Implementation of Special Equipment Design and Verification System Based on OPENCASCADE

A Thesis in Software Engineering

Presented to

The Faculty of the

**College of Computer Science and Technology, Zhejiang University of Technology**

In Fulfillment of the Requirements for the Degree Bachelor of Science in Software Engineering

By

Kaicheng Lou

June, 2021

## 摘要

计算机辅助设计(CAD)是指使用计算机(或者工作站)中的创建,修改,分析或者优化以此帮助设计.CAD软件系统通常被用来提高设计人员的产品设计效率,提高设计质量并且通过文档改善和创造制造数据库.随着计算机科学技术的发展和兴起,CAD已经成为一种重要的工业技术,并且被广泛应用于各种行业比如汽车,造船和航天航空工业,工业和建筑设计等.除了制造业之外,CAD还经常被用于制作电影,广告和技术手册中具有特殊效果的计算机动画.CAD强大的功能性和经济重要性使得它成为许多行业的必不可少的设计软件.CAD软件起源于美国麻省理工学院的交互式图形计划,在经过六十多年的蓬勃发展之后,CAD在国外的应用已经相当成熟.而国内对于3D CAD技术的研究应用相比之下,还存在着许多问题.

本文将基于OpenCASCADE(OCC)几何内核,在分析OpenCASCADE三维造型内核的开发模式和数据结构基础上,开发一款具有特定功能的压力容器设计软件,以模型数据和算法为基础调用OpenCASCADE造型函数,结合PyQT5打造可视化界面并将建模结果显示出来,针对开孔筒体和普通筒体可以进行参数化设计和调整.对于设计出的模型,根据输入的参数进行校核流程,校核条件和计算流程公式依照附录给出的条件

**关键词:OpenCASCADE, PV Elite, Modeling**

## Abstract

CAD is to help design by using the creation, modification, analysis or optimization of computer (or workstation). CAD software system is usually used to improve the product design efficiency of designers, improve the design quality and create manufacturing database through documents. With the development and rise of computer science and technology, CAD has become an important industrial technology, And it is widely used in various industries such as automobile, shipbuilding and aerospace industry, industry and architectural design. Besides manufacturing, CAD is often used in film making, The powerful function and economic importance of CAD make it an essential design software for many industries. CAD software originated from the interactive graphics program of MIT, after more than 60 years of development of the boom, CAD has been used in foreign countries quite mature, but there are still many problems in the research and application of 3D CAD technology in China

Based on the OpenCASCADE (OCC) geometric kernel, based on the analysis of the development mode and data structure of OpenCASCADE 3D modeling core, a Pressure vessel design software with specific functions is developed. The OpenCASCADE modeling function is called based on the model data and algorithm. The visualization interface is created and the modeling results are displayed in combination with pyqt5, For the open hole cylinder and ordinary cylinder, parametric design and adjustment can be carried out. For the designed model, the check process is carried out according to the input parameters, and the checking conditions and calculation flow formula are in accordance with the conditions given in the appendix

**Keywords: OpenCASCADE, PvElite, modeling**

Content

[摘要 2](#_Toc1365)

[Abstract 3](#_Toc8513)

[CHAPTER 1. Introduction 5](#_Toc2004)

[1.1 CAD Understanding 5](#_Toc19977)

[1.2 Origin of CAD 5](#_Toc21197)

[1.4 Literature Review 7](#_Toc11980)

[1.4.1 OpenCASCADE 7](#_Toc19100)

[1.4.2 OCAF 9](#_Toc23479)

[1.4.3 QT Framework 11](#_Toc24309)

[1.5 Code for design of pressure vessels 12](#_Toc17664)

[1.6 Conclusion 13](#_Toc18748)

[CHAPTER 2. Problem and Requirement 13](#_Toc18524)

[2.1 Problem Analysis 13](#_Toc23220)

[2.1.1 Problem Motivation 13](#_Toc6902)

[2.2 Requirement Specification 14](#_Toc16639)

[2.2.1 Purpose 14](#_Toc29802)

[2.2.2 Scope 14](#_Toc7108)

[2.2.3 Definitions, Acronyms, and Abbreviations 14](#_Toc2643)

[2.2.4 Vision 15](#_Toc21632)

[2.2.5 Functional Perspective 15](#_Toc10590)

[2.3 Modeling 18](#_Toc29122)

[CHAPTER 3. Experiments 22](#_Toc24593)

[3.1 Parameters 23](#_Toc10595)

[3.1.1 Cylinder Parameters 23](#_Toc4256)

[3.1.2 Perforated Cylinder Parameters 23](#_Toc20865)

[3.2 Modeling 24](#_Toc10882)

[3.2.1 Cylinder 24](#_Toc3650)

[3.2.2 Perforated Cylinder 25](#_Toc4522)

[3.3 Comparison of experimental data 25](#_Toc19108)

[3.3.1 Comparison of Cylinder Data 26](#_Toc10385)

[3.3.2 Comparison of Perforated Cylinder Data 27](#_Toc15594)

[3.4 Experiment Data Conclusion 29](#_Toc23405)

[CHAPTER 4. Conclusion 30](#_Toc31596)

[CHAPTER 5. References 31](#_Toc15948)

[APPENDIX 32](#_Toc3272)

[1.内压圆柱形壳体开孔补强计算程序 32](#_Toc7408)

[2.筒体壁厚计算 36](#_Toc30414)

## CHAPTER 1. Introduction

### 1.1 CAD Understanding

CAD(Computer Aided Design) is mean that help design product by using computer’s analysis and optimization function.CAD software is used to improve the productivity of designers, improve design quality, improve communication through documents and create manufacturing database.[1]CAD could be used to design curves and graphics in two-dimensional(2D) space or curves ,surface and solids in three-dimensional(3D) space.CAD is a major industrial technology which is widely applied to many fields including automobile, shipbuilding and aerospace industry, industrial and architectural design so on. And it’s also used to make computer effect in movies, advertisement and technical manuals , commonly known as DCC digital content creation.Because of its great economic importance, CAD has always been the main driving force for the research of computational geometry, computer graphics(including hardware and software) and discrete different geometry.[2]Nowadays, CAD has become a very important technology in the field of computer aided technology, which has the advantages of reducing the cost of product development and greatly shortening the design cycle. CAD system enables designers to lay out and develop work on the screen, print and save it for future editing, thus saving drawing time.

### 1.2 Origin of CAD

Between the mid-1940s and 1950s, various developments were made in computer software. Some of these developments include servo-motors controlled by generated pulse , a digital computer with built-in operations to automatically coordinate transforms to compute radar related vectors , and the graphic mathematical process of forming a shape with a digital machine tool .[3]

CAD appeared in 1960s, which is a research project of interactive graphics proposed by MIT.In 1953, Douglas T. Ross, a researcher at the Massachusetts Institute of technology, saw the "interactive display device" being used by radar operators and thought that it was just what his sage related data reduction team needed.Ross and other researchers at MIT's Lincoln lab are the only users of complex display systems installed for sage's pre Cape Cod system. Rose claimed in an interview that they "use it for their own personal workstations."These computer designers built utilities to ensure that programmers could debug software using flowcharts in the display area, and turn on and off logic switches during debugging.They found it’s possible to create electronic symbols and geometry to create simple circuit diagrams and flowcharts. [4] These programs allow objects to be copied if necessary,they can also change their orientation, link (flux, mechanical, lexical scope) or scale. This brings them countless possibilities.

The invention of CAD is attributed to the French engineer Pierre Bézier.From 1966 to 1968, after completing mathematical work on surfaces, he developed Unisurf to simplify the design of parts and tools for the automotive industry. Then, Unisurf becomes the foundation of the next generation CAD software.

With the growing of IBM drafting system in the mid-1960s, Computer Aided Design began to provide more function, not just the function of reproducing manual drawing with electronic drawing.In order to lower the cost, some companies switched to use CAD system.The first commercial application of CAD was in large companies in the automotive and aerospace industries as well as in the electronic field.This is because only large companies can have enough computers that can perform the necessary calculations.With the price of computer becoming cheaper and cheaper, the application of CAD gradually expands to new fields. The development of CAD software for personal desktop computer is the driving force for its universal application in almost all construction fields.

**1.3 Historical status at home and abroad**

CAD is the abbreviation of computer-aided design. It is a kind of technology that makes use of the graphics processing ability and computing ability of computer to assist technical personnel in the design and analysis of engineering or products, complete the construction of target products, and achieve innovation.In the early 1960s, Ivan Sutherland, a doctor of Massachusetts Institute of technology, developed the world's first interactive graphics system Sketchpad which uses light pen to interactive.But at the time,graphic equipment was very expensive, only large companies with strong financial resources could develop and use this technology.As the basis of CAD technology, computer graphics has developed rapidly in this period.In the 1970s, the CAD system based on minicomputer was released.At the same time, the graphics software and CAD application supporting software were constantly enriched and improved, and the graphics equipment, such as raster scanning display, graphics input board, plotter, etc. were launched and improved one after another.After that, CAD commercialization system for small and medium-sized enterprises has been pushed.

In foreign countries, after 50 years of development, CAD technology’s level has reached a quite mature stage. Japan, France and Germany have also widely used CAD technology in machinery manufacturing, aerospace, automotive industry, construction and chemical industry. CAD technology has become an important pillar of national economy in developed countries.In China, the application of CAD technology started in the 1960s. After more than 40 years of research, development and application, CAD technology has been widely used in various industries.But from a comprehensive point of view, there are still many problems in the application of CAD technology in China.There are many problems in our country's CAD technology from 2D CAD to 3D CAD, such as less development innovation and more imitation. And parametric design and data management problems also need to be solved.These problems make some domestic CAD software have some problems in data exchange and product design.Our information integration technology also lags behind foreign countries. The extensive integration of information technology is based on product data management (PDM) and process management (PM) to realize the organic integration of CAD / CAPP / CAM and ERP. PDM is also an important foundation in concurrent engineering. The research in this field in China is only in the initial stage, Our CAD integration technology is only based on the basic system of foreign countries. As for the standardization of CAD technology, our country has not put forward a unified standard for CAD because of backward technology and insufficient capital investment.So many domestic enterprises use foreign CAD software, but these software are expensive and some functions are not used at all.

Generally speaking, there is still a big gap in the research and development of CAD technology between China and foreign countries

### 1.4 Literature Review

#### 1.4.1 OpenCASCADE

Nowadays , most of developments of CAD software are based on Geometric kernel.For example, AutoCAD is based on ACIS kernel and Pro/E,UG,Solid Works are based on Parasolid kernel.

Actually, the geometric kernel is a kind of class library which defines the storage format of graphic data and a lot of graphic algorithms.OpenCASCADE is a open-source geometric kernel.Open cascade is a set of open source CAD / CAM / CAE geometric model core, which originated from France's Matra Datavision company. In 1999, Datavision announced the source code of open cascade, which can be downloaded and used by anyone on the company's website.

Open cascade is a powerful 3D modeling tool, which provides the display and interactive operation of point, line, surface, body and complex body. After deep development, it can realize the graphic operations such as texture, lighting, element filling and rendering, and dynamic operations such as zooming in, zooming out, rotating, roaming, simulating flight and simulating crossing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Foundation Classes | Modeling Data | Modeling Algorithms | Visualization | Data Exchange | Application Framework |
| Kernel Classes  Math Utilities | 2D Geometry  3D Geometry  Geometry Utilities  Topology | Construction of Primitives  Boolean Operations  Fillets and Chamfers  Offsets, Drafts Sewing and Sweeps | Services Common to 2D and 3D  2D Visualization  3D Visualization | IGES  STEP AP203 AP214  Extended data  exchange (XDE) | Data Framework  Data Storage  Application Desktop |

Chart 1-Content of Class Library

1. **Foundation Classes**:

The basic class module provides a series of basic services, such as defining entity type, root class, memory management, exception handling, array operation, etc

1. **Modeling Data**:

The module provides data structure definitions of various 2D and 3D geometric models. The geom2d class package collects various 2D model data structure classes, and each data class name is prefixed with geom2d\_ Various data structure classes of 3D model are collected in geom class package, and each data class is prefixed with geom\_ , such as geom\_ Curve defines three-dimensional curve and provides various corresponding operation methods, including parameter space

1. **Modeling Algorithms**:

(1) Geometric tools: provide intersection calculation between 2D curves, 3D curves and surfaces, projection calculation from point to curve, point to surface and curve to surface

(2) Topology tool: it provides local and global property definition of topological body, geometric transformation operation and conversion operation from topological body to NURBS geometric expression

(3) Basic geometric structure: mainly including prism, cylinder, cone and torus

(4) Feature operation: provide shape and mechanical feature operation based on existing topological body, such as adding cylinder, draft, stretching, feature rotation, tube body, etc

1. **Visualization**

provides interactive services for displaying geometry in 3D Viewer; implements a compact OpenGL / OpenGL ES renderer, supporting conventional Phong, real-time PBR metal-roughness shading models as well as interactive Ray-Tracing/Path-Tracing engine.

1. **Data Exchange**

provides possibility to import/export various CAD formats.

STEP, IGES, glTF, OBJ, STL and VRML are supported natively.[5] Other formats can be imported by using plug-ins.[6] Extended Data Exchange (XDE) components rely on a unified XCAF document definition, which includes an assembly structure of CAD shapes, color/name/material/metadata/layer attributes as well as other supplementary information like PMI.

1. **Application Framework**

offers solutions for handling application-specific data.

#### 1.4.2 OCAF

OCAF(the Open CASCADE Application Framework) is an easy-to-using platform which is used to rapidly developing sophisticated domain-specific design application.Typical applications developed involve trade specific CAD systems,manufacturing or analysis applications,simulation applications or illustration tools.

The ready-to-use solution provided by OCAF make users could concentrate on the functionality specific for application.And the underlying mechanisms required to support the application are already provided.So the final application could be developed by industrializing the prototype and there is no necessary to restart the development from scratch.

OCAF provides many models:Graphical user interface framework, Modeling data, Modeling algorithm, Visualization module,etc.GUI framework is a series of tools which can be used to design system with application framework.Modeling data mainly includes data structure of three-dimensional geometric model and topological model.Modeling algorithm provides a series of topology algorithms needed in the process of modeling.Users can quickly build the model based on the algorithm provided.

The relationship between OCAF and the Open CASCADE Technology (OCCT) Object Libraries can be seen in the image below.

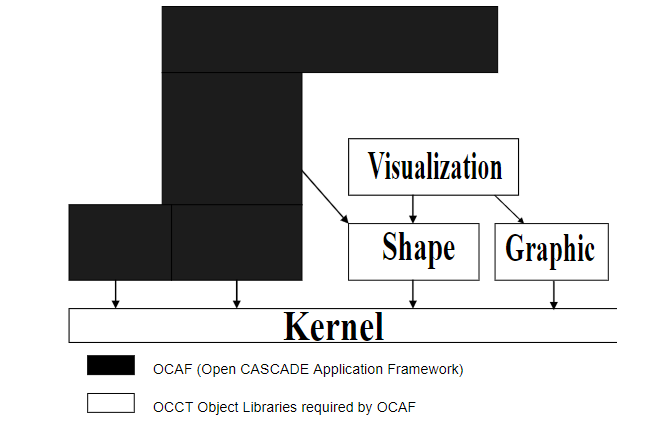


Figure-1 The relationship between OCAF and the OCCT

Architecture Overview

OCAF provides you with an object-oriented Application-Document-Attribute model consisting of C++ class libraries.

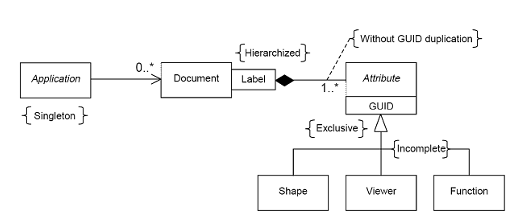


Figure2-The Application-Document-Attribute model

1. **Application**:

The Application is an abstract class in charge of handling documents during the working session, namely:

* Create documents
* Save documents and open
* Initializing document views

1. **Document**:

Documents implemented by the concrete class document are containers for application data. Documents provide access to the data framework and are used for the following purposes::

* Manage the notification of changes
* Update external links
* Manage the saving and restoring of data
* Store the names of software extensions.
* Manage command transactions
* Manage Undo and Redo options.

1. **Attributes:**

Application data is described by Attributes,which are instances of classes derived from the Attribute abstract class, organized according to the OCAF Data Framework.

OCAF includes many attributes:

* Standard attributes,it is allowed to operate with simple general data (such as integer, real number, string, array type data) in the data framework, realize auxiliary functions (such as label source attribute of the offspring of label counter), and create dependencies (such as reference, tree node)
* Shape attributes contain the geometry of the whole model or its elements including reference to the shapes and tracking of shape evolution;
* Other geometric attributes such as Datums (points, axis and plane) and Constraints (tangent-to, at-a-given-distance, from-a-given-angle, concentric, etc.)
* User attributes, that is, attributes typed by the application
* Visualization properties allow the viewer information to be placed in the data frame, the visual representation of objects, and other auxiliary visual information, which is necessary for graphic data representation.
* Function services — The purpose of these attributes is to reconstruct the object after modifying it (parameterization of the model). At the same time that the document manages change notifications, the function manages the propagation of these changes. Function mechanism provides links between functions and calls to various algorithms.

#### 1.4.3 QT Framework

Qt is a widget toolkit for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms such as Linux, Windows, macOS, Android or embedded systems with little or no change in the underlying codebase while still being a native application with native capabilities and speed.

Qt is used for developing graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms. Most GUI programs created with Qt have a native-looking interface, in which case Qt is classified as a widget toolkit. Also non-GUI programs can be developed, such as command-line tools and consoles for servers. An example of such a non-GUI program using Qt is the Cutelyst web framework.[7]

**Qt concepts:**

-Complete abstraction of the GUI

When QT was released for the first time, it used its own rendering engine and controls to simulate the appearance of different platforms when drawing widgets. This makes porting easier because few classes in QT really depend on the target platform

-Signals and slots

QT introduces a language construction for communication between objects [8], which makes it easy to implement observer pattern and avoid template code. The concept is that GUI widgets can send signals containing event information that can be received by other controls using a special feature called slots.

-Meta Object compiler

The meta object compiler called MOC is a tool that runs on the source code of QT program. It interprets some macros from C + + code as comments and uses them to generate additional C + + code with meta information about the classes used in the program. QT uses this meta information to provide programming functions that C + + does not have: signals and slots, introspection and asynchronous function calls

-Language bindings

Qt can be used in several programming languages other than C++, such as Python, Javascript, C# and Rust via language bindings

Qt support various compilers including GCC C++ compiler.Qt also provide QT quick which includes a declarative scripting language called QML.The language allows you to use JavaScript to write logic.It has to be said that strong cross platform capability and good compatibility make QT widely used in various programming.

### 1.5 Code for design of pressure vessels

There are two kinds of pressure vessel design methods, corresponding to two kinds of design specifications.China's GB150 adopts the traditional design method, which is called design according to rules. Its basic idea is that it does not need to carry out detailed stress analysis on each part of the pressure vessel, but combines the classical mechanical theory and empirical formula to make some regulations on the design of pressure vessel components, such as material selection, safety factor, characteristic size, etc.

Another design specification is JB4732, which adopts the method of design analysis, which requires stress analysis and fatigue analysis of pressure vessels. Because of the quantitative analysis results, the structure tends to be reasonable, therefore, the vessels designed with the specification can achieve higher allowable stress without weakening safety margin.

In general, the design method according to rules is simple, easy to be mastered by designers and has certain experience. The design method according to analysis puts forward higher requirements for designers. Designers must have certain ability of stress analysis and master the finite element software and other stress analysis means. The core of the analysis and design process is to do the necessary stress analysis for components, Then according to the results of stress analysis, the stress classification is checked, and then the optimization is carried out according to the check results

### 1.6 Conclusion

The aim of CAD(Computer Aided Design) is to help designer design target product more efficiently in a shorter period, improve design quality, improve communication through documentation and create manufacturing database.And with the continuous innovation of modern technology, CAD has been widely used in various industries.Now most CAD design software on the market is based on geometric kernel.As an open source geometric kernel, the open source and powerful function library of OCC make it the first choice for many software designs.With the help of OCAF(the Open CASCADE Application Framework),we could use OpenCASCADE to develop a powerful CAD design system with specific target function.Moreover, we could design a user-friendly interface by using QT framework.The framework and software mentioned above are of great reference for the design of new CAD software system in the market.

## CHAPTER 2. Problem and Requirement

In this chapter, the problem needs to be solved is defined and formulated in section 3.1. And the solution is given to figure it out in section 3.2.

### 2.1 Problem Analysis

This section will provide a brief analysis of the problem that needed to be solved and needs of requirement.

#### 2.1.1 Problem Motivation

According to the first chapter above, with the rise and development of computer science and technology, CAD system has more and more significant influence on the production and manufacturing of various industries. In foreign countries, CAD technology has become one of the indispensable technologies for designers' daily work, and has also become the economic pillar of various industries. In China, the application of CAD technology is becoming more and more popular, However, there is a big gap between China's CAD technology research and that of foreign countries. Most enterprises have to pay a high price for using foreign genuine software although some functions are not used in actual production, they also have to pay for them,.Some small companies use the system based on the secondary development of CAD commercial software. These systems have great defects in parametric design and data management. There is an urgent need for a self-developed CAD design system in China. The above-mentioned open source geometric kernel opencascade provides us with a good direction and tool to design a CAD system with specific functions

### 2.2 Requirement Specification

#### 2.2.1 Purpose

For the description of the above existing problems, the purpose of this paper is to develop a 3D CAD design system software based on OpenCascade. The specific requirements and functions of this software are based on the analysis of the existing CAD software.

#### 2.2.2 Scope

In detail are specified the following features:

-Parametric Design:Generates the specified model based on the target parameters and model types entered by the user

-Export model report:Generate reports based on user input parameters versus acceptable part requirements in standard conditions

#### 2.2.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| OCC | OpenCASCADE, a geometric kernel to build the software |
| Parametric Design | The casting process adopts parametric design technology to avoid wasting time and energy in repeatedly drawing the same three-dimensional structure solid model. Parametric design is a parameterized modification technique that can generate, move, modify or delete any form of geometric element, thus affecting its associated elements |
| Combinatorial modeling technique | Combinatorial modeling technique is a holistic modeling technique or object grouping technique. In the process of casting process design, there are many similar structures with similar characteristics, types and shapes. According to the characteristics of these structures, they are divided into a group to complete parametric design and one-time modeling |
| Export Report | Different specifications of parts under different temperature and pressure conditions have different requirements, according to the user's input parameters and standard state for comparison, generate a report |
| Personalized positioning technology | According to the needs of designers, it is divided into automatic positioning and custom positioning. When selecting the method of automatic positioning, the designer only needs to select the face on which to place the desired element model. Then, depending on the type of design unit, the center face or other faces are set. In the custom positioning method, the designer can place the model according to his own operation and calculation habits |

Chart1-Definitions

#### 2.2.4 Vision

At present, the existing CAD design software system lacks the design of specific target parts, and designers need to adjust the parameters and design after the design of parts, which is lack of efficiency in design.

We hope that specific parts can be designed quickly according to the target parameters, so as to avoid repeatedly drawing the same structure of the three-dimensional body. In addition, we also want to be able to detect whether the input parameters of the target parts meet the requirements of the standard conditions at design time.

#### 2.2.5 Functional Perspective

a.Parametric Design

This use case diagram shows the function of Parametric Design

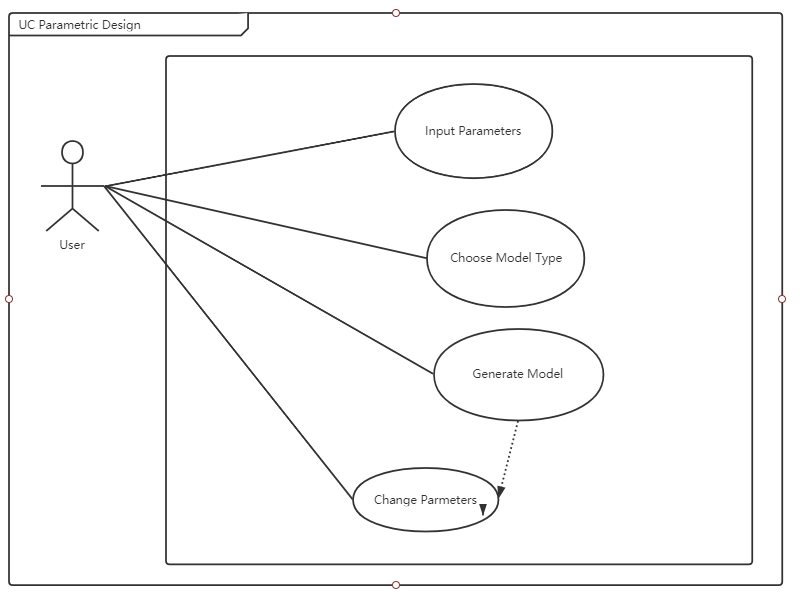


Figure3-Use Case Diagram for Parametric Design

|  |  |
| --- | --- |
| **ID and Name** | UC1.001 Input Parameters |
| **Actors** | individual |
| **Preconditions** | System launched |
| **Scenarios** | 1. individual input parameters to design model |
| **Post-conditions** | - |
| **Alternatives** | - |
| **Exceptions** | Parameter format error |

|  |  |
| --- | --- |
| **ID and Name** | UC1.002 Choose Model Type |
| **Actors** | individual |
| **Preconditions** | System launched |
| **Scenarios** | 1. individual choose model type to design |
| **Post-conditions** | - |
| **Alternatives** | - |
| **Exceptions** | - |

|  |  |
| --- | --- |
| **ID and Name** | UC1.003 Generate Model |
| **Actors** | individual |
| **Preconditions** | 1. Choose Model Type 2. Input Parameters correctly |
| **Scenarios** | 1. System generates the model based on model type and parameters |
| **Post-conditions** | - |
| **Alternatives** | - |
| **Exceptions** | Parameter format error |

|  |  |
| --- | --- |
| **ID and Name** | UC1.004 Change Parameters |
| **Actors** | individual |
| **Preconditions** | Generated Model |
| **Scenarios** | 1. individual wants to adjust model by parameters |
| **Post-conditions** | - |
| **Alternatives** | - |
| **Exceptions** | Parameter format error |

b.Export Data Report

This use case diagram shows the function of Parametric Design

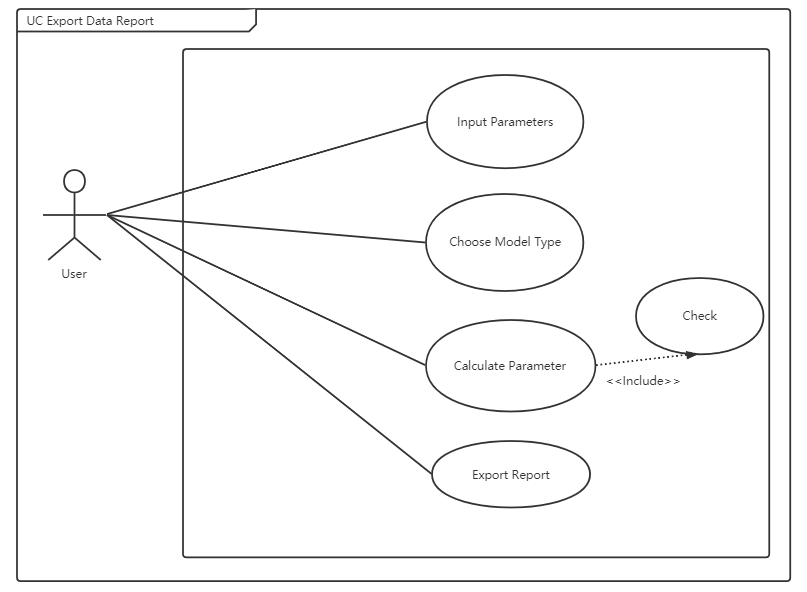


Figure4-Use Case Diagram for Export Report

|  |  |
| --- | --- |
| **ID and Name** | UC2.001 Calculate Parameter |
| **Actors** | System |
| **Preconditions** | Generated Model |
| **Scenarios** | 1. The system calculates according to the input parameters |
| **Post-conditions** | - |
| **Alternatives** | - |
| **Exceptions** | Calculation Error |

|  |  |
| --- | --- |
| **ID and Name** | UC2.002 Check |
| **Actors** | System |
| **Preconditions** | Calculate Parameters |
| **Scenarios** | 1. The system checks whether the input parameters meet the standard |
| **Post-conditions** | - |
| **Alternatives** | - |
| **Exceptions** | Calculation Error |

|  |  |
| --- | --- |
| **ID and Name** | UC2.003 Export Report |
| **Actors** | Individual |
| **Preconditions** | Calculation Correctly and Checked |
| **Scenarios** | 1. Individual wants to export model report |
| **Post-conditions** | - |
| **Alternatives** | - |
| **Exceptions** | Export Error |

### 2.3 Modeling

This section will introduce how to build the basic model

**2.3.1 Cylinder**

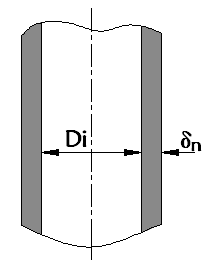


Figure5- Schematic diagram of cylinder

We will use the following code to create the basic model of the cylinder

class Cylinder(threading.Thread):  
 def \_\_init\_\_(self, R = 500, t = 30, L = 100):  
 threading.Thread.\_\_init\_\_(self):  
 self.R = R   
 self.t = t   
 self.L = L   
 self.new\_thing1 = ''  
 def run(self) -> None:  
 P1 = gp\_Pnt(-self.R, 0, 0)  
 P2 = gp\_Pnt(0, self.R, 0)  
 P3 = gp\_Pnt(self.R, 0, 0)  
 P4 = gp\_Pnt(self.R - self.t, 0, 0)  
 P5 = gp\_Pnt(0, self.R - self.t, 0)  
 P6 = gp\_Pnt(self.t - self.R, 0, 0)  
 P7 = gp\_Pnt(0, self.t - self.R, 0)  
 P8 = gp\_Pnt(0, -self.R, 0)  
  
 # In[3]:  
 Circle1 = GC\_MakeArcOfCircle(P1, P2, P3)  
 Circle2 = GC\_MakeArcOfCircle(P4, P5, P6)  
 Circle3 = GC\_MakeArcOfCircle(P4, P7, P6)  
 Circle4 = GC\_MakeArcOfCircle(P3, P8, P1)  
  
 # In[4]:  
 aEdge3 = BRepBuilderAPI\_MakeEdge(Circle1.Value())  
 aEdge4 = BRepBuilderAPI\_MakeEdge(Circle2.Value())  
 aEdge5 = BRepBuilderAPI\_MakeEdge(Circle3.Value())  
 aEdge6 = BRepBuilderAPI\_MakeEdge(Circle4.Value())  
  
 # In[5]:  
 W1 = BRepBuilderAPI\_MakeWire(aEdge3.Edge(), aEdge6.Edge())  
 W2 = BRepBuilderAPI\_MakeWire(aEdge4.Edge(), aEdge5.Edge())  
 S1 = BRepPrimAPI\_MakePrism(BRepBuilderAPI\_MakeFace(W1.Wire()).Face(), gp\_Vec(0., 0, self.L))  
 S2 = BRepPrimAPI\_MakePrism(BRepBuilderAPI\_MakeFace(W2.Wire()).Face(), gp\_Vec(0., 0, self.L))  
 # In[6]:  
 my\_cylinder = S2.Shape()  
 my\_box = S1.Shape()  
 self.new\_thing1 = BRepAlgoAPI\_Cut(my\_box, my\_cylinder).Shape()

**2.3.2 Perforated cylinder**

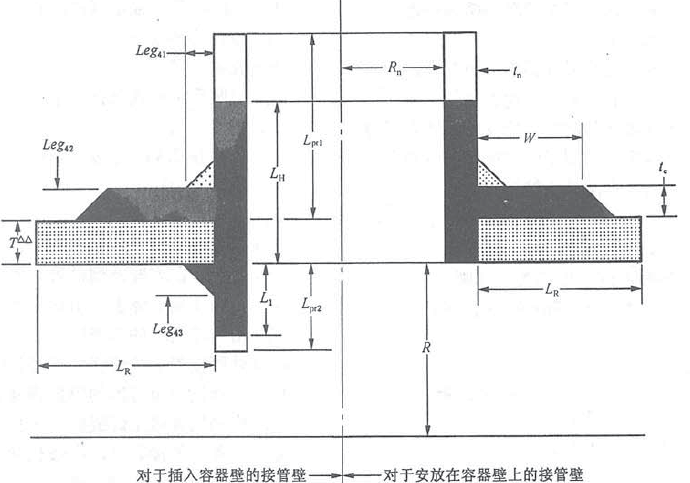


Figure6-Sketch of perforated cylinder

We will use the following code to create the basic model of the perforated cylinder

def \_\_init\_\_(self, D\_i = 1000, t = 35, l = 2000, R\_n = 100, L\_pr1 = 50, t\_n = 50, t\_n2 = 0, t\_e = 0, L\_pr2 = 0, L\_pr3 = 0, Hole\_pos = 1000):  
 threading.Thread.\_\_init\_\_(self)   
 self.D\_i = D\_i   
 self.R = D\_i / 2   
 self.t = t   
 self.l = l   
 self.r = self.R - self.t   
 if self.r <= 0:  
 self.r = 10  
  
 self.R\_n = R\_n   
 self.L\_pr1 = L\_pr1   
 self.t\_n = t\_n   
 self.t\_n2 = t\_n2   
 self.t\_e = t\_e   
 self.L\_pr2 = L\_pr2   
 self.L\_pr3 = L\_pr3   
   
 self.Hole\_pos = Hole\_pos  
 self.new\_thing0 = ''  
  
def run(self) -> None:  
 edge = BRepBuilderAPI\_MakeEdge(gp\_Pnt(self.r, 0, 0), gp\_Pnt(0, 0, 0)).Edge()  
   
 cylinder1 = BRepPrimAPI\_MakeCylinder(gp\_Ax2(gp\_Pnt(0, 0, 0), gp\_Dir(0, 1, 0)), self.R, self.l).Shape()  
 cylinder2 = BRepPrimAPI\_MakeCylinder(gp\_Ax2(gp\_Pnt(0, 0, 0), gp\_Dir(0, 1, 0)), self.r, self.l).Shape()  
 new\_thing1 = BRepAlgoAPI\_Cut(cylinder1, cylinder2).Shape()  
  
   
 my\_cylinder = BRepPrimAPI\_MakeCylinder(gp\_Ax2(gp\_Pnt(0, self.Hole\_pos, 0), gp\_Dir(1, 0, 0)), self.R\_n + self.t\_n,  
 self.l).Shape()

new\_thing2 = BRepAlgoAPI\_Cut(new\_thing1, my\_cylinder).Shape()  
  
   
 cylinder3 = BRepPrimAPI\_MakeCylinder(gp\_Ax2(gp\_Pnt(0, self.Hole\_pos, 0), gp\_Dir(1, 0, 0)), self.R\_n + self.t\_n,  
 self.r + self.t + self.L\_pr1).Shape()  
 cylinder4 = BRepPrimAPI\_MakeCylinder(gp\_Ax2(gp\_Pnt(0, self.Hole\_pos, 0), gp\_Dir(1, 0, 0)), self.R\_n,  
 self.r + self.t + self.L\_pr1).Shape()  
 new\_thing3 = BRepAlgoAPI\_Cut(cylinder3, cylinder4).Shape()  
 new\_thing = BRepAlgoAPI\_Fuse(new\_thing2, new\_thing3).Shape()  
  
 Circle = GC\_MakeCircle(gp\_Ax2(gp\_Pnt(0, 0, 0), gp\_Dir(0, 1, 0)), self.r).Value()  
 aEdge = BRepBuilderAPI\_MakeEdge(Circle)  
 aWire = BRepBuilderAPI\_MakeWire(aEdge.Edge())  
 AddFace = BRepBuilderAPI\_MakeFace(aWire.Wire())  
  
 splitter = BOPAlgo\_Splitter()  
 splitter.AddArgument(AddFace.Face())  
 splitter.AddTool(edge)  
 splitter.Perform()  
  
 for shape in TopologyExplorer(splitter.Shape()).faces():  
 True  
 B = BRepPrimAPI\_MakeCylinder(gp\_Ax2(gp\_Pnt(0, 0, 0), gp\_Dir(0, 1, 0)), self.r, self.l).Shape()  
 self.new\_thing0 = BRepAlgoAPI\_Cut(new\_thing, B).Shape()

The above code is rotated, topologically and stretched through the functions in the OCC library to get the three-dimensional model and present it. These two models will be used as the main experimental objects in this paper

## CHAPTER 3. Experiments

This chapter will briefly describe the experimental process, mainly divided into the requirements of model input parameters and the generation of experimental results, compared with other pressure vessel design software calculation program

### 3.1 Parameters

#### 3.1.1 Cylinder Parameters

**Input:**

**-**Design Pressure

-Temperature

-Inner radius

-Cylinder length

-Design thickness

-Corrosion thickness

-Welding coefficient

**Parameters to be calculated:**

**-**Minimum thickness

-Maximum allowable working pressure

#### 3.1.2 Perforated Cylinder Parameters

**Input:**

|  |  |  |  |
| --- | --- | --- | --- |
| Inner diameter of shell | (mm) | Shell thickness | (mm) |
| Cylinder length | (mm) | Inner radius of nozzle | (mm) |
| Length of outer nozzle | (mm) | Thickness of nozzle wall | (mm) |
| Variable thickness nozzle | (mm) | Thickness of reinforcing plate | (mm) |
| Length of nozzle inside vessel | (mm) | Outside variable thickness length | (mm) |
| Design Pressure | (MPa) | Temperature | (centigrade) |
| Shell corrosion allowance | (mm) | Nozzle corrosion allowance | (mm) |
| Width of reinforcing plate | (mm) | Length of weld leg for vessel | (mm) |

**Parameters to be calculated:**

-Reinforcement range along vessel wall

-Reinforcement range of outer nozzle vessel wall

-Reinforcement range of vessel wall with inner tube

-Effective area near nozzle opening

-Effective radius of shell

-The force of internal pressure on the outer nozzle of the vessel

-The force of internal pressure on the shell

-Discontinuous force caused by internal pressure

-Effective thickness of shell

-Average primary membrane stress on vessel

-Overall primary membrane stress on vessel

-Maximum local primary membrane stress at nozzle intersection

*Refer to appendix for calculation process*

### 3.2 Modeling

This section will show the model generated after calculation according to the above input parameters

#### 3.2.1 Cylinder

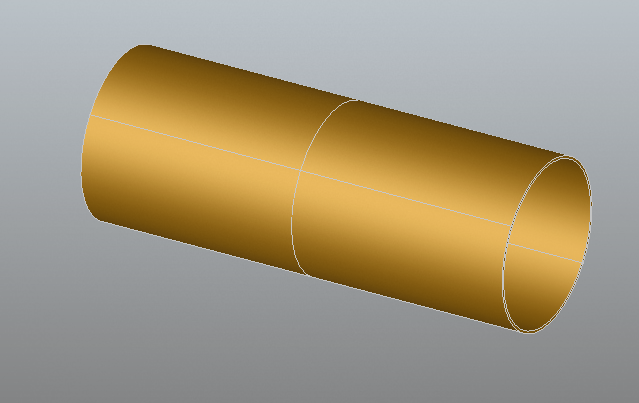


Figure7-Cylinder Model

#### 3.2.2 Perforated Cylinder

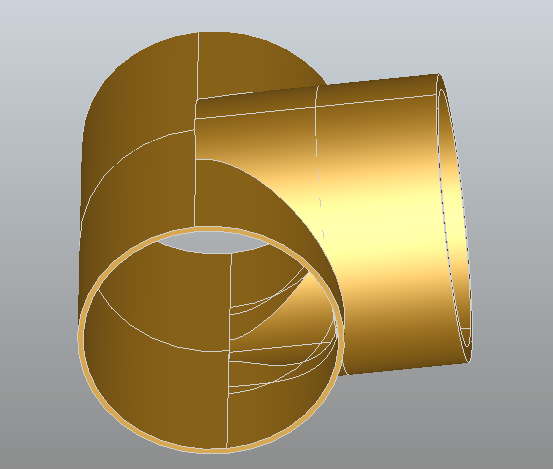


Figure8-Perforated Cylinder Model

### 3.3 Comparison of experimental data

This part will compare the model data generated by the software developed in this paper with the model of pressure vessel design software design already available on the market. Some differences may be found in the experimental data due to the different parts material selection

#### 3.3.1 Comparison of Cylinder Data

**Input Parameters:**

|  |  |
| --- | --- |
| Parameter symbol | Parameter |
| Inner diameter of cylinder /mm | 95 |
| Design Pressure /MPa | 3.1 |
| Design Temperature/ Centigrade | 232 |
| Material | SA-414G |
| Allowable stress of material /MPa | 147.6 |
| Welding coefficient | 0.8 |
| Design Thickness / mm | 3 |
| Corrosion allowance | 0 |

**Export results:**

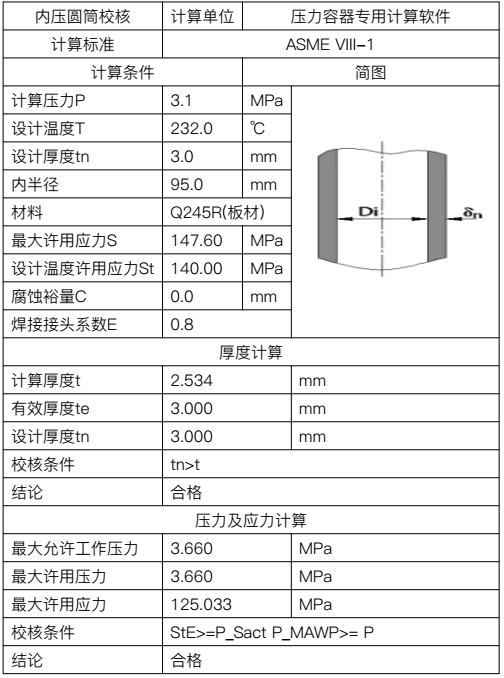


Figure9-Export Chart

**External software results:**

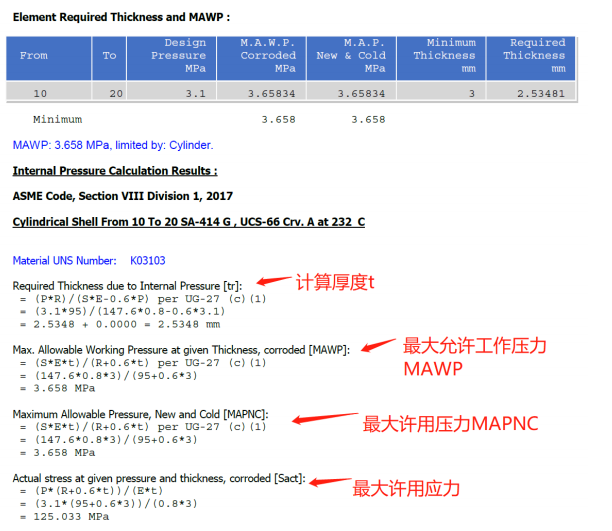


Figure10-External software results

#### 3.3.2 Comparison of Perforated Cylinder Data

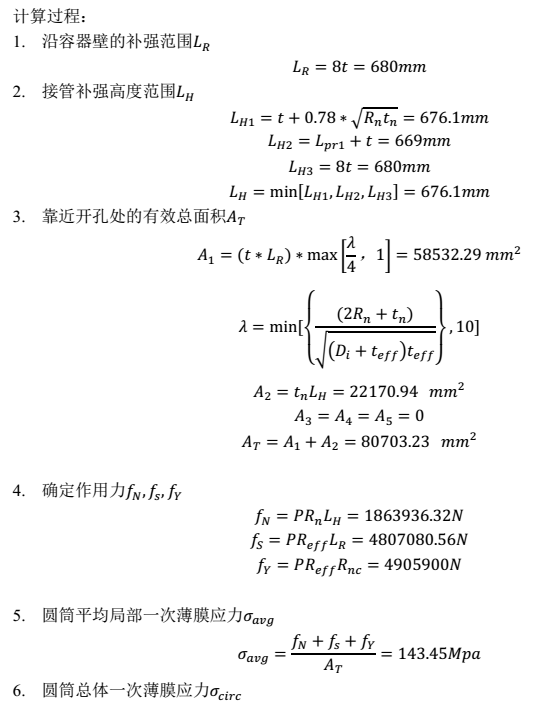
**Input Parameters:**

|  |  |
| --- | --- |
| Parameter symbol | Parameter |
| Design Pressure/MPa | 10 |
| Design Temperature/Centigrade | 20 |
| Corrosion allowance/mm | 2 |
| Welding coefficient | 1 |
| Allowable stress of material/ MPa | 203 |
| Tensile strength of materials/ MPa | 490 |
| Inner diameter of cylinder/ mm | 1422 |
| Inner radius of cylinder | 711 |
| Nominal thickness of cylinder | 85 |
| Inner diameter of nozzle | 1380 |
| Inner radius of nozzle | 690 |
| Nominal thickness of nozzle | 82 |
| Outside height of nozzle | 584 |

**Export results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 沿容器壁补强范围 | 680.00 | mm | 最大一次局部薄膜应力 | 126.15 | MPa |
| 容器外接管外表面沿接管补强范围 | 270.54 | mm | 平均一次薄膜应力 | 104.90 | MPa |
| 容器内接管内表面沿接管补强范围 | 0.00 | mm | 总体一次薄膜应力 | 83.65 | MPa |
| 壳体有效半径 | 711.00 | mm | 壳体有效厚度 | 85.00 | MPa |
| 接管有效外伸长度 | 584.00 | mm | 接管有效内伸长度 | 0.00 | mm |
| 接管开口处有效总面积 | 64870.00 | mm2 | 内压在壳体上引起的力 | 5147820.00 | N |
| 内压在容器外侧接管上引起的力 | 1280192.17 | N | 内压引起的不连续的力 | 106650.00 | N |
| 许用应力 |  |  | 接管最大许用应力 | 6.94 | N |
| 结论:合格 | | | | | |

**External software results:**



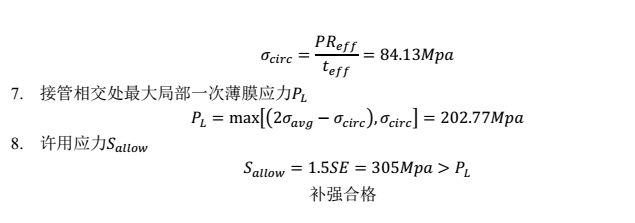


Figure11-External software results

### 3.4 Experiment Data Conclusion

According to the above two types model that we have mentioned.We input the required parameters.Comparing the data report generated by the software in this paper with the calculation example generated by the external software, we can get the results shown above. From the calculation results of the cylinder, the calculation results obtained by the software developed in this paper and the external software are basically consistent, but there are some differences in the experimental data results of the perforated cylinder, this may be due to the difference between the steps of the calculation program and the selection of materials for the parts.The results of model generation and experimental data show that the pressure vessel calculation software developed in this paper has a certain practical effect.

## CHAPTER 4. Conclusion

## CHAPTER 5. References

[1] Narayan, K. Lalit (2008). Computer Aided Design and Manufacturing. New Delhi: Prentice Hall of India. p. 3. ISBN 978-8120333420.

[2] Pottmann, H.; Brell-Cokcan, S. and Wallner, J. (2007) "Discrete surfaces for architectural design" Archived 2009-08-12 at the Wayback Machine, pp. 213–234 in Curve and Surface Design, Patrick Chenin, Tom Lyche and Larry L. Schumaker (eds.), Nashboro Press, ISBN 978-0-9728482-7-5.

[3] Susskind, Alfred Kriss; McDonough, James O. (March 1953). "Numerically Controlled Milling Machine" (PDF). Review of Input and Output Equipment Used in Computing Systems. International Workshop on Managing Requirements Knowledge. New York City: American Institute of Electrical Engineers. p. 136. LCCN 53-7874. Retrieved 2015-02-24.

[4] Hurst, J. (1989) Retrospectives II: The Early Years in Computer Graphics, pp. 39–73 in SIGGRAPH 89 Panel Proceedings, ACM New York, NY, USA, ISBN 0-89791-353-1 doi:10.1145/77276.77280

[5] https://www.opencascade.com/content/data-exchange

[6] <https://www.opencascade.com/content/data-exchange>

[7] <https://cutelyst.org/>

[8] "Signals & Slots - QtCore 5.1". Qt Project. 4 July 2013. Retrieved 10 April 2015.

## APPENDIX

### 1.内压圆柱形壳体开孔补强计算程序

**Step 1 计算沿容器壁的补强范围 L\_R**

1.if 接管类型=整体补强式接管

1)输入：t - 容器壁公称厚度

2)输出：L\_R

3)计算规则

*LR=8\*t*

2.if 接管类型= 带补强板的接管

1)输入：t - 容器壁公称厚度; t\_e - 补强板厚度； W - 补强板宽度

2)输出：L\_R

3）计算规则

1)te<0.5t or W<2t, LR=8\*t

2)te>=0.5t or W<8(t+te), LR=10\*t

3)te>=0.5t or W>=8(t+te), LR=8(t+te)

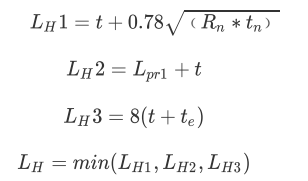
**Step 2 计算伸出在容器外表面沿接管壁的补强范围 L\_H**

1.if 接管臂不是变厚度

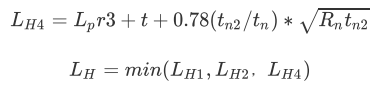
1）输入：R\_n - 接管内半径 ；t - 容器壁公称厚度； t\_n- 接管壁厚度；L\_pr1-容器壁外侧接管伸出长度；t\_e - 补强板厚度；L\_pr3 - 由容器外侧改变厚度t的长度;t\_n2 - 变厚度接管部分较薄部分的公称厚度

2）输出：L\_H

3）计算规则：



2.if 接管壁是变厚度，且L\_H>L\_pr3+t

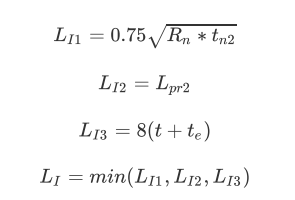


**Step 3 计算伸入在容器内表面沿接管壁的补强范围 L\_I（有则计算，没有则跳过）**

1.）输入：R\_n- 接管内半径;t\_nt\_n- 接管壁厚度;L\_pr2 - 容器壁内测的接管伸出长度;t\_e - 补强板厚度

2) 输出：L\_I

3）计算规则：



**Step 4 确定靠近接管开孔处的有效总面积 A\_T**

1.输入：t- 容器壁公称厚度; L\_R-容器壁有效长度; d\_n-接管内径; t\_n- 接管壁厚度;

t\_eff-用于计算接近开孔处压力应力的有效厚度,t\_eff=t D\_i-壳体内径; L\_R-容器壁有效长度，Step1计算得出

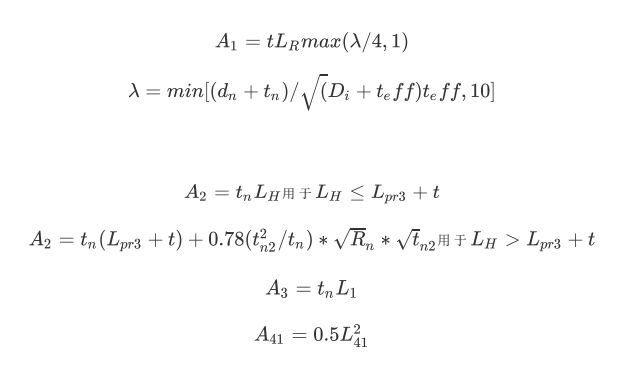
L\_H-容器外侧接管壁的有效长度;Step2计算得出

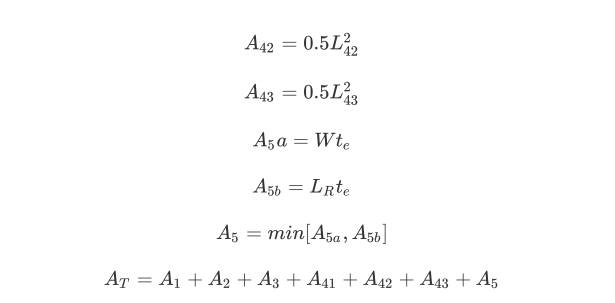
L\_pr3-由容器壁外侧改变厚度t的长度; t\_n2-变厚度接管较薄部分的公称厚度; L\_I-容器内侧接管壁的有效长度;Step3计算得出

L\_42 - 补强板焊制容器的填角焊缝的焊脚长; L\_43 - 内测接管填角焊缝的焊脚长度; W- 补强板宽度; t\_e- 补强板厚度;

2.输出：A\_T所提供的总面积;A\_1-壳体提供的面积;A\_2- 外伸接管所提供的面积;A\_3- 由内插接管所提供的面积;A\_41 - 由外部焊缝所提供的面积;A\_42 - 由补强板对容器焊缝所提供的面积;A\_43 - 由内部焊缝所提供的面积;A\_5 - 由补强板所提供的面积;

1. 计算规则:





**Step5 确定壳体有效半径 R\_eff**

1.输入：D\_i - 圆筒内径

2.输出：R\_eff

3.计算规则：



**Step6确定各作用力**

1.输入：P-设计压力;

L\_H- 容器外侧接管壁的有效长度;Step2计算得出；

L\_R-容器壁有效长度，Step1计算得出;

t- 容器壁公称厚度;

t\_n- 接管壁厚度;

R\_eff-壳体有效半径 ;

R\_n - 接管内半径;

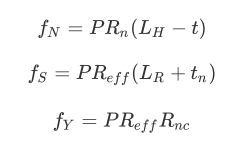
R\_nc-在容器上沿接管开孔长轴方向的接管半径

2.输出：f\_N-内压在容器外侧接管上 引起的力;

f\_S-内压在壳体上引起的力；

f\_Y-内压引起的不连续力

1. 计算规则:



**Step7 确定壳体有效厚度 t\_eff**

1.输入: t- 容器壁公称厚度; t\_e- 补强板厚度;

2.输出: t\_eff

3.计算规则:





**Step8 确定容器上的平均局部一次薄膜应力sigma\_avg和总体一次薄膜应力sigma\_eire**

1.输入：

Step6中计算结果

f\_N-内压在容器外侧接管上 引起的力; f\_S-内压在壳体上引起的力；f\_Y-内压引起的不连续力；

Step4中计算结果

A\_T-所提供的总面积

Step5 计算结果

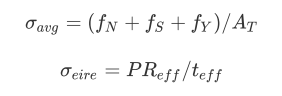
R\_eff -壳体有效半径

Step7 计算结果

t\_eff - 壳体有效厚度

2.输出：sigma\_avg,sigma\_eire

3.计算规则:



**Step9 确定在接管相交处的最大局部一次薄膜应力P\_L**

1.输入：Step8中计算结果容器上的平均局部一次薄膜应力sigma\_avg和总体一次薄膜应力

sigma\_eire

2.输出：P\_L

3.计算规则:



**Step10 校核最大局部一次薄膜应力**

1.输入：step9计算结果P\_L；

S-设计温度下接管和壳体材料的许用应力

E-焊接接头系数

2.输出：合格/不合格

3.计算规则：



**STEP11 确定接管最大许用工作压力P\_max**

1.输入：

S-设计温度下接管和壳体材料的许用应力;E-焊接接头系数;t- 容器壁公称厚度;t\_n- 接管壁厚度; Step4中计算结果

A\_T-所提供的总面积

Step5 计算结果

R\_eff -壳体有效半径；

Step7 计算结果

t\_eff - 壳体有效厚度;

Step2计算得出

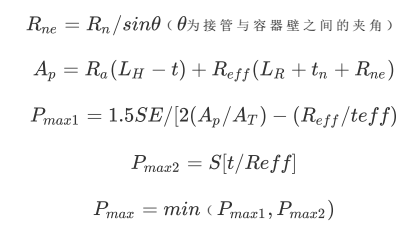
L\_H-容器外侧接管壁的有效长度;

Step1计算得出

L\_R-容器壁有效长度

2.输出P\_max

3.计算规则:



### 2.筒体壁厚计算

**1. 圆筒形壳体的最小厚度或最大许用压力应按照下列（1）或（2）计算出的较大厚度或较小压力。**

**(1)环向应力（纵向接头），if 厚度t不超过内半径的1/2或P不超过0.385SE时，应采用下列计算公式。**

Case A)已知设计压力P,求最小厚度t

输入：P-设计内压，R-壳体内半径，S-材料最大许用应力值，E-焊接接头系数

输出：最小厚度t

计算规则：



Case B)已知设计厚度t\_n，求最大许用工作压力P

输入：t\_e - 有效厚度=设计厚度减去腐蚀裕量，R-壳体内半径，S-材料最大许用应力值，E-焊接接头系数

输出：最大许用工作压力P\_MAWP（MAWP）



**（2）纵向应力（环向接头）。当厚度不超过内半径的1/2活P不超过1.25SE时，采用下列公式**

Case C)已知设计压力P,求计算厚度t

输入：P-设计内压，R-壳体内半径，S-材料最大许用应力值，E-焊接接头系数

输出：计算厚度t\_min



Case D)已知设计厚度t\_n，求最大许用工作压力P

输入：t\_e - 有效厚度=设计厚度减去腐蚀裕量，R-壳体内半径，S-材料最大许用应力值，E-焊接接头系数

输出：最大许用工作压力P\_MAWP（MAWP）



**最终输出的 t\_min 为max(t\_min--(1),t\_min--(2)) ,**

**最大许用工作压力 P\_MAWP 为min(P\_MAWP--(1),P\_MAWP--(2))**

**校核条件**

（1）厚度计算校核条件

设计厚度t\_n大于等于计算厚度t，成立则合格



(2) 压力应力计算校核条件

压力计算校核条件：最大许用工作压力P\_MAWP大于等于设计压力P



应力计算校核条件：最大许用应力P\_Sact大于等于材料最大许用应力S

