plate with different failure strains.....	41
3.11	Relationship between charge weight and hole diameter for reinforced concrete explosive test.	42
3.12	Relationship between explosive charge weight and hole diameter in aircraft explosion.....	42
3.13	Panel of snapshot of explosion. (With $8 \times 0.6 \times 8 \text{ cm}^3$ size of TNT explosive) The unit of time is millisecond. See the video animation in http://goo.gl/aUwqJx	44
3.14	Comparison between the real airplane damage photo and the simulation result (With $24 \times 1.5 \times 24 \text{ cm}^3$ size of TNT explosive).....	45
3.15	A real fuselage and design.	46
3.16	The top two panels are the cabin fuselage with only two ribs and the bottom two are the one fully supported by more ribs. The left two are the ribs setting, and the right two are the results of damage after 14 ms.	47
3.17	Snapshots of explosion showing damage to the aircraft body, whose airframe is fully supported by ribs. For video animation, see http://goo.gl/3NmV2u	48
3.18	The left panel corresponds to is the case without ribs, and the right top one is the case with ribs. The right bottom one is the real photo.....	49
3.19	The result of damage after mesh refinement.	50

LIST OF TABLES

TABLE	Page
3.1 Parameters for the aircraft.....	30
3.2 Parameters for acrylic windows.	32
3.3 Parameters for TNT in JWL equation.	34
3.4 Parameters for high explosive (C4).....	40
3.5 Parameters for steel plate.	40

1. INTRODUCTION

This dissertation is built on the following two main topics:

(P1) 3D shape optimization by a finite volume method using OpenFOAM as the platform;

(P2) computation of a laptop bombing case by the finite element method based on the mechanics of solid, fracture and explosion, using LS-DYNA software.

These two themes look, at first, to be disjoint and dissimilar, as they deal with seemingly totally different topics. However, there is a unifying theme, namely, the application and development of the state-of-the-art computational methodologies and software to solve problems of contemporary interest in applications.

Both problems (P1) and (P2) are challenging, at least as far as actual computations are concerned, because we are dealing with 3D geometries. Regarding (P1), nearly *all* computational studies published earlier dealt with 2D geometry. The reason is quite simple - 3D geometries are far more complex in general, requiring a significant amount of computer code development even for an experienced programmer. But, with the advent of OpenFOAM, available less than a decade ago using the finite volume method, the treatment of 3D geometries has been greatly simplified. Better still, the developed codes in OpenFOAM are open-source and, thus, can be easily modified for our own purposes. Those existing codes are written in object-oriented C++ language enabling the researcher to easily understand and adapt. Such a luxury was not available at all a few years ago, and even now, we still have not seen much (or, possibly, any) shape-optimization computation work using the finite volume method and OpenFOAM. In this regard, the author may likely be entitled the claim of priority for the advance in this direction by her work here.

For (P1), the author will first introduce the background and motivation to solve a coupled systems for the optimization problem. Next, in Section 2.2 we will introduce the software OpenFOAM and also the way to use OpenFOAM for the computational modeling when the domains are time-varying. Section 2.3 deals with the 2D cases, but the author extends the study to 3D in Section 2.4.

Next, with regard to (P2), our work here is probably also one of the first to study the mathematical modeling and supercomputing of an airplane bombing case, which would be of strong interest to homeland security. Bombing is perhaps the most favorite tactics by terrorists, and it can cause a large number of casualties up to several hundreds. Everywhere in the world, terrorists have not given up their attempts to blow up airplanes; the most recent incident happened in 2016. Therefore, it is necessary to elucidate what would happen when an aircraft is bombed. However, a real physical experiment of aircraft bombing would be very expensive. On the other hand, the computer-based high-fidelity simulation is much cheaper and easier to achieve the result. It is hard to imagine that computational mechanics can do such a wonderful job on a real physical phenomenon simulation. Numerical simulation of explosion involves high complexities that have to be captured in order to successfully predict their behavior. These complex phenomena include aerodynamics, solid and fracture mechanics, detonation process, crack and fracture (erosion).

In (P2), the authors will first present examples of several past airplane bombing cases. We explore what may happen when a large amount of explosives are detonated inside a cabin; some photographic evidence will also be discussed. The authors will then introduce the modeling and computational forensics Regarding this particular Daallo Airlines bombing case, in which only a small Amount of explosives were used. Mathematical and computer modeling of viscoplasticity, fracture and explosion by LS-DYNA will be used to study this problem.

In summary, the above work done by the author represents new advances in computational mathematics and mechanics. The work of shape optimization (i.e., (P1)) is useful in selecting the best shape that minimizes a given cost, and the simulation of Daallo Airline laptop bombing will be helpful to study of homeland security. The author is excited and gratified that she has learned these state-of-the art methodologies and software (such as OpenFOAM and LS-DYNA) for treating problems of contemporary interest.

2. NUMERICAL MODELING AND COMPUTATION OF 3D SHAPE OPTIMIZATION PROBLEMS BY OPENFOAM

2.1 Introduction

Shape optimization is a problem belong to the optimal control theory field and it is an important problem in engineering design and optimization. For example, ([4]) has helped the (real-world) problem of the design of a Canadian satellite. This work [4] finds the optimal shape of a minimum-weight *thermal diffuser* with a priori specifications on the inward thermal power flux (TPF) and a bound on the outward TPF. There are, naturally, many other applications. Structural engineer might want to know what is the shape of an airplane wing has minimal drag when a minimal wing strength is required. Generally speaking, in shape optimization, the domain needs to be optimized according to certain distributed or boundary control forces and design specifications. The classical problem is to find the best shape which is optimal and which can simultaneously minimize the cost functional J while satisfying given constraints. For the airplane wing problem, the cost functional is the wind drag and the given constraint is the wing strength. The most important part of the optimization problem is to determine the shape gradient. When computing the gradient, it is common to use the finite element method for implementation.

There are three ways to treat of shape optimization. The first is a Lagrangian approach. It considers a sufficiently large number of point to get accurate outline of the shape then moving the boundary points to evolve the shape. The second is an Eulerian approach. It defines a function to be zero on the boundary of the shape, negative outside of the shape and positive inside of the region. Then it considers a rectangular grid on the box around the shape and value the functional at the grid points. When the shape is evolving, only the functional will change, but not the grid points. The last one uses a flow function. It considers that any point belonging to the shape can be traced back to a point of the original shape in a one-to-one mapping. That is, a diffeomorphism f_t exist. $f_t : \Omega_0 \rightarrow \Omega_t, t \geq 0$.

The finite element method is perhaps the most standard way to treat non-rectangular geometry in multiple dimensions. In Sokolowski et al. [5], chapter 7 the important Hadamard formula shows that the Eulerian upper semiderivative of the shape functional J is a boundary integral. The boundary element method (BEM) utilizes this and reduces all calculation on the boundary. BEM avoids internal discretization, which provides more efficiency and accuracy.

This chapter follows Chen et al's [6] and extends their examples into 3D cases via an open source software OpenFOAM [7].

The shape optimization problem requires problem-solving on a changing domains Ω^t according to the flow

$$\begin{cases} \frac{d}{dt}x(X, t) = V(x(X, t), t), & x(X, t) \in \Omega^t \subset \mathbf{R}^N, t \geq 0, \\ x(X, 0) = X \in \Omega^0, \end{cases} \quad (2.1)$$

where the $V(x, t)$ is the velocity (vector) field. The time-varying Ω^t can also be defined as

$$\Omega^t = \{x(X, t) | \forall X \in \Omega^0\}$$

Let a target function z_d be defined in \mathbf{R}^N , and

$$z_d \in H_{loc}^{r+\frac{1}{2}}(\mathbf{R}^N), r > 0. \quad (2.2)$$

Therefore, $z_d \in H_{loc}^r(M)$ for any (N-1) dimensional manifold M in \mathbf{R}^N . Let f be a given distribution function defined on \mathbf{R}^N . Now consider the linear-quadratic-regulator (LQR) problem on the moving domains Ω^t :

$$(LQR) \begin{cases} \inf_{t \geq 0} \inf_u J(\Omega^t, u) \equiv \int_{\partial\Omega^t} [|w(x) - z_d(x)|^2 + \gamma u^2(x)] d\sigma, \\ \text{subject to} \\ \Delta w(x) = f, \quad x \in \Omega^t, \\ \frac{\partial w}{\partial n} = u \in L^2(\partial\Omega^t). \end{cases} \quad (2.3)$$

On a fixed Ω^t with sufficiently smooth boundary $\partial\Omega^t$ and sufficient by regular f , the LQR problem has a unique optimal control $\hat{u}^t \in L^2(\partial\Omega^t)$ (by Chen et al [6]). Define

$$J^t(\Omega^t) \equiv J(\Omega^t, \hat{u}^t).$$

For problem (2.3) with a fixed $t \geq 0$, the solution w will satisfy the following compatibility condition:

$$\int_{\partial\Omega^t} w d\sigma = \int_{\Omega^t} f(x) dx$$

Problem (2.3) has infinitely many solutions $w(x)$ as shown below if $w_p(x)$ is a particular solution of problem (2.3).

$$w(x) = w_p(x) + c, \quad \forall c \in \mathbf{R}.$$

Therefore, the cost function J in problem (2.3) as defined by infinitely many $w(x)$, it has the effect of determining the free constant c in a unique way.

We first study the problem (LQR) on the fixed domain $\Omega^t \equiv \Omega$. By introducing an adjoint state p and using calculus of variations (by [8]), we obtain the following optimality system (OPT) characterizing the unique optimal control \hat{u} :

$$(OPT) \left\{ \begin{array}{l} \text{primal system } \left\{ \begin{array}{ll} \Delta w = f & \text{on } \Omega, \\ \frac{\partial w}{\partial n} = \hat{u} & \text{on } \partial\Omega; \end{array} \right\} \\ \text{adjoint system } \left\{ \begin{array}{ll} \Delta p = 0 & \text{on } \Omega, \\ \frac{\partial p}{\partial n} = w - z_d & \text{on } \partial\Omega; \end{array} \right\} \\ \text{optimality condition } p + \gamma \hat{u} = 0 & \text{on } \partial\Omega \end{array} \right. \quad (2.4)$$

In the rest of the chapter, we will simply discuss the case where $\hat{u} = -\frac{1}{\gamma}p(x)$. Therefore, this chapter is handles with the coupled system p, w as follows:

$$\begin{cases} \Delta p(x) = 0 & \text{on } \Omega^t, \\ \frac{\partial p(x)}{\partial n} = w(x) - z_d(x), & x \in \partial\Omega^t; \end{cases} \quad (2.5)$$

and

$$\begin{cases} \Delta w(x) = f(x) & \text{on } \Omega^t, \\ \frac{\partial w(x)}{\partial n} = -\frac{1}{\gamma}p(x), & x \in \partial\Omega^t; \end{cases} \quad (2.6)$$

Then we have

$$J^t(\Omega^t) = J(\Omega^t, \hat{u}^t) = \int_{\partial\Omega^t} \left[\frac{\partial p(x)^2}{\partial n} + \frac{1}{\gamma}p^2(x) \right] d\sigma$$

The main purpose of this chapter is to calculate the shape gradient δJ^t , in order to determine when the shape functional J will be minimized and what the value of J is. We will redo the two examples (from Chen [6]) in OpenFOAM so we can visualize them in an easier way to validate their examples.

2.2 The Computational Method

2.2.1 Software OpenFOAM

OpenFOAM (Open source Field Operation and Manipulation) [7] is a free, open-source software for solving continuum mechanics problems, including computational fluid dynamics (CFD). The name, OpenFOAM, was registered by OpenCFD Ltd. in 2007. It is based on the finite volume method (FVM) and it develops a syntactical model of equation mimicking by using C++ and object-oriented programming. Thus, people can easily create their own customized numerical solvers with OpenFOAM. One feature of OpenFOAM is its syntax. The syntax for tensor operations and partial differential equations are close to the mathematical representation of the equation. One can easily translate between the syntax and the equation.

There are many number of solvers that have been developed in OpenFOAM, including basic CFD solvers, DNS and LES, buoyancy-driven flow solvers, multiphase flow solvers, solvers for conjugate heat transfer, molecular dynamics solvers, particle-tracking solvers, electromagnetics solvers, direct simulation Monte Carlo solvers, solvers for combustion problems, compressible

flow solvers with RANS and LES capabilities, incompressible flow solvers with RANS and LES capabilities and solid dynamics solvers.

One can use OpenFOAM to generate their mesh and then pick a proper solver and utilities (for example, the setup for boundary conditions, parallel processing, pre-processing, post-processing, etc.)

There's many research done by OpenFOAM. It can works on some real world problems like wind flows simulation, [9] to test the interaction between wind turbine in order to get the best way position turbines. [10] studying cavitation nuisance for hydrodynamic machinery by using Large Eddy Simulation (LES) and finite rate mass transfer modeling together with a mixture assumption. It simulate the presence of several cavitation mechanisms. The OpenFOAM can also solve some pure PDE problems, like want I will show in this chapter. For more cases, [11, 12] using OpenFOAM to get the numerical solution on the Kirchhoff type problem.

2.2.2 Modeling

OpenFOAM allows users to change and define their own programs. The version that we use to solve the coupled system is OpenFOAM - 2.3.0. To solve PDE problems, we first need to define and mesh a domain.

When meshing with OpenFOAM, we use **BlockMesh** utility. The first step is to describe the whole domain with grids and to separate the gridded domain into many hexahedrons or wedge shaped blocks. For the shape optimization problems, this is not an easy step. The next step is to use all the vertices from blocks describing the boundary face, which is needed for setting-up boundary condition. Notice that user needs to define the coordinate of all the vertices and describe the function for the edge to connect between vertices. Since there are many types of domains that I have tried for shape optimization problems, I have written several customized edge utilities to satisfy the requirement.

I noted that there are three ways to track the shape and the last one is the reason we can track our mesh with OpenFOAM. I not only need to mesh a domain, but also have to find a way to make the domain change corresponding to the give changing function. Since the domain is not fixed, the

dynamicMeshDict file in *constant* directory is needed. In order to let the domain changing as the way we like. I write boundary editing tool for different changing domains. For example, the first 2D case has the following codes describing the motion.

```

void libMyChangePatchVectorField::updateCoeffs ()
{
    if (this ->updated ()) { return; }
    const polyMesh& mesh = this ->dimensionedInternalField ().mesh ();
    const Time& t = mesh.time ();
    const pointPatch& p_ = this ->patch ();
    vectorField pNew_=p_.localPoints ();
    vectorField p0Rel = p0_ - origin_;
    vector p0rot;
    vector p0New;
    vectorField sd(p0Rel,0);
    forAll(p0_, iter )
    {
        p0rot = p0Rel[ iter ];
        p0New = pNew_[ iter ];
        sd[ iter ] = vector( cosh(t.value ())* p0rot [0]/ cosh(0.5) - p0rot [0],
            sinh(t.value ())* p0rot [1]/ sinh(0.5) - p0rot [1], 0);
    };
    vectorField::operator=(sd);
    fixedValuePointPatchField <vector >::updateCoeffs ();
}

void libMyChangePatchVectorField::write (Ostream& os) const
{
    pointPatchField <vector >::write (os);
    os.writeKeyword("origin")<< origin_ << token::END_STATEMENT << nl;
    p0_.writeEntry ("p0", os);
}

```

```

    writeEntry("value", os);
}

```

Next, it is also important to write a solver to solve the coupled system. We set up the given function f as fun and use OpenFOAM - 2.3.0 to solve the system by:

```

    solve(fvm::laplacian(P));
    solve(fvm::laplacian(W)-fun);

```

We also set up the equation for boundary condition in the *system/codeDict* directory with the coded streams: codedP, and codedW.

```

codedP
{
    code
    #{
        volScalarField valW=this->db().lookupObject<volScalarField>("W");
        label patchI=patch().index();
        scalarField pW(valW.boundaryField()[patchI]);
        scalarField one(patch().Cf().size(),1.0);
        scalarField Zd_ = Given Function;
        scalarField gradient_ = pW* one-Zd_;
        operator==(this->patchInternalField()
                +gradient_/this->patch().deltaCoeffs());
    #};
}

```

```

codedW

```

```

{
code
#{
    volScalarField valP=this ->db().lookupObject<volScalarField>("P");
    scalarField one(patch().Cf().size(),1.0);
    label patchI=patch().index();
    scalarField pP(valP.boundaryField()[patchI]);
    float coef_=1;
    scalarField gradient_ =(-1)/coef_*pP*one;
    operator==( this ->patchInternalField()
                +gradient_/ this ->patch().deltaCoeffs());
#};
}

```

It is also important to setup the numerical method with each operator and tolerance in *fvSolution* and *fvSchemes* file in *system* directory.

After solving the coupled system of PDE, we use the calculus of the variations to find and to record the values of cost functional J and δJ . By doing so, we are able to find the minimal value of J when δJ changes its sign. After we know WHEN the minimal value of J occur, we can back to find the shape that we need. The found domain which is optimized with the minimal cost value and satisfying given constraints.

2.3 Example in 2D

Example 1. This example defines the domain Ω^t according to the flow

$$\left\{ \begin{array}{l} \frac{d}{dt}x_1(X, t) = x_1 \tanh t, \\ \frac{d}{dt}x_2(X, t) = x_2 \coth t, \end{array} \right\} \quad t \geq \frac{1}{2} \quad (2.7)$$

$$\left\{ \begin{array}{l} \left[\begin{array}{c} x_1(X, \frac{1}{2}) \\ x_2(X, \frac{1}{2}) \end{array} \right] = \left[\begin{array}{c} X_1 \\ X_2 \end{array} \right] = \left[\begin{array}{c} r_1 \cos(\theta) \\ r_2 \sin(\theta) \end{array} \right] = X, r_2 = \sqrt{r_1^2 - 1}, \\ 1 \leq r_1 < \cosh \frac{1}{2}, 0 \leq \theta < 2\pi \end{array} \right.$$

Therefore, the domain Ω^t is formed as

$$\left\{ \begin{array}{l} x_1(X, t) = r_1 \cosh t \cos \theta / \cosh \frac{1}{2}, \\ x_2(X, t) = r_2 \sinh t \sin \theta / \sinh \frac{1}{2}, \end{array} \right\} \quad 0 \leq \theta < 2\pi, t \geq \frac{1}{2} \quad (2.8)$$

Hence,

$$\partial\Omega^t = \{(x_1, x_2) | x_1 = \cosh t \cos \theta, x_2 = \sinh t \sin \theta\}$$

For the cost functional J as in (LQR), we choose the target function as follows:

$$\gamma = 1, f = 0, \quad (2.9)$$

$$z_d(x) = e^{-(1-2|x|)^2} - \frac{1}{2} + \frac{1}{|x|^2} - x_1x_2, \quad x = (x_1, x_2) \in \mathbf{R}^2 \quad (2.10)$$

The profile of the z_d function can be found in Figure 2.1. Since the $\partial\Omega^t$ avoids the center $(0, 0)$ when $t \geq \frac{1}{5}$, it causes no effect even though Z_d has a singularity at $x = (0, 0)$. Figure 2.2 is present the mesh when I select 64 points on the boundary to build the mesh.

In order to check there's only a little difference between the mesh refinement Figure 2.3 present the results of \hat{u}_h^t where the number of the boundary points selected is 64, 128 and 256.

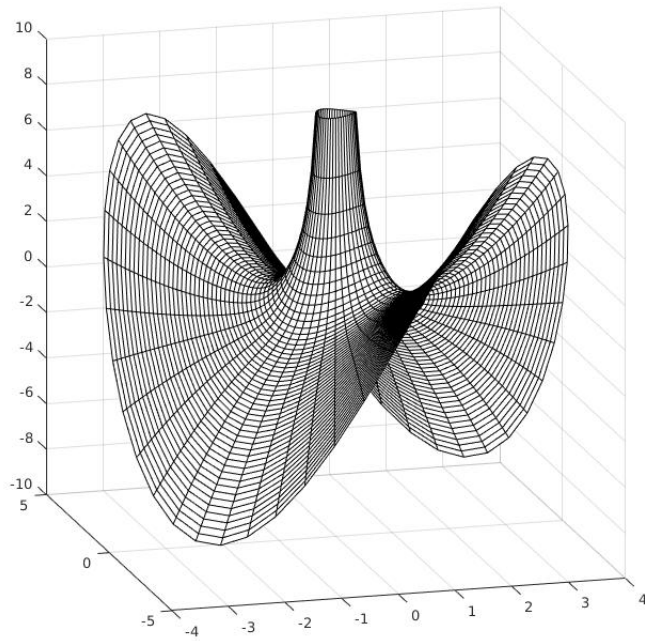


Figure 2.1: The graphics of Z_d , the desired target shape.

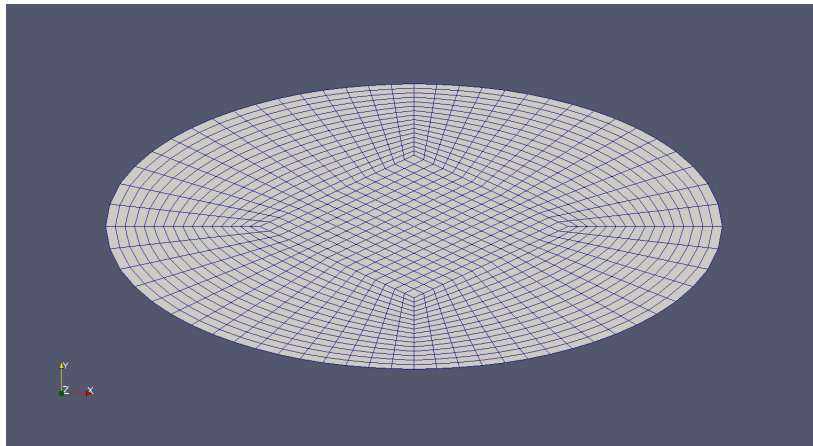


Figure 2.2: The mesh where the number of the boundary points selected is 64.

Figure 2.4 2.5 show the observation for p_h^t and w_h^t when there' 128 points selected on the boundary at time $t = 0.84$.

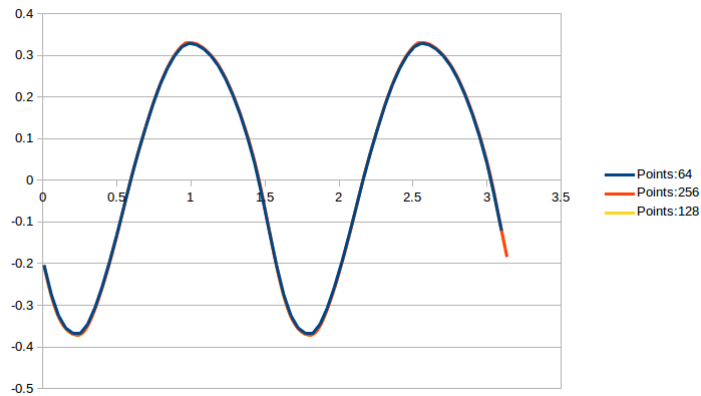


Figure 2.3: The graph of \hat{u}_h^t as a function of θ , at time $t=0.84$, where the number of the boundary points selected is 64, 128 and 256.

By checking the record list of δJ (see Figure 2.6), got the minimal value occurred between $t=0.84283$ and $t=0.84284$. Therefore, 1.86216 is the approximation of J when $n_h = 128$. Therefore, the required domain is occurred between $t=0.84283$ and $t=0.84284$. The Figure 2.7 indicates the result of J is consistent with different level of mesh refinement.

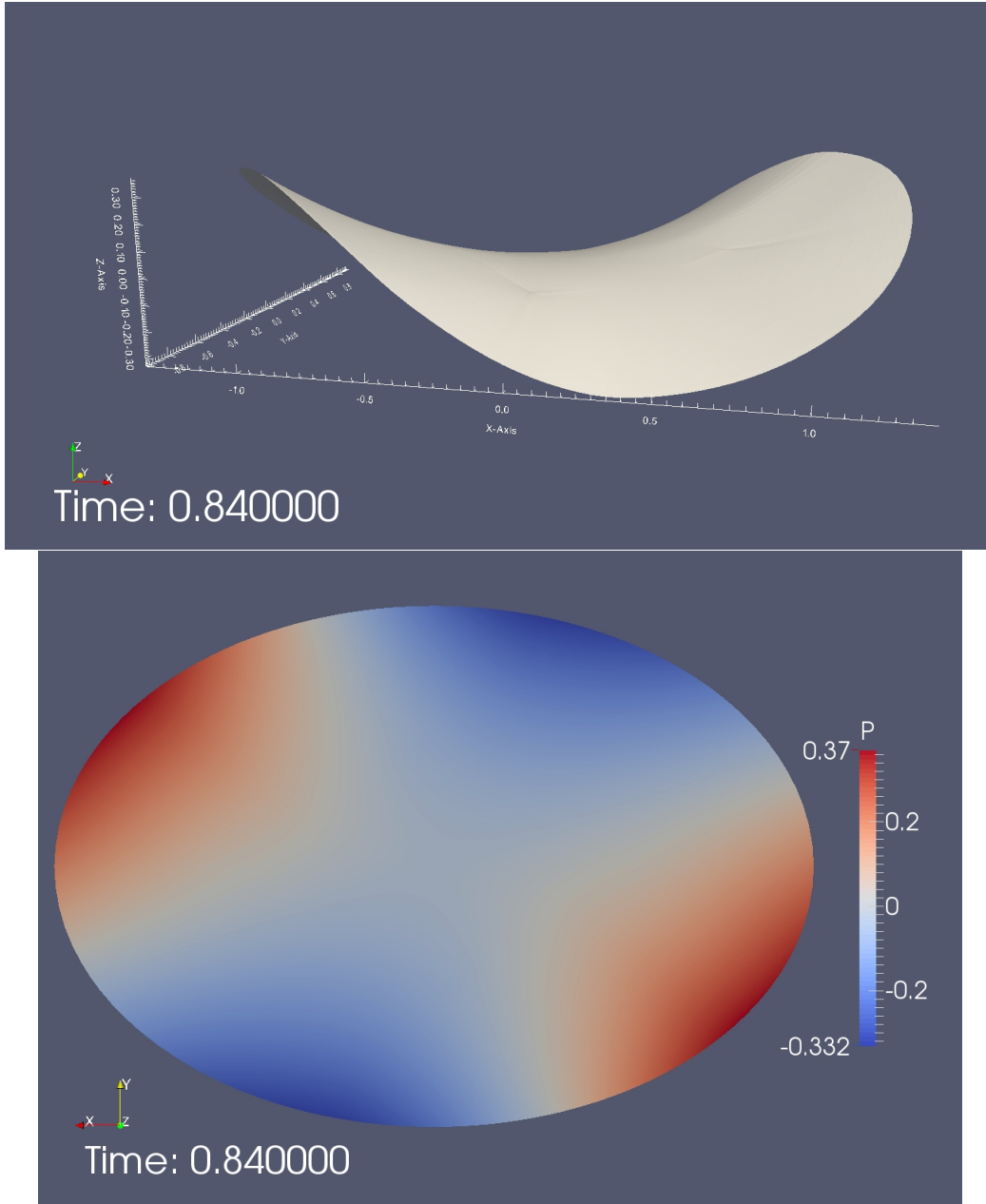


Figure 2.4: The graphics of p_h^t on $\hat{\Omega}_h^{t,h}$ where the number of the boundary points selected is 128.

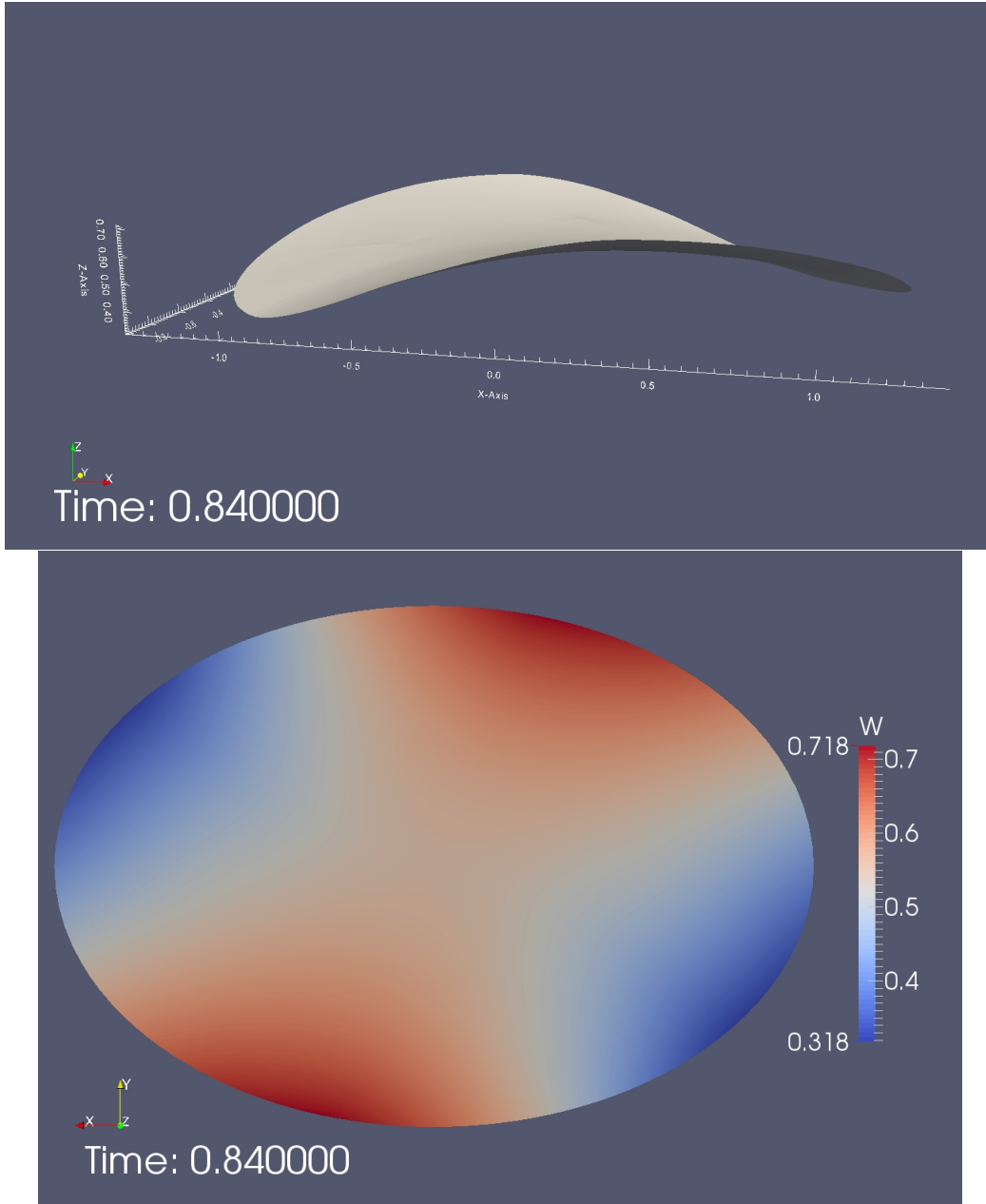


Figure 2.5: The graphics of w_h^t on $\hat{\Omega}^{\hat{t}_h, h}$ where the number of the boundary points selected is 128.

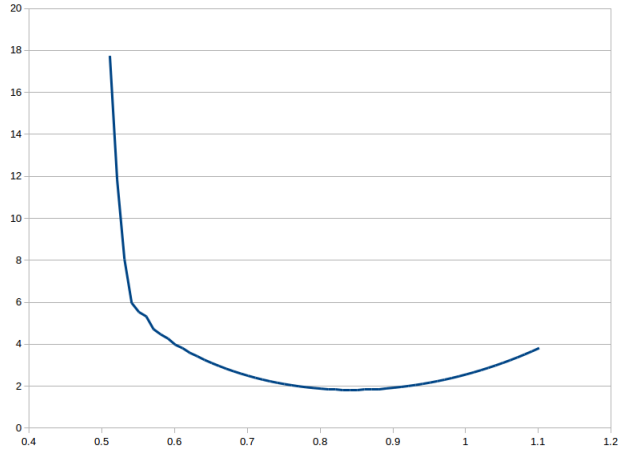


Figure 2.6: The graph of J_h^t as a function of t , where the number of the boundary points selected is 128.

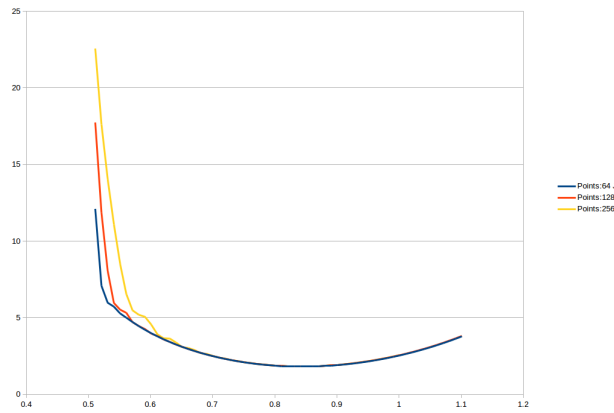


Figure 2.7: The graph of J_h^t as a function of time t , where the number of the boundary points selected is 64, 128 and 256.

2.4 Example in 3D

Example 2. In this example, the domain Ω^t changes as a growing sphere. The domain Ω^t is formed

as

$$\left\{ \begin{array}{l} x_1(X, t) = r * t * \sin \theta * \cos \phi, \\ x_2(X, t) = r * t * \sin \theta * \sin \phi, \\ x_3(X, t) = r * t * \cos \theta \end{array} \right\} \quad 0 \leq \theta < \pi, 0 \leq \phi < 2\pi, t \geq \frac{1}{2} \quad (2.11)$$

Thus, we know that the Ω^t is a sphere centered at origin with radius rt . For the cost functional J as in (LQR), we set the target function in (2.13) as the following

$$\gamma = 1, f = 0, \quad (2.12)$$

$$z_d(x) = e^{-(1-2|x|)^2} - \frac{1}{2} + \frac{1}{|x|^2} - x_1x_2x_3, \quad x = (x_1, x_2, x_3) \in \mathbf{R}^3 \quad (2.13)$$

For the 3D case, we build mesh with the package `blockMeshBodyFit` [13] in `OpenFOAM`, which can project a cubic mesh onto a 3D stl file. The Figure 2.8 is a $13*13*13$ cubic projected onto a sphere.

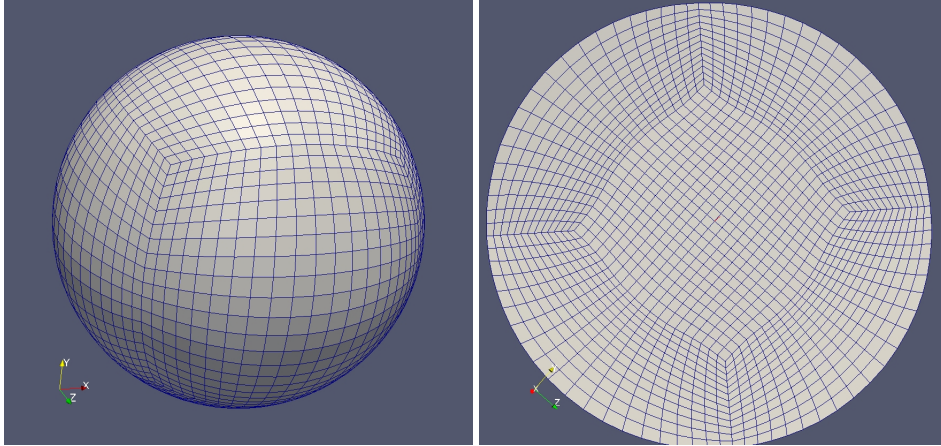


Figure 2.8: The mesh when a $13*13*13$ cube is projected to a sphere. The right one is the cross-sectional surface.

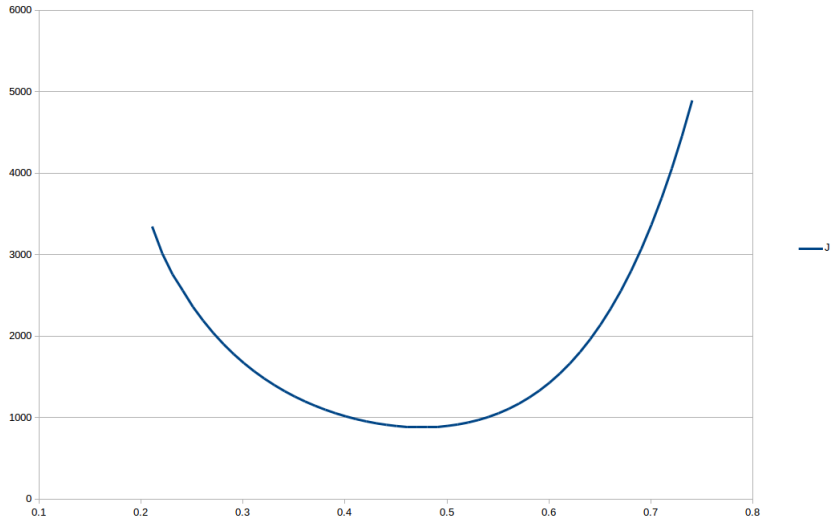


Figure 2.9: The graph of J_h^t as a function of time t , where z_d in (2.13) is used in (2.3)

By checking the values of δJ (see Figure 2.9), we can determine that the minimal value is happens between $t=0.47452$ and $t=0.47453$. The approximation value of J is 889.286 and the required domain is occurred between $t = 0.47452$ and $t = 0.47453$. The Figure 2.10 , 2.11 and 2.12 are present the results for $p_h^{\hat{t}}$, $w_h^{\hat{t}}$ and $\hat{u}_h^{\hat{t}}$ when time t is 0.47.

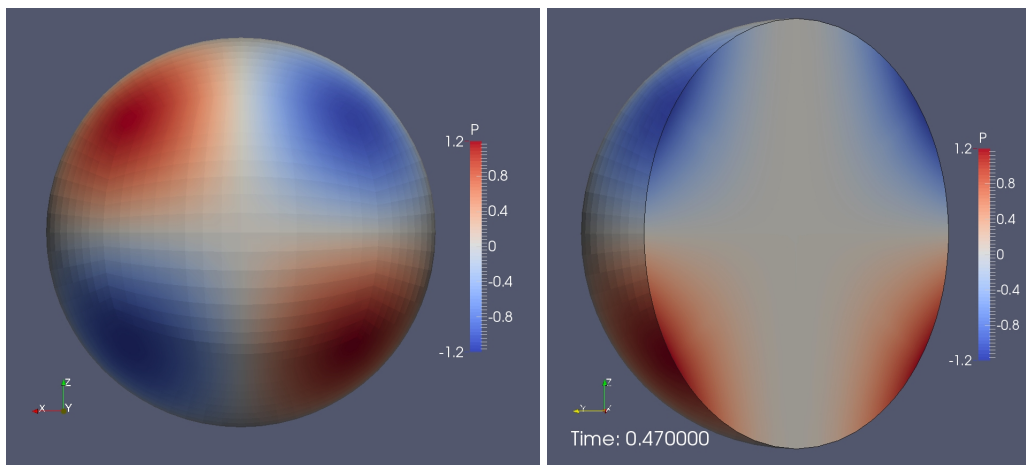


Figure 2.10: The left graphics shows $p_h^{\hat{t}}$ on $\Omega^{\hat{t}}$. The right graphics is the semi sphere with $(1, 1, 0)$ as normal vector on the cross-sectional surface.

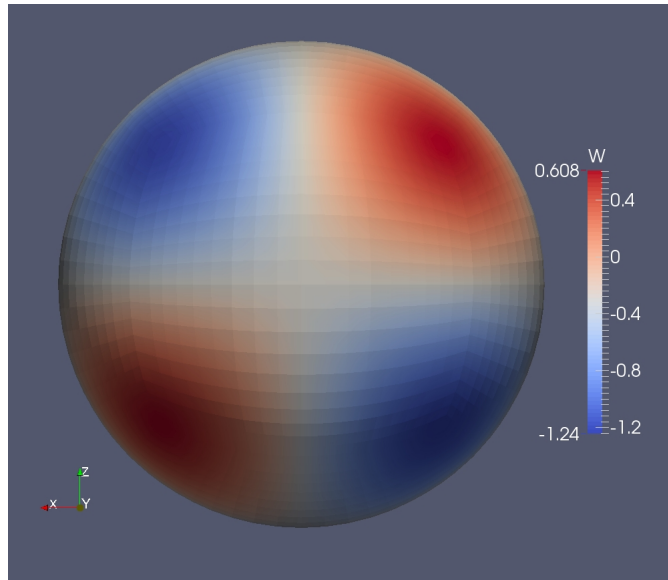


Figure 2.11: The graphics shows w_h^t on Ω^t .

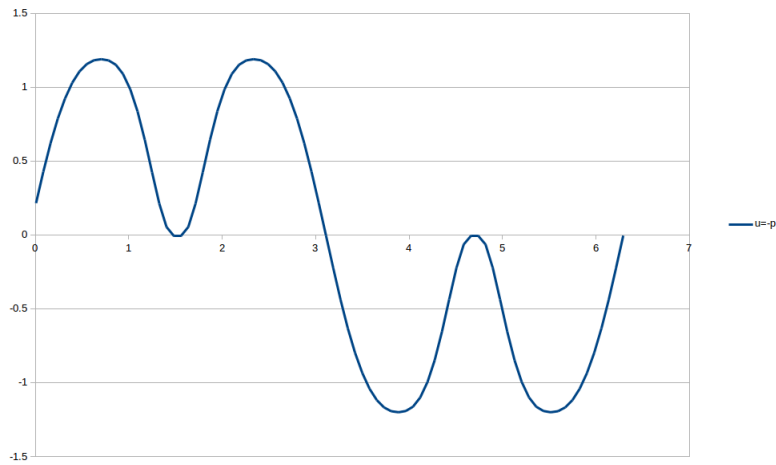


Figure 2.12: The graph of \hat{u}_h^t as a function of θ on with the intersection of $x+y=0$ and the sphere, at time $t=0.47$.

2.5 Conclusions

Shape optimization problem is important in many engineering design problems. We will try to evolve the shape when satisfying given constraints and tracking the shape to find when it has the minimal cost function.

In this Chapter, we have studied the shape optimization problem when the Ω_t is given. It is really interesting to simulate the result with OpenFOAM. We can intuitively observe the cost functional and other condition on the domain for every changing moment. This gives us the idea of what we can expect on the numerical result. Such an ease of use and a simplicity for observation for 3D time-varying domains have never been achieved before.

We confirmed that there is little difference between mesh refinement as long as the grid are sufficiently small. And then we've successfully presented the result of the cost function in both 2D and 3D domains.

3. TOWARD COMPUTATIONAL FORENSICS FOR THE LAPTOP BOMBING TERRORISM CASE OF DAALLO AIRLINES FLIGHT 159, FEBRUARY 2, 2016

3.1 Introduction

Bombing is perhaps the most common tactic done by terrorists. Airplane bombing is particularly favored by terrorists as it can cause a large number of casualties up to several hundred. In addition, it can destroy an airplane, often worth hundreds of million dollars for a modern passenger airliner. Ultra tight inspection procedures in airport boarding processes have been implemented in order to prevent airplane bombings. If an airplane bombing occurs, aviation authorities want to examine all the forensics of such a crime in order to understand the motives and modus operandi of the perpetrators. Such a study will help prevent future airplane bombings. In the U.S., we have not seen an airplane bombing for decades, thanks to the success of homeland security measures and law enforcement.

Elsewhere in the world, terrorists have not given up their attempts to blow up airplanes. The most recent incident happened in 2016. Daallo Airlines is a Somalia-based airline with a hub located in the Djibouti-Ambouli International Airport in Djibouti. On February 2, 2016, Flight 159 (DAO 159) en route from Mogadishu to Djibouti (see Figure 3.1) carried 74 passengers and 7 crew members. The plane was a 19 year-old Airbus A321-111 model. A suicide bomber from the Islamic militant group Al-Shabaab set off a laptop bomb with a built-in explosive device at an altitude of about 14,000ft (4,300m) five minutes after taking off from Mogadishu, Somalia. The explosion blew a large hole in the fuselage, and the suspected bomber was sucked out of the hole. His body was found in Balad, Somalia. Aviation sources said that it would have been a disaster had the bomb detonated after the flight reached its cruising altitude, and there could have been a secondary explosion in the fuel tank. The bombing of Daallo Airlines Flight 159 will be the focal topic of this chapter of the PhD dissertation. The author present some real-world photos of this accident from internet in Figure 3.2.

Wikipedia search for "Timeline of airliner bombing attacks"[14], will turn up tens of instances. Casualties can easily reach hundreds. On June 23 1985, an explosion happened in the forward cargo area of Air India Flight 182 on its path from Montreal to London. The bombing killed 329 people, making it the largest mass murder in Canadian history and the deadliest aircraft bombing. The most famous airplane bombing occurred on Pan Am Flight 103. On December 21 1988, a bomb destroyed a Boeing 747 flying from Frankfurt to Detroit via London. The aircraft crashed onto residential areas of Lockerbie, Scotland, killing everyone on board and also 11 people on the ground.



Figure 3.1: Flight origin: MGQ (Aden Adde International Airport). Destination: JIB (Djibouti-Ambouli International Airport). The location where a burnt body of the terrorist was found: Balad.



Figure 3.2: The photos of DAO159 taken from the Internet [1, 2, 3].

After these attacks, people wanted to know what happened on the airplane. The flight recorder (also known as the Black Box) can help confirm what happened, but the recorder can't tell us everything. Of course, police can run some experiments to recover the truth, but aircrafts are expensive! Running realistic bombing experiments on aircraft explosions would be highly expensive for every single incident. Computational mechanics can simulate the results at a much lower cost; all one needs is a supercomputer and the required software, and no aircraft is destroyed in the computational processes; digital tests can save time and money. That is the beauty of computational forensics.

The English idiom says "a picture is worth a thousand words." We can similarly say that a

video is worth a thousand pictures. Suppose one is running some experiment. There are many physical variables and quantities to observe and measure, but data collection is only a part of the experiment. Thanks to modern video recording technology, one can review a video of every important moment. However, it is beyond video's capacity to record results from multiple directions or observe something in the blind spot of the video. With scientific computations, one can check every little piece of a computational simulation from any needed perspective. Moreover, computational simulations can also present some results not easily achievable by experiments. In reality, one can never observe an aircraft crash test without the flying dust and smoke from an explosion or the breach in the craft without the fragmented shell. However, these observations are possible with computer simulations. For example, one could zoom-in on the important part of the simulation from any direction and any perspective - this is something one could never do in physical experiments. Computational simulation is not just a substitution for an experiment; it has its own value.

3.1.1 Airplane bombing experiments by MythBusters

Our computational simulation aims to test the result of a mid-air airplane damaged by a bomb. We will simplify the case as an explosion happening on a pressurized aircraft. We have found a similar experiment on an episode of MythBusters [15].

MythBusters is an Australian-American science entertainment television program that premiered on the Discovery Channel on January 23, 2003. The creator is Peter Rees and the producer are Australia's Beyond Television Productions. The series was broadcast by SBS Australia and other Discovery channels worldwide. The show were hosted by Adam Savage and Jamie Hyneman, these special effects experts used the scientific method to verify a variety of rumors, myths, news stories, Internet videos and movie scenes. The show was one of the most popular on the Discovery Channel.

The 10th episode of MythBusters, aired on January 18, 2004, was concerned with explosive decompression. The hosts tested some explosives on a pressurized airplane, and checked whether the hole caused by a bullet or a small explosive would grow dramatically to cause a breakup of the

plane as seen in movies. They experimented with three different cases on a pressurized airplane. The airplane was pressurized at 8 pounds per square inch (8 psi), making the pressure difference between the inside and outside of the airplane the same as it would be at a height of 35,000 feet. The first case dealt with bullet damage to a window, and the results was a normal bullet hole. In the second case, the hosts used a special charge to damage the circumference of the window, which resulted in a hole in the window. A passenger's arm would have been blown out from the hole, but no other damage occurred (see Figure 3.3) In the last case, hosts used a large charge to try to break up the airplane (see Figure 3.4)



Figure 3.3: The result by a special explosive charge on the circumference of the window by a MythBusters' experiment. For video, see https://youtu.be/Fi1_117M8FA?t=2m30s



Figure 3.4: Explosion by a large explosive charge by a MythBusters' experiment. For video, see https://youtu.be/Fi1_117M8FA?t=4m17s

3.2 Computer modeling for the airplane, explosion and validation

We study the computer modeling of the problem in this section.

3.2.1 Software LS-DYNA

LS-DYNA is a numerical modeling tool developed by LSTC [16]. It started from the Lawrence Livermore National Laboratory (LLNL) in 1976, a 3D FEA program DYNA3D developed by Dr. John O. Hallquist. It is generally designed for transient dynamic analysis of highly nonlinear problems. As a general-purpose multiphysics simulation software, LS-DYNA can be applied to metal forming, drop tests, crash and occupant simulations and other contact-related applications. LS-DYNA has numerous capabilities and functions, especially for transient dynamic problems such as automotive crashes and explosions. More recent enhancements concern fluid-structure inter-

actions. Much of LS-DYNAs finite element (FE) discretization of real industrial problems deals with thin-shell elements, for example the beam and solid elements. Based on the Mindlin theory, a large number of element formulations for shells are available. Therefore, LS-DYNA offers choices between improved accuracy and computational efficiency. Because of the efficiency of the element vectors' formation and robustness in the case of large element distortions, reduced order integration is often preferred in large deformation analyses. There are also a choice of methods and formulations to avoid hourglassing problems. Moreover, the software provide more than 100 different material models to represent many types of highly nonlinear material behaviors. It can simulate real-world problems and show the results in 3D video animations.

A large number of problems can be solved or simulated by LS-DYNA. For example, to improve the industrial manufacturing process, a factory may need to try different spray nozzle positions for optimal efficiency. Car companies need crash tests to define safe design standards in crash worthiness and crash compatibility. Of course, they still need actual vehicles to run the final crash tests, but computer simulations can save a lot of time and money by allowing them to establish the perfect settings for the physical tests. Reference [17] shows an experiment investigating the impact conditions for components and tensile tests on specimens with simulations. For safety design of civil engineering structures under fire conditions, one needs to investigate various modeling approaches of composite actions between the concrete slab and the steel beams; it is not realistic to build many buildings with various structures, and burn them. Reference [18] shows that LS-DYNA can to capture key phenomena of heated structures. It can also be used in biomechanics. Reference [19] shows a method for building a Hill-type muscle model with LS-DYNA.

When Malaysia Airlines flight MH370 disappeared on March 8, 2014, no one could find the aircraft, and no one had any idea of what had happened. Chen [20] ran several emergency water landing simulations from various angles. The simulation results showed an angle at which the aircraft could have ditched safely in water, as well as a possible landing angle that could explain the plane's mysterious disappearance.

In our case, we use LS-DYNA to simulate the DAO 159 crash through *finite element analysis*

(FEA) and *smoothed particle hydrodynamics* (SPH). LS-DYNA can model the process of how explosives breached the aircraft fuselage and the interaction between all the fragmented shells.

3.2.2 Mechanical models and simplification

Our first goal is to see the video animations for the whole process of the explosion while DAO 159 was crashing. The second goal was to establish the minimum amount of explosives that could breach the aircraft. Thus, we need to build an aircraft model with LS-DYNA and test it with different amounts of explosive.

To be able to do so, we use the same aircraft geometry as Chen et al. [20, 21] and build a slightly smaller inner layer as the cabin wall. But we note that a real aircraft is far more complicated. A horizontal floor separates the passenger area and cargo bay. The aircraft also has many sections such as the cockpit, passenger area, bathroom, flight attendants area, etc. Other minor supporting or decorative structures also exist besides the exterior fuselage skin. There are additional varying factors such as the number of passengers and seats, the amount of luggage, etc. Our numerical experiments are an attempt to investigate the explosion, focusing on the hole caused by the bomb. We focus on the bomb damage and the breached walls. Therefore, this experiment only required an exterior fuselage skin, an inner cabin wall and a bomb (TNT explosive).

The aircraft material primarily utilized in transport aircraft today is an aluminum alloy, 7075-T6, which is one of the strongest among all such alloys. Many plastic pieces form the inner cabin wall, and the differences in strength between plastic pieces are omitted (under explosion circumstances). We choose a common plastic as the material. In LS-DYNA simulations, the exterior fuselage skin was structured as an aluminum alloy 7075-T6 shell with 1.2 mm thickness, and the inner cabin wall is a plastic shell with the same thickness. LS-DYNA has more than 100 material models. The fundamental material models for the aircraft explosion experiment are viscoplasticity and fracture mechanics. We'll give a brief review for the technical equation behind the concept; the complete theory is in the LS-DYNA Theory Manual [22].

The classical linear elasticity theory is based on the constitutive equation

$$\bar{\sigma} = C\bar{\epsilon},$$

where $\bar{\sigma}$ is the Cauchy stress tensor, C is the fourth-order stiffness tensor, $\bar{\epsilon} = \frac{1}{2}[\nabla u + (\nabla u)^T]$, is the infinitesimal strain tensors, and u is the Lagrangian displacement.

The equation of motion is

$$\nabla \cdot \bar{\sigma} + F = \rho \frac{\partial^2(u)}{\partial t^2}.$$

where F is the body force per unit volume, and ρ is the mass density. In the case of nonlinear viscoplasticity, such as in our investigation of the break-up of an aircraft and the impact effects of fracture, the strain is generally large. Therefore, the time rates of change of both stress and strain should be considered within the constitutive equation. We have $\bar{\epsilon} = \bar{\epsilon}_e + \bar{\epsilon}_{vp}$ where $\bar{\epsilon}_{vp}$ and $\bar{\epsilon}_e$ are the viscoplastic and elastic strain tensor components of $\bar{\epsilon}$. For viscoplasticity described in terms of spring-dashpot-slider elements, there are three constitutive models: the elastic perfectly viscoplastic solid, the elastoviscoplastic hardening solid and the perfectly viscoelastic solid and their serial or parallel connections.

In this paper, we choose a simplified effective material model, MAT_PLASTIC_KINEMATIC, for the fuselage and cabin. This model is designed for modeling isotropic and kinematic hardening plasticity with the option of including rate effects. In this model, for the yield stress σ_y we use the following Cowper and Symonds model:

$$\sigma_y = \left[1 + \frac{\dot{\epsilon}^{1/p}}{C} \right] (\sigma_0 + \beta E_p \epsilon_{eff}^p),$$

where σ_0 is the initial yield strength, $E_p = \frac{E_t E}{E - E_t}$ is the plastic hardening modulus, β is a parameter satisfying $0 \leq \beta \leq 1$, $\dot{\epsilon} = \sqrt{\dot{\epsilon}_{ij} \dot{\epsilon}_{ij}}$ is the strain rate, $\epsilon_{eff}^p = \int_0^t \sqrt{\frac{2}{3} \dot{\epsilon}_{ij}^p \dot{\epsilon}_{ij}^p} dt$ is the effective plastic strain, $\dot{\epsilon}_{ij}^p$ is plastic strain rate which is the difference between the total and elastic strain rates and p and C are user-defined input constants. The parameters for the material model are listed in Table



(3.1).

To compute the a case of whole aircraft would take up a lot of computing resources and time, especially when we use larger amounts of explosives. During numerical experiments, we have noticed that some parts of the fuselage skin were not impacted by the blast particles. Since we only need to check whether the explosion was large enough to cause certain damage to the aircraft fuselage, we have selected only a portion part of the fuselage to run the test. At the end of each simulation, we checked that the boundary of the selected fuselage cell partition was not impacted by the blast particles. We run still a few more cases with the whole aircraft fuselage to make sure that the simplified case was representative of the original test.

Parameter for Aluminum	Value	Parameter for Plastic	Value
Density	3000 kg/m^3	Density	1400 kg/m^3
Young's modulus	70 GPa	Young's modulus	1.4 GPa
Poisson's ratio	0.35	Poisson's ratio	0.39
Yield stress	0.47 GPa	Yield stress	0.016 GPa

Table 3.1: Parameters for the aircraft.

3.2.2.1 Whole Plane Simulation

When using the whole aircraft model to run explosion simulations, the grids from the original model may not have sufficient resolution. There are two ways to split the mesh (check Figure 3.5): the first type is  (type A), another type is  (type B). Type A remained one edge from the original grid. That means that after triple refinements, the grids look thin and long, and still maintain a long edge. This is not ideal because we hope to minimize the longest edge from each grid. Type B causes a free edge. The red edges from Figure 3.5c are the free edges. Free edge is when an element edges in the mesh is not shared with other elements. Therefore, free edges may indicate connectivity problems in the mesh and/or the underlying model geometry.

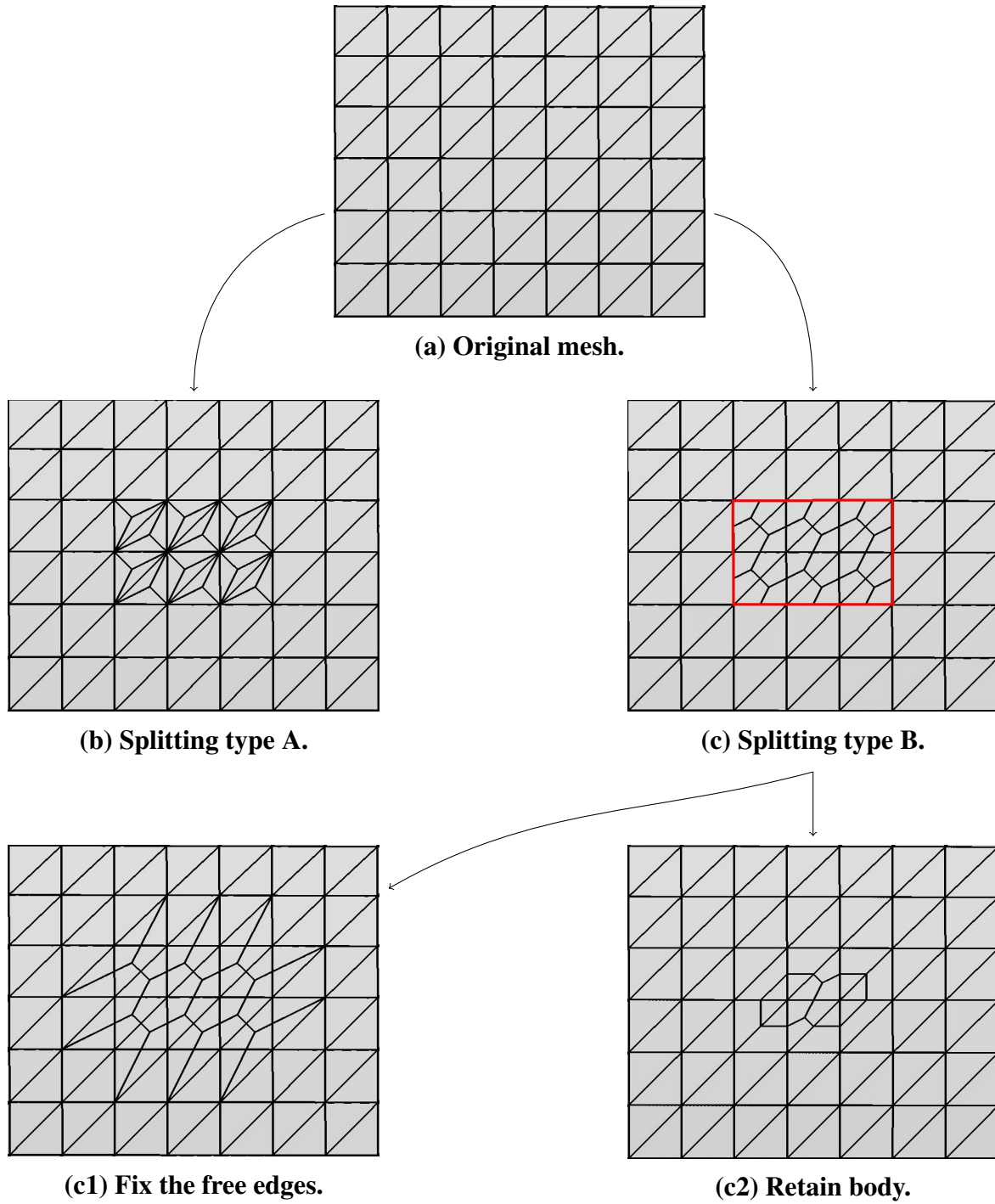


Figure 3.5: Mesh refinements.

3.2.2.2 Aircraft Windows

Generally, aircraft windows are made of acrylic plastics colloquially called plexiglass. This material is light, relatively strong and has decent optical properties. The window is elliptic with a major radius 225 mm and a minor radius 150 mm on the aircraft shell. We use the material model `MAT_PLASTIC_KINEMATIC`, where Table (3.2) displays its parameters list.

Parameter for Acrylic	Value
Density	1200 kg/m^3
Young's modulus	3.2 GPa
Poisson's ratio	0.37
Yield stress	0.071 GPa

Table 3.2: Parameters for acrylic windows.

There are two ways to connect the window to the aircraft: `CONTACT_TIED_SURFACE_TO_SURFACE_FAILURE` or `CONTACT_TIEBREAK_SURFACE_TO_SURFACE_ONLY`. The results are slightly different during these transient response to a blast, but they look the same after the window parts are blown out or away.

3.2.3 Modeling the explosive

We are now in a position to simulate the laptop bombing on DAO 159. We choose TNT as the explosive and store it in the laptop. The principal part of the laptop bomb is the explosive. We use only a box-shaped TNT explosive to represent the laptop bomb. The assumed distance between the TNT explosive and the cabin shell is 1 cm. We use SPH to model the explosive. Not that the blast wave draws a large amount of energy to a small, very localized volume to increase pressure and flow. The material model for explosives in LS-DYNA is `MAT_HIGH_EXPLOSIVE_BURN`.

(See Table (3.1).) The idea of SPH is essentially an N-body integration scheme, whose major departure from the classical method is the absence of a grid. It uses conservation equations to approximate particles.

In the material model of explosives, the pressure satisfies

$$p = F p_{eos}(V, E),$$

where F is the burn fraction and p_{eos} is the pressure from the equation of state (see below). The value of burn fraction is

$$F = \max(F_1, F_2),$$

where

$$F_1 = \frac{2tDA_{e_{max}}}{3v_e}, \quad F_2 = \frac{1 - V}{1 - V_{CJ}},$$

where t is current time, D is detonation velocity, and V_{CJ} Chapman-Jouguet relative volume.

The equation of state of high explosives is chosen to be EOS_JWL (the empirical Jones-Wilkins-Lee equation of state), given by

$$p_{eos}(V, E) = A \left(1 - \frac{\omega}{R_1 V}\right) e^{-R_1 V} + B \left(1 - \frac{\omega}{R_2 V}\right) e^{-R_2 V} + \frac{\omega E}{V},$$

where V is the relative volume, E is internal energy per initial volume, and ω, A, B, R_1 and R_2 are user defined input parameters. The parameters for TNT explosive needed when using the EOS_JWL can be formed in Table 3.3.

Parameter for TNT	Value
Density	1630 kg/m^3
Detonation velocity	6930 m/s
Chapman-Jouget pressure	21 GPa
A in JWL	371.2 GPa
B in JWL	2.23 GPa
R_1 in JWL	4.15
R_2 in JWL	0.95
ω in JWL	0.3
Initial E in JWL	7 GJ/m^3
Initial V in JWL	1.0

Table 3.3: Parameters for TNT in JWL equation.

The number of SPH is important. With the same size of explosive, different arrangements of the SPH might lead to different results. The smooth particle mesh must be uniform. If the smooth particle represents a $1 \times 1 \times 100$ rectangle space, the explosion will expand mainly in one direction instead of exploding uniformly. A lack of sufficient smooth particles causes poor accuracy but a large amounts of smooth particles may take too long to compute.

3.2.4 Cabin Pressurization

To model the cabin pressurization, we have tried a few methods, but only one has worked. First, we use `PARTICLE_BLAST` to simulate the way that an explosion can fill air particles to a domain automatically. However, there is no way to fill a domain with two kinds of air particles — the kind with higher pressure inside the cabin and the one with lower pressure outside the fuselage. The keyword `PARTICLE_BLAST` can only contain one kind of air particle in a rectangle domain or the entire domain. Therefore, we have achieved no success. Modeling air particles with

SPH has another possibility way; the keyword `EOS` defines an equation of state. The keyword `EOS_IDEAL_GAS` for air particles could set up ideal gas properties. Therefore, one can describe the pressure by $P = \frac{nRT}{V}$. However, such air particles won't flow based on the pressure difference alone; the left graphics in Figure 3.6 sets up two kinds of air particles, one fuselage skin and one TNT explosive. The right graphics in Figure 3.6 shows the result after detonation. The air particles moved because they were bumping into blast particles not because of the pressure difference.

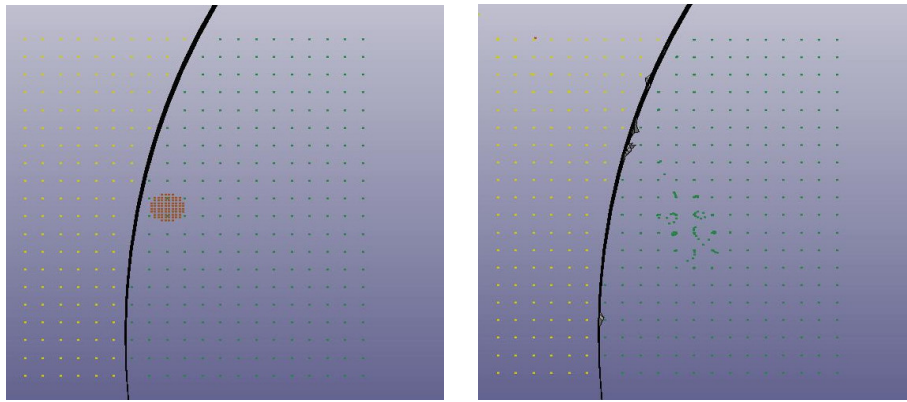


Figure 3.6: The left figure shows the set-up of the inner and outer air particles using SPH. The right figure shows the air particles have been disturbed without flow after the fuselage has been breached.

Finally, we have decided to use the keyword `LOAD` to load the pressure difference (8 psi) uniformly on the surface with the normal direction of fuselage skin. The effect of pressure difference on the surface is a force acting on the surface uniformly, and its magnitude is the degree of pressure difference. The white arrows in Figure 3.7 show how `LOAD` works on the surface.

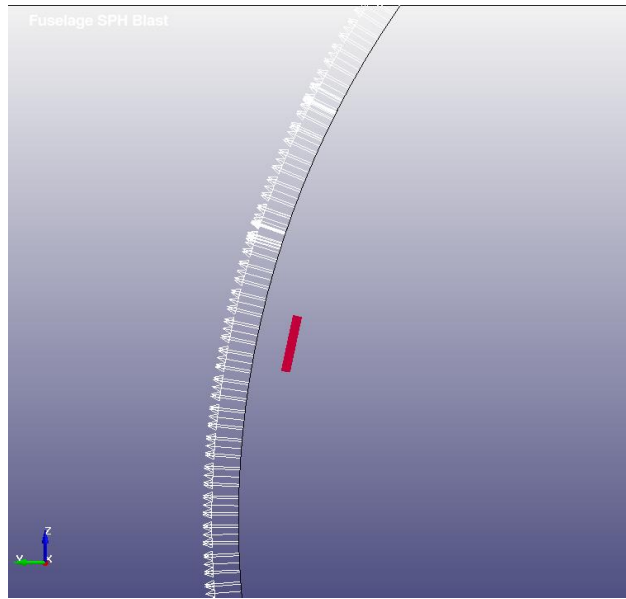


Figure 3.7: The arrows indicate how the LOAD works on the surface.

3.2.5 Cracks and fracture criteria

When a blast occurs, the impacts between objects follow. Contact conditions are necessary when impact is present. The keyword `CONTACT` uses the *contact-impact algorithms* to handle sliding and impact along interfaces. The impact between the blast SPH nodes and the aircraft is modeled by `CONTACT_AUTOMATIC_NODES_TO_SURFACE` since this keyword describes the contact force. We also modeled the contact force between the fragmentation of fuselage and cabin with `CONTACT_ERODING_SINGLE_SURFACE`.

The erosion criteria are available for each model, and are independent from the choices of material. The parameter f_s is normally used as the effective plastic strain for eroding elements or the so-called the failure strain. There are many independent conditions, and the element will be deleted by the calculation if any of the conditions are satisfied. The criteria for failure according to LS-DYNA (page 18.15 from [22]) are:

- $p \geq p_{min}$ where P is the pressure (positive in compression), and p_{min} is the pressure at failure.

- $\sigma_1 \geq \sigma'_{max}$, where σ_1 is the maximum principal stress, and σ_{max} is the principal stress at failure..
- $\sqrt{\frac{3}{2}\sigma'_{ij}\sigma'_{ij}} \geq \bar{\sigma}_{max}$, where σ'_{ij} are the deviatoric stress components, and $\bar{\sigma}_{max}$ is the equivalent stress at failure.
- $\epsilon_1 \geq \epsilon_{max}$, where ϵ_1 is the maximum principal strain, and ϵ_{max} is the principal strain at failure.
- $\gamma_1 \geq \gamma_{max}$ where γ_1 is the shear strain, and γ_{max} is the shear strain at failure.
- The Tuler-Butcher criterion,

$$\int_0^t [\max(0, \sigma_1 - \sigma_0)]^2 dt \geq K_f,$$

where γ_1 is the maximum principal stress, γ_0 is a specified threshold stress, $\sigma_1 \geq \sigma_0 \geq 0$, and K_f is the stress impulse for failure. Stress values below the threshold value are too low to cause fracture even for very long duration loadings.

3.2.6 Validation of SPH Blast Simulation

We validated the SPH blast computational modeling by simulating Shirey's explosion test [23]. The test was used to investigate the breach of a steel plate after a direct explosion by circular disks of sheet explosive C4, in this case (see Figure 3.8 for the set-up, and Figure 3.9 for the simulation processes). Shirey has concluded that there is a threshold of explosive thickness needed for breaching the plate. If the diameter of the explosive disk is small, it needs to be thicker in order to breach the steel plate. However, when the diameter is large enough, there's a minimum thickness requirement to breach the steel plate. (see the red curve in Figure 3.10 for the property). Since Shirey didn't describe details of the material properties of the steel used in the test, instead we choose a property for some common steel and experimented with failure strains of 0.28, 0.31 and 0.41. For all these failure strains, one simulation results show qualitatively causes of similar shapes.

Figure 3.10 shows the thresholds of explosive thickness from both the LS-DYNA simulation and Shirey's experiment. The material model for the steel plate part is MAT_PLASTIC_KINEMATIC, and we used MAT_HIGH_EXPLOSIVE_BURN for C4 explosive. See the parameters in Table (3.4) and Table (3.5).

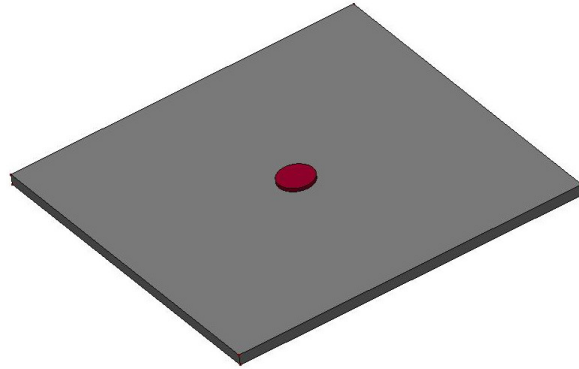


Figure 3.8: Explosive set-up geometry.

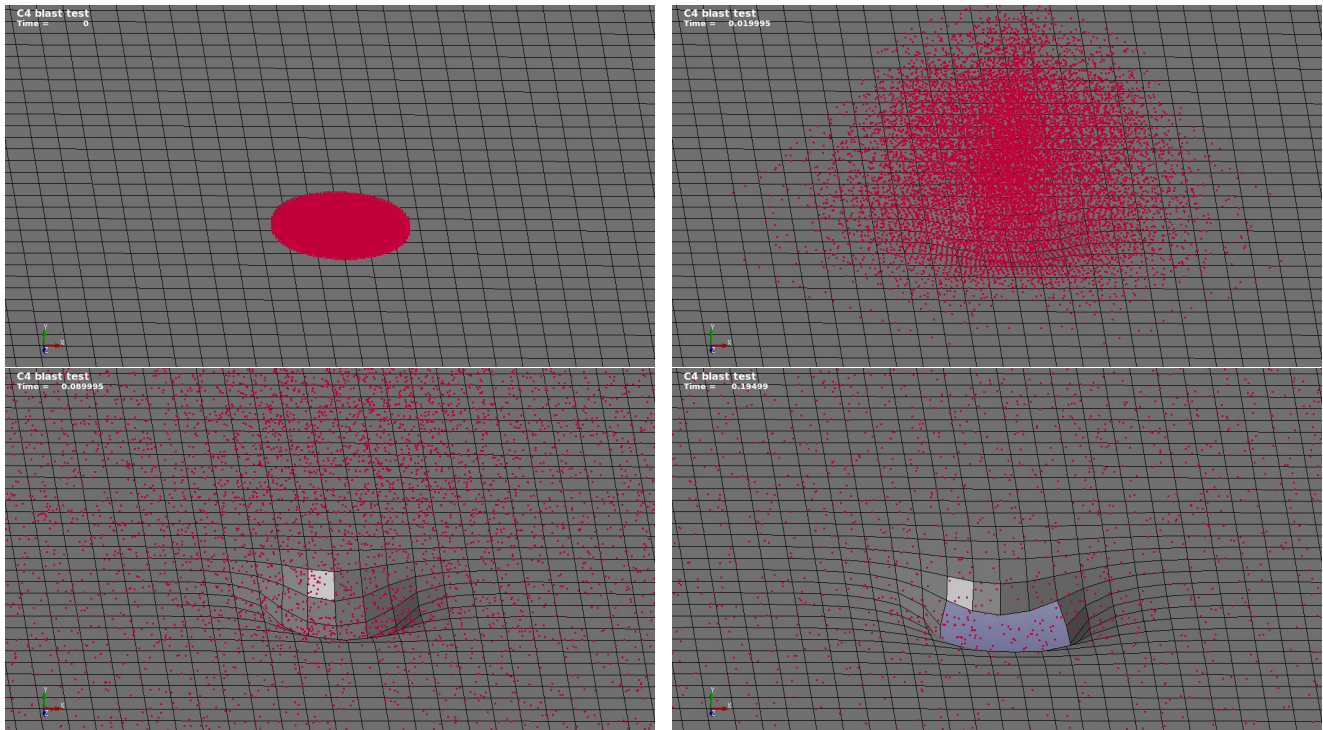


Figure 3.9: Snapshots of simulation for the Shirey's explosion test.

Parameters	Value
Density	1601 kg/m^3
Detonation velocity	8193 m/s
Chapman-Jouget pressure	28 GPa
A in JWL	609.7999 GPa
B in JWL	13 GPa
R_1 in JWL	4.5
R_2 in JWL	1.4
ω in JWL	0.25
Initial E in JWL	9 GJ/m^3
Initial V in JWL	1.0

Table 3.4: Parameters for high explosive (C4).

Parameters	Value
Density	7691 kg/m^3
Young's modulus	210 GPa
Poisson's ratio	0.3
Yield stress	0.3 GPa

Table 3.5: Parameters for steel plate.

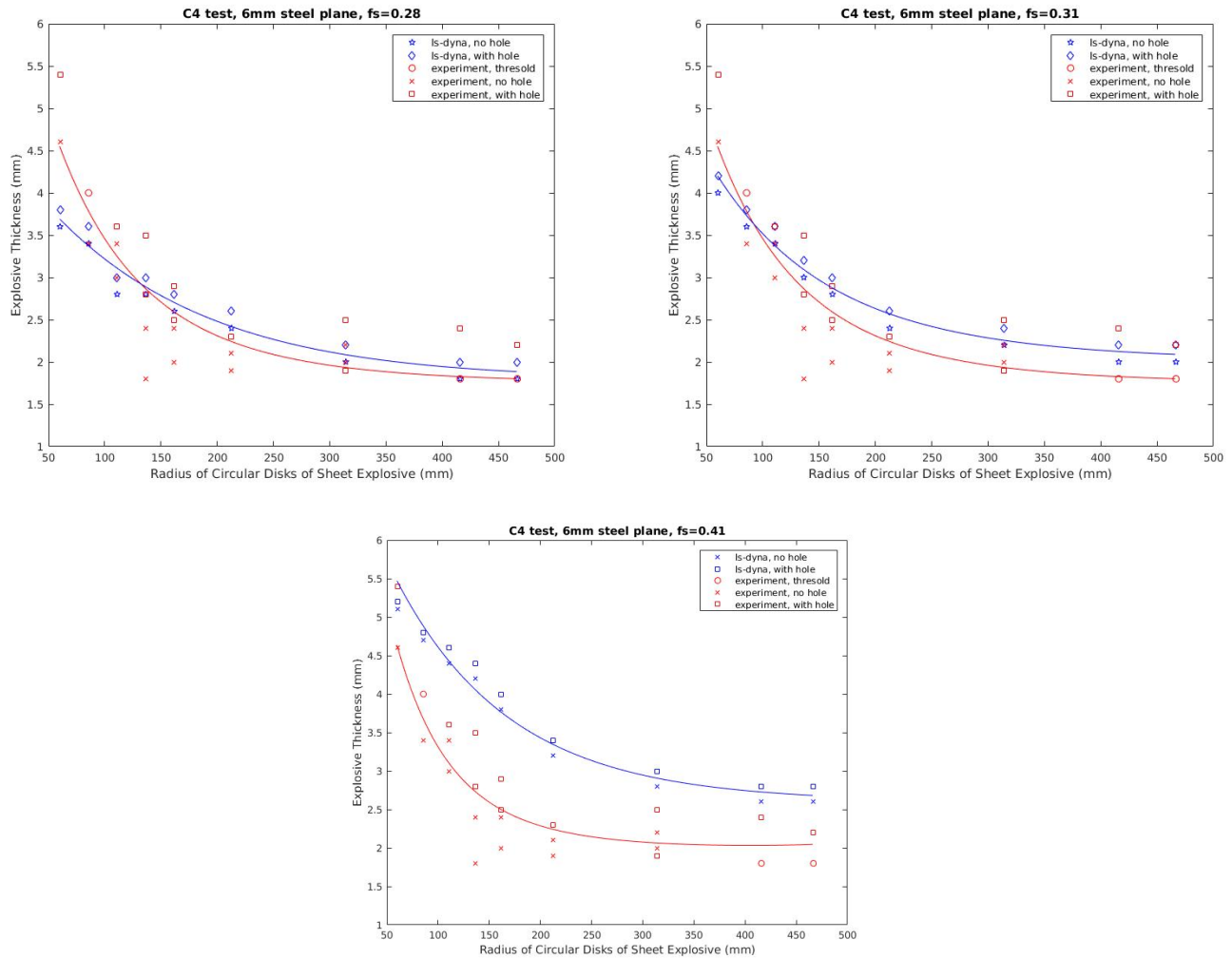


Figure 3.10: Threshold causes for 6 mm steel plate with different failure strains.

Jasak, et al's book [24] mentions two explosion tests with reinforced concrete structures. One test was conducted by Jonasson and reported by Forsen [25] who found that the relationship between charge weight and hole diameter is linear. Lonnqvist [26] extended this research with additional parameters and combined them with the previous work by Jonasson and Persson. Here we quote Figure 3.11 directly from the book to show the conclusions from these two explosion tests. We did not actually simulate the reinforced concrete explosion test with LS-DYNA, as there are too many parameters to be considered that are not present in the aircraft explosive case. We just want to point out that the simulation of the aircraft explosion case can also establish that the

relationship between charge weight and hole diameter is linear (Figure 3.12).

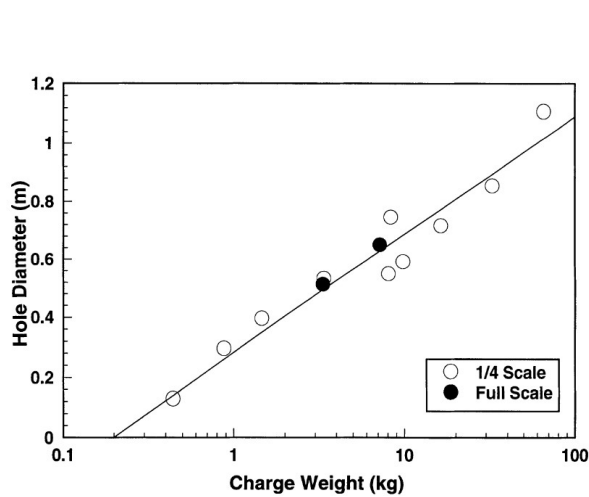


Figure 10.18. Experimental results from contact charges [data from Forsen (1990)].

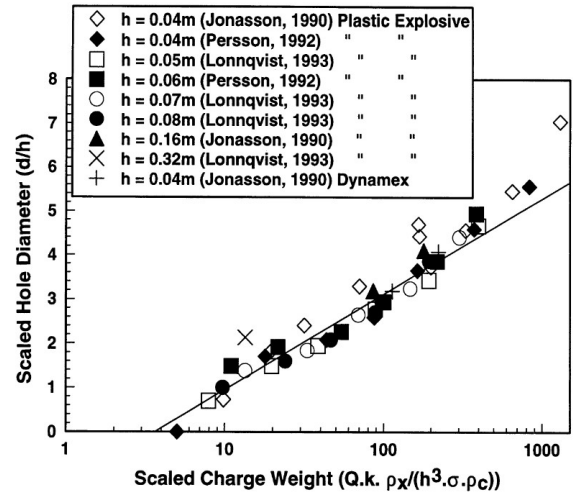


Figure 10.19. Dimensionless charge weight/hole diameter relationship [data from Lonnqvist (1993)].

Figure 3.11: Relationship between charge weight and hole diameter for reinforced concrete explosive test.

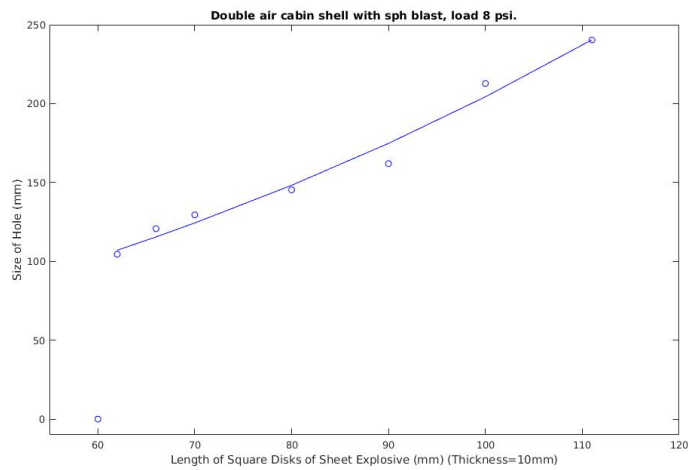


Figure 3.12: Relationship between explosive charge weight and hole diameter in aircraft explosion.

3.3 Computer simulation of the Daallo Airline Flight 159 bombing

The first case we have tried is a cargo plane, which has no windows. In this case, we still put an extra layer inside the fuselage as the plastic cabin wall. We have obtained the results below. We have also established the relationship between the size of the charge and the dimension of the hole. Figure 3.13a is a cross section of the aircraft. The inner square layer is the cabin. Note that we didn't need to build the whole cabin for this test. Figure 3.13b-3.13f are snapshot photos of the explosion. In these figures, the left layer represents the fuselage, the right one is the cabin and the red particles are the TNT explosive.

Figure 3.12 shows that the relationship between charge weight and hole diameter is linear, and that TNT explosive measuring $6\text{ cm} \times 6\text{ cm} \times 1\text{ cm}$ is not enough to breach the aircraft based on our simulation result. However, TNT explosive measuring $6.2\text{ cm} \times 6.2\text{ cm} \times 1\text{ cm}$ was enough to breach it. Figure 3.14 compares the real aircraft damage photo with the LS-DYNA simulation result.

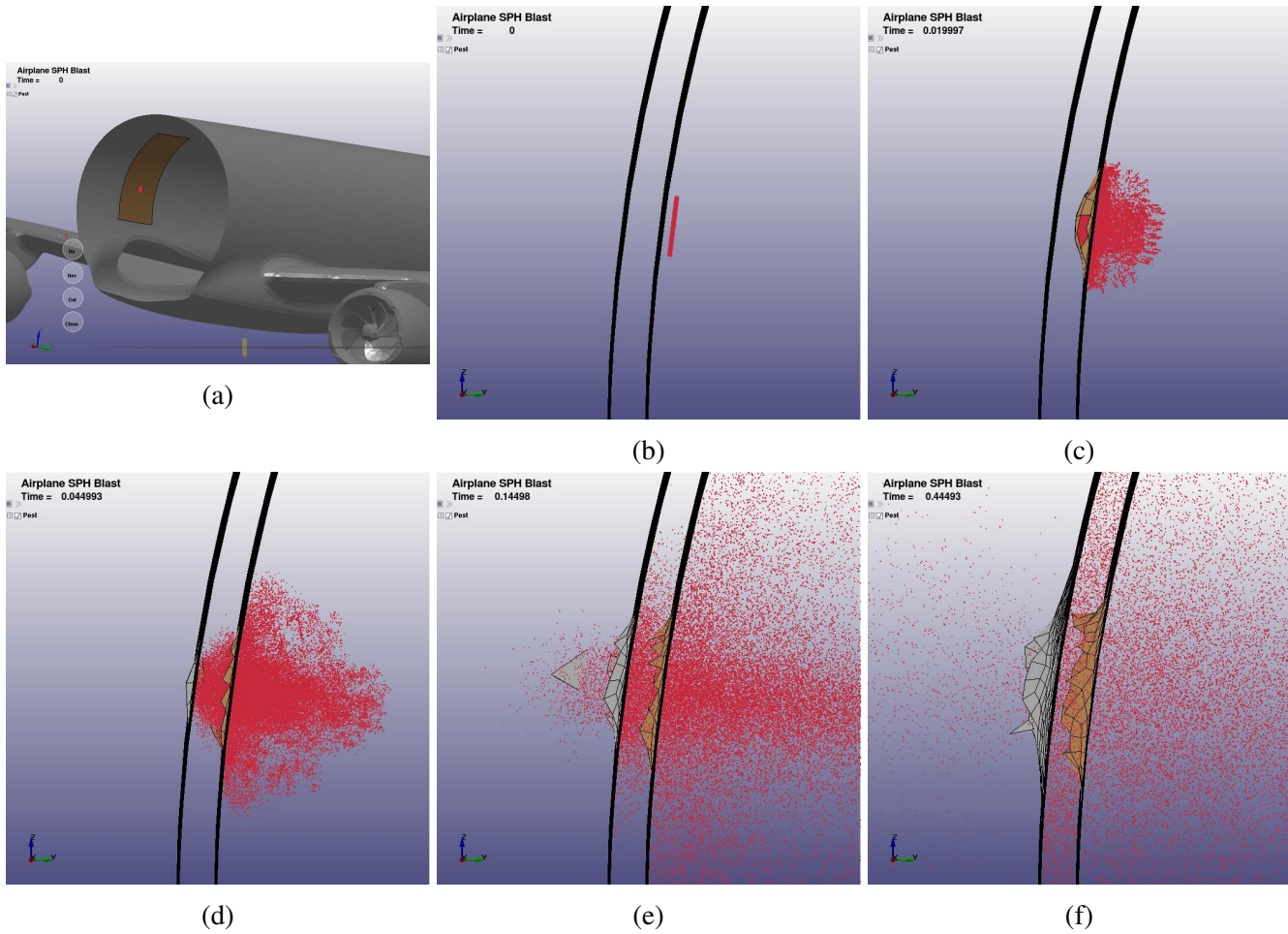


Figure 3.13: Panel of snapshot of explosion. (With $8 \times 0.6 \times 8 \text{ cm}^3$ size of TNT explosive) The unit of time is millisecond. See the video animation in <http://goo.gl/aUwqJx>

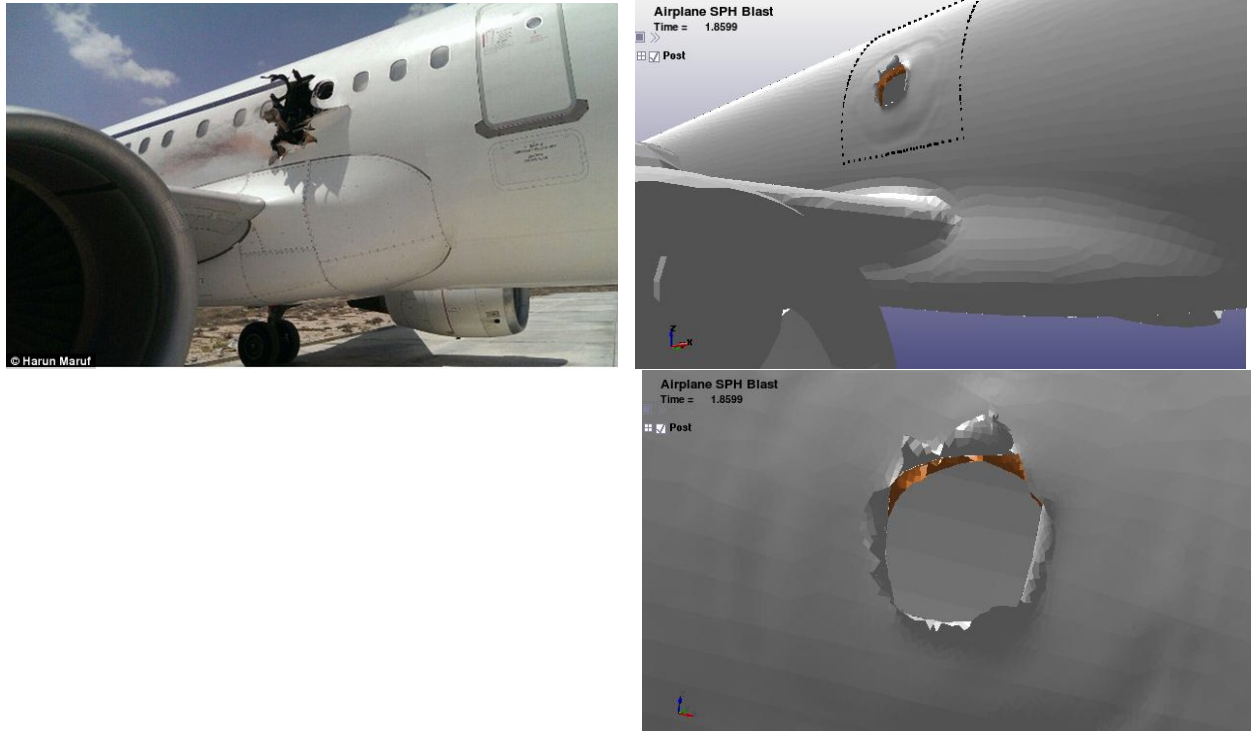


Figure 3.14: Comparison between the real airplane damage photo and the simulation result (With $24 \times 1.5 \times 24 \text{ cm}^3$ size of TNT explosive).

By comparing the simulated result and the real photo in Figure 3.14, we have noticed that the real hole was nearly rectangular in shape, but ours is circular. After we have seen the photo of the inside view of the airplane (Figure 3.2), we have found that airframes' ribs were the cause. As seen in the bottom right photo in Figure 3.2, the ribs obviously bounded the hole. Therefore, our next, refined model is to treat a fuselage with ribs.

After searching online, we have formed the structure of an airframe. (Figure 3.15) We want to build a cylinder shaped passenger cabin with only two rings of ribs, since the hole was between two ribs. However, as shown in Figure 3.16 having only two rings of ribs for a cabin would have suffered extra deformation. Therefore, the final model is a cylinder-shaped passenger cabin fully supported by many rings of ribs. Figure 3.17 shows snapshot of the process of explosion on the fully supported cabin model. The left one in Figure 3.18 shows the result from no-ribs case, and the right-top one in Figure 3.18 shows the result when the fuselage fully supported by ribs. A few

segments of the LS-DYNA computer codes are shown in Appendix 1 with comments. Moreover, it is important to refine the mesh in a proper way. With the same structure as in Figure 3.17, the result will be different if we refine the mesh. If we split every element in Figure 3.17 into 16, i.e. four rounds of refinements, then the result will be as the left panel in Figure 3.19. If we split every element in the previous case into 16, then the result will be as the right panel in Figure 3.19.

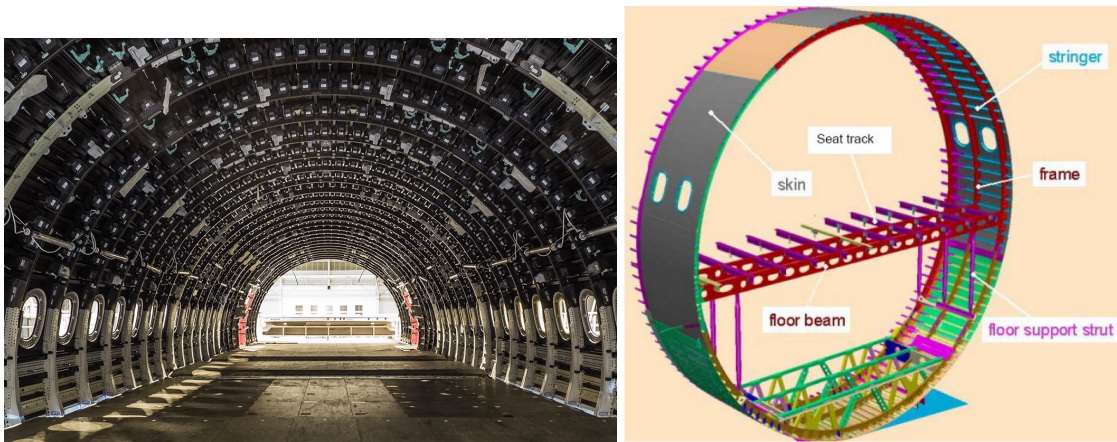


Figure 3.15: A real fuselage and design.

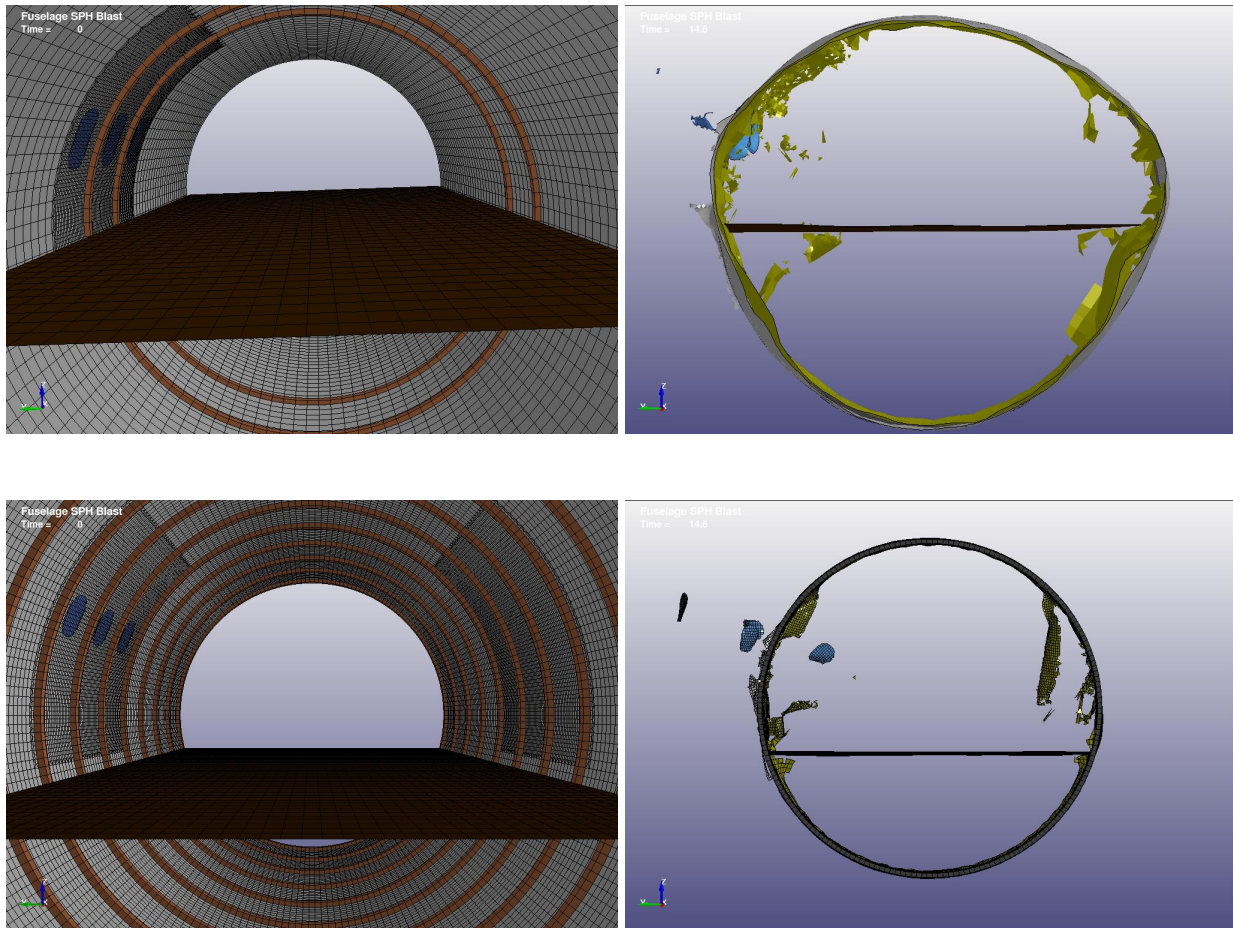


Figure 3.16: The top two panels are the cabin fuselage with only two ribs and the bottom two are the one fully supported by more ribs. The left two are the ribs setting, and the right two are the results of damage after 14 ms.

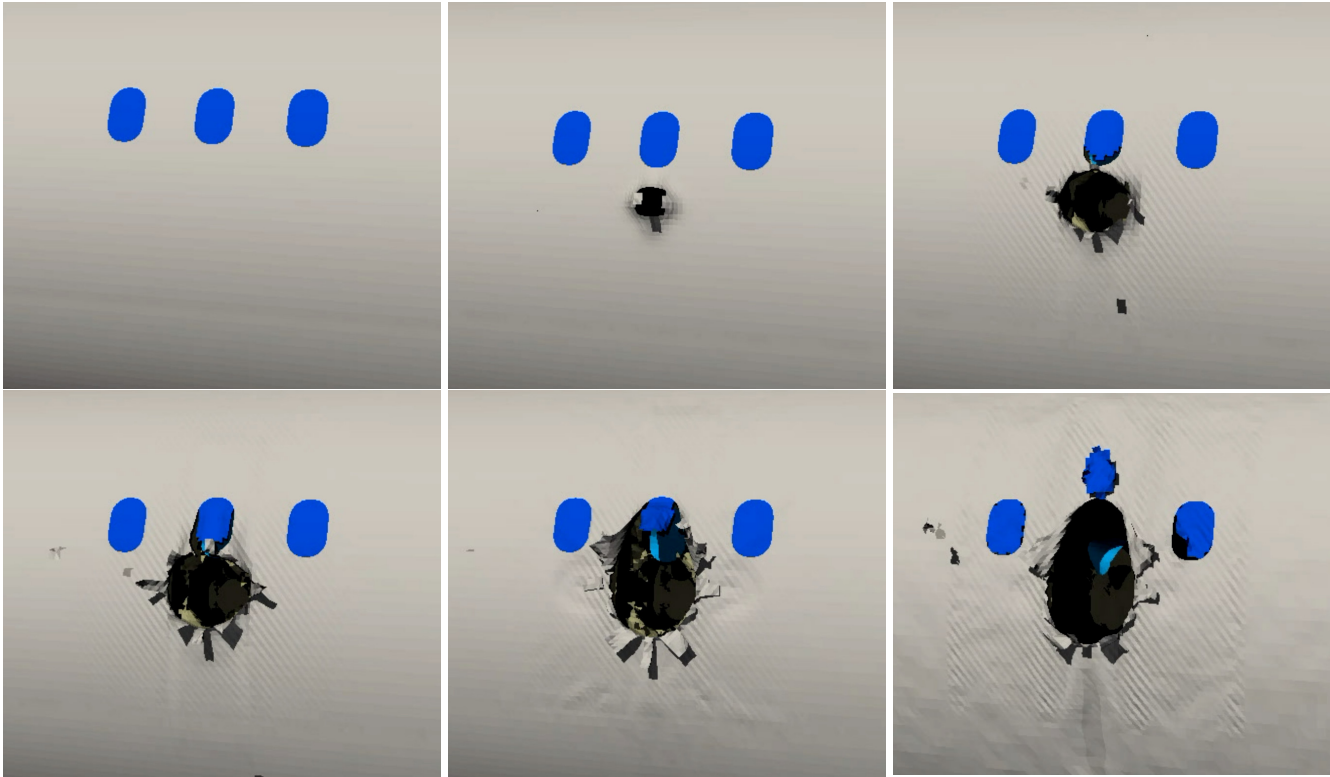


Figure 3.17: Snapshots of explosion showing damage to the aircraft body, whose airframe is fully supported by ribs. For video animation, see <http://goo.gl/3NmV2u>

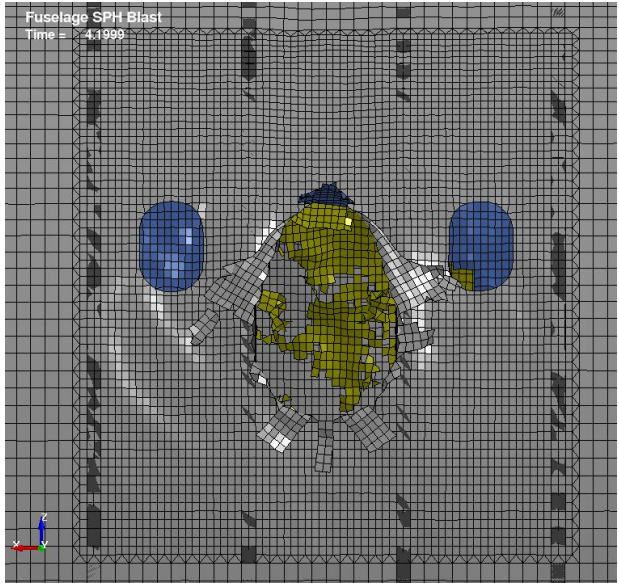
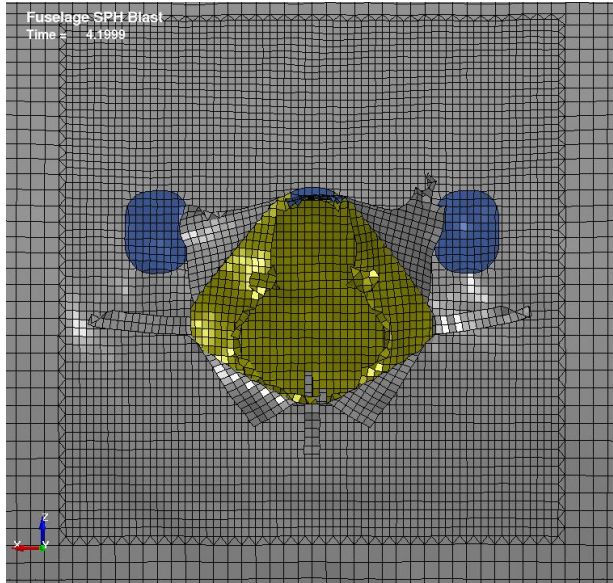


Figure 3.18: The left panel corresponds to is the case without ribs, and the right top one is the case with ribs. The right bottom one is the real photo.

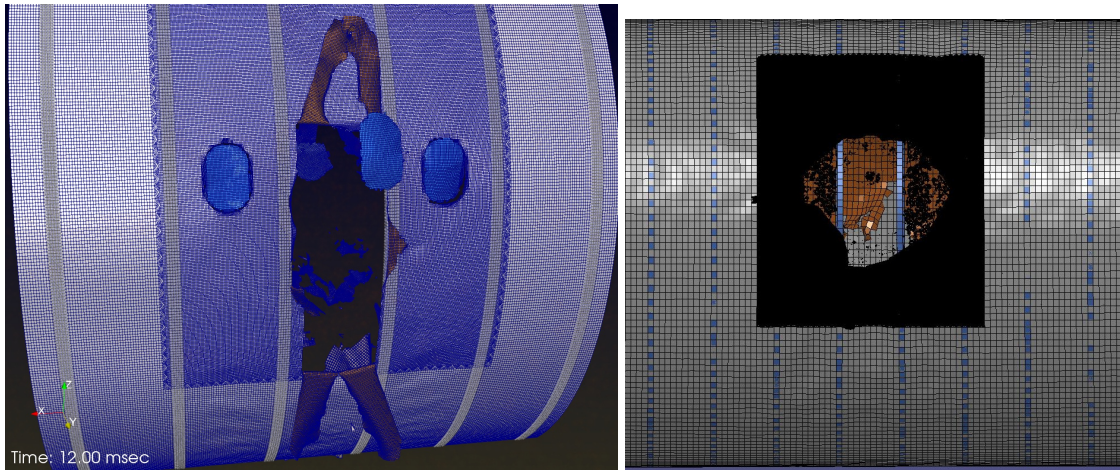


Figure 3.19: The result of damage after mesh refinement.

3.4 Conclusions

Antiterrorism is a major undertaking by governments all over the world. A substantial part of antiterrorism revolves around the prevention of airplane bombings. We hope our study on airplane bombing will contribute to the understanding of and forensics for antiterrorism.

A modern airliner has a sophisticated airframe structure, allowing it to fly at high speed with minimum weight. The supporting structure, such as rings of ribs, the floor, and even restrooms and passenger seats can play some role in protecting an airplane in the event of a small scale bombing. This is an interesting finding in our work.

We have also learned that in the new generation of passenger airliners, Boeing is able to fabricate the fuselage hull of an aircraft in one piece with lightweight composite materials, without the need to build ribs. Under normal operating conditions, we believe there are no structural safety concerns. However, if a small size bomb were detonated, the lack of protection provided by ribs in the hull could result in a larger hole; this could be rather problematic.

The mathematical modeling for an inside-cabin explosion of a large passenger airplane flying at a high altitude remains a challenging task. For Daallo Flight 159, the altitude of 4,300m was only a fraction of the cruising altitude. It is our understanding that there is not yet an accepted

theoretical model for decompression collapse of an airframe structure; building a fundamental model allowing the exchange of airflows between the inside and the outside of airplane cabin once the hull is breached would constitute the most important step. This model must be dynamic, because the high flight and wind speeds tend to open up the hole and eventually lead to the breakup of the fuselage.

Our computations for a refined model in section 3.3 of Chapter 3, even though ribs have been built in, still do not constitute a full-size airplane. We don't expect that this would cause a large discrepancy in numerical computations and outcomes. However, one should still use as complete an airplane model as possible. The building of such a model is important, because we can use this refined model to compute problems in this dissertation and for a large class of other similar problems in aviation involving explosions and impacts. This will be part of our future research.

4. SUMMARY

In this thesis, two problems are solved by computational mathematics and mechanics. One is the work of shape optimization and the other is the simulation of Daallo Airline laptop bombing.

We use OpenFOAM to work on shape optimization which is important in many engineering design problems. After validate there is little difference between mesh refinement as long as the grid is sufficiently small, the author evolves the shape when satisfying given constraints and tracking the shape to find when it has the minimum cost function. The author can achieve 3D time-varying domains which never been done before and we can expect to use it for more complex domain latter and might able to apply to the real world problems.

We use LS-DYNA to simulate the Daallo Airline laptop bombing case. Our computer simulation has not included wind speed. Our aircraft is only damaged by the blast particle and there's no wind fore to drag the fragment to enlarge the size of the hole. The aircraft model does not contain seats, engine, and fuel tank. Therefore, we would not consider the case if the bomb explosion near the engine or fuel tank (which might cause larger damage). Although we believe there would not cause a large discrepancy in numerical computations and outcomes if we use a full-size airframes airplane. Our simulations still need to be improved to be close to the real case.

The computational simulation is favorable to saving long and expensive processes of laboratory setup and measurements. It is easier for mathematicians to conduct interdisciplinary collaboration with physicists and engineers for the modeling and computation of real-world problems when there are more abundant free and open-source computational tools and user-friendly software been developed. However, there still have challenges such as get the perfect quality of mesh or pick the good-enough model when the situation is not ideal. The computational research sometimes can become very empirical and require lots of trial-and-error. Under these circumstances, the mathematical analysis might able to lead some best practice guidelines and give more insights.

REFERENCES

- [1] T. Wyke and G. M. for MAILONLINE, “Laptop bomb used to blow a hole in somalian plane was so sophisticated it managed to get through x-ray scanners undetected.” <http://www.dailymail.co.uk/news/article-3443756/Laptop-bomb-used-blow-hole-Somalian-plane-sophisticated-managed-X-ray-scanners-undetected.html>, (accessed December 02, 2017).
- [2] S. Tomlinson and O. N. for MAILONLINE, “Man sucked out of passenger jet after bomb exploded was suicide bomber who smuggled his device on board in his wheelchair, claim investigators.” <http://www.dailymail.co.uk/news/article-3431295/Islamist-terror-group-Al-Shabaab-named-prime-suspects-bomb-blast-tore-hole-Somali-plane-sucked-passenger-death.html>, (accessed December 02, 2017).
- [3] J. Z. for TIME, “Man sucked from somali jet may have been suicide bomber.” <http://time.com/4208448/suicide-bomber-somalia-flight/>, (accessed December 02, 2017).
- [4] M. Delfour, G. Payre, and J.-P. Zolésio, “Optimal design of a minimum weight thermal diffuser with constraint on the output thermal power flux,” *Applied Mathematics and Optimization*, vol. 9, pp. 225–262, 1983.
- [5] J. Sokolowski and J.P.zolesio, “Introduction to shape optimization,” *Heidelberg : Springer-Verlag*, 1992.
- [6] G. Chen, J. Zhou, and R. C. McLean, “boundary element method for shape (domain) optimization of linear quadratic elliptic boundary control problems,” *Boundary Control and Variation*, 1994.
- [7] *OpenFOAM*. <http://www.openfoam.com/>,.

- [8] J. L. L. (translated by S.K. Mitter), “Optimal control of systems governed by partial differential equations,” *New York : Springer-Verlag*, 1971.
- [9] G. Chen, C. Gu, H. Hajaiej, P. J. Morris, E. G. Paterson, and A. Sergeev, “Openfoam computation of interacting wind turbine flows: the free rotating case,” 2015.
- [10] R. E. Bensow and G. Oran Bark, “Simulating cavitating flows with les in openfoam,” *V European Conference on Computational Fluid Dynamics*, 2010.
- [11] H. Zhang, C. Gu, C.-M. Yang, J. Yeh, and J. Jiang, “Positive solutions for the kirchhoff-type problem involving general critical growth - part ii: 3d numerical solutions,” *Journal of Mathematical Analysis and Applications*, vol. 460, no. 1, 2018.
- [12] C. Gu, C.-M. Yang, T.-C. Lin, J. Yeh, and H. Zhang, “Positive solutions for the kirchhoff-type problem involving general critical growth - part i: Existence theorem involving general critical growth,” *Journal of Mathematical Analysis and Applications*, vol. 460, no. 1, 2018.
- [13] nogenmyr, *blockMeshBodyFit*. Available from the github website, <https://github.com/nogenmyr/blockMeshBodyFit>.
- [14] Wikipedia, *Timeline of airliner bombing attacks*. https://en.wikipedia.org/wiki/Timeline_of_airliner_bombing_attacks.
- [15] Mythbusters, *Explosive Decompression*. <https://www.discovery.com/tv-shows/mythbusters/>.
- [16] Livermore Software Technology Company, *LS-DYNA*. <https://www.lstc.com/products/ls-dyna>.
- [17] A. Arriaga, R. Pagaldai, A. M. Zaldua, A. Chrysostomou, and M. O'Brien, “Impact testing and simulation of a polypropylene component. correlation with strain rate sensitive constitutive models in ansys and ls-dyna,” *Polymer Testing*, vol. 29, pp. 170–180, 2010.

- [18] E. Rackauskaite, P. Kotsovinos, and G. Rein, “Model parameter sensitivity and benchmarking of the explicit dynamic solver of ls-dyna for structural analysis in case of fire,” *Fire Safety Journal*, vol. 90, pp. 123–138, 2017.
- [19] J. Fehr, F. Kempfer, C. Kleinbach, and S. Schmitt, “Guiding strategy for an open source hill-type muscle model in ls-dyna and implementation in the upper extremity of a hbm,” tech. rep., IRCOBI conference, 2017.
- [20] G. Chen, C. Gu, P. J. Morris, E. G. Paterson, A. Sergeev, Y.-C. Wang, and T. Wierzbicki, “Malaysia airlines flight mh370: Water entry of an airliner,” *Notices of the American Mathematical Society*, vol. 62, no. 4, pp. 300–334, 2015.
- [21] G. Chen, Y.-C. Wang, A. Perronnet, C. Gu, P. Yao, B. Bin-Mohsin, H. Hajaiej, and M. O.Scully, “The advanced role of computational mechanics and visualization in science and technology: analysis of the germanwings flight 9525 crash,” *Physica Scripta*, vol. 92, 2017.
- [22] *LS-DYNA Theory Manual*. <https://www.dynasupport.com/manuals>.
- [23] D. L. Shirey, “Breaching of structural steel plates using explosive disks,” tech. rep., The Shock and Vibration Bulletin, 1980.
- [24] J. A. Zukas and W. Walters, *Explosive Effects and Applications*. Springer, 2013.
- [25] R. Forsen, “Experiments used for comparison of blast damage to full-scale and one fourth scale reinforced concrete structures,” tech. rep., 24th DoD Explosives Safety Seminar, Saint Louis, ML., 1990.
- [26] L. Lonnqvist, “The effects of high explosives in contact with reinforced concrete slabs,” tech. rep., Proceedings of the Sixth International Symposium on Interaction of Nonnuclear Munitions with Structures, Panama City Beach, FL., 1993.

APPENDIX A

PART OF THE K-FILE

A few segments of the input k-file for the LS-DYNA simulation:

```

*CONTROL_CONTACT
$#  slsfac  rwpnal  islchk  shlthk  penopt  thkchg  orien  enmass
    0.1      0.0      1        0        1        0        1        2
$#  usrstr  usrfrc  nsbcs  interm  xpene  ssthk  ecdt  tiedprj
    0        0        0        0        4.0      0        0        0
$#  sfrc   dfrc   edc    vfc    th    th_sf  pen_sf
    0.0     0.0     0.0    0.0    0.0    0.0    0.0
$#  ignore  frceng  skiprwg  outseg  spotstp  spotdel  spothin
    0        0        0        0        0        0        0.0
$#  isym   nserod  rwgaps  rwgdt  rwksf  icov   swradf  ithoff
    0        0        1        0.0    1.0    0      0.0     0
$#  shldg  pstiff  ithcnt  tdcnof  ftall  unused  shltrw
    0        0        0        0        0      0      0.0

*CONTROL_SPH
$#  ncbs   boxid   dt      idim   memory  form   start  maxv
    1      01.00000E20  3      150   0      0.01.00000E15
$#  cont   deriv   ini     ishow  ierod  icont  iavis  isymp
    0      0        0      0      0      0      0      100
$#  ithk
    0

$
*CONTROL_ENERGY
$#  hgen   rwen   slnten  rrlen
    2      2      1      1

```

The part for the passenger cabin: the material is plastic and the thickness for the skin is 1.2 mm.

```

*PART
$#                                     title
inner fuselage
$#  pid   secid   mid   eosid   hgid   grav   adpopt  tmid
    2     1     2     0     1     0     0     0
*MAT_PLASTIC_KINEMATIC_TITLE
plastic
$#  mid   ro     e     pr   sigy   etan   beta
    21.40000E-6  1.4  0.39  0.016  0.0   1.0
$#  src   srp   fs    vp
    0.0   0.0   0.067  0.0

```

The part for the fuselage: the material is Aluminum Alloy 7075-T6 and the thickness for the skin is also 1.2 mm.

```

*PART
$#                                     title
fuselage
$#   pid      secid      mid      eosid      hgid      grav      adpopt      tmid
      1         1         1         0         1         0         0         0
*MAT_PLASTIC_KINEMATIC_TITLE
Aluminium Alloy 7075-T6
$#   mid      ro         e         pr         sigy      etan      beta
      13.00000E-6  70.0      0.35     0.47      0.0       0.11
$#   src      srp         fs         vp
      0.0       0.0       0.3       0.0
*SECTION_SHELL
$#   secid    elform      shrf      nip      propt    qr/irid    icomp      setyp
      1         2      0.833      5        1.0       0         0         1
$#   t1       t2       t3       t4      nloc     marea     idof      edgset
      1.2     1.2     1.2     1.2     0.0       0.0       0.0       0
*HOURLGLASS
$#   hgid      ihq      qm      ibq      q1      q2      qb/vdc      qw
      1         4      0.1      0        1.5     0.06     0.1       0.1

```

The part for the ribs: the material is Aluminum Alloy 2024 and the thickness for the skin is 6 mm.

```

*PART
$#                                     title
ribs
$#   pid      secid      mid      eosid      hgid      grav      adpopt      tmid
      7         2         7         0         1         0         0         0
*MAT_SIMPLIFIED_JOHNSON_COOK_TITLE
2024_aluminum for the stringers and ribs
$#   mid      ro         e         pr         vp
      72.77000E-6  70.0      0.33     0.0
$#   a         b         n         c      psfail    sigmax    sigsat    epso
      26.5     42.6     0.41     0.011.00000E171.00000E281.00000E28  1.0
*SECTION_SHELL
$#   secid    elform      shrf      nip      propt    qr/irid    icomp      setyp
      2         2      0.833      5        1.0       0         0         1
$#   t1       t2       t3       t4      nloc     marea     idof      edgset
      6.0     6.0     6.0     6.0     0.0       0.0       0.0       0

```

The part for the TNT explosive and its JWL equation.

```

*PART
$#                                     title
HighExp
$#   pid      secid      mid      eosid      hgid      grav      adpopt      tmid
      3         3         3         3         0         0         0         0
*SECTION_SPH
$#   secid      cslh      hmin      hmax      sphini      death      start
      3         1.2       0.2       2.0       0.01.00000E20  0.0
*MAT_HIGH_EXPLOSIVE_BURN_TITLE
HighExp
$#   mid      ro      d      pcj      beta      k      g      sigy
      31.63000E-6  6930.0  21.0    0.0    0.0    0.0    0.0
*EOS_JWL_TITLE
HIGHEXP
$#   eosid      a      b      r1      r2      omeg      e0      vo
      3         371.2  2.231  4.15  0.95  0.3    7.0    1.0

```

The part for the windows: the material is Acrylic and the thickness for the skin is also 1.2 mm.

```

*PART
$#                                     title
window
$#   pid      secid      mid      eosid      hgid      grav      adpopt      tmid
      4         1         4         0         1         0         0         0
*MAT_PLASTIC_KINEMATIC_TITLE
Acrylic
$#   mid      ro      e      pr      sigy      etan      beta
      41.20000E-6  3.2    0.37  0.071  0.0    0.0
$#   src      srp      fs      vp
      0.0     0.0    0.067  0.0

```

Add the erosion effect on the airplane, so the airplane will crack after blasting.

*CONTACT_ERODING_SINGLE_SURFACE_MPP

cid	title
ignore	cparm8
2	0
ssid	mpr
1	0
fs	dt
0.0	0.01.00000E20
sfs	vsf
1.0	1.0
isym	
0	
soft	frcfrq
1	1
penmax	sldstf
0.0	0.0

Let the blast SPH work on the airplane

*CONTACT_AUTOMATIC_NODES_TO_SURFACE_MPP

cid	title
ignore	cparm8
2	0
ssid	mpr
3	0
fs	dt
0.0	0.01.00000E20
sfs	vsf
1.0	1.0
soft	frcfrq
1	1
penmax	sldstf
0.0	0.0

Use TIEBREAK so the ribs will weld on the fuselage.

*CONTACT_TIEBREAK_SURFACE_TO_SURFACE_ONLY

cid	title
ssid	mpr
4	0
fs	dt
0.3	0.01.00000E20
sfs	vsf
1.0	1.0
nfls	
1.0	
soft	frcfrq
0	1
penmax	sldstf
0.0	0.0

To LOAD 8 psi uniformly only on the surface of aircraft fuselage.

```
*DEFINE_CURVE_TITLE
8 psi
$#   lcid   sidr   sfa   sfo   offa   offo   dattyp   lcint
      1     0     1.0   1.0   0.0    0.0     0         0
$#           al     o1
      0.0   5.5158059695e-05
      0.1   5.5158059695e-05
      0.11   0
      20.0   0

*LOAD_SHELL_SET_ID
$#   id                                     heading
      1
$#   esid   lcid   sf   at
      1     1   -1.0  0.0
```

Use TIED so the windows and the fuselage will glued together.

```
*CONTACT_TIED_SURFACE_TO_SURFACE
$#   cid                                     title
$#   ssid   msid   sstyp   mstyp   sboxid   mboxid   spr   mpr
      7     1     3     3     0     0     0     0
$#   fs     fd     dc     vc     vdc     penchk   bt     dt
      0.8   0.0   0.0   0.0   0.0     0     0.01.00000E20
$#   sfs     sfm     sst     mst     sfst     sfmt     fsf     vsf
      1.0   1.0   5.0   5.0   1.0     1.0     1.0     1.0
$#   soft   sofsc1  lcidab  maxpar  sbopt   depth   bsort   frcfrq
      0     0.1     0     1.025   2.0     2     0     1
$#   penmax thkopt  shlthk  snlog   isym   i2d3d   sldthk  sldstf
      0.0   0     0     0     0     0     5.0     0.0

*CONTACT_TIED_SURFACE_TO_SURFACE_ID
$#   cid                                     title
$#   ssid   msid   sstyp   mstyp   sboxid   mboxid   spr   mpr
      7     2     3     3     0     0     0     0
$#   fs     fd     dc     vc     vdc     penchk   bt     dt
      0.8   0.0   0.0   0.0   0.0     0     0.01.00000E20
$#   sfs     sfm     sst     mst     sfst     sfmt     fsf     vsf
      1.0   1.0   5.0   5.0   1.0     1.0     1.0     1.0
$#   soft   sofsc1  lcidab  maxpar  sbopt   depth   bsort   frcfrq
      0     0.1     0     1.025   2.0     2     0     1
$#   penmax thkopt  shlthk  snlog   isym   i2d3d   sldthk  sldstf
      0.0   0     0     0     0     0     5.0     0.0
```