ARCH DAM DESIGN AND OPTIMIZATION

MASTERARBEIT

Zur Erlangung des akademischen Grades eines

DIPLOM-INGENIEURS

der Studienrichtung

BAUINGENIEURWISSENSCHAFTEN – GEOTECHNIK UND WASSERBAU

eingereicht am

INSTITUT FÜR WASSERBAU UND WASSERWIRTSCHAFT

TECHNISCHE UNIVERSITÄT GRAZ

Graz, Mai 2012

Arch Dam Design and Optimization

Master Thesis

In partial fulfilment of the requirements for the academic degree

Diplom-Ingenieur

In the field of study

Civil Engineering – Geotechnics und hydraulic Engineering

submitted to the

Institute of Hydraulic Engineering and Water Resources Management

Graz University of Technology

Graz, May 2012

Author:

Manuel PAGITSCH 0530414

....

Head of Institute:

Univ.-Prof. Dipl.-Ing. Dr. techn. Gerald ZENZ

....

Assistance:

Dipl.-Ing. Markus GOLDGRUBER

Dipl.-Ing. Rupert FELDBACHER

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Ich versichere, dass ich dieses Masterarbeitsthema bisher weder im In- noch im Ausland (einer Beurteilerin oder einem Beurteiler) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Graz am, …………………………………….. ……………………………………………………….

(Unterschrift)

Beschluss der Curricula-Kommission für Bachelor-,Master- und Diplomstudien vom 10.11.2008. Genehmigung des Senates am 1.12.2008.

DANKSAGUNG

Das Studium der Bauingenieurwissenschaften mit der Vertiefung Geotechnik und Wasserbau umfasst zahlreiche technische Facetten, welche neben dem theoretischen Lernprozess auch die dafür notwendige Praxisnähe erfordern, um sie begreifen und erfassen zu können. Grundlegend erfordert das Bauingenieurwesen jedoch, neben der eigenen eisernen Disziplin sich stetig neuem zu öffnen, auch die Bereitwilligkeit anderer sich einem anzunehmen und diese technischen Aspekte näher zu bringen.
Die Diskussion und das „Teamwork", sei es mit Professoren, Assistenten, Studienkollegen oder auch Freunden, erleichtert dabei den Lernprozess und öffnet den Horizont um auch komplexe Vorgänge erfassen und beschreiben zu können. Ihnen gebührt an dieser Stelle mein Dank.

Insbesondere möchte ich aber Univ.-Prof. Dipl.-Ing. Dr. techn. Gerald Zenz danken, der es mir ermöglicht hat meine Masterarbeit am Institut für Wasserbau und Wasserwirtschaft zu verfassen. Dem Thema meiner Arbeit, der automatischen Ermittlung einer Talsperren Geometrie stand er von Beginn an aufgeschlossen gegenüber und brachte auch die notwendige Unterstützung für die Ausarbeitung in die Arbeit mit ein.

Des Weiteren gilt mein Dank Dipl.-Ing Markus Goldgruber, der mich schon in meinem Studium als Kollege unterstützte und auch bei der Ausarbeitung der Masterarbeit mit Rat und Tat zur Seite stand, sowie Dipl.-Ing Rupert Feldbacher, der mit mir schon bei einem vorangegangen Master Projekt das Thema der Koordinatenberechnung für Gewölbestaumauern erörterte und somit den Grundstein für diese Masterarbeit legte.

Zuletzt möchte ich natürlich meiner Familie, mit meinen Eltern Hubert und Ernestine, welche mir mein Studium, sei es in finanzieller aber auch logistischer Hinsicht, erst möglich gemacht haben, sowie meiner Frau Sabine, meiner Tochter Sarah aber auch meiner Schwester Bettina und meinem Schwager Michael, danken. Sie brachten die notwendige Geduld und das erforderliche Verständnis auf und gaben mir immer die sichere Unterstützung und den unerlässlichen Rückhalt wenn dies vonnöten war.

An dieser Stelle, vielen Dank.

Meiner Familie

in Dankbarkeit gewidmet

KURZFASSUNG

In Zeiten der Diskussion über die sichere Nutzung von atomarer Energie zur Erzeugung von elektrischem Strom und der anhaltenden Endlagerungsproblematik, gewinnt die Nutzung Wasserkraft wieder zunehmend an Bedeutung. Des Weiteren konnten durch die Liberalisierung des Strommarktes neue technische Methoden, wie die Nutzung von Pumpspeicherkraftwerken, wirtschaftlich realisiert werden, stellen jedoch nach wie vor einen großen Planungsaufwand dar. Somit ist die Nutzung von software-gestützten Berechnungsmethoden, zur Analyse und Planung neuer wasserbaulicher Anlagen ein notwendiger Schritt dieser Entwicklung Rechnung zu tragen.

Diese Masterarbeit beschäftigt sich mit der Berechnung der 3D Koordinaten von doppelt gekrümmten Bogenstaumauern und der Weiterverarbeitung dieser ermittelten Koordinaten mithilfe eines, auf der Methode der finiten Elemente, basierenden Programms. Für die Berechnung der Koordinaten wurde ein Programm mittels Visual Basic for Applications, welches in Microsoft Excel integriert ist, entwickelt. Durch die Eingabe unterschiedlicher Parameter und die Nutzung unterschiedlicher Berechnungsmethoden, ist es dabei möglich die Geometrie von Gewölbemauern zu ermitteln und für Optimierungszwecke zu verändern.

Darüber hinaus wurde eine automatisierte Script-Erstellung realisiert, welche die Integration dieser ermittelten Geometrie der Staumauer in das Finite Elemente Programm Abaqus CAE ermöglicht

Mit dem Programm Abaqus CAE konnte abschließend die Strukturanalyse zur Erfassung der auftretenden Spannungen und Verschiebungen anhand des Beispiels einer großen Bogenstaumauer durchgeführt werden. Zur Untersuchung des Tragverhaltens kamen dabei die Lastfälle Eigengewicht, Wasserdruck und Temperatur zum Einsatz.

ABSTRACT

At a time of discussion about the safe use of nuclear power to generate electricity and the unsolved problem of radioactive waste deposit, the importance of hydropower is increasing. Furthermore the power market liberalization enabled the development of power generation methods, like the use of pumped storage hydro power plants, which can now be realized economically. Nevertheless the planning effort is still considerable when constructing hydro power plants and therefore the use of computer aided design is indispensable.

This master thesis deals with the software based calculation of the 3D coordinates of double curved arch dams and the additional processing of these coordinates using Finite Element software. To calculate these 3D coordinates, a program based on Visual Basic for Applications, which is a Microsoft Excel tool, has been developed. By varying the input parameters and using different calculation methods, it is possible to determine the geometry of double curved arch dams and modify it for optimization purposes.

As this calculation and optimization is done, an automated Python script export has been implemented. This script enables the import of the chosen dam geometry into the Finite Element program Abaqus CAE.

Finally a structural analysis investigating the stresses and displacements of a given arch dam example is done within Abaqus CAE. The structural behavior of the arch dam is determined using the assigned loads dead weight, water pressure and temperature distribution.

Table of Contents

*The numbers [1], linked to the heading represent the bibliography entry

1 A BRIEF INTRODUCTION INTO ARCH DAMS

1.1 BENEFIT AND PURPOSE OF DAMS

Water, as common known giver of life, has always played a central role to guarantee the existence of human being on planet earth. To ensure drink water supply and irrigation in agriculture, humans were forced to settle nearby bodies of water. But with all the benefits, that water provides, also dangers concerning flood risk arise and therefore it was necessary to design structures, which were able to regulate the water level. Building dams become an appropriate method and for this reason a continuous development of dam structures began. The much later growing demand for electric energy gained from Hydropower also promoted this progress.

Dams were built of soil and rock for thousands of years, until concrete became one of the most common building material. Nowadays, structural and topographic as well as economic reasons specify the geometry and building material a dam structure is built with. In the special topographical case of a narrow valley with stable valley sides, engineers started to design dams with a special shape and therefore special structural behaviour. Simultaneously they reduce the volume of the building significantly by using this new design, which is known as arch dam.

1.2 ARCH DAM TYPES*[11]

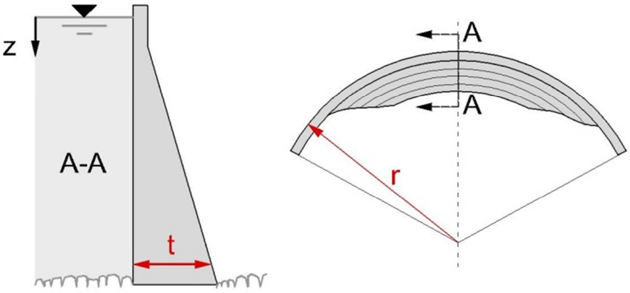
In history, different types of arch dams were developed due to the prevailing ability of doing the structural engineering calculation of the geometry of the dam. As for example constant radius or constant angle arch dams allow a simplified structural analysis, which is basically explained in the chapter below, they represent the initial arch dam types. Due to the fact, that this simple geometry arch dams are hardly appropriate to asymmetric topography, further geometry development was necessary. By the use of the Finite Element Method, which is mentioned in chapter 3.3 it become possible to do a structural analysis for complex geometries and therefore also double curved arch dams arise.

1.2.1 CONSTANT RADIUS ARCH DAM

The geometry of the constant radius arch dams is realized with arches of equal radius, regarding the upstream face of the dam (fig.1.1). Using a constant radius at any height of the dam causes the upstream face to be vertical. Considering the linear increasing water pressure by depth, the dam's thickness is increasing linear by depth too. Therefore, the vertical sections of a constant radius arch dam are similar to the shape of a gravity dam. To determine the thickness of a constant radius arch dam Barlow's formula (3-20) can be used. With known radius, water pressure and compressive strength of the concrete the required thickness is

$$t = \frac{p_i * r}{\sigma_\varphi} \tag{1-1}$$

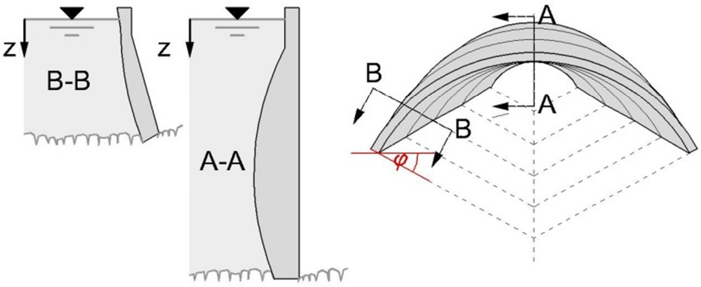
To ensure a minimum of volume an opening angle of 133° should be used, but varying the opening angle from 120° to 140° causes the volume to increase merely slightly.



(fig. 1.1) Cross-section and layout of a constant radius arch dam [11]

1.2.2 CONSTANT ANGLE ARCH DAM

The geometry of a constant angle arch dams is characterized by equal angles of the surfaces of the abutment (fig.1.2). By transferring forces, having the same angle, to the ground, the thickness of the arch dam and therefore the volume, can be reduced. Although the volume is decreasing when using that arch dam type, effort is increasing caused by curved upstream surfaces and thereby resulting additional amount of formwork. To realize a constant angle arch dam, the opening angle should lie between 100° and 130°.

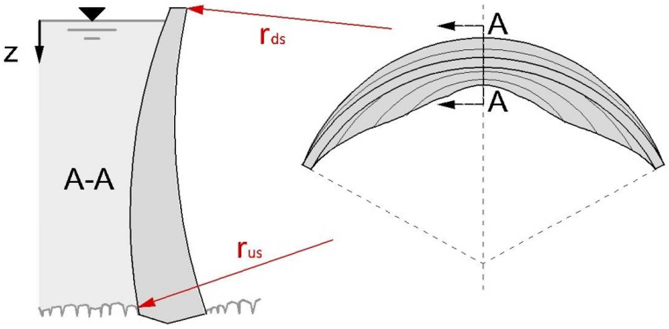


(fig. 1.2) Cross-section and layout of a constant angle arch dam [11]

1.2.3 DOUBLE CURVED ARCH DAM

Double curved arch dams feature horizontal as well as vertical arches (fig.1.3). Therefore, the geometry of double curved arch dams is the most suitable concerning load transfer and volume reduction and can be fitted to nearly every asymmetric valley shape. The vertical sections are commonly realized with circles, whereas the horizontal arches are often realized with conic sections, basket-handle arches or logarithmic helixes.

Despite the fact that double curvature dams have the most convenient structural behaviour, the effort, regarding design and construction is enormous. Without using modern calculation methods, such as the Finite Element Method, it would be hardly possible to investigate strains, displacements and stresses for such a complex geometry.



(fig. 1.3) Cross-section and layout of a double curved arch dam [11]

1.3 ENGINEERING ASPECTS*[8][12][15][17]

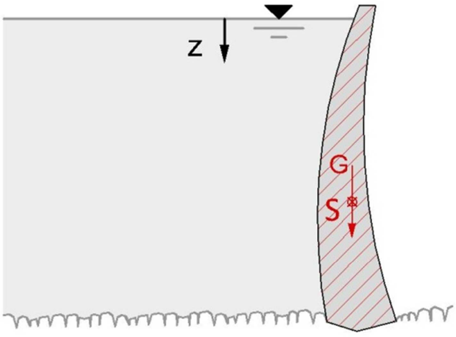
Beside the analysis of the strains and stresses to give some indication on the structural behaviour of arch dams, some other engineering aspects should be considered for preliminary design of arch dams. Some of those issues are explained in the chapters below to give a general overview for basic design.

1.3.1 LOADS

When applying loads to the arch dam, it is essential to consider the current state of construction. As for example arch dams are constructed to separate blocks, the structural behaviour concerning the dead load differs, whether the blocks are grouted, or not. Further, when combining loads, impossible assumptions such as temperature loads caused by a surface temperature of 20°C and simultaneously occurring ice load, should be avoided.

-Dead load

The dead load of an arch dam can be calculated form the volume of the arch dam times the specific weight of concrete. To gain the maximum value of the dead load, the weight of all appurtenances must be added too. As mentioned in the introduction the structural behaviour of the arch dam, concerning the dead load, differs according to whether the certain blocks are grouted, or not. When blocks are ungrouted no shear stresses can be transferred and the dead load is transmitted vertically only, directly to the foundation. Grouted Blocks are able to withstand shear stresses and therefore, the dead load is transmitted horizontally to the valley sides too.



(fig. 1.4) Dead load of an arch dam

The dead load of the arch dam can be calculated with

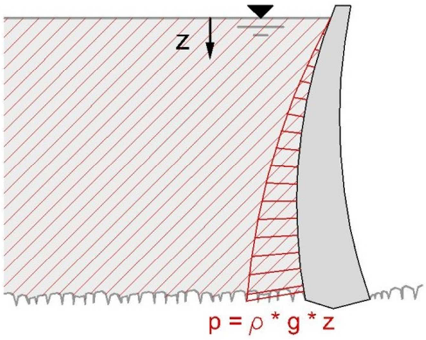
$$G = m_d * g \tag{1-2}$$

with

G	[N]	.. dead load
m_d	[kg]	.. mass of the dam
g	[m/s²]	.. acceleration due to gravity

-Hydrostatic Pressure

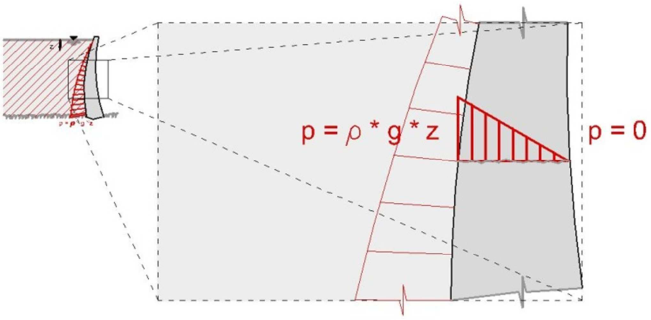
The hydrostatic pressure, which is applied to the arch dam, can be calculated from the hydrostatic pressure distribution (3-17). As the distribution does only depend on the height z , the maximum value of hydrostatic pressure occurs at the bottom of the arch dam. Water pressure is applied in direction perpendicular to the surface and therefore a curved surface causes vertical and horizontal water pressure.



(fig. 1.5) Hydrostatic pressure of an arch dam

-Internal hydrostatic Pressure

Internal hydrostatic pressure can also be calculated from the hydrostatic pressure distribution. Whereas the hydrostatic pressure is applied to the upstream surface the internal hydrostatic pressure acts on cracks and joints within the dam and foundation. The maximum value of the internal hydrostatic pressure occurs on the upstream surface and has the same value as the hydrostatic pressure at that height. Within the dam it is increasing linearly until it becomes zero at the downstream surface. To determine the internal hydrostatic pressure distribution through the foundation, also drain size and depth as well as rock porosity, faulting and jointing must be observed.



(fig. 1.6) Internal hydrostatic pressure of an arch dam

-Temperature

Temperature loads occur on concrete arch dams, when temperature changes from the value at the end of the construction phase, which is called closure temperature, to another value, either higher or lower. Therefore, this change is called temperature difference. That temperature difference can be generated by the difference between the temperature of the air at the downstream surface and the water temperature at the upstream surface, as well insolation, which is heating the concrete mass. As a temperature difference exists, concrete experience a volumetric change, based on its thermal coefficient of expansion. Caused by the fact that this expansion is supressed, temperature originated stresses arise.

-Earthquake

Earthquake caused loads applied to an arch dam depend on the magnitude and the frequency of the earthquake and the resonant frequency of the dam itself. To analyse earthquake originated loads, complex dynamic models, processed with Finite Element software, are necessary. To simplify the analysis a "'pseudo static"-method, which can be written as

$$F_e = m * a_e \tag{1-3}$$

with

F_e	[N]	.. force caused by earthquake
m	[kg]	.. mass of the dam/water
a_e	[m/s²]	.. acceleration due to earthquake

can be used for preliminary design and subordinated structures.

-Silt

Silt induced loads take place on the upstream surface of the arch dam. Analogous to the hydrostatic pressure distribution, the earth pressure distribution caused by silt is linear, but vertical and horizontal pressure differs from each other. The vertical earth pressure can be written as

$$e_{0v,s} = \gamma_{u,s} * z \tag{1-4}$$

and the horizontal pressure, modified with K_0 as simplified estimation, can be written as.

$$e_{0h,s} = e_{v,s} * K_0 = \gamma_{u,s} * z * K_0 \tag{1-5}$$

with

$e_{0v,s}$	[MPa]	.. vertical earth pressure caused by silt
$e_{0h,s}$	[MPa]	.. horizontal earth pressure caused by silt

K_0	[-]	.. earth pressure coefficient $K_0 = 1 - \sin(\varphi_s)$
φ_s	[°]	.. friction angle of silt
$\gamma_{u,s}$	[MN/m³]	.. specific weight of the uplifted silt

To apply earth pressure on the arch dam for analysis, separate investigation of the location is necessary, due to the fact that silt do not occur at every arch dam.

-Ice

Loads, which occur due to ice, are not well investigated yet. Ice pressure, which is applied to the arch dam, is created by thermal expansion of the ice sheet and wind drag acting on the surface of the ice. When adding ice loads to the analysis of the arch dam, it has to be considered, that ice is not a durable load and does not affect every arch dam, depending on its location.

1.3.2 Foundation

The stability of the foundation is one of the most important aspects, when doing the design of arch dams and therefore the investigation of the foundation require extraordinary treatment. In situ observation should give an overview over joints and faults and laboratory tests should determine the uniaxial compressive strength and the tensile strength as well as shear strength of the intact rock. One of the results of the laboratory test is the shear strength, which can be formulated with the Coulomb equation and written as

$$\tau_f = c' + \sigma_n{}' * \tan(\varphi') \tag{1-6}$$

with

τ_f	[MPa]	.. shear strength of the intact rock
c'	[MPa]	.. cohesion
$\sigma_n{}'$	[MPa]	.. effective normal stress
φ'	[°]	.. friction angle of the intact rock

The results of the laboratory tests represent strength values for intact rock only and have therefore be modified when analysing rock mass, containing joints and faults. It has also to be mentioned that different types of rock mass have different behaviour and therefore failure mechanism. That is another reason why accurate investigation is inevitable.

1.3.3 Grouting

As the foundation of the arch dam has high permeability, or is highly fractured it is necessary to do some grouting operations. Grouting can be divided into consolidation and curtain grouting, where consolidation grouting represents low pressure injections to fill cracks and curtain grouting is accomplished with high pressure injection to avoid seepage under the arch dam. To determine the permeability of the foundation Darcy's law

$$v = k * i \tag{1-7}$$

with

v	[m/s]	.. shear strength of the intact rock
k	[m/s]	.. cohesion
i	[-]	.. effective normal stress

can be applied. With known flow Q and area A the permeability can be written as

$$k = \frac{Q}{A * i} \tag{1-8}$$

using the equation of continuity for flow, written as

$$Q = v * A \tag{1-9}$$

A high amount of seepage may cause erosion within the foundation and therefore curtain grouting is fundamental to provide a durable stability of the foundation.

2 Definition of scope

The scope of this master thesis is the development of a software based method to calculate 3D arch dam coordinates. Design guidelines, obeying the rules of USBR, as well as 2 alternative calculation methods represent the foundation of the calculation of the coordinates. As the coordinates of the arch dam are calculated, a python script should provide the necessary commands, to import the geometry of the developed arch dam into Abaqus CAE, which is Finite Element software. Finally an analysis, which should give some information about the displacements and stresses of the given arch dam example, represent the completion of this master thesis.

2.1 Given Example of a large Arch Dam*[14]

The given example of a large arch dam is located at the province Artvin, which is placed at the northeast of Turkey (fig.2.1). Bounded by mountains and traversed by the Çoruh River, this location represents a suitable location for constructing an arch dam.

(fig. 2.1) The province Artvin at the northeast of Turkey [14]

The given double curved arch dam is 220 m high. The width at the crest is about 440m and the width at the bottom is about 110 m. The thickness of the horizon at the crest of the dam is 8 m and the thickness of the horizon at the bottom is 55 m. The horizons of the dam are realized with circles and ellipses, with radii from 55 to 254m. The arch dam is built of concrete, with an estimated specific weight of 24 kN/m³, a Young's modulus of 25000 MPa and a Poisson's ratio of 0,2. The surrounding foundation contains rock, with an estimated specific weight of 26 kN/m³, a Young's modulus of 26000 MPa and a Poisson's ratio of 0,2. The table below shows detailed information on the data of the arch dam.

Table 2-1: Data of the arch dam

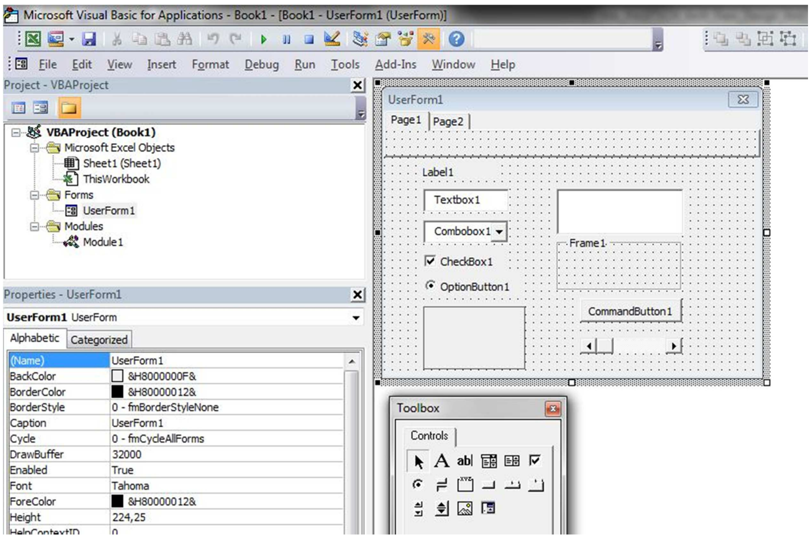
Height	220 m	
Geometry	**Crest**	**Bottom**
Altitude	495 AMSL	715 AMSL
Width	440 m	110 m
Thickness	8 m	55 m
Radii US	254 m	155 m
Radii DS	246 m	55 m
Material	**Concrete**	**Rock**
Specific weight	24 kN/m³	26 kN/m³
Young's modulus	25000 MPa	26000 MPa
Poisson's ratio	0.2	0.2

Beside the geometric and material pre-sets also the loads are given to investigate the resulting displacements and stresses. The loads applied to the arch dam for analysis are the dead load of the construction, the hydrostatic pressure and temperature originated stresses.

2.2 The Creation of a Program to Calculate 3D Dam Coordinates[10]

Due to the fact, that the geometry of an arch dam has to be fitted to the given local topography, the calculation of the 3D coordinates of the dam is an iterative process. To simplify this process, the use of a software based calculation method is obvious.

The program, used for the single calculation steps, is Visual Basic for Applications, briefly VBA, which is integrated in the Microsoft Office Program Excel and therefore common software. Beside the possibility to write scripts, VBA provides the integration of visual user forms (fig.2.2), which are known from user interfaces of earlier Microsoft Windows versions.



(Name)	UserForm1
BackColor	□ &H8000000F&
BorderColor	■ &H80000012&
BorderStyle	0 - fmBorderStyleNone
Caption	UserForm1
Cycle	0 - fmCycleAllForms
DrawBuffer	32000
Enabled	True
Font	Tahoma
ForeColor	■ &H80000012&
Height	224,25
HelpContextID	0

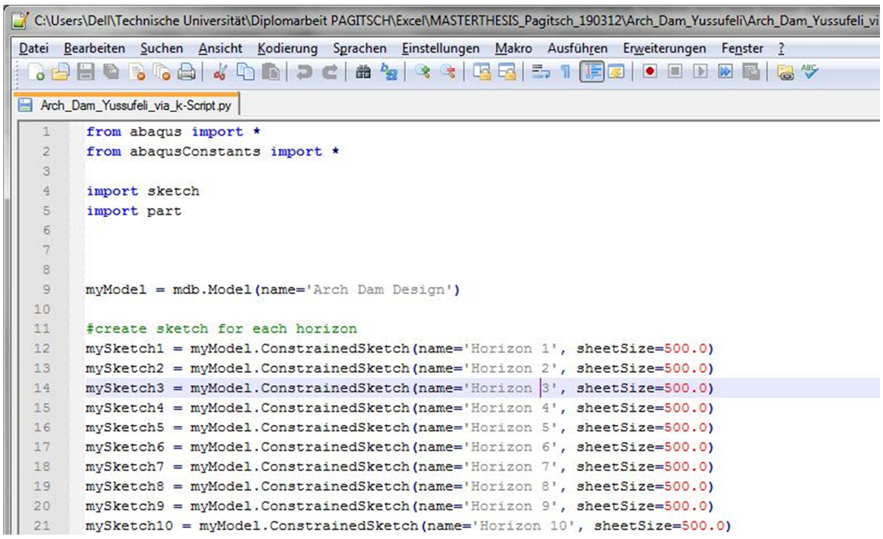
(fig. 2.2) Visual Basic for Applications user interface

The program developed for this master thesis, is divided into several of this user forms. The main user form enables the user to navigate through the program by the use of a menu and a tool bar, which are placed at the top of the user form. Seven additional sub user forms provide the possibility, to put in the given geometry and material data of the arch dam and to do the post processing. As the whole input is done the calculation of the coordinates can be done by clicking the corresponding button. When the coordinates are calculated, the exportation via CSV file as well as Python Script is provided for further processing. Detailed information on the VBA based program can be found within chapter 4.

2.3 PYTHON SCRIPT AS INTERFACE BETWEEN THE CALCULATED COORDINATES AND THE ABAQUS MODEL[2]

As opposed to Microsoft Excel, which uses VBA Scripts to regulate the program, Abaqus CAE uses Python Script to manage commands within the program. Python Script is an object orientated programming language, which provides commands (fig.2.3) that can be processed from Abaqus with the "Run Script" function.

Within the developed VBA program, the Python Script commands can be written to a file with the filename extension ".py" after the calculation of the coordinates, by clicking on the corresponding button. As the script is exported, it contains the definition of the coordinates, sketches, wires and datum planes, as well as commands for the lofting and the assembly. These definitions are necessary to create a solid model of the arch dam within Abaqus CAE, using geometrical information, calculated with another program.


C:\Users\Dell\Technische Universität\Diplomarbeit PAGITSCH\Excel\MASTERTHESIS_Pagitsch_190312\Arch_Dam_Yussufeli\Arch_Dam_Yussufeli_vi
Datei
Bearbeiten
Suchen
Ansicht
Kodierung
Sprachen
Einstellungen
Makro
Ausführen
Erweiterungen
Fenster
?
Arch_Dam_Yussufeli_via_k-Script.py

```python
from abaqus import *
from abaqusConstants import *

import sketch
import part



myModel = mdb.Model(name='Arch Dam Design')

#create sketch for each horizon
mySketch1 = myModel.ConstrainedSketch(name='Horizon 1', sheetSize=500.0)
mySketch2 = myModel.ConstrainedSketch(name='Horizon 2', sheetSize=500.0)
mySketch3 = myModel.ConstrainedSketch(name='Horizon 3', sheetSize=500.0)
mySketch4 = myModel.ConstrainedSketch(name='Horizon 4', sheetSize=500.0)
mySketch5 = myModel.ConstrainedSketch(name='Horizon 5', sheetSize=500.0)
mySketch6 = myModel.ConstrainedSketch(name='Horizon 6', sheetSize=500.0)
mySketch7 = myModel.ConstrainedSketch(name='Horizon 7', sheetSize=500.0)
mySketch8 = myModel.ConstrainedSketch(name='Horizon 8', sheetSize=500.0)
mySketch9 = myModel.ConstrainedSketch(name='Horizon 9', sheetSize=500.0)
mySketch10 = myModel.ConstrainedSketch(name='Horizon 10', sheetSize=500.0)
```

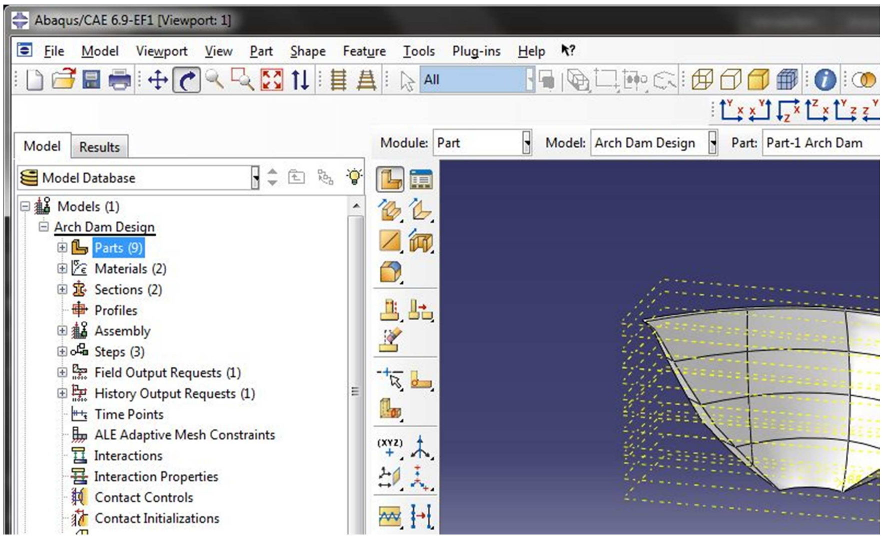
(fig. 2.3) Python Script opened with Notepad + +

Beside the geometry of the arch dam, script commands for the geometry of a so called virtual terrain can be exported too. Except the geometry of the abutment of the arch dam, which must also be provided by virtual terrain, the remaining coordinates of the terrain are artificially predefined. The creation of a virtual terrain represents an approximation of the real topography, which is sufficiently accurate for the analysis of the arch dam, providing that the area of the virtual terrain is widespread enough, to neglect boundary influences. The structure of a Python Script and an explanation of the different commands are placed within chapter 5.

2.4 THE CALCULATION OF STRESSES AND DISPLACEMENTS WITH ABAQUS CAE[1]

As mentioned in the introduction Abaqus CAE is Finite Element software. Finite Element can be used to calculate the strains and displacements of loaded structures approximately, by using a numerical calculation method, which is briefly explained within chapter 3.3. As the calculated geometry of the arch dam is imported into Abaqus via Python Script, several steps must be done before the analysis of the stresses and displacements. The discretisation of the model and the definition of the material as well as the setting of the boundary conditions are just a few of them.

As the analysis is done the stresses and displacements can be visualised within the results screen of the program. Chapter 6 would explain how to prepare a solid model to do a structural analysis and to evaluate the obtained results.



Module: Part
Model: Arch Dam Design
Part: Part-1 Arch Dam

(fig. 2.4) Abaqus CAE user interface

3 TECHNICAL ESSENTIALS

3.1 HYDRO-MECHANICAL ESSENTIALS*[8][17]

3.1.1 THE EULER EQUATION IN FLUID DYNAMICS

To obtain the hydrostatic pressure distribution of a stationary liquid it is necessary to establish the Euler equation for fluids. Considering an infinitesimal fluid particle, let's assume a cuboid (fig.3.1) with the mass *dm* where

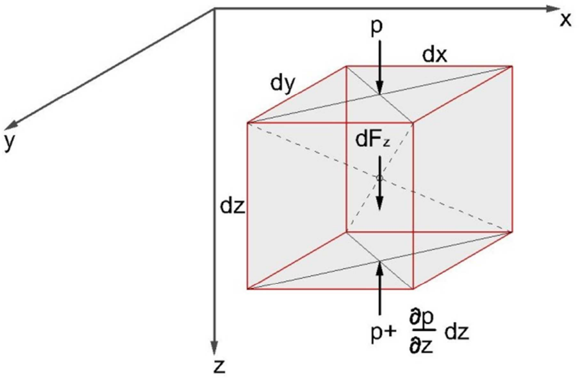
$$dm = \rho * dx * dy * dz \tag{3-1}$$

and the resulting force after Newton's second law regarding this mass as

$$dF_m = dm * a \tag{3-2}$$

the equilibrium of forces where the sum of the forces is zero can be formed taking account of the pressure p effective on the fluid particle.

$$\Sigma\, F = 0 \tag{3-3}$$



(fig. 3.1) Infinitesimal cuboid fluid particle [17]

Separating the equilibrium of forces to the 3 directions of the coordinate system x, y and z we obtain the following 3 equations

$$p * dy * dz + \rho * a * dx * dy * dz = \left(p + \frac{\partial p}{\partial x} dx\right) dy * dz \tag{3-4}$$

$$p * dx * dz + \rho * a * dx * dy * dz = \left(p + \frac{\partial p}{\partial y} dy\right) dx * dz \tag{3-5}$$

$$p * dx * dy + \rho * a * dx * dy * dz = \left(p + \frac{\partial p}{\partial z} dz\right) dx * dy \tag{3-6}$$

Applying Pascal’s law where pressure is transferred equally in all directions within a stationary incompressible fluid, written as

$$p = p_x = p_y = p_z \tag{3-7}$$

these equations can be transformed into

$$\rho * a_x *= \frac{\partial p}{\partial x} \tag{3-8}$$

$$\rho * a_y = \frac{\partial p}{\partial y} \tag{3-9}$$

$$\rho * a_z = \frac{\partial p}{\partial z} \tag{3-10}$$

Forming the total derivative with

$$dp = \frac{\partial p}{\partial x} * \mathrm{dx} + \frac{\partial p}{\partial y} * \mathrm{dy} + \frac{\partial p}{\partial z} * \mathrm{dz} \tag{3-11}$$

leads to the Euler equation in fluid dynamics, written as

$$dp = \rho\left(a_x * dx + a_y * dy + a_z * dz\right) \tag{3-12}$$

with

ρ	[kg/m³]	.. density of the fluid
a_z	[m/s²]	.. acceleration in the x direction
a_y	[m/s²]	.. acceleration in the y direction
a_z	[m/s²]	.. acceleration in the z direction

3.1.2 HYDROSTATIC PRESSURE DISTRIBUTION

The former shown Euler equation in fluid dynamics (3-12) and the assumption that only gravity is considered, which means that

$$a_z = g \ , \ a_x = a_y = 0 \tag{3-13}$$

lead to

$$dp = \rho * g * dz \tag{3-14}$$

With the integration of dz which is given by

$$p = \int \rho * g * dz \tag{3-15}$$

and the boundary condition

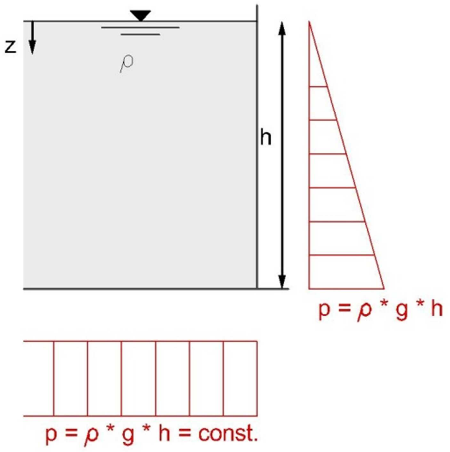
$$if \ \ z = 0 \, ; \, p = p_a \tag{3-16}$$

the hydrostatic pressure distribution (fig.3.2) can be written as

$$p = \rho * g * z + p_a \tag{3-17}$$

with

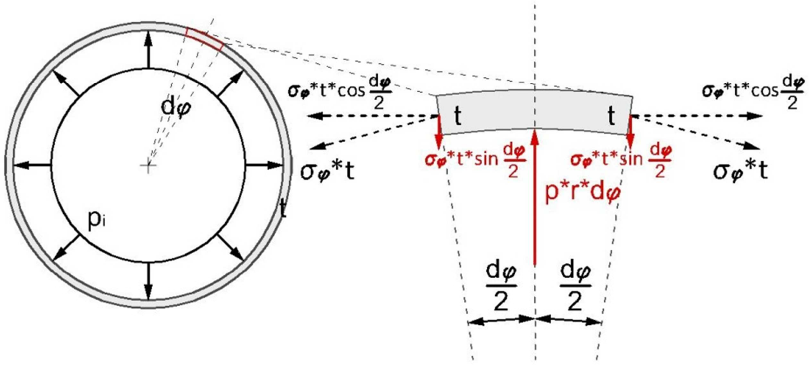
p	[Pa]	.. hydrostatic pressure
ρ	[kg/m³]	.. density of the fluid
z	[m]	.. vertical distance to the reference point
p_a	[Pa]	.. reference pressure



(fig. 3.2) Hydrostatic pressure distribution [17]

3.1.3 Barlow's formula

To receive an estimation of the arch stresses of an arch dam, circumferential stresses of a thin shell, which are covered with Barlow's formula can be used.



(fig. 3.3) Equilibrium of forces in radial direction

Considering an infinitesimal ring shell element (fig.3.3) with thickness t, which is loaded with uniform internal pressure pi and forming the equilibrium of forces in radial direction we obtain

$$2 * \sigma_\varphi * t * \sin\frac{d_\varphi}{2} = p_i * r * d_\varphi \qquad (3\text{-}18)$$

Using the small angle approximation the sinus of the angle can be written as the angle itself

$$\sin\frac{d_\varphi}{2} \sim \frac{d_\varphi}{2} \qquad (3\text{-}19)$$

Applying (3-19) with (3-18) leads to

$$\sigma_\varphi = \frac{p_i * r}{t} \qquad (3\text{-}20)$$

with

σ_φ	[MN/m²]	.. circumferential stress
p_i	[MPa]	.. internal pressure
r	[m]	.. radius of the ring element
t	[m]	.. thickness of the ring element

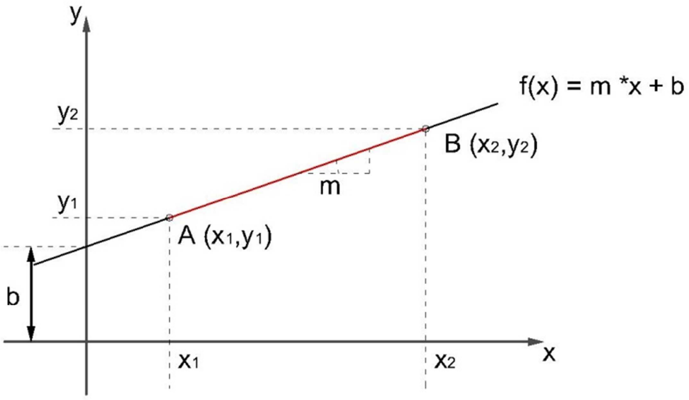
3.2 GEOMETRIC ESSENTIALS*[3][9]

3.2.1 GENERAL FORM OF THE EQUATION OF A STRAIGHT LINE

Two points A(x_1,y_1) and B(x_2,y_2) lying on one plane are given. By connecting these points they shape a straight line (fig.3.4) of the form

$$f(x) = m * x + b \tag{3-21}$$

with so called gradient *m* and y-intercept *b.* To define that function it is necessary to determine these parameters m and b.



(fig. 3.4) Graph of a straight line [3]

To set the parameter m you have to calculate the gradient of the line between A and B with

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)} \tag{3-22}$$

As the gradient m is known the calculation of the y-intercept b can be done by converting (3-21) into

$$b = y - x * m \tag{3-23}$$

with the known points x_1 and y_1 or x_2 and y_2 inserted for x and y. As m and b is determined the equation of a straight line with two known points A(x_1,y_1) and B(x_2,y_2) can be written as

$$f(x) = \frac{(y_2 - y_1)}{(x_2 - x_1)} * x + \left(y_1 - \frac{(y_2 - y_1)}{(x_2 - x_1)} * x_1\right) \tag{3-24}$$

3.2.2 VERTEX FORM OF THE EQUATION OF A CONIC SECTION

To establish the vertex equation for conic sections it is necessary to shift the vertex of the conic section to the origin of the coordinate system. Therefore the common known conic section equations, except the equation for the parabola, which is still formulated in vertex equation form, must be modified. The equation for a circle centred at the origin is

$$r^2 = x^2 + y^2 \tag{3-25}$$

To centre the vertex at the origin the x coordinate has to be modified with the radius to get

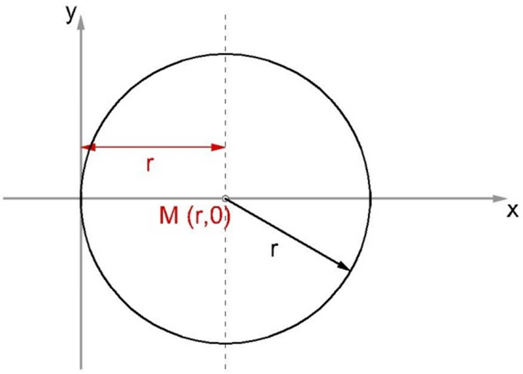
$$r^2 = (x-r)^2 + y^2 \tag{3-26}$$

As we solve this equation, the vertex equation of a circle can be written as

$$y^2 = 2 * r * x - x^2 \tag{3-27}$$

If the formula (3-23) is extended with r=p and (ε²-1), we obtain the general vertex form of the equation of a conic section regarding a circle

$$y^2 = 2 * p * x + (\varepsilon^2 - 1) * x^2\,,\ if\ \varepsilon = 0 \tag{3-28}$$



(fig. 3.5) Decentred circle [3]

The common known equation for an ellipse centred at the origin of the coordinate system is written as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{3-29}$$

To shift the vertex to the centre, the x coordinate has to extended with (*-a)* so that (3-29) become

$$\frac{(x-a)^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{3-30}$$

Solving the equation (3-30) leads to

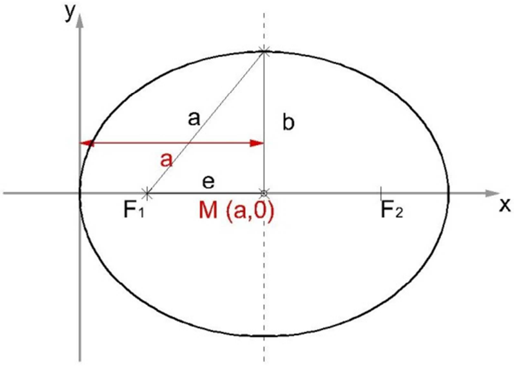
$$y^2 = 2 * \frac{b^2}{a} * x + \frac{b^2}{a^2} * x^2 \tag{3-31}$$

As we add the arguments

$$p = \frac{b^2}{a} \; and \; \varepsilon = \frac{\sqrt{a^2-b^2}}{a} \tag{3-32}$$

we get the general vertex form of the equation of a conic section regarding an ellipse, written as

$$y^2 = 2 * p * x + (\varepsilon^2 - 1) * x^2\,,\; if\; 0 < \varepsilon < 1 \tag{3-33}$$



(fig. 3.6) Decentred ellipse [3]

Analogous to the ellipse the equation for a hyperbola centred at the origin is

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \tag{3-34}$$

As we shift the hyperbola with (+a) and solve the developed equation we get

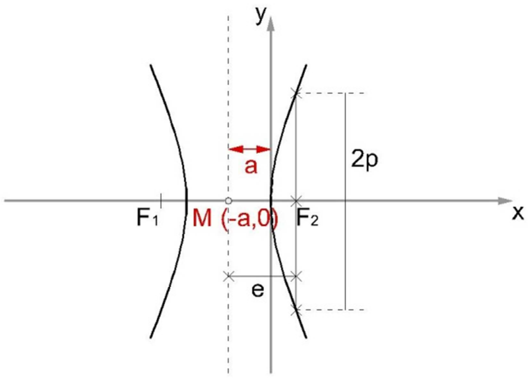
$$y^2 = 2 * \frac{b^2}{a} * x - \frac{b^2}{a^2} * x^2 \tag{3-35}$$

Adding the arguments

$$2p = \frac{2b^2}{a} \; and \; \varepsilon = \frac{\sqrt{a^2+b^2}}{a} \tag{3-36}$$

leads to the general vertex form of the equation of a conic section regarding a hyperbola, written as

$$y^2 = 2 * p * x + (\varepsilon^2 - 1) * x^2\,,\ if\ \varepsilon > 1 \tag{3-37}$$



(fig. 3.7) Decentred hyperbola [3]

If the general equation of a parabola known as

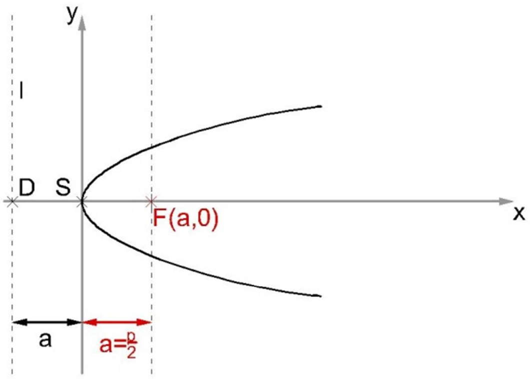
$$y^2 = 2 * p * x \tag{3-38}$$

is extended with

$$(\varepsilon^2 - 1) * x^2 \tag{3-39}$$

we also obtain the general vertex form of the equation of a conic section regarding a parabola, written as

$$y^2 = 2 * p * x + (\varepsilon^2 - 1) * x^2\,,\ if\ \varepsilon = 1 \tag{3-40}$$



(fig. 3.8) Parabola with horizontal axis of symmetry [3]

As established above the vertex form of the equation of conic section can be commonly written as

$$y^2 = 2 * p * x + (\varepsilon^2 - 1) * x^2 \qquad (3\text{-}41)$$

with varying parameter ε to take account of the different types of conic sections.

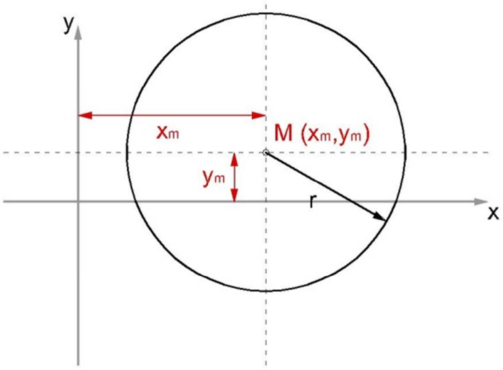
3.2.3 General form of the equation of a circle

Equation (3-25) describes a circle centred at the origin of the coordinate system. If the centre of the circle is shifted along the x axis as well as the y axis (fig.3.9) the equation must be modified to

$$r^2 = (x - x_m)^2 + (y - y_m)^2 \qquad (3\text{-}42)$$

Converting the formula to extract the y term leads to

$$\mathrm{y} = \sqrt{r^2 - (x - x_m)^2} - y_m \qquad (3\text{-}43)$$

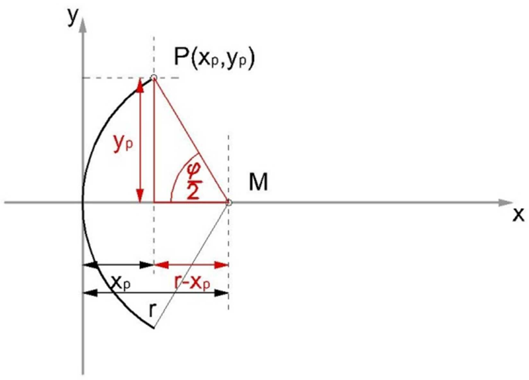


(fig. 3.9) Circle shifted along x and y axis [3]

3.2.4 Opening angle

The opening angle of an arc with the vertex at the origin of the coordinate system (fig.3.10) can be calculated from the arc tangent of the proportion of vertical and the horizontal distance of the endpoint to the centre of the arc and can be written as

$$\varphi = 2 * \arctan \frac{y_p}{r - x_p} \qquad (3\text{-}44)$$



(fig. 3.10) Opening angle of an arc [3]

3.2.5 Angle sum identities

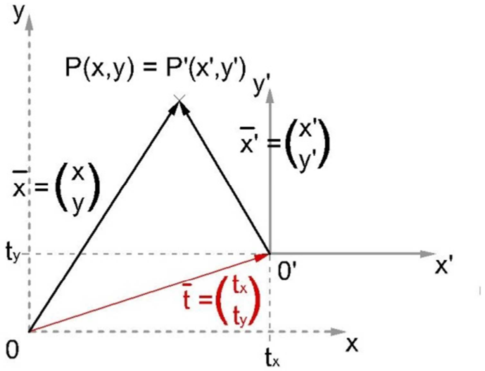
To obtain the sum of the tangent of two angles it is not possible to simply sum up the angles itself and form the tangent. Therefore angle sum identities have to be used. The angle sum identity for the tangent of two angles is written as

$$\tan(\alpha + \beta) = \frac{tan(\alpha)+tan(\beta)}{1-tan(\alpha)*tan(\beta)} \tag{3-45}$$

3.2.6 Coordinate transformation

To transform the coordinate system there are 2 possibilities given. Either the coordinate system can be translated or it can be rotated. Translation (fig.3.11) means that every point is moved for a certain amount and rotation (fig.3.12) describes the rotation through an angle ϕ around the origin of the coordinate system. The translation of the coordinate system can be written as

$$\bar{x'} = \bar{x} - \bar{t} \qquad \begin{cases} x' = x - t_x \\ y' = y - t_y \end{cases} \tag{3-46}$$



(fig. 3.11) Translation of the coordinate system [3]

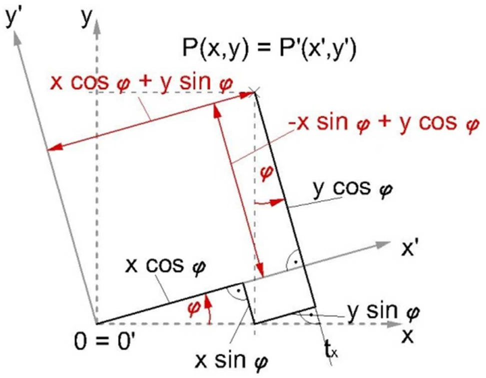
And to obtain a rotation of the coordinate system, the relation

$$\overline{\mathrm{x}'} = \overline{\overline{\mathrm{R}}} * \overline{\mathrm{x}} \quad \begin{cases} x' = x * \cos(\varphi) + y * sin(\varphi) \\ y' = -x * \sin(\varphi) + y * cos(\varphi) \end{cases} \tag{3-47}$$

with

$$\overline{\overline{\mathrm{R}}} = \begin{bmatrix} cos(\varphi) & sin(\varphi) \\ -sin(\varphi) & cos(\varphi) \end{bmatrix} \tag{3-48}$$

have to be used.

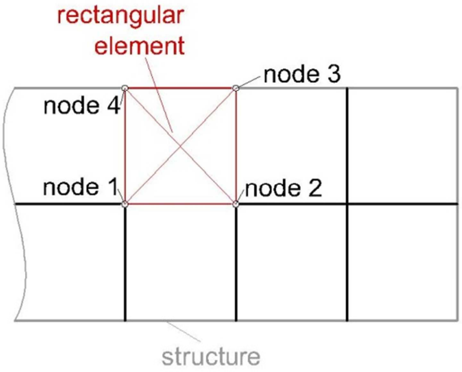


(fig. 3.12) Rotation of the coordinate system [3]

3.3 THE FINITE ELEMENT METHOD – FEM*[4][5][16]

The following chapter should give a brief introduction into Finite Element Method and explain its essential calculation steps. Due to the problem that it is hardly possible to calculate the strains and stresses of loaded structures, which have complex geometry, the method of the Finite Element, which provides an estimated solution of the problem, arises.
Walter Ritz, a Swiss scientist, first introduced an approximation procedure, which does not analytically solve Bernoulli's differential equation to calculate displacements but give an estimated solution by only considering geometric boundary and matching conditions. To calculate the strains and stresses out of the displacements he was using the principle of the minimum of potential energy. Ritz's method is able to solve mechanical problems as long as it is able to define a function, which fulfils all boundary conditions. Therefore the Finite Element Method, where the geometry is divided into partial areas also known as discretisation, was introduced, as it is easier to find a function, which fulfils the boundary conditions of partial areas as for the whole structure. Although this calculation method provides a solution of the problem and fulfils the conditions of compatibility it has to be mentioned that not all conditions of equilibrium are fulfilled exactly.

In addition the essential steps to calculate the stains and stresses of a 2-dimensional element are shown. As mentioned above the whole geometry has to be discretised (fig.3.13).



(fig. 3.13) Discretisation of a 2-dimensional structure

For the discretisation of the geometry of the whole structure, different types of elements can be used. To solve a 2-dimensional problem elements with rectangular shape as well as elements with triangular shape are used generally. Analogous to the 2-dimensional problem, tetrahedron or hexahedron elements are used for a 3-dimensional problem. To simplify the integration which is necessary to calculate the potential energy, a local coordinate system has to be introduced (fig.3.14). To transform the global coordinates of each node of each element the formula

$$\left\{x_{(\xi,\eta)}\right\}^{e} = \sum_{n} N_{n(\xi,\eta)} * \{x\}_{n}^{e} \tag{3-49}$$

with

$x_{(\xi,\eta)}$	.. local coordinates of each node
$N_{n(\xi,\eta)}$	.. shape function
x	.. global coordinates of each node
n	.. number of nodes of each element

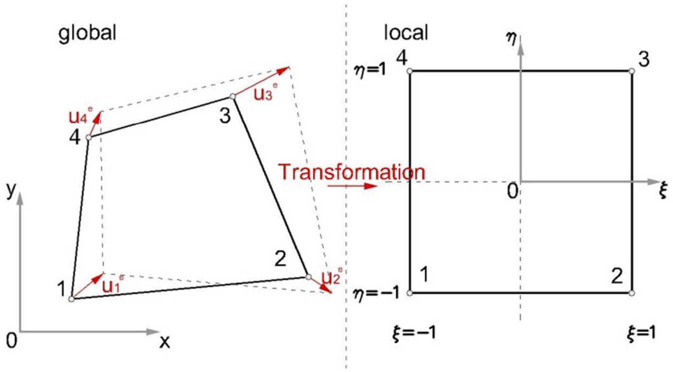
For the next step it is fundamental to describe the formulation for the displacements, which can be written as

$$\{u_{(\xi,\eta)}\}^e = \sum_n N_{n(\xi,\eta)} * \{u\}_n^e \tag{3-50}$$

with

$u_{(\xi,\eta)}$	.. local displacement of the node
u	.. global displacement of each node

The used shape function has a lower order than the analytically solution, therefore the element become stiffer. When using the same shape function to calculate the displacements of an element as for the transformation to the local coordinate system the element is called isoparametrical element.



(fig. 3.14) 2-dimensional element in the global and local coordinate system [5]

As the displacements are defined the strains and can be formulated as

$$\varepsilon_x = \frac{\delta u_x}{\delta x} = \sum_n \frac{\delta N_{n(\xi,\eta)}}{\delta x} * \{u_x\}_n^e \tag{3-51}$$

$$\varepsilon_y = \frac{\delta u_y}{\delta y} = \sum_n \frac{\delta N_{n(\xi,\eta)}}{\delta y} * \{u_y\}_n^e \tag{3-52}$$

and

$$\gamma_{xy} = \frac{\delta u_x}{\delta y} + \frac{\delta u_y}{\delta x} = \sum_n \frac{\delta N_{n(\xi,\eta)}}{\delta y} * \{u_x\}_n^e + \sum_n \frac{\delta N_{n(\xi,\eta)}}{\delta x} * \{u_y\}_n^e \tag{3-53}$$

As sum the strains of each direction x and y up to a vector, we need to introduce a so called B matrix, which contains the derivatives of the shape functions.

$$\mathbf{B} = \begin{bmatrix} \frac{\delta N_{1(\xi,\eta)}}{\delta x} & 0 & \dots \\ 0 & \frac{\delta N_{1(\xi,\eta)}}{\delta y} & \dots \\ \frac{\delta N_{1(\xi,\eta)}}{\delta y} & \frac{\delta N_{1(\xi,\eta)}}{\delta x} & \dots \end{bmatrix} \tag{3-54}$$

With the B-matrix the strains can be written as vector of the following form

$$\{\varepsilon\}^e = [B]^e * \{u\}^e \tag{3-55}$$

As it is not possible to derive the local shape function from above by global dx and dy the Jacobi matrix of the form

$$\mathbf{J} = \begin{bmatrix} \frac{\delta x}{\delta \xi} & \frac{\delta y}{\delta \xi} \\ \frac{\delta x}{\delta \eta} & \frac{\delta y}{\delta \eta} \end{bmatrix} \tag{3-56}$$

has to be established. For the next step it is necessary to calculate the stresses of each element. As the strains are known the stresses can be calculated after Hook’s law with

$$\sigma_x = \frac{E}{(1-\nu^2)} * (\varepsilon_x + \nu * \varepsilon_y) \tag{3-57}$$

$$\sigma_y = \frac{E}{(1-\nu^2)} * (\varepsilon_y + \nu * \varepsilon_x) \tag{3-58}$$

and

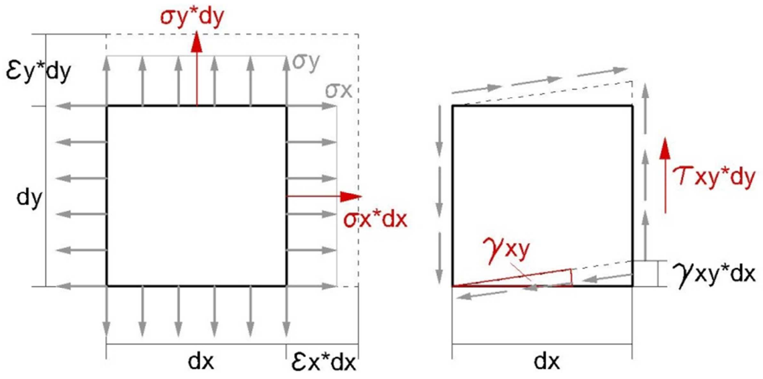
$$\tau_{xy} = \frac{E}{2*(1+\upsilon)} * \gamma_{xy} \tag{3-59}$$

with

E	[MPa]	.. Young’s modulus
υ	[-]	.. Poisson ratio

Analogous to the strains the stresses can be written as vector. Therefore the material dependant parameters E and v have to be separated from the strains. To segregate the material dependant parameters a so called D-matrix is introduced.

$$\mathbf{D} = \frac{E}{(1-\nu^2)} * \begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & \frac{(1-\upsilon)}{2} \end{bmatrix} \tag{3-60}$$



(fig. 3.15) Stresses of a 2-dimensional element [5]

As mentioned in the introduction the principle of the minimum of the potential energy is used for the calculation. For this reason it is required to formulate the internal and external potential energy. The external potential energy can be written as

$$\Pi_{ext} = \{P\}^e * \{u\}^e \tag{3-61}$$

with

P	[MN]	.. External forces

and the internal potential energy which consist of the stresses and strains formulated above, written as

$$\Pi_{int} = \frac{1}{2} * \int_{-1}^{1} \int_{-1}^{1} \{\varepsilon\}^T * \{\sigma\} * [J] * d\xi d\eta \tag{3-62}$$

To calculate the displacements as well as the strains and stresses the internal and external potential energy has to be summed up. To fulfil the conditions of equilibrium the minimum of the potential energy have to be built by deriving the potential energy by the displacement and set it equal to zero.

$$\frac{\Pi}{\delta\{u\}^e} = \frac{\Pi_{int}}{\delta\{u\}^e} + \frac{\Pi_{ext}}{\delta\{u\}^e} = 0 \tag{3-63}$$

The equation above can be substituted with

$$\frac{\Pi_{int}}{\delta\{u\}^e} = [K]^e * \{u\}^e \tag{3-64}$$

and

$$\frac{\Pi_{ext}}{\delta\{u\}^e} = \{P\}^e \tag{3-65}$$

and therefore written as

$$[K]^e * \{u\}^e - \{P\}^e = 0 \tag{3-66}$$

with [K], also known as stiffness matrix, is

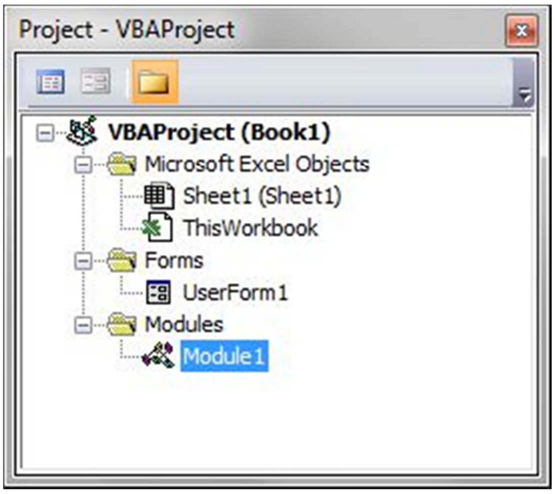
$$[K] = \int_{-1}^{1}\int_{-1}^{1}[B]^T * [D] * [B] * [J] * d\xi d\eta \tag{3-67}$$

As the stiffness of an element as well as its external forces are known the displacements of the element can be calculated from. Further the strains and stresses can be computed from these displacements.

4 VBA BASED PROGRAMMING

4.1 VBA – VISUAL BASIC FOR APPLICATIONS - BASICS*[10]

To understand the programming with VBA a fundamental declaration is needed. As VBA is an object orientated programming language, a VBA program contains Objects, which allow the user to navigate through the program and Code, which is in the background of the program and comprises all functions, calculations and declarations. Further a VBA program is summarized in the project explorer, where all objects and code parts are listed. Figure 4.1 shows the project explorer of a default project.



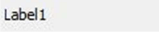
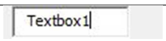
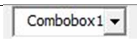
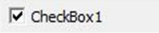
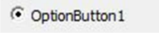
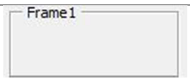
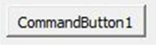
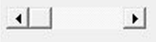
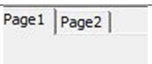
(fig. 4.1) The VBA project explorer

4.1.1 OBJECTS

As mentioned above objects are used to navigate through the program. Examples of objects are textboxes, where text and numbers can be put in, command buttons, which are used to execute code, checkboxes and radio buttons, which are used to set possible options true or false, or list boxes, which can be used as table with rows and columns. The behaviour of these objects can be manipulated with predefined properties and methods. The Properties of an object are shown in the corresponding properties window. Further each object can be allocated with events, which execute parts of the code as the condition of the event is fulfilled. The functional principle of the properties, method and event syntax is explained in the sub chapters below.

Another type of objects are forms, which are used to summarize and arrange objects within a window, which represent the user interface. These forms can be considered as parent objects, whereas objects integrated in a form can be considered as child objects.

Table 4-1: Child objects of VBA

Control type	Appearance	Function
Label		Labels show descriptions or calculation results, which should not be editable.
Textbox		Textboxes are used to put in numbers or text, which should be computed.
Combo Box		Combo boxes are some kind of dropdown menu. They are used for the selection of different given entries.
Checkbox		Checkboxes are used to mark or unmark possible given options.
Radio Button		Radio Buttons are used set possible options true or false. As opposed to Checkboxes at least one option has to be set true.
List box		List boxes are useful to show tables.
Frame		Frames summarize objects. Option Buttons, which belong together, should be within a frame.
Command Button		Command Buttons are used to execute code. They are the most commonly control used in combination with the “Click” event.
Scroll bar		Scroll bars are used to change values from a given minimum to a given maximum. To scroll the screen up and down is the most commonly use of scroll bars
Picture box		Picture boxes are used to show pictures.
Multi Page		Multi pages are used within a user form, to show different pages. That is useful to ensure efficient navigation and reduce the amount of user forms.

4.1.1.1 Properties

The properties syntax *object.property* enables the possibility to change the properties of an object. The name of an object, its caption and font as well as height and width or back colour are some of the properties of an object. It has to be mentioned that each object has default properties for the initialization phase of the program but can later be changed within the program to enable or disable the object for example.

property syntax example

ParentObject.ChildObject.Property=Definition

arch_dam_design.label_bael.Caption = "Base elevation:"

4.1.1.2 Methods

The method syntax *object.method* is used to perform object actions. The Set Focus, Move or Hide as well as the Cut, Copy or Paste methods are some of them. The example bellow shows the Set Focus method, which is used to set the focus to the “label_bael” child object.

method syntax example

ParentObject.ChildObject.Method

arch_dam_design.label_bael.SetFocus

4.1.1.3 Events

Events are used to execute some parts of the code. The Click or MouseMove event can be instanced as example. With the event syntax *object_event*, a condition is set, which triggers the code underneath this event, if the condition is fulfilled. The example below shows the Click event, which is enabled as the user clicks on the object linked to this event.

event syntax example

Declaration ChildObject_Event

Private Sub tool_csv_Click()

4.1.2 Code

The code of a VBA program contains the syntax referred to objects as well as the declaration of the variables or functions, which are used for processing. The declaration of variables is needed to specify the variables, which are used within the program code. Beside the Prefix Public or Dim the declaration comprises the identifier, which is a unique label and the variable type, which defines the type of the variable. Integer, String or Boolean as well as Single or Double are such variable types. The use of functions, such as If-Else or For-Next as well as Do-Until is essential to verify given conditions or to start loops, which are needed for the calculation.

Code can also be provided within modules, which represent special project elements. Beside code, which is embedded in parent objects and does only refer to this object, code within Modules contain, so called sub functions or global declarations, which are declared as public to be available for all objects of a project.

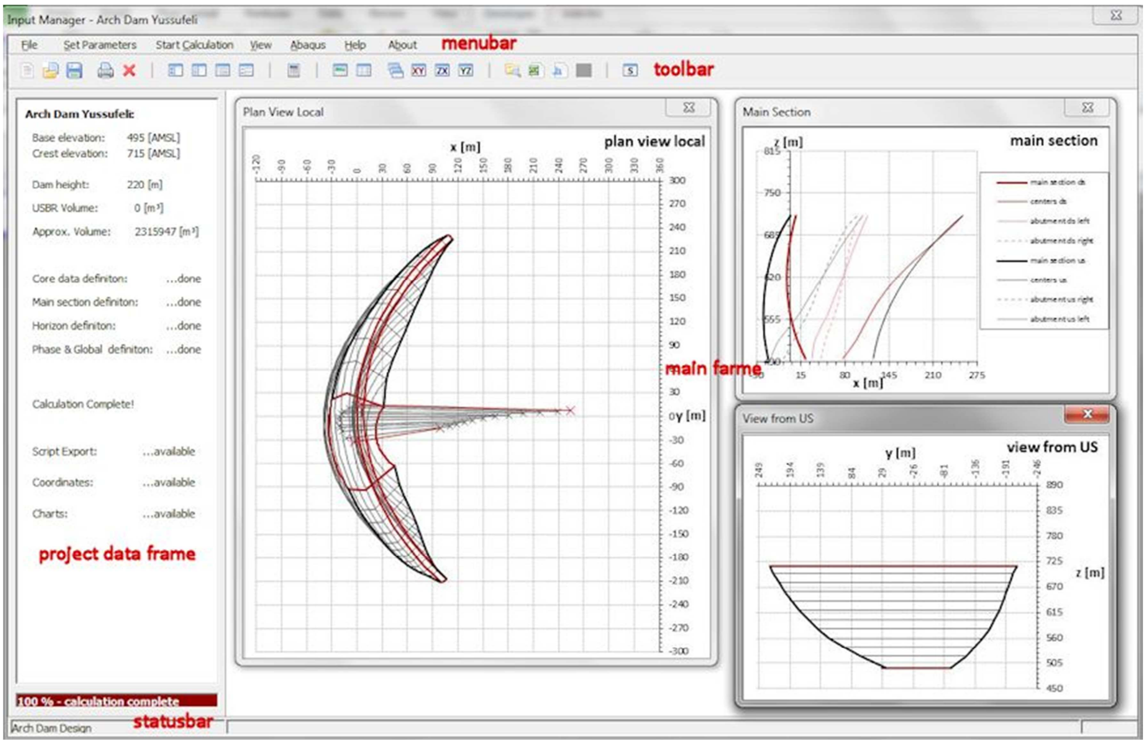
To gain detailed information on VBA programming specialised literature should be consulted.

4.2 PROGRAM SEQUENCE OF THE ARCH DAM DESIGN – INPUT MANAGER*[3][6][7][13]

The program, created for the calculation of the 3D coordinates of an arch dam, is called Arch Dam Design - Input Manager, briefly ADD. ADD is divided into several forms containing different types of child objects. The forms are the main form, four forms to do the input and three forms for the post processing. How to use the different forms and child objects is explained in the chapters below.

4.2.1 MAIN USER FORM

The main user form comprises the menu- and the toolbar for the navigation through the program as well as the project data frame, which shows some information concerning the current project and the main frame, which contain the sub user forms of the program. At the bottom of the main user form the status bar give some information on the current project progress (fig.4.2).



Input Manager - Arch Dam Yussufeli

Arch Dam Yussufeli:

Base elevation: 495 [AMSL]
Crest elevation: 715 [AMSL]

Dam height: 220 [m]

USBR Volume: 0 [m³]

Approx. Volume: 2315947 [m³]

Core data definiton: ...done

Main section definiton: ...done

Horizon definiton: ...done

Phase & Global definiton: ...done

Calculation Complete!

Script Export: ...available

Coordinates: ...available

Charts: ...available

100 % - calculation complete

Arch Dam Design

(fig. 4.2) The main user form of ADD

4.2.1.1 THE MENU BAR

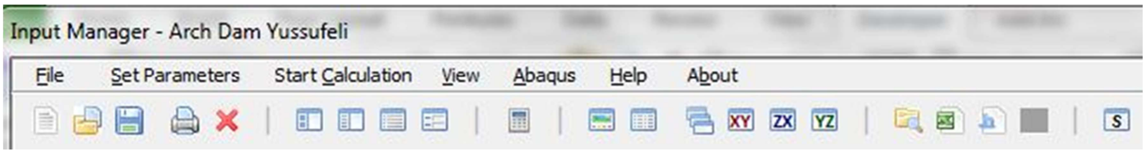
As known from other programs the menu bar can be used by moving the mouse over on the entries to drop down a submenu, which contains objects to call different program functions. From left to the right the entries in the menu bar are, File, Set Parameters, Start Calculation, View, Abaqus, Help and About. The different objects of the menu entries and the called functions are listed in the table below.

Table 4-2: Menu bar entries

File	
New	Create a new project
Import	Import project from existing .add file
Export	Export current project to .add file
Print	Print input data
Close	Close main user form
Exit	Exit program
Set Parameters	
Core Data Definition	Open the Core Data sub form
Main Section Definition	Open the Main Section Definition sub form
Horizon Definition	Open the Horizon Definition sub form
Phase & Global Definition	Open the Phase & Global Definition sub form
Start Calculation	Start Calculation
View	
View 3D Wireframe	Show 3D Wireframe of the arch dam
View Coordinates	Show calculated coordinates of each horizon
View All Charts	Show all 3 Charts
View Plan View	Show the plan view of the arch dam
View Main Section	Show the main section of the arch dam
View US Surface	Show the upstream surface of the arch dam
Abaqus	
Set Abaqus Directory	Set directory of the Abaqus exe file (e.g.:abq69ef1.bat)
Write CSV	Export a CSV file containing the calculated coordinates
Write Abaqus Script	Export a script to create a solid in Abaqus
Start Abaqus	Start Abaqus (directory has to be set!)
View Stresses	View arch stresses caused by hydrostatic pressure
Help	Open help file
About	Show recent version number as well as program author

4.2.1.2 THE TOOLBAR

The toolbar (fig.4.3) provides the same objects which are used in the menu bar. As opposed to the menu bar the different events are called by clicking on the appropriate symbol in the toolbar. The table below shows the different symbols with the corresponding functions.



(fig. 4.3) The menu and the toolbar of the program

Table 4-3: Toolbar entries

Symbol	Event
📄	Create a new project
📂	Import project from existing .add file
💾	Export current project to .add file
🖨	Print input data
❌	Exit program
▤	Open the Core Data sub form
▤	Open the Main Section Definition sub form
▤	Open the Horizon Definition sub form
▤	Open the Phase & Global Definition sub form
🖩	Start Calculation
🖼	Show 3D Wireframe of the arch dam
▦	Show calculated coordinates of each horizon
🗗	Show all 3 Charts
XY	Show the plan view of the arch dam
ZX	Show the main section of the arch dam
YZ	Show the upstream surface of the arch dam
🔍	Set directory of the Abaqus exe file (e.g.:abq69ef1.bat)
📊	Export a CSV file containing the calculated coordinates
📄	Export a script to create a solid in Abaqus
>_	Start Abaqus (directory has to be set!)
S	View arch stresses caused by hydrostatic pressure

4.2.1.3 THE PROJECT DATA FRAME

The project data frame gives an overview of the core data of the current project. The base and crest elevation of the arch dam, the height as well as the volume is listed on the left side of the window. Beside the core data of the arch dam, the current state of the input and, as the calculation is done, the availability of the script export, the coordinates and the charts of the arch dam is shown in the data frame too

4.2.1.4 THE MAIN FRAME

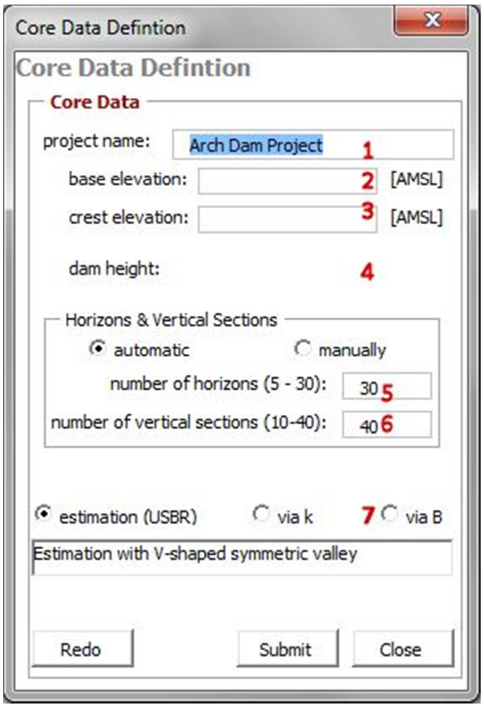
The main frame contains all the forms of the program. The input forms such as the core data form, the main section definition form, the horizon definition form and the phase & global definition form as well as the charts, which display the plan view, the main cross section and the upstream surface are shown within this frame. To do the post processing the main frame comprises the coordinates, 3D wireframe and the arch stresses form too.

4.2.1.5 THE STATUS BAR

The status bar shows the name of the program and the progress bar. The progress bar gives an overview of the current input progress as well of the progress during the calculation of the coordinates. It is scaled from zero to 100 percent.

4.2.2 CORE DATA DEFINITION

The Core Data form (fig.4.4) can be opened by clicking on the corresponding button in the menu or toolbar.



project name: Arch Dam Project

base elevation: [AMSL]

crest elevation: [AMSL]

dam height:

Horizons & Vertical Sections

● automatic ○ manually

number of horizons (5 - 30): 30

number of vertical sections (10-40): 40

● estimation (USBR) ○ via k ○ via B

Estimation with V-shaped symmetric valley

(fig. 4.4) The core data form

To do the core data definition it is necessary to put in a project name, the base elevation and the crest elevation of the arch dam as well as the number of horizons, the number of vertical sections and the calculation method.

The project name (1) is the name of your current project. A unique name has to be chosen for the project name due to the fact, that all the created files like charts, scripts and CSV files would get this name as default save name with its appropriate filename extension.

The base elevation (2) is the elevation at the base of the arch dam. To calculate the coordinates of the arch dam from a relative elevation level, zero has to be put in for the base elevation. To gain absolute coordinates the absolute elevation above mean sea level [AMSL] has to be inserted for the base elevation. It has to be mentioned that no elevation level below zero is allowed within the base elevation text box.

The crest elevation (3) is the elevation at the crest of your arch dam structure. The elevation of the crest of the arch dam has to be submitted corresponding to the input of the base elevation. As described above either from a relative or an absolute level.

The dam height (4) is calculated from your base and crest elevation. As the crest elevation is determined with a relative elevation level, the dam height corresponds to the crest elevation, while the dam height is calculated as the difference between the crest and the base elevation, when the input is done with absolute elevation levels.

$$\mathrm{h}_{dam} = elev_{crest} - elev_{base} \tag{4-1}$$

with

h_{dam}	[m]	.. dam height
$elev_{crest}$	[m]	.. crest elevation
$elev_{base}$	[m]	.. base elevation

The number of horizons (5) determines the quantity of calculated horizontal sections. A horizontal section, or horizon, is composed of a closed horizontal polygon with coordinates of the same elevation. Further a horizon is used as cross section of a certain elevation level of the arch dam. At least 5 and at most 30 horizons has to be determined for the calculation of the coordinates

As the option manually is chosen, the distance between the horizons can be inserted manually, whereas the option automatic calculates the distance between the horizons from the dam height divided by the number of horizons. It has to be remarked that more horizons refine the geometry of the arch dam but slow the calculation.

The number of vertical sections (6) determines the quantity of sections of the upstream and downstream arch of each horizon. The more vertical sections are chosen the more coordinates a calculated for the upstream and downstream arch. At least 10 and at most 40 sections can be determined for each arch, which means that the arch of the upstream and downstream surface is divided into that number of sections. As 40 sections are chosen 41 coordinates are calculated for the each arch and two phase coordinates for each abutment, which gives a maximum amount of 84 calculated coordinates per horizon.

There are three different calculation methods (7), to investigate the coordinates of an arch dam, which are the estimation obeying the rules of the USBR design guide, the calculation with focus on the input parameter k and the calculation with focus on the parameter B. The three different methods differ from each other in the amount of input and the type of the parameters, which are used for the calculation.

The estimation, also called the basic mode presupposes the assumption of a V-shaped symmetric valley. The basic mode can be used to pre calculate your project and get default parameters to detail your project with expert mode later.

The so called expert mode enables the input of asymmetric dam geometry. The option via k or via B determines the type of the input parameters only, but does not have any influence on the amount of the input. As the input via k is chosen the inclination of your abutment illustrates the central parameter whereas as the input via B requires the width of the valley at the elevation of each horizon. Detailed information on the different parameters is provided within chapter 4.2.4 horizon definition.

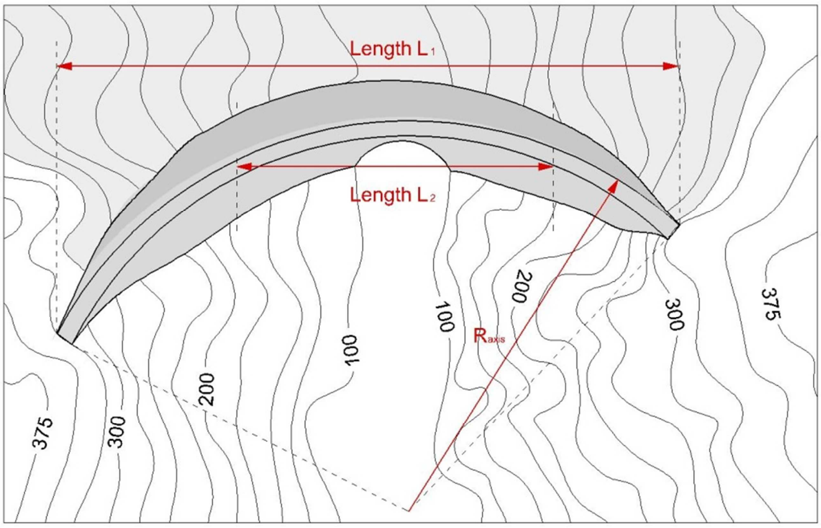
4.2.3 MAIN SECTION DEFINITION

As the core data definition is done, the main section definition is available. The main section definition form can be opened by clicking on the corresponding button in the menu or toolbar.

The main section of an arch dam represents the vertical cross section at the axis of the dam. To determine this cross section the x coordinates of the upstream and downstream surface and the thickness of the dam as well as the horizontal downstream and upstream radii has to be set for each horizon.

4.2.3.1 THE MAIN SECTION DEFINITION USING THE BASIC MODE

The USACE - US Army Corps of Engineers developed a guide to generate the main section depending on the width of the valley (fig.4.5) at the crest (L1) and 15 percent of the height of the dam (L2). The coordinates of the surface and the thickness of the dam will calculated from these parameters.



(fig. 4.5) Definition of the lengths L1 and L2 and the radius R_{axis} [13]

As the height of the dam as well as the lengths L1 and L2 are known the thickness of the arch dam can be calculated with

$$T_C = 0.01 * (H + 1.2 * L_1) \tag{4-2}$$

$$T_B = 0.10626586 * \sqrt[3]{H * L_1 * L_2 \left(\frac{3.28*H}{400}\right)^{\left(\frac{3.28*H}{400}\right)}} \tag{4-3}$$

$$T_{0.45} = 0.95 * T_B \tag{4-4}$$

with

T_C	[m]	.. thickness of the dam at the crest
T_B	[m]	.. thickness of the dam at the base
$T_{0.45}$	[m]	.. thickness of the dam at 45 percent of the dam height
L_1	[m]	.. width of the valley at the crest
L_2	[m]	.. width of the valley at 15 percent of the dam height
H	[m]	.. dam height USBR

When the thickness of the dam is known the downstream coordinates are

$$\mathrm{DSP}_{crest} = T_C \tag{4-5}$$

$$\mathrm{DSP}_{base} = 0.33 * T_B \tag{4-6}$$

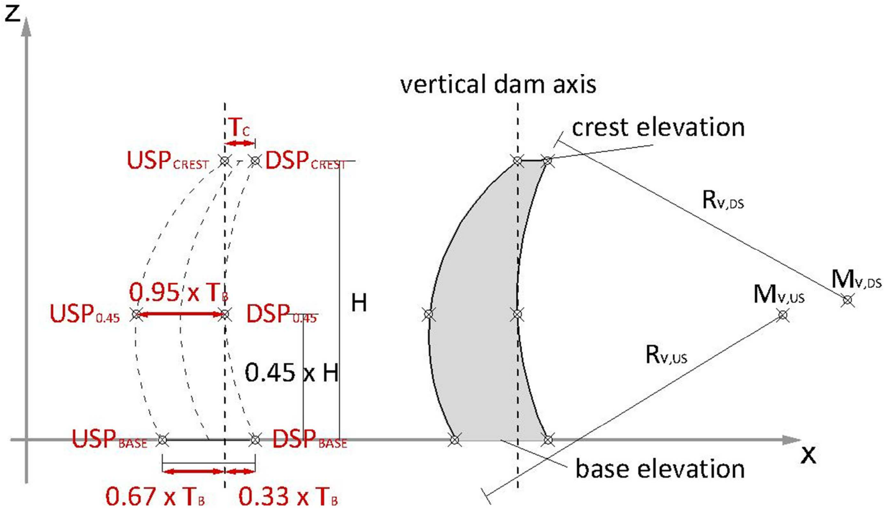
$$\mathrm{DSP}_{0.45} = 0 \tag{4-7}$$

and the upstream coordinates can be written as

$$\mathrm{USP}_{crest} = 0 \tag{4-8}$$

$$\mathrm{USP}_{base} = 0.67 * T_B \tag{4-9}$$

$$\mathrm{USP}_{0.45} = 0.95 * T_B \tag{4-10}$$



(fig. 4.6) Coordinates definition of the main section [13]

After the definition of the thickness and the main section coordinates the determination of the horizontal upstream and downstream radii has to be done. To gain the radius of the upstream surface at the crest a correlation with the length L1 can be written as

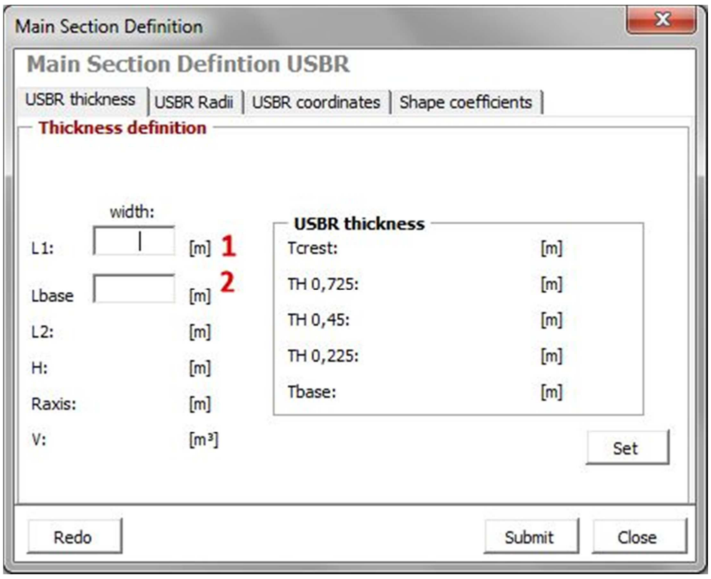
$$\mathrm{R}_{axis} = 0.6 * L_1 \tag{4-11}$$

with

R_{axis} [m] .. radius of the upstream surface at the crest

To obtain the remaining radii of the upstream and downstream surface, the principle of trial and error must be applied. For detailed information on the rules of design the USBR design guide should be consulted

Further within the program ADD the main section definition (fig.4.7) is depending on the length L1 too, but uses the length L_{base}, which represents the width of the valley at the base of the arch dam, instead of the length L2. As there is a linear correlation between L_{base} and L2, caused by the symmetric V-shaped valley, the equations (4-2) to (4-11) from the USBR guide can be used within ADD. It has to be noted that the input, which is given in metres has to be converted to feet.



Main Section Definition

Main Section Defintion USBR

Thickness definition

width:

L1: [m] 1

Lbase [m] 2

L2: [m]

H: [m]

Raxis: [m]

V: [m³]

USBR thickness

Tcrest: [m]

TH 0,725: [m]

TH 0,45: [m]

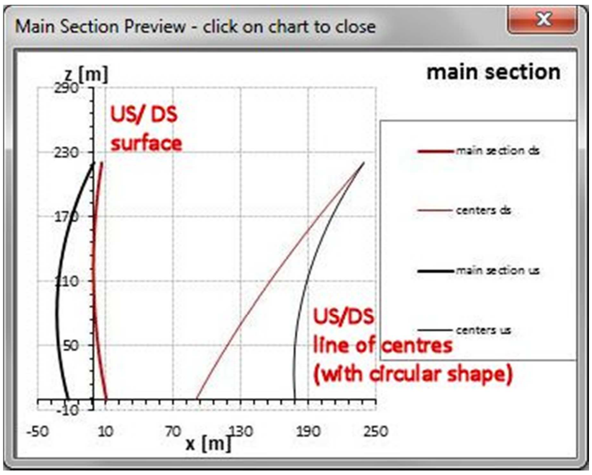
TH 0,225: [m]

Tbase: [m]

(fig. 4.7) Basic mode main section definition form using the basic mode

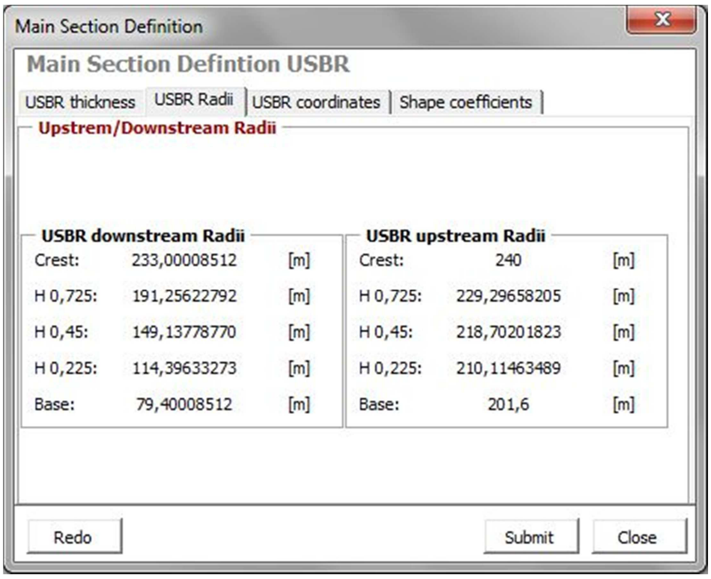
As the parameters L1 (1) and L_{base} (2) are determined the requested parameters for the thickness, coordinates of the surfaces and radii are calculated automatically at five elevations. As mentioned above three of those elevations are the crest, the base and the elevation at 45 percent of the dam height. The remaining two elevations are at 22.5 percent of the dam height and 72.5 percent of the dam height. It is necessary to calculate these parameters at five elevations, to enable a 4th degree polynomial regression.

As a result of the polynomial regression, the regression coefficients, which are also called shape coefficients, determine the shape of the upstream and downstream surface as well as the shape of the upstream and downstream line of centres, which allows the calculation of those parameters at any height of the dam. The mentioned shapes are displayed within a preview screen after the input of the parameters L1 and L_{base} (fig.4.8).



(fig. 4.8) Preview screen within the main section definition form

Beside the first register of the main section definition form, where the input can be done and the calculated thicknesses of the dam are listed, there are three other registers. Within the second register the upstream and downstream radii are listed (fig.4.9).



USBR downstream Radii			USBR upstream Radii		
Crest:	233,00008512	[m]	Crest:	240	[m]
H 0,725:	191,25622792	[m]	H 0,725:	229,29658205	[m]
H 0,45:	149,13778770	[m]	H 0,45:	218,70201823	[m]
H 0,225:	114,39633273	[m]	H 0,225:	210,11463489	[m]
Base:	79,40008512	[m]	Base:	201,6	[m]

(fig. 4.9) Basic mode main section definition form showing the calculated radii

As noted above, the USBR design guide follows the principle of trial and error concerning the radii definition, which is obviously not useful within a program. Due to that fact a radii definition is processed within the program code based on the shape of the line of centres of known arch dam projects. With the assumptions

$$\mathrm{R}_{US,crest} = R_{axis} \tag{4-12}$$

$$\mathrm{R}_{US,base} = \left(\frac{4}{5} * R_{axis} + \frac{1}{5} * \frac{L_{base}}{L_1} * R_{axis}\right) \tag{4-13}$$

$$\mathrm{R}_{DS,crest} = R_{axis} - T_C \tag{4-14}$$

$$\mathrm{R}_{DS,base} = \left(\frac{1}{5} * R_{axis} + \frac{4}{5} * \frac{L_{base}}{L_1} * R_{axis}\right) - T_c \tag{4-15}$$

with

L_{base}	[m]	.. width of the valley at the base of the arch dam
$\mathrm{R}_{US,crest}$	[m]	.. upstream radius at the crest
$\mathrm{R}_{US,base}$	[m]	.. upstream radius at the base
$\mathrm{R}_{DS,crest}$	[m]	.. downstream radius at the crest
$\mathrm{R}_{DS,base}$	[m]	.. downstream radius at the base

the upstream and downstream radii at the crest and the base represent two known points of the upstream and downstream line of centres. For the development of the shape of the line of centres, circles with two known points and the radii

$$\mathrm{R}_{DS,LC} = 400 * \frac{H * L_{base}}{L_1} \tag{4-16}$$

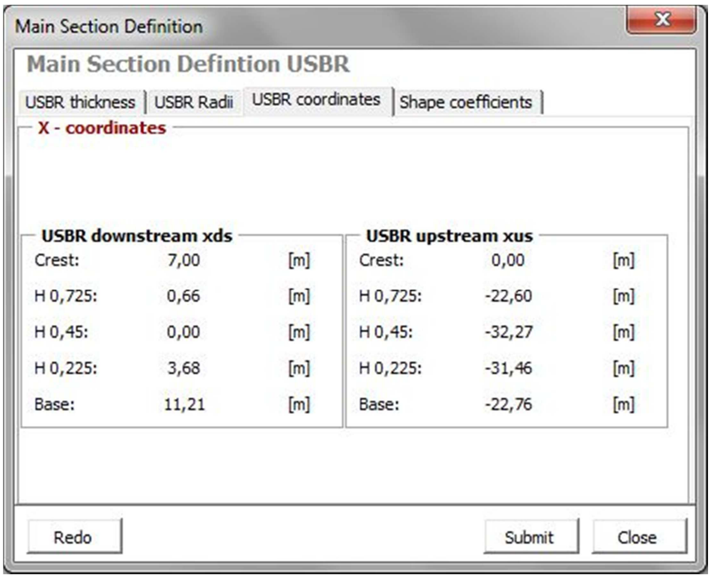
$$\mathrm{R}_{US,LC} = 800 * \frac{H * L_{base}}{L_1} \tag{4-17}$$

with

$\mathrm{R}_{DS,LC}$	[m]	.. radius of the downstream line of centres
$\mathrm{R}_{US,LC}$	[m]	.. radius of the upstream line of centres

are applied. It has to be mentioned that the definition of the line of centres accomplishes the requirements of the preliminary design, but should later be edited to detail the arch dam project.

The third register shows the calculated coordinates of the upstream and downstream surface (fig.4.10).

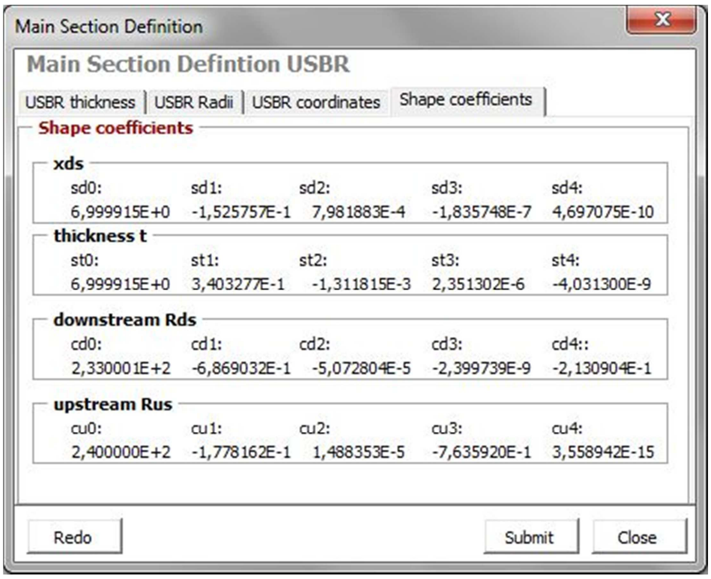


X - coordinates

USBR downstream xds			USBR upstream xus		
Crest:	7,00	[m]	Crest:	0,00	[m]
H 0,725:	0,66	[m]	H 0,725:	-22,60	[m]
H 0,45:	0,00	[m]	H 0,45:	-32,27	[m]
H 0,225:	3,68	[m]	H 0,225:	-31,46	[m]
Base:	11,21	[m]	Base:	-22,76	[m]

(fig. 4.10) Basic mode main section definition form showing the calculated coordinates

The fourth and last register of the main section definition form using the basic mode shows the coefficients of the 4th degree polynomial regression (fig.4.11).



Shape coefficients

xds

sd0:	sd1:	sd2:	sd3:	sd4:
6,999915E+0	-1,525757E-1	7,981883E-4	-1,835748E-7	4,697075E-10

thickness t

st0:	st1:	st2:	st3:	st4:
6,999915E+0	3,403277E-1	-1,311815E-3	2,351302E-6	-4,031300E-9

downstream Rds

cd0:	cd1:	cd2:	cd3:	cd4::
2,330001E+2	-6,869032E-1	-5,072804E-5	-2,399739E-9	-2,130904E-1

upstream Rus

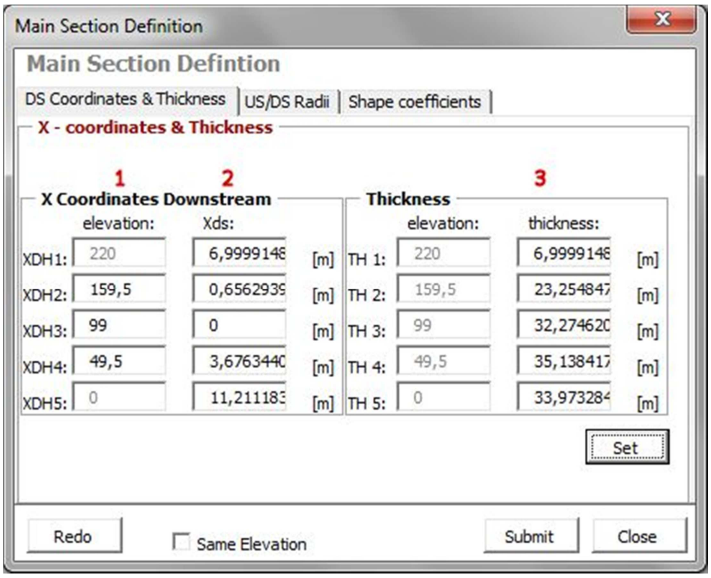
cu0:	cu1:	cu2:	cu3:	cu4:
2,400000E+2	-1,778162E-1	1,488353E-5	-7,635920E-1	3,558942E-15

(fig. 4.11) Basic mode main section definition form showing the shape coefficients

4.2.3.2 THE MAIN SECTION DEFINITION USING THE EXPERT MODE

Beside the main section definition using the basic mode, which is only appropriate to do the preliminary design of arch dam, due to the fact that the USBR design guide primarily uses circles for the shape of the arch dam, which do not represent the optimal arch dam geometry, the expert mode provides the input of the main section definition parameters to be manually and therefore more suitable for the design of an arch dam. But using the expert mode efficiently, expect the arch dam project to be pre calculated using the basic mode to gain default parameters within the expert main definition form.

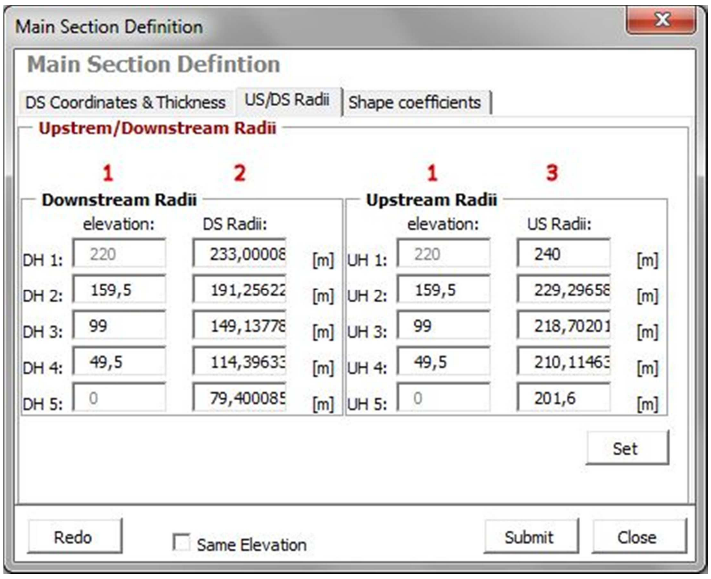
As the main section form is opened (fig.4.12), the user is constrained to put in the elevation (1) as well as the downstream surface coordinates (2) and the thickness (3) of five horizons within the first register. Caused by the fact that the elevation of the crest and the base with the corresponding coordinate and thickness parameters are needed to do the 4th degree polynomial regression, the input fields are locked to prevent input.



X Coordinates Downstream	elevation:	Xds:		Thickness	elevation:	thickness:	
XDH1:	220	6,9999148	[m]	TH 1:	220	6,9999148	[m]
XDH2:	159,5	0,6562939	[m]	TH 2:	159,5	23,254847	[m]
XDH3:	99	0	[m]	TH 3:	99	32,274620	[m]
XDH4:	49,5	3,6763440	[m]	TH 4:	49,5	35,138417	[m]
XDH5:	0	11,211183	[m]	TH 5:	0	33,973284	[m]

(fig. 4.12) Expert mode main section definition form showing the first register

As the coordinate and thickness definition is done, the elevation (1) as well as the upstream (3) and downstream (2) radii have to be set manually within the second register (fig.4.13). According to the input of first register the elevation with the corresponding upstream and downstream radii has to be set manually, obeying the fact that the input fields of the crest and base elevation are locked too.



Main Section Definition

Main Section Defintion

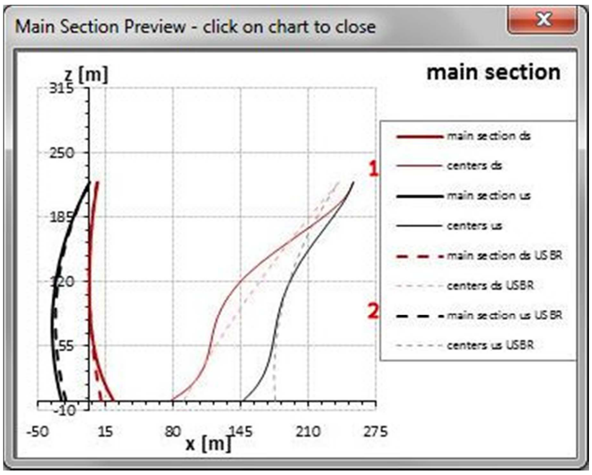
DS Coordinates & Thickness | US/DS Radii | Shape coefficients

Upstrem/Downstream Radii

Downstream Radii	elevation:	DS Radii:		Upstream Radii	elevation:	US Radii:	
DH 1:	220	233,00008	[m]	UH 1:	220	240	[m]
DH 2:	159,5	191,25622	[m]	UH 2:	159,5	229,29658	[m]
DH 3:	99	149,13778	[m]	UH 3:	99	218,70201	[m]
DH 4:	49,5	114,39633	[m]	UH 4:	49,5	210,11463	[m]
DH 5:	0	79,400085	[m]	UH 5:	0	201,6	[m]

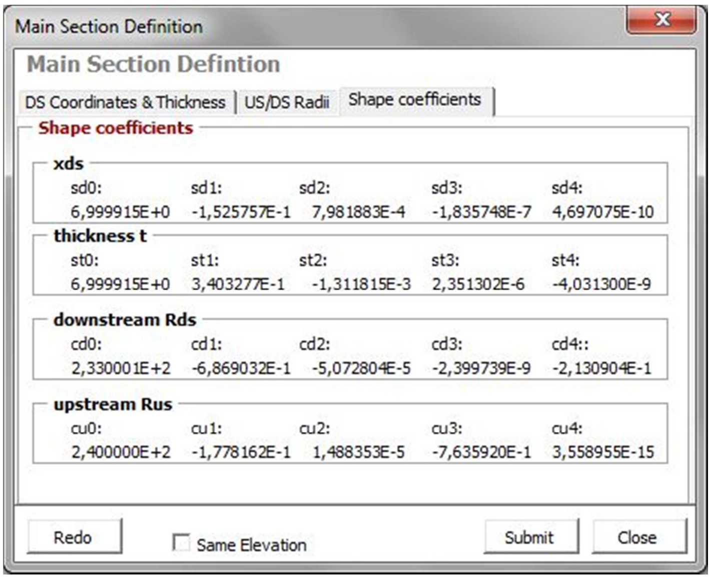
(fig. 4.13) Expert mode main section definition form showing the second register

As the input of the coordinates and the thickness as well as the upstream and downstream radii is done and submitted the main section definition preview screen shows the shape of the upstream and downstream surface as well as the upstream and downstream line of centres. The difference between shape of the expert mode input (1) and the shape of the basic mode input (2) is displayed by using a dashed line for shape of the basic mode input.



(fig. 4.14) Expert mode main section definition preview screen

At least, analogous to the basic mode, the last register of the expert mode shows the coefficients of the 4th degree polynomial regression (fig.4.15).



Shape coefficients

xds

sd0:	sd1:	sd2:	sd3:	sd4:
6,999915E+0	-1,525757E-1	7,981883E-4	-1,835748E-7	4,697075E-10

thickness t

st0:	st1:	st2:	st3:	st4:
6,999915E+0	3,403277E-1	-1,311815E-3	2,351302E-6	-4,031300E-9

downstream Rds

cd0:	cd1:	cd2:	cd3:	cd4::
2,330001E+2	-6,869032E-1	-5,072804E-5	-2,399739E-9	-2,130904E-1

upstream Rus

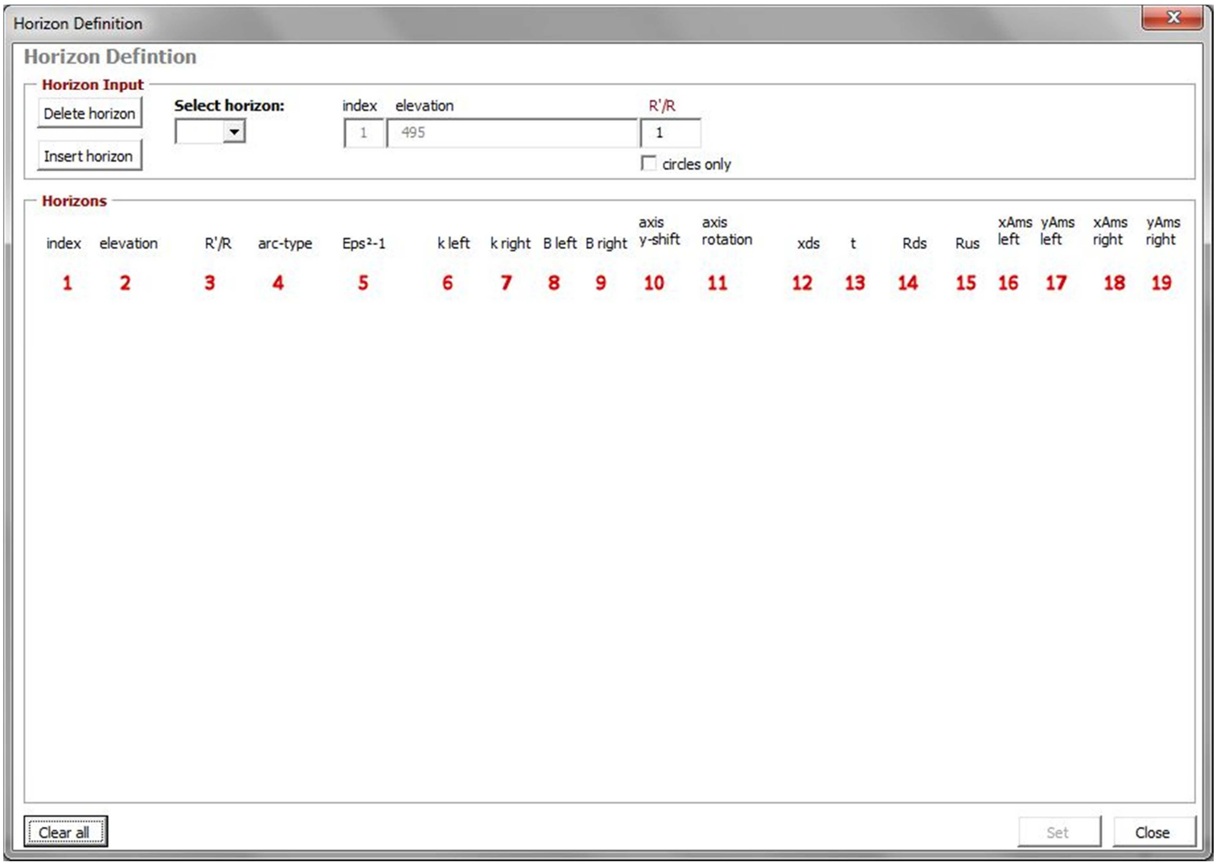
cu0:	cu1:	cu2:	cu3:	cu4:
2,400000E+2	-1,778162E-1	1,488353E-5	-7,635920E-1	3,558955E-15

(fig. 4.15) Expert mode main section definition form showing the third register

4.2.4 HORIZON DEFINITION

When the main section definition is done, the horizon definition form can be opened by clicking on the corresponding button in the menu or toolbar.

The Horizon definition (fig.4.16) provides all the parameters, which are needed for the calculation of the coordinates of the arch dam. Some of these parameters have to be put in, depending on the calculation method and some of the parameters are calculated automatically. The 19 parameters within the Horizon definition are listed below.


Horizon Definition

Horizon Defintion

Horizon Input

index 1

elevation 495

R'/R 1

□ circles only

(fig. 4.16) Horizon definition form

The index (1), the first column of the horizon definition table, represents the consecutive numbering of the horizons. Considering the index of the crest horizon, the index is equal to the number of horizons determined at the core data definition.

The second column contains the elevation (2) of each horizon. As the option automatically is chosen at the core data definition form, the difference in altitude between the horizons is calculated automatically. The difference in altitude can be written as

$$\Delta \mathrm{d}_{horizon} = \frac{h_{dam}}{n_{horizon-1}} \qquad (4\text{-}18)$$

with

$\Delta \mathrm{d}_{horizon}$	[m]	.. difference in altitude between each horizon
$n_{horizon}$	[-]	.. number of horizons

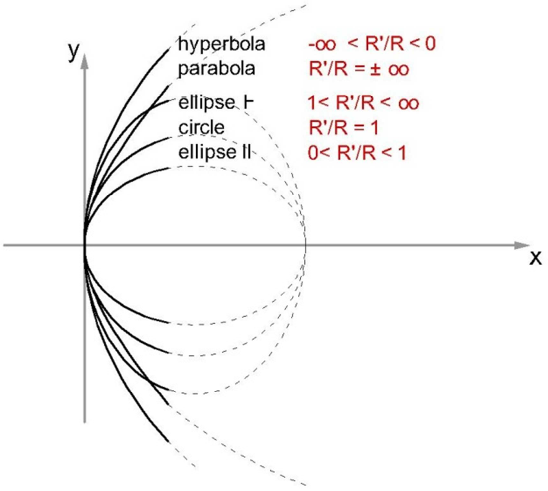
When the option manually is chosen the elevation of each horizon can be put in manually. It has to be mentioned that the crest elevation should be reached, as the index reaches the number of horizons, chosen at the core data definition, otherwise the input may cause an error.

The radii ratio, which can be written as R'/R (3), represents the ratio between the radius of the minor axis to the radius of the major axis of a conic section. Considering a circle, the radii ratio is one, because the radius of the minor axis and the major axis are equal. As other conic sections are used, the radii ratio differs from one. The radii ratio has to be inserted no matter which calculation option is chosen. The list below shows the radii ratio of the different conic sections.

Table 4-4: Radii ratio of conic sections

Conic Section	Radii ratio
parabola	-∞
hyperbola	*-∞ < R'/R < 0*
ellipse with major axis in direction of the valley	*0 < R'/R < 1*
circle	*1*
ellipse with major axis transverse to the valley	*1 < R'/R < ∞*
parabola	∞

The arc-type (4) is listed in the fourth column of the horizon definition table. Depending on the inserted radii ratio, the arc type column shows the corresponding conic section. Figure 4.17 illustrates the different arc-types with the related radii ratio.



(fig. 4.17) Different conic sections corresponding to the radii ratio

The expression ε^2-1 (5) is known form the vertex form of the equation of a conic section (3-41). ε^2-1 can be calculated from the radii ratio, which can be written as

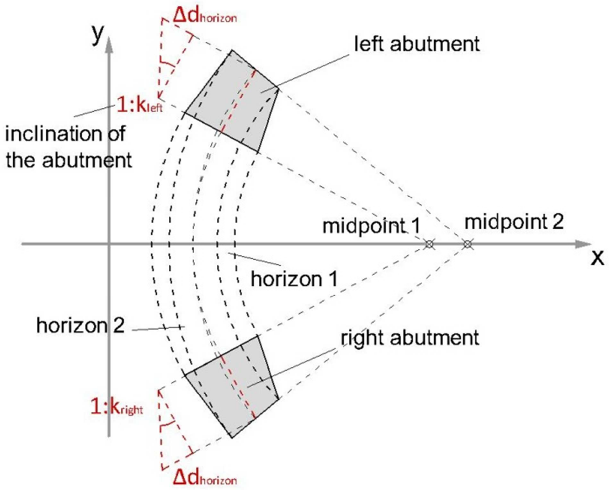
$$(\varepsilon^2 - 1) = \pm \frac{1}{\sqrt[3]{\left(\frac{R'}{R}\right)^2}} \qquad (4\text{-}19)$$

with

$\frac{R'}{R}$ [m] .. radii ratio of the conic section

The parameter k (fig.4.18) describes the inclination of the abutment between two horizons. When using the basic mode k_{left} (6) and k_{right} (7) are equal, due to the assumption that the valley is symmetric and V-shaped, whereas k_{left} and k_{right} differ from each other, as the valley is asymmetric. The sixth and seventh column of the horizon definition table contains the parameters k_{left} and k_{right}.

When choosing the calculation method via k, k_{left} and k_{right} have to be inserted for each horizon except the base horizon, where the input of the parameter B is necessary, due to the fact that the parameter k does not exist for this horizon. For the calculation of the coordinates the parameter k will be automatically transferred into the parameter B. The relation between k and B is explained within chapter 4.3.5.



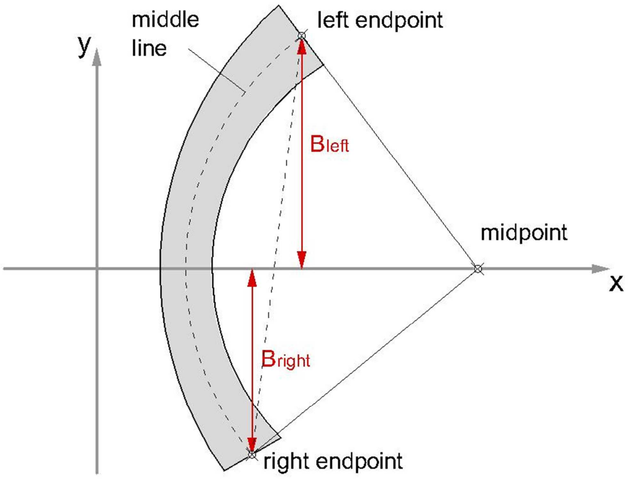
(fig. 4.18) The inclination of the abutment

The eighth and ninth column comprises the parameter B, which represents the projected chord length of the middle line of the horizon (fig.4.19). B_{left} (8) can be taken form the distance between the left endpoint of the arc and the x axis of the dam, analogous B_{right} (9) can be taken form the distance between the right endpoint of the arc and the x axis of the dam. As mentioned above the parameter B is one of the essential parameters for the calculation of the coordinates. When choosing the calculation option via B, B_{left} and B_{right} have to be inserted directly, whereas the other calculation options require the calculation of the parameter B. The calculation of the parameter B, when using the calculation option estimation (USBR) will be done automatically and can be written as

$$\mathrm{B}_{left,right,i} = L_{base} + \frac{\frac{L_{!}-L_{base}}{2}}{h_{dam}} * \Delta h_{base,i} \tag{4-20}$$

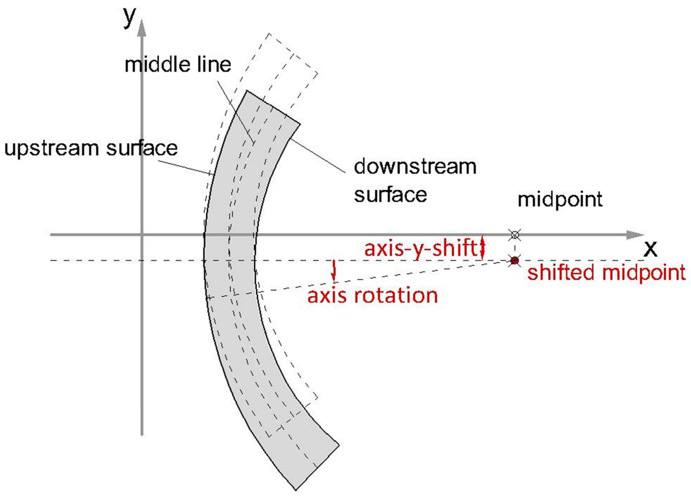
with

$\Delta h_{base,i}$ [m] .. difference in altitude between the current and the base horizon



(fig. 4.19) The projected chord length B

The axis-y-shift (10) describes the shift of the midpoint of the arc from the x axis of the dam and the axis rotation (11) represents the rotation of this shifted axis (fig.4.20). As the calculation is done using the basic mode, the axis-y-shift and axis rotation are set to zero, according to the symmetric valley condition, whereas, when using the expert mode, the axis-y-shift and axis rotation can be changed to refine the dam geometry. Inserting a negative value for the axis rotation causes the horizon to rotate counter clockwise, analogous a positive value causes a clockwise rotation. The tenth and eleventh column of the horizon definition table show the axis-y-shift and axis rotation.



(fig. 4.20) The axis-y-shift and the axis rotation

The x coordinate of the downstream surface (fig.4.21) is referred to as $\underline{x_{ds}}$ (12). The parameter x_{ds} can be calculated form the shape coefficients from the main section definition and the distance in altitude from the elevation of current horizon to the crest elevation, which can be written as

$$\mathrm{x}_{ds,i} = sd_4 * \Delta h_{crest,i}{}^4 + sd_3 * \Delta h_{crest,i}{}^3 + sd_2 * \Delta h_{crest,i}{}^2 + sd_1 * \Delta h_{crest,i} + sd_0 \qquad \text{(4-21)}$$

with

$\Delta h_{crest,i}$	[m]	.. difference in altitude between the crest and the current horizon
sd_4	[m^{-3}]	.. 5th x coordinate shape coefficient
sd_3	[m^{-2}]	.. 4th x coordinate shape coefficient
sd_2	[m^{-1}]	.. 3rd x coordinate shape coefficient
sd_1	[m^{0}]	.. 2nd x coordinate shape coefficient
sd_0	[m^{1}]	.. 1st x coordinate shape coefficient

Beside the x coordinate of the downstream surface, the x coordinates of the vertical middle line x_{ms} and the upstream surface x_{us} are necessary to do the calculation. The x coordinate of the upstream surface can be calculated form the x coordinate of the downstream surface and thickness of the current horizon, which can be written as

$$\mathrm{x}_{us,i} = \mathrm{x}_{ds,i} - \mathrm{t}_i \qquad \text{(4-22)}$$

And the x coordinate of the vertical middle line can be calculated form the x coordinate of the upstream and downstream surface, which can be written as

$$\mathrm{x}_{ms,i} = \frac{\mathrm{x}_{ds,i}+\mathrm{x}_{us,i}}{2} = \mathrm{x}_{ds,i} - \frac{\mathrm{t}_i}{2} \qquad \text{(4-23)}$$

The thirteenth column of the horizon definition table contains the thickness $\underline{t}$ (13) of the main section of the horizon (fig.4.21). Analogous to the x coordinate the thickness of each horizon can be calculated from the corresponding shape coefficients of the main section definition, which can be written as

$$\mathrm{t}_i = st_4 * \Delta h_{crest,i}{}^4 + st_3 * \Delta h_{crest,i}{}^3 + st_2 * \Delta h_{crest,i}{}^2 + st_1 * \Delta h_{crest,i} + st_0 \qquad \text{(4-24)}$$

with

st_4	[m^{-3}]	.. 5th thickness shape coefficient
st_3	[m^{-2}]	.. 4th thickness shape coefficient
st_2	[m^{-1}]	.. 3rd thickness shape coefficient
st_1	[m^{0}]	.. 2nd thickness shape coefficient
st_0	[m^{1}]	.. 1st thickness shape coefficient

The fourteenth and fifteenth column of the horizon definition table comprise the radii of the downstream $\underline{R_{ds}}$ (14) and the upstream $\underline{R_{us}}$ (15) surface (fig.4.21). According to the calculation of the x coordinate and the thickness, the calculation of the downstream and the upstream radii is done by using the coefficients of the main section definition. With

$$\mathrm{R}_{ds,i} = cd_4 * \Delta h_{crest,i}{}^4 + cd_3 * \Delta h_{crest,i}{}^3 + cd_2 * \Delta h_{crest,i}{}^2 + cd_1 * \Delta h_{crest,i} + cd_0 \tag{4-25}$$

with

cd_4	[m^{-3}]	.. 5th downstream radii shape coefficient
cd_3	[m^{-2}]	.. 4th downstream radii shape coefficient
cd_2	[m^{-1}]	.. 3rd downstream radii shape coefficient
cd_1	[m^{0}]	.. 2nd downstream radii shape coefficient
cd_0	[m^{1}]	.. 1st downstream radii shape coefficient

and

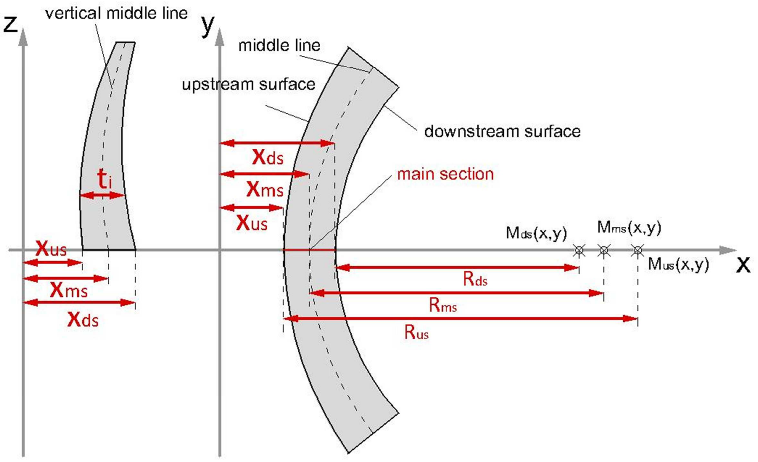
$$\mathrm{R}_{us,i} = cu_4 * \Delta h_{crest,i}{}^4 + cu_3 * \Delta h_{crest,i}{}^3 + cu_2 * \Delta h_{crest,i}{}^2 + cu_1 * \Delta h_{crest,i} + cu_0 \tag{4-26}$$

with

cu_4	[m^{-3}]	.. 5th upstream radii shape coefficient
cu_3	[m^{-2}]	.. 4th upstream radii shape coefficient
cu_2	[m^{-1}]	.. 3rd upstream radii shape coefficient
cu_1	[m^{0}]	.. 2nd upstream radii shape coefficient
cu_0	[m^{1}]	.. 1st upstream radii shape coefficient

the radii of the upstream and downstream surface can be calculated. To gain the radius of the middle line of the horizon R_{ms}, which represents another calculation parameter, a correlation with the upstream and downstream radii can be used.

$$\mathrm{R}_{ms,i} = \frac{\mathrm{R}_{ds,i}+\mathrm{R}_{us,i}}{2} \tag{4-27}$$



(fig. 4.21) The main section parameters

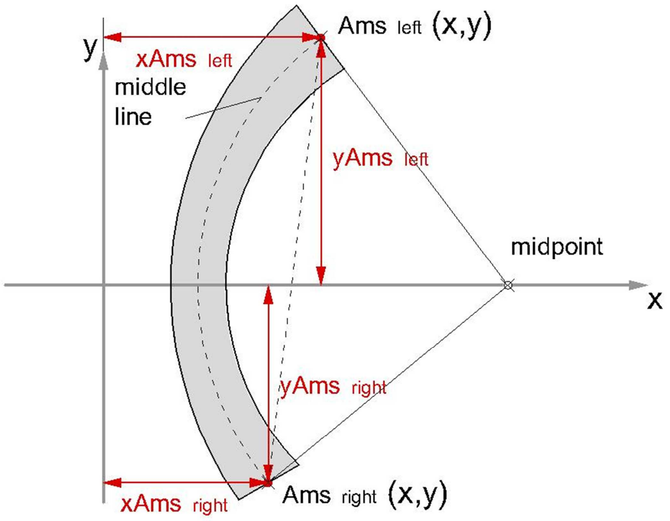
The last four columns, from sixteen to nineteen, contain the x and y coordinates of the endpoint of the middle line of the horizon. $\underline{\text{xAms}_{\text{left}}}$ (16) represents the x coordinate of the left endpoint and $\underline{\text{yAms}_{\text{left}}}$ (17) is the y coordinate of the left endpoint. Analogous to the left side the coordinates of the right side are $\underline{\text{xAms}_{\text{right}}}$ (18) and $\underline{\text{yAms}_{\text{right}}}$ (19). The expression Ams is used as abbreviation for the term abutment middle section.

Independent from the chosen calculation option, the Ams parameters can be calculated from the parameter B. With the correlation

$$\mathrm{Ams}_{y,i} = B_i \tag{4-28}$$

the yAms can be calculated for the left and the right side using the corresponding parameter B. To calculate the xAms parameter it is necessary to rearrange the vertex form of the equation of a conic section (3-41). Inserting the corresponding parameter B, the equation can be written as

$$\mathrm{Ams}_{x,i} = \frac{\pm\sqrt{{R_{ms,i}}^2+(\varepsilon^2-1)_i * {B_i}^2}-R_{ms,i}}{(\varepsilon^2-1)_i} \tag{4-29}$$



(fig. 4.22) The x and y coordinate of the endpoint of the middle line

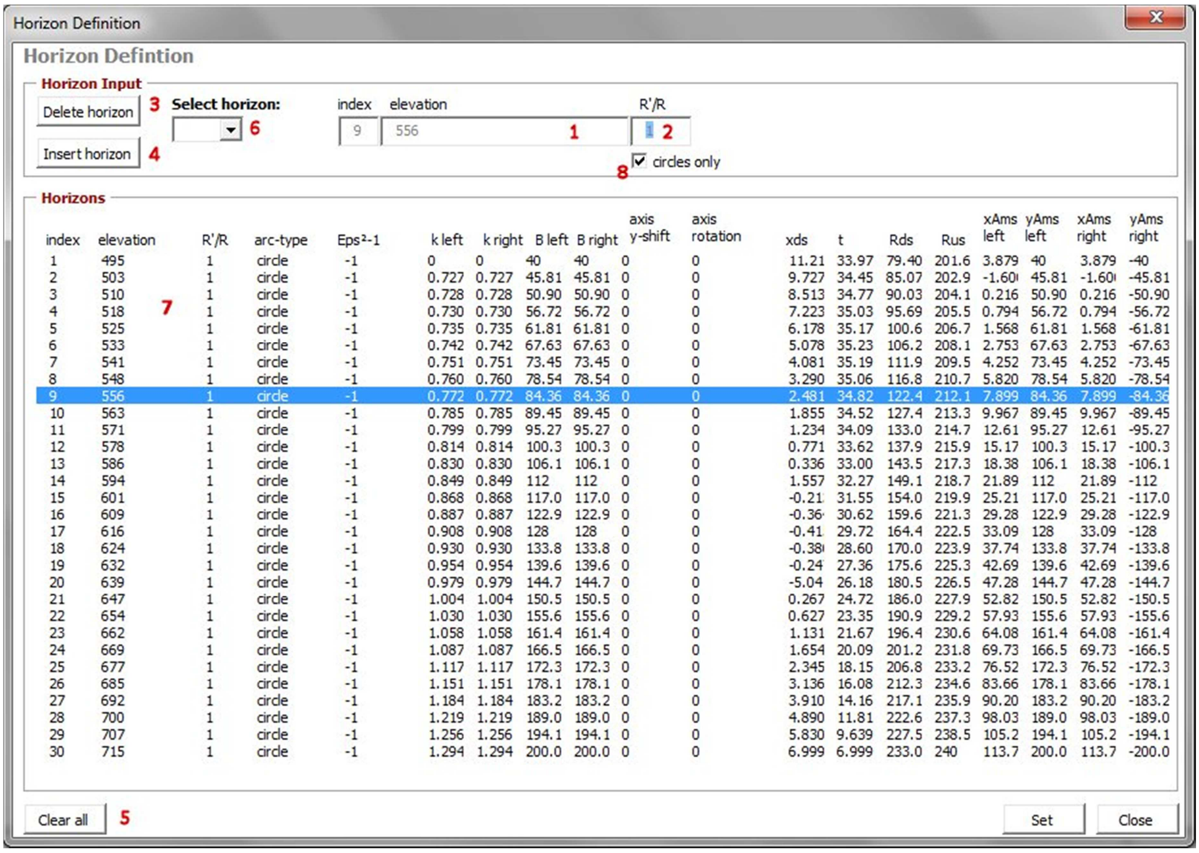
4.2.4.1 THE HORIZON DEFINITION USING THE BASIC MODE

When doing the horizon definition using the basic mode (fig.4.23), there are two fields to do the input. The first parameter, which has to be inserted, is the elevation (1) of the horizon and second parameter is the radii ratio (2). As the number of horizons option is set automatically at the core definition window, only the radii ratio has to be inserted. Beside the input fields, there are five controls to navigate through the horizon definition. The delete horizon button (3), the insert horizon button (4) and the clear all button (5), as well as the select horizon dropdown (6) and the list (7), which is showing the parameter table.

As the input for the current horizon is done, the insert horizon button has to be clicked. The inserted parameters for the current horizon appear within the list and additionally the remaining parameters are calculated automatically. To finalize the horizon definition, the input has to be done for all parameters separately, until the "crest level reached" message appears.

To edit an inserted horizon, the select horizon dropdown can be used by choosing the index of the requested horizon. Beside the dropdown control, horizons can be chosen by clicking on the appropriate entry within the list. Analogous selected horizons can be deleted by clicking the corresponding button.

When using the basic mode, the horizon definition can be fully automatized, using the circles only check box (8). It has to be mentioned, that the number of horizons option has to be set automatically to enable that feature.



Select horizon:

index: 9 elevation: 556 R'/R: 1

☑ circles only

index	elevation	R'/R	arc-type	Eps^2-1	k left	k right	B left	B right	axis y-shift	axis rotation	xds	t	Rds	Rus	xAms left	yAms left	xAms right	yAms right
1	495	1	circle	-1	0	0	40	40	0	0	11.21	33.97	79.40	201.6	3.879	40	3.879	-40
2	503	1	circle	-1	0.727	0.727	45.81	45.81	0	0	9.727	34.45	85.07	202.9	-1.60	45.81	-1.60	-45.81
3	510	1	circle	-1	0.728	0.728	50.90	50.90	0	0	8.513	34.77	90.03	204.1	0.216	50.90	0.216	-50.90
4	518	1	circle	-1	0.730	0.730	56.72	56.72	0	0	7.223	35.03	95.69	205.5	0.794	56.72	0.794	-56.72
5	525	1	circle	-1	0.735	0.735	61.81	61.81	0	0	6.178	35.17	100.6	206.7	1.568	61.81	1.568	-61.81
6	533	1	circle	-1	0.742	0.742	67.63	67.63	0	0	5.078	35.23	106.2	208.1	2.753	67.63	2.753	-67.63
7	541	1	circle	-1	0.751	0.751	73.45	73.45	0	0	4.081	35.19	111.9	209.5	4.252	73.45	4.252	-73.45
8	548	1	circle	-1	0.760	0.760	78.54	78.54	0	0	3.290	35.06	116.8	210.7	5.820	78.54	5.820	-78.54
9	556	1	circle	-1	0.772	0.772	84.36	84.36	0	0	2.481	34.82	122.4	212.1	7.899	84.36	7.899	-84.36
10	563	1	circle	-1	0.785	0.785	89.45	89.45	0	0	1.855	34.52	127.4	213.3	9.967	89.45	9.967	-89.45
11	571	1	circle	-1	0.799	0.799	95.27	95.27	0	0	1.234	34.09	133.0	214.7	12.61	95.27	12.61	-95.27
12	578	1	circle	-1	0.814	0.814	100.3	100.3	0	0	0.771	33.62	137.9	215.9	15.17	100.3	15.17	-100.3
13	586	1	circle	-1	0.830	0.830	106.1	106.1	0	0	0.336	33.00	143.5	217.3	18.38	106.1	18.38	-106.1
14	594	1	circle	-1	0.849	0.849	112	112	0	0	1.557	32.27	149.1	218.7	21.89	112	21.89	-112
15	601	1	circle	-1	0.868	0.868	117.0	117.0	0	0	-0.21	31.55	154.0	219.9	25.21	117.0	25.21	-117.0
16	609	1	circle	-1	0.887	0.887	122.9	122.9	0	0	-0.36	30.62	159.6	221.3	29.28	122.9	29.28	-122.9
17	616	1	circle	-1	0.908	0.908	128	128	0	0	-0.41	29.72	164.4	222.5	33.09	128	33.09	-128
18	624	1	circle	-1	0.930	0.930	133.8	133.8	0	0	-0.38	28.60	170.0	223.9	37.74	133.8	37.74	-133.8
19	632	1	circle	-1	0.954	0.954	139.6	139.6	0	0	-0.24	27.36	175.6	225.3	42.69	139.6	42.69	-139.6
20	639	1	circle	-1	0.979	0.979	144.7	144.7	0	0	-5.04	26.18	180.5	226.5	47.28	144.7	47.28	-144.7
21	647	1	circle	-1	1.004	1.004	150.5	150.5	0	0	0.267	24.72	186.0	227.9	52.82	150.5	52.82	-150.5
22	654	1	circle	-1	1.030	1.030	155.6	155.6	0	0	0.627	23.35	190.9	229.2	57.93	155.6	57.93	-155.6
23	662	1	circle	-1	1.058	1.058	161.4	161.4	0	0	1.131	21.67	196.4	230.6	64.08	161.4	64.08	-161.4
24	669	1	circle	-1	1.087	1.087	166.5	166.5	0	0	1.654	20.09	201.2	231.8	69.73	166.5	69.73	-166.5
25	677	1	circle	-1	1.117	1.117	172.3	172.3	0	0	2.345	18.15	206.8	233.2	76.52	172.3	76.52	-172.3
26	685	1	circle	-1	1.151	1.151	178.1	178.1	0	0	3.136	16.08	212.3	234.6	83.66	178.1	83.66	-178.1
27	692	1	circle	-1	1.184	1.184	183.2	183.2	0	0	3.910	14.16	217.1	235.9	90.20	183.2	90.20	-183.2
28	700	1	circle	-1	1.219	1.219	189.0	189.0	0	0	4.890	11.81	222.6	237.3	98.03	189.0	98.03	-189.0
29	707	1	circle	-1	1.256	1.256	194.1	194.1	0	0	5.830	9.639	227.5	238.5	105.2	194.1	105.2	-194.1
30	715	1	circle	-1	1.294	1.294	200.0	200.0	0	0	6.999	6.999	233.0	240	113.7	200.0	113.7	-200.0

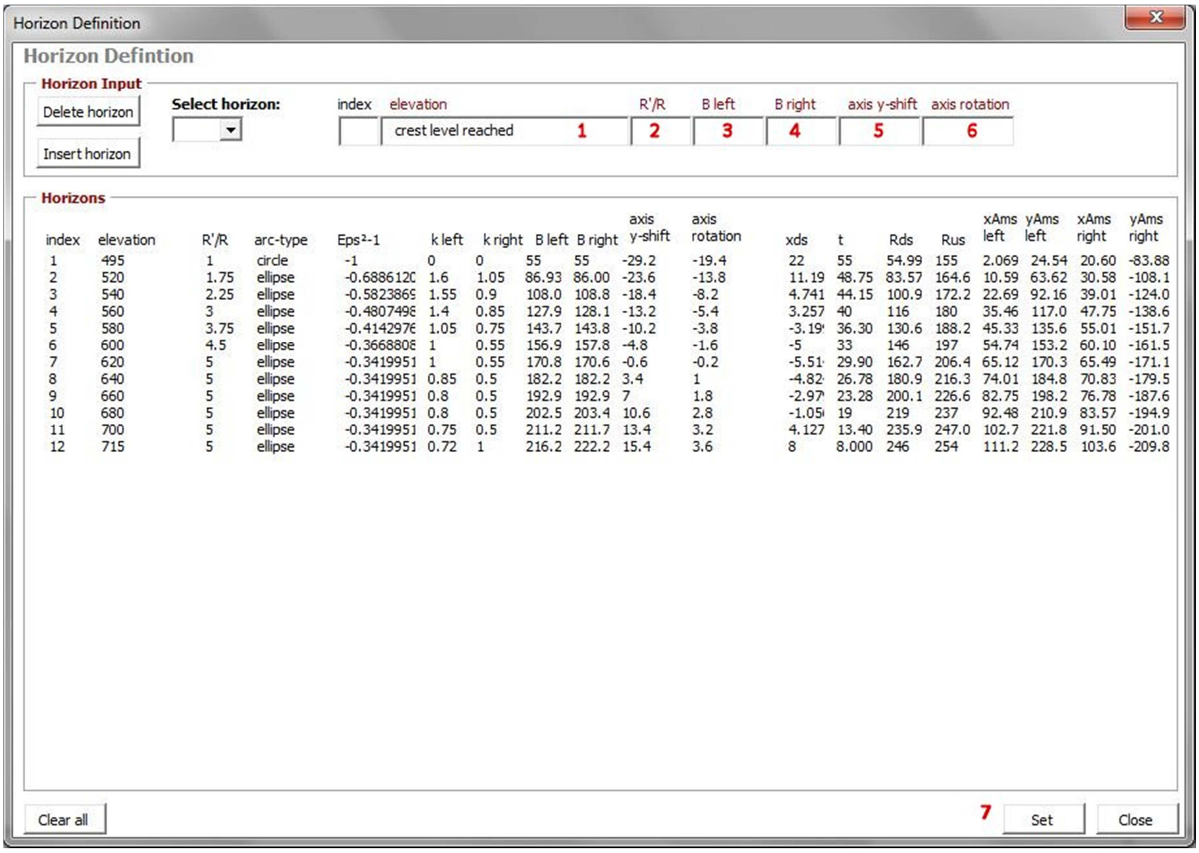
(fig. 4.23) Horizon definition using the basic mode

4.2.4.2 The horizon definition using the expert mode

The expert mode of the horizon definition (fig.3.24) provides the same controls as the basic mode, but requires six instead of two parameters to be inserted. Beside the elevation (1) and the radii ratio (2), the parameter B or k of the left (3) and the right (4) side, according to the chosen expert calculation method and the axis-y-shift (5) as well as the axis rotation (6) has to be put in.

As noted above the projected chord length B has to be inserted for the base horizon, independent of the expert calculation method, due to the fact the parameter k does not exist for the base horizon. For the other horizons the parameter k has to be put in, if the method via k is chosen and analogous the parameter B has to be inserted, if via B is set as calculation method.

The navigation within the horizon definition using expert mode is correspondent to the navigation using the basic mode, except the circles only checkbox, which is not available for the expert mode, due to the fact that no reasonable automation is possible, caused by the amount of necessary input.



elevation: crest level reached

index	elevation	R'/R	arc-type	Eps²-1	k left	k right	B left	B right	axis y-shift	axis rotation	xds	t	Rds	Rus	xAms left	yAms left	xAms right	yAms right
1	495	1	circle	-1	0	0	55	55	-29.2	-19.4	22	55	54.99	155	2.069	24.54	20.60	-83.88
2	520	1.75	ellipse	-0.6886120	1.6	1.05	86.93	86.00	-23.6	-13.8	11.19	48.75	83.57	164.6	10.59	63.62	30.58	-108.1
3	540	2.25	ellipse	-0.5823869	1.55	0.9	108.0	108.8	-18.4	-8.2	4.741	44.15	100.9	172.2	22.69	92.16	39.01	-124.0
4	560	3	ellipse	-0.4807498	1.4	0.85	127.9	128.1	-13.2	-5.4	3.257	40	116	180	35.46	117.0	47.75	-138.6
5	580	3.75	ellipse	-0.4142976	1.05	0.75	143.7	143.8	-10.2	-3.8	-3.19	36.30	130.6	188.2	45.33	135.6	55.01	-151.7
6	600	4.5	ellipse	-0.3668808	1	0.55	156.9	157.8	-4.8	-1.6	-5	33	146	197	54.74	153.2	60.10	-161.5
7	620	5	ellipse	-0.3419951	1	0.55	170.8	170.6	-0.6	-0.2	-5.51	29.90	162.7	206.4	65.12	170.3	65.49	-171.1
8	640	5	ellipse	-0.3419951	0.85	0.5	182.2	182.2	3.4	1	-4.82	26.78	180.9	216.3	74.01	184.8	70.83	-179.5
9	660	5	ellipse	-0.3419951	0.8	0.5	192.9	192.9	7	1.8	-2.97	23.28	200.1	226.6	82.75	198.2	76.78	-187.6
10	680	5	ellipse	-0.3419951	0.8	0.5	202.5	203.4	10.6	2.8	-1.05	19	219	237	92.48	210.9	83.57	-194.9
11	700	5	ellipse	-0.3419951	0.75	0.5	211.2	211.7	13.4	3.2	4.127	13.40	235.9	247.0	102.7	221.8	91.50	-201.0
12	715	5	ellipse	-0.3419951	0.72	1	216.2	222.2	15.4	3.6	8	8.000	246	254	111.2	228.5	103.6	-209.8

(fig. 4.24) Horizon definition using the expert mode

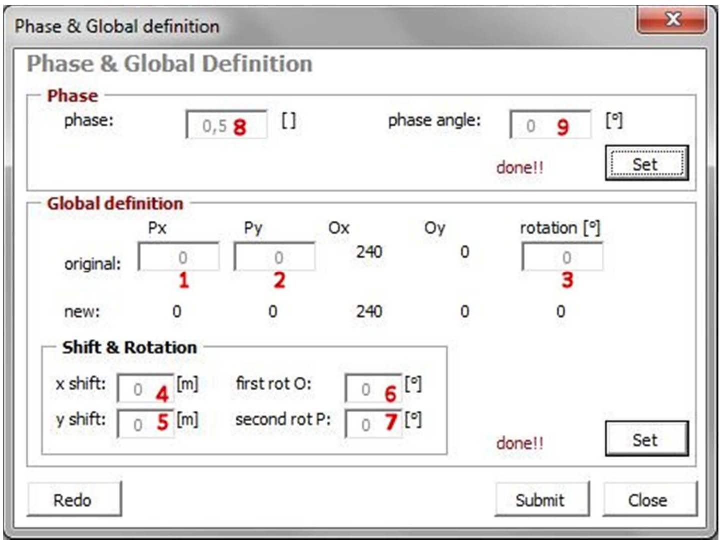
The set button (7) on the bottom of the horizon definition window, which is available for the expert and the basic mode, is used to submit the parameters and close the horizon definition window.

4.2.5 PHASE AND GLOBAL DEFINITION

The last input window, the phase & global definition form, can be opened by clicking on the corresponding button in the menu or toolbar.

The phase & global definition (fig.4.25) form contains two types of input. The first type of input treats the phase, which is needed to bevel the abutment of the arch dam. The upstream abutment can be cut off, which is called phase, due to the fact, that the upstream side of the abutment do not have decisive structural behaviour.

The second type of input treats the global definition of the coordinates. As there is a known global point of the arch dam, it can be put in to replace the local origin of the coordinates system. Beside the global point, a global rotation of the axis of the dam can be added too.



Phase & Global definition

Phase & Global Definition

Phase

phase: 0,5 [] phase angle: 0 [°]

done!!

Global definition

	Px	Py	Ox	Oy	rotation [°]
original:	0	0	240	0	0
new:	0	0	240	0	0

Shift & Rotation

x shift: 0 [m] first rot O: 0 [°]

y shift: 0 [m] second rot P: 0 [°]

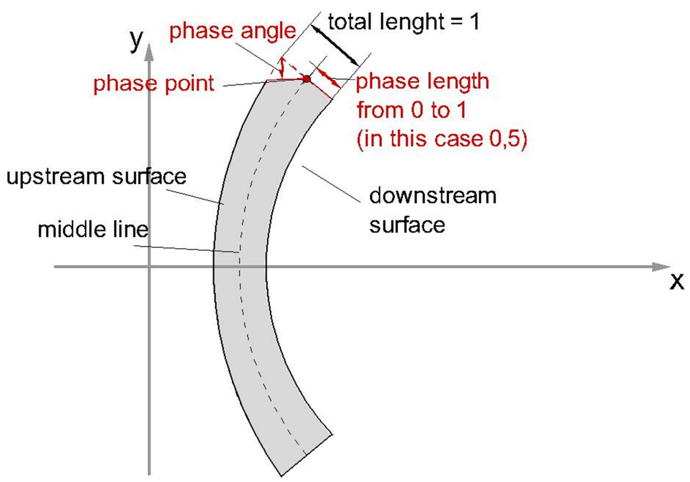
done!!

(fig. 4.25) Phase & global definition form

4.2.5.1 THE PHASE DEFINITION

To add a phase to the arch dam two parameters are needed. The so called phase (8) determines the length of the abutment from the downstream side of the abutment to the cut-off point at the abutment. The phase has to be inserted form 0 to 1, where 0 represents no length and 1 represents the total abutment length. Further the so called phase angle (9) has to be put in. The phase angle is needed to determine the angle of the phase (fig.4.26).

It has to be mentioned that the adding of a phase represents a possible but not fundamental input to calculate the coordinates of the arch dam.



(fig. 4.26) Phase length and phase angle

4.2.5.2 The global definition

The global definition enables the possibility to add a global reference point P, with Px (1) and Py (2), to replace the origin of the global coordinates system (0,0) and a global rotation (3) to rotate the axis of the dam. To calculate the shifted and rotated coordinates, the equations from (3-46) and (3-47) are used.

Beside the reference point P, a virtual point O is calculated, which is needed to refine the global shift and rotation. The virtual point O, with Ox and Oy, is laying on the dam axis with a distance from the point P calculated from the upstream radius of the crest.

The refinement of the global definition can be done by using an x-shift (4) and y-shift (5) as well as a first rotation (6) of the dam axis around point O and a second rotation (7) of the dam axis around point P.

For the calculation of the local 3D coordinates of the arch dam, the global definition is not needed. As the default entries are not replaced by other values the global coordinates are equal to the local coordinates.

4.2.6 CALCULATION

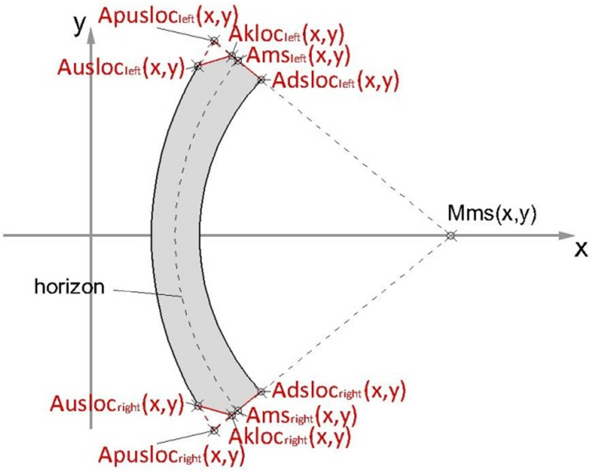
As the input of the arch dam is done, the calculation of the coordinates as well as the calculation of the volume of the arch dam can be triggered by clicking on the corresponding button in the menu or toolbar.

4.2.6.1 THE CALCULATION OF THE 3D COORDINATES OF THE ARCH DAM

The calculation of the coordinates of an arch dam is divided into several steps and has to be done for each horizon separately. The calculation starts with the determination of the abutment coordinates. The abutment coordinates are the downstream surface coordinates Adsloc(x,y), the phase coordinates Akloc(x,y) the pseudo upstream surface coordinates Apusloc(x,y) and the real upstream surface coordinates Ausloc(x,y) (fig.4.27).

As the abutment of the horizon is determined for the left and the right side the calculation of the section coordinates of the arches represents the next step. To calculate the section coordinates, the opening angle of the downstream and upstream arch is divided into the chosen number of sections, from the core data definition. Dependant on this divided angle, the remaining coordinates of the downstream and upstream surface can be determined. As mentioned above the calculation of the abutment and the section coordinates has to be done for each horizon separately, using the corresponding parameters.

The detailed calculation steps, which are necessary to investigate the coordinates of each horizon, are explained in the corresponding chapter 4.3.



(fig. 4.27) The abutment coordinates of an arch dam

4.2.6.2 The calculation of the volume

The volume of the arch dam, which is listed within the project data frame, can be estimated with two different methods. The first method is provided within the USBR design guide, where the volume can be calculated from a correlation of the parameters L1, L2 and the dam height H. The USBR volume which is given in cubic yards, can be written as

$$\mathrm{V}_{USBR} \approx 0.00017699 * H^2 * L_2 * \left[\frac{(H+0.8*L_1)^2}{L_1-L_2}\right] + 0.01079178 * H * L_1 * [H + 1.1 * L_1] \qquad \text{(4-30)}$$

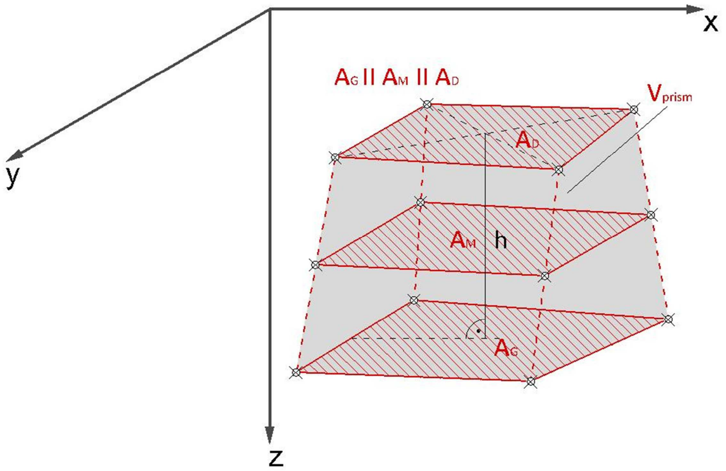
It is necessary to pre calculate the project using the basic mode, to have access to the USBR volume estimation.

The second method of volume estimation is done by calculating the sum of partial volumes of the arch dam. The partial volume between two horizons (fig.4.28), linearized as prism, can be calculated with the equation

$$\mathrm{V}_{prism} \approx \frac{h}{6} * (A_G + A_D + 4 * A_M) \qquad \text{(4-31)}$$

with

A_G	[m²]	.. base surface of the prism
A_D	[m²]	.. top surface of the prism
A_M	[m²]	.. mean surface of the prism

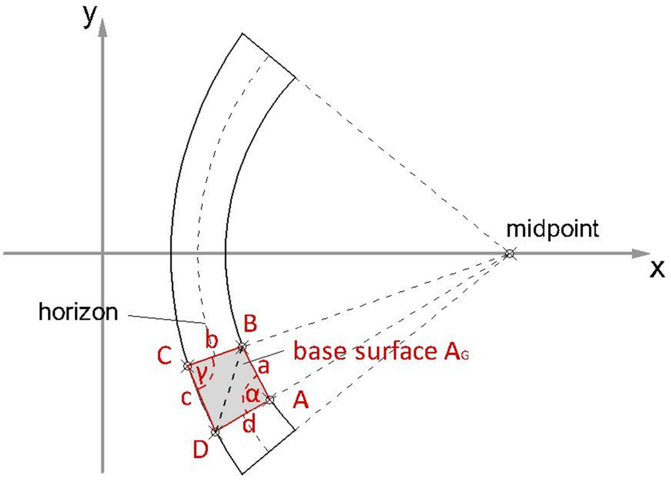


(fig. 4.28) The volume of a general prism [3]

To gain the volume of the partial prism it is necessary to determine the base surface A_G and the top surface A_D as well as a mean surface A_M represented by a cross section in the middle of the volume. The area of the base surface can be calculated from two known coordinates of the downstream arch, A and B and two known corresponding coordinates of the upstream arch, C and D, of one horizon, which represent the corners of the base surface (fig4.29). As the lengths of the edges a, b, c and d and the corresponding angles α and γ are determined, the base surface can calculated with the equation

$$\mathrm{A}_G = \frac{a*d*\sin(\alpha)}{2} + \frac{b*c*\sin(\gamma)}{2} \tag{4-32}$$

Analogous to the base surface, the area of the overlaying mean and the top surface can be calculated using the corresponding coordinates of the upstream and downstream surface of the respective horizon.



(fig. 4.29) The base surface of the prism

4.2.7 Post processing

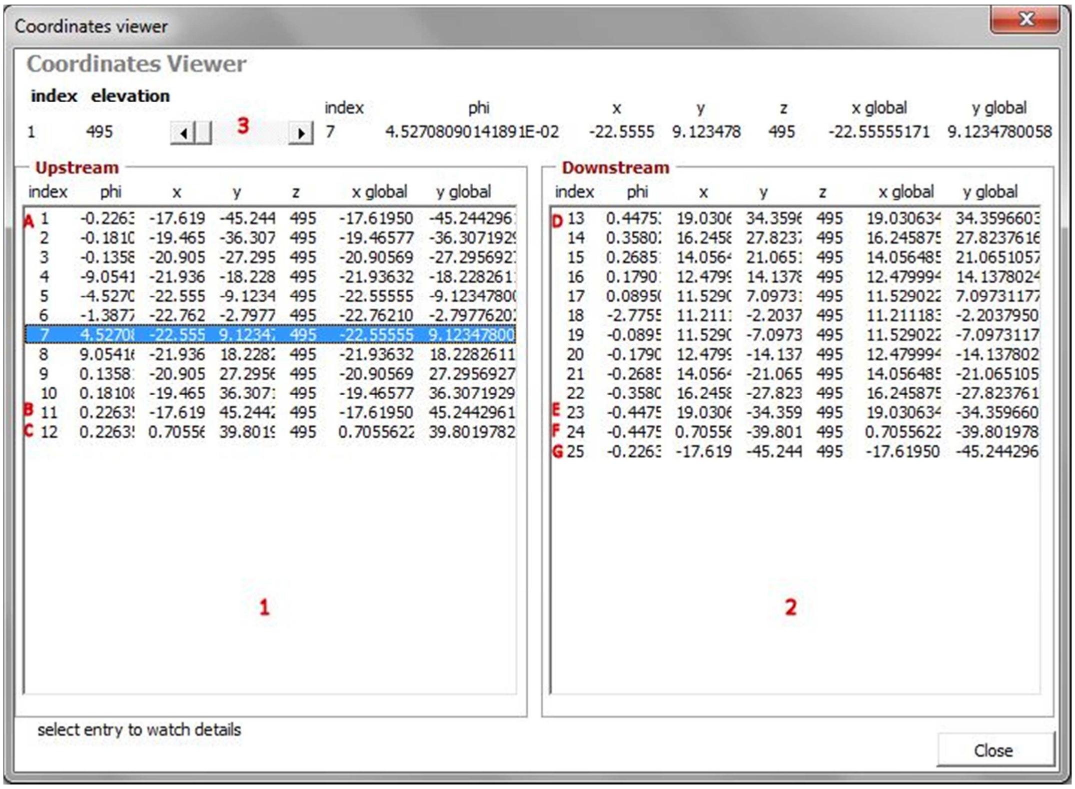
4.2.7.1 The coordinates form

After the calculation, the coordinates form can be opened by clicking on the corresponding button in the menu or toolbar. Within the coordinates form (fig.4.30), there are two lists showing the upstream (1) and downstream (2) coordinates of each horizon. From the left to the right the columns of each list show the index, the angle phi, the x, y and z coordinate as well as the global x and global y coordinate.

The first entry in the upstream list shows the coordinates of the right abutment of the upstream surface (A) followed by the coordinates of the sections between the abutments of the upstream surface until the penultimate entry, which represents the coordinates of the left abutment of the upstream surface (B). The last entry of the upstream list shows the coordinates of the left phase (C).

Analogous the downstream list starts with the coordinates of the left abutment of the downstream surface (D) followed by the coordinates of the sections between the abutments of the downstream surface. The last three entries of the downstream list represent the coordinates of the right abutment (E), the coordinates of the right phase (F) and the endpoint (G), whose coordinates are equal to the coordinates of the right abutment of the upstream surface. The implementation of an endpoint is necessary to create a closed polygon.

When opening the coordinates form, the coordinates of the base horizon are show as default. To list the coordinates of the other horizons, the scrollbar (3) at the top to the form should be used.



Coordinates viewer

Coordinates Viewer

index	elevation
1	495

index	phi	x	y	z	x global	y global
7	4.52708090141891E-02	-22.5555	9.123478	495	-22.55555171	9.1234780058

Upstream

	index	phi	x	y	z	x global	y global
A	1	-0.2263	-17.619	-45.244	495	-17.61950	-45.244296
	2	-0.1810	-19.465	-36.307	495	-19.46577	-36.307192
	3	-0.1358	-20.905	-27.295	495	-20.90569	-27.295692
	4	-9.0541	-21.936	-18.228	495	-21.93632	-18.228261
	5	-4.5270	-22.555	-9.1234	495	-22.55555	-9.1234780
	6	-1.3877	-22.762	-2.7977	495	-22.76210	-2.7977620
	7	4.52708	-22.555	9.1234	495	-22.55555	9.12347800
	8	9.05416	-21.936	18.228	495	-21.93632	18.2282611
	9	0.1358	-20.905	27.2956	495	-20.90569	27.2956927
	10	0.18108	-19.465	36.307	495	-19.46577	36.3071929
B	11	0.2263	-17.619	45.244	495	-17.61950	45.2442961
C	12	0.2263	0.70556	39.801	495	0.7055622	39.8019782

Downstream

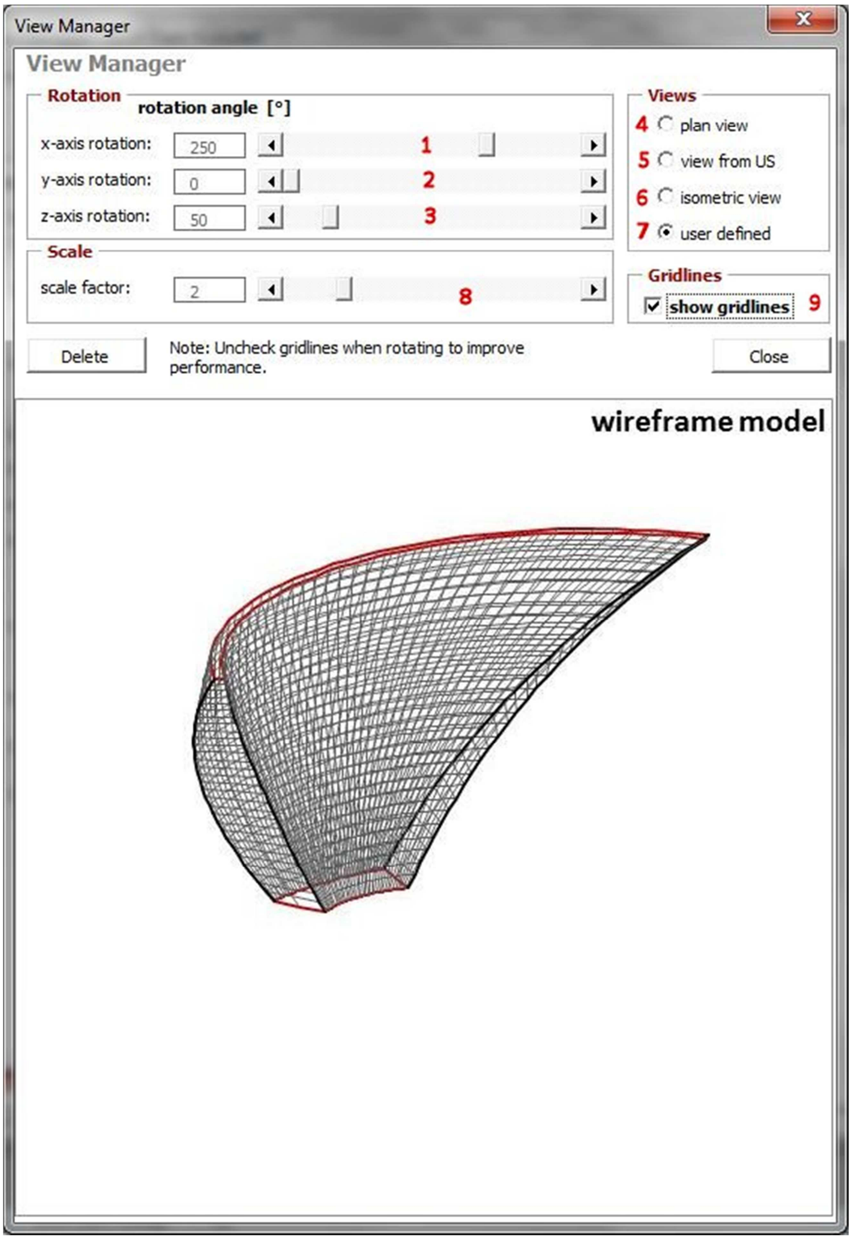
	index	phi	x	y	z	x global	y global
D	13	0.4475	19.0306	34.3596	495	19.030634	34.3596603
	14	0.3580	16.2458	27.8237	495	16.245875	27.8237616
	15	0.2685	14.0564	21.0651	495	14.056485	21.0651057
	16	0.1790	12.4799	14.1378	495	12.479994	14.1378024
	17	0.08950	11.5290	7.0973	495	11.529022	7.09731177
	18	-2.7755	11.211	-2.2037	495	11.211183	-2.2037950
	19	-0.0895	11.5290	-7.0973	495	11.529022	-7.0973117
	20	-0.1790	12.4799	-14.137	495	12.479994	-14.137802
	21	-0.2685	14.0564	-21.065	495	14.056485	-21.065105
	22	-0.3580	16.2458	-27.823	495	16.245875	-27.823761
E	23	-0.4475	19.0306	-34.359	495	19.030634	-34.359660
F	24	-0.4475	0.70556	-39.801	495	0.7055622	-39.801978
G	25	-0.2263	-17.619	-45.244	495	-17.61950	-45.244296

select entry to watch details

(fig. 4.30) The coordinates form

4.2.7.2 THE 3D WIREFRAME FORM

Additionally to the coordinates form, the 3D wireframe form can be opened by clicking on the corresponding button in the menu or toolbar. The 3D wireframe form illustrates a 3D preview of the arch dam. With the three scrollbars at the top of the form the axes of the coordinates system x (1), y (2) and z (3) can be rotated, to show different views of the arch dam. It has to be mentioned that the option user defined (7) has to be chosen when rotating the arch dam manually. On the right side of the form three predefined views are given. The plan view (4) and the view from US (5) are equal to the charts of the main program and the isometric view (6) shows a 3D preview of the dam. To change the scale (8) of the 3D wireframe the scrollbar underneath the rotation frame can be used. The show gridlines checkbox (9) is used to display the vertical sections of the dam. It is recommended to unmark the show gridlines checkbox, when rotating the arch dam to improve the performance of the program.



View Manager

Rotation — rotation angle [°]

x-axis rotation: 250

y-axis rotation: 0

z-axis rotation: 50

Scale

scale factor: 2

Views

○ plan view

○ view from US

○ isometric view

● user defined

Gridlines

☑ show gridlines

Note: Uncheck gridlines when rotating to improve performance.

(fig. 4.31) The 3D wireframe form

To generate a 3D view within a 2D chart it is necessary to transform the coordinates of the arch dam. The coordinates of the arch dam can be rotated by using the equation (3-47). The used 3D rotation matrix can be written as

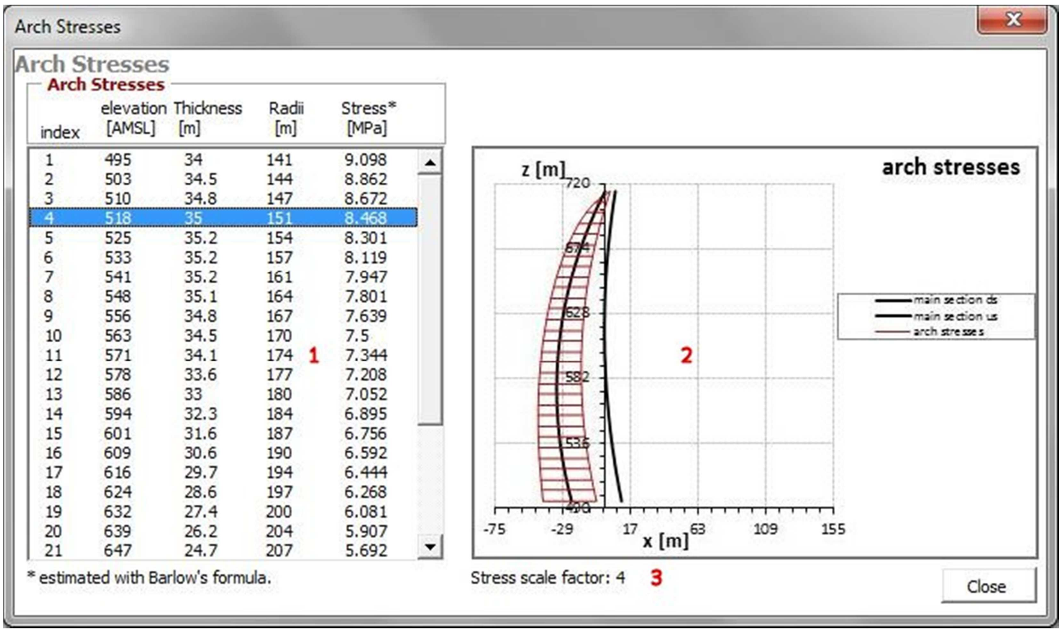
$$\overline{\overline{\mathrm{R}}}_{3D} = \begin{bmatrix} \mathrm{d} * f & d * (-e) & c \\ b * e + (-a) * (-c) * f & b * f + (-a) * (-c) * (-e) & (-a) * d \\ a * e + b * (-c) * f & a * f + b * (-c) * (-e) & b * d \end{bmatrix} \tag{4-33}$$

with

a	[-]	.. $\sin\varphi$ sinus of the angle phi
b	[-]	.. $\cos\varphi$ cosines of the angle phi
c	[-]	.. $\sin\theta$ sinus of the angle theta
d	[-]	.. $\cos\theta$ cosines us of the angle theta
e	[-]	.. $\sin\psi$ sinus of the angle psi
f	[-]	.. $\cos\psi$ cosines of the angle psi
φ	[°]	.. rotation around the x axis
θ	[°]	.. rotation around the y axis
ψ	[°]	.. rotation around the z axis

4.2.7.3 THE ARCH STRESSES FORM

To gain an overview over the arch stresses, the arch stresses form can be opened by clicking on the corresponding button in the menu or toolbar. S The form shows the arch stresses of the dam, taken from the main section. On the left side of the form the main parameters of each horizon are listed (1), which are necessary to calculate the arch stresses with the equations (3-17) and (3-20). From the left to the right the parameters are the index, the elevation, the thickness and the radii as well as the resulting arch stress.
The right side of the form shows a preview (2) of the arch stresses of the main section assigned to the vertical middle line of the dam. To improve the illustration of the stresses, the preview is scaled automatically with a scale factor (3), which is show on the bottom of the preview.



Arch Stresses

Arch Stresses

Arch Stresses

index	elevation [AMSL]	Thickness [m]	Radii [m]	Stress* [MPa]
1	495	34	141	9.098
2	503	34.5	144	8.862
3	510	34.8	147	8.672
4	518	35	151	8.468
5	525	35.2	154	8.301
6	533	35.2	157	8.119
7	541	35.2	161	7.947
8	548	35.1	164	7.801
9	556	34.8	167	7.639
10	563	34.5	170	7.5
11	571	34.1	174	7.344
12	578	33.6	177	7.208
13	586	33	180	7.052
14	594	32.3	184	6.895
15	601	31.6	187	6.756
16	609	30.6	190	6.592
17	616	29.7	194	6.444
18	624	28.6	197	6.268
19	632	27.4	200	6.081
20	639	26.2	204	5.907
21	647	24.7	207	5.692

* estimated with Barlow's formula.

Stress scale factor: 4

(fig. 4.32) The arch stresses form

4.2.7.4 THE EXPORT OF THE GEOMETRY SCRIPT

The python script, which is explained within chapter 5, can be exported by clicking on the corresponding button in the menu or toolbar. The export of the coordinates using a python script ensures that the necessary syntax, which is needed within Abaqus CAE, is provided too.

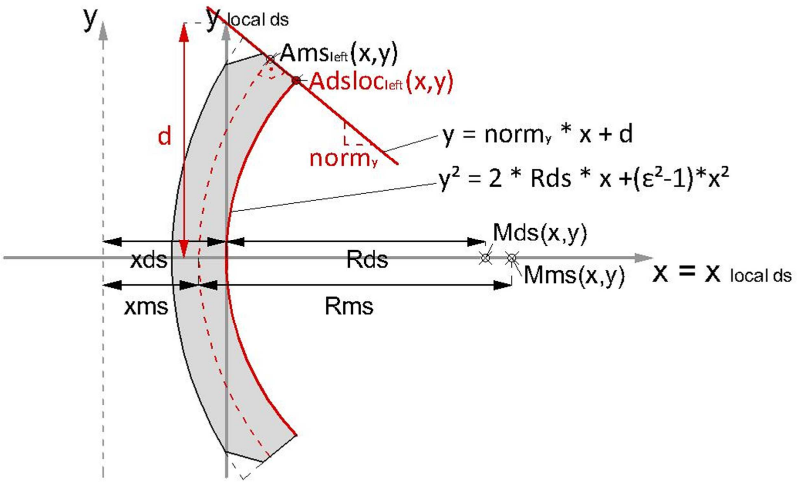
4.2.7.5 THE EXPORT OF A CSV FILE

To export the coordinates via CSV file, the corresponding button in the menu or toolbar has to be clicked. The export of the coordinates via CSV file is useful for the manipulation of the coordinates with other programs, except Abaqus CAE. Within the CSV file the x, y and z coordinates are listed starting with the coordinates of the base horizon up to the coordinates of the crest horizon of the arch dam.

4.3 CALCULATION OF 3D DAM COORDINATES

4.3.1 THE CALCULATION OF THE DOWNSTREAM ABUTMENT COORDINATES

The calculation of the coordinates of each horizon starts with the calculation of the downstream abutment coordinates (fig.4.33) of the left and the right side. To gain the downstream abutment coordinates it is necessary to determine the intersection of the downstream arch and the straight line which is orthogonal to the middle line and passing through the point Ams(x,y).



(fig. 4.33) The calculation of the downstream abutment coordinates

The gradient of the orthogonal straight line, which is called norm$_y$, can be found by taking the negative reciprocal of the first derivative of the vertex form of the equation of a conic section (3-41). The first derivate of the equation represents the gradient of the tangent of the function at a certain point. When forming the negative reciprocal, the resulting direction is orthogonal to this gradient. With

$$-\frac{1}{\frac{\mathrm{d}}{dx}\sqrt{2*p*x+(\varepsilon^2-1)*x^2}} = -\frac{\sqrt{2*p*x+(\varepsilon^2-1)*x^2}}{p+(\varepsilon^2-1)*\mathrm{x}} \tag{4-34}$$

and the known point Ams(x,y) as well as the radius Rms the gradient of the orthogonal line can be written as

$$\mathrm{norm}_y = -\frac{yAms}{Rms+(\varepsilon^2-1)*\mathrm{xAms}} \tag{4-35}$$

with the equation (3-41)

$$\mathrm{y} = \sqrt{2*p*x+(\varepsilon^2-1)*x^2} \tag{4-36}$$

and therefore

$$\mathrm{yAms} = \sqrt{2 * Rms * xAms + (\varepsilon^2 - 1) * xAms^2} \tag{4-37}$$

Beside the gradient of the equation of the straight line, it is necessary to determine the y-intercept d. When solving the equation of the straight line (3-21) with the known point Ams(x,y) which has to be transferred to the local downstream coordinates system, the y intercept can be written as

$$d = \mathrm{yAms} - norm_y * (xAms + xms - xds) \tag{4-38}$$

As the gradient and the y-intercept of the straight line are known the equation of the straight line, which is orthogonal to the middle line and passing through the point Ams(x,y), can be written as

$$y = -\frac{yAms}{Rms+(\varepsilon^2-1)*xAms} * x + yAms + \frac{yAms}{Rms+(\varepsilon^2-1)*xAms} * (xAms + xms - xds) \tag{4-39}$$

With the input parameters form the horizon definition form, the vertex form of the equation of a conic section (3-41) of the downstream arch can be written as

$$y = \sqrt{2 * Rds * x + (\varepsilon^2 - 1) * x^2} \tag{4-40}$$

When forming the intersection of the straight line and the vertex form of the equation of the conic section the coordinates of the downstream abutment can be calculated. The x coordinate xAdsloc can be written as

$$xAdsloc = \frac{\sqrt{d^2*(\varepsilon^2-1)-2*d*norm_y*Rds+Rds^2}+d*norm_y-Rds}{(\varepsilon^2-1)-norm_y{}^2} \tag{4-41}$$

and the y coordinate yAdsloc can be written as

$$yAdsloc = norm_y * \; xAdsloc + d \tag{4-42}$$

It has to be noticed that the calculation of the downstream abutment coordinates is done within a corresponding local downstream coordinates system. To use the coordinates within the superior local coordinates system, which is also called dam coordinate system, the downstream coordinates are transferred using the equation form (3-46). An additional transformation of the coordinates is required as an axis y-shift and an axis rotation is added to the horizon.

4.3.2 The calculation of the phase coordinates

The calculation of the phase coordinates Akloc(x,y) requires a prior calculation of the pseudo upstream abutment coordinates Apusloc(x,y) (fig.4.34).

Analogous to the calculation of the downstream abutment coordinates the pseudo upstream abutment coordinates can be calculated form the intersection of the vertex form of the equation of a conic section of the upstream arch and the same straight line passing through the point Ams(x,y). As using the local upstream coordinates system the y-intercept of the equation of the straight line has to be modified. With

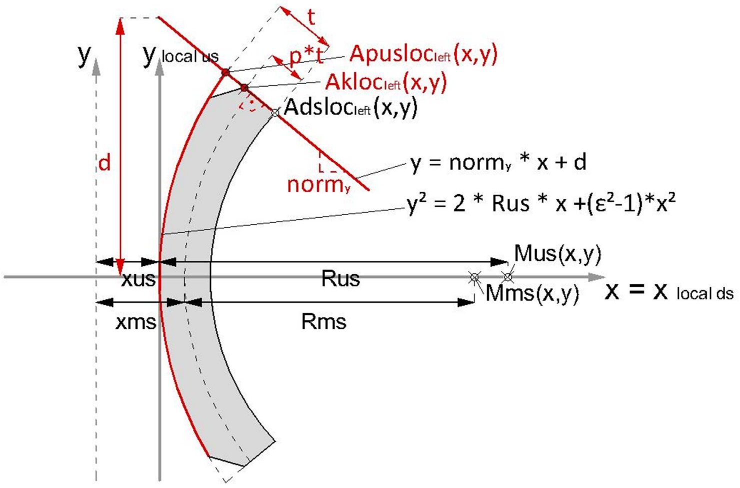
$$d = \mathrm{yAms} - norm_y * (xAms + xms - xus) \tag{4-43}$$

the x coordinate of the pseudo upstream surface xApusloc can be written as

$$xApusloc = \frac{\sqrt{d^2 * (\varepsilon^2 - 1) - 2 * d * norm_y * Rus + Rus^2} + d * norm_y - Rus}{(\varepsilon^2 - 1) - norm_y{}^2} \tag{4-44}$$

and the y coordinate yApusloc is

$$yApusloc = norm_y * \ xApusloc + d \tag{4-45}$$



(fig. 4.34) The calculation of the pseudo upstream abutment and phase coordinates

As the downstream abutment coordinates and the pseudo upstream abutment coordinates are known the phase coordinates can be calculated from the correlation

$$Akloc(x,y) = \begin{cases} xAdsloc(x,y)\,, if\ p = 0 \\ xApusloc(x,y)\,, if\ p = 1 \end{cases} \tag{4-46}$$

with

p [-] .. phase of the abutment

When forming an equation out of (4-46) the x-coordinate of the phase, can be written as

$$xAkloc = p * xApusloc + (1 - p) * (xAdsloc + xds - xus) \tag{4-47}$$

considering the transformation of xAdsloc form the local downstream coordinates system to the local upstream coordinates system. According to the x-coordinate the y coordinate of the phase can be written as

$$yAkloc = p * yApusloc + (1 - p) * yAdsloc \tag{4-48}$$

4.3.3 The calculation of the Upstream abutment coordinates

The last abutment coordinates, which have to be calculated are the real upstream abutment coordinates Ausloc(x,y) (fig.4.35). Analogous to the other abutment coordinates the real upstream abutment coordinates are calculated from the intersection of the vertex form of the equation of a conic section of the upstream arch and a straight line. As opposed to the other abutment coordinates the straight line passing through the phase coordinate Akloc(x,y). The gradient of this straight line can be calculated form the angle sum identity of the tangent of the given phase angle and the gradient $norm_y$, of the previous straight line. With the equation from (3-45) the gradient of the straight line, which is called tanphase, can be written as

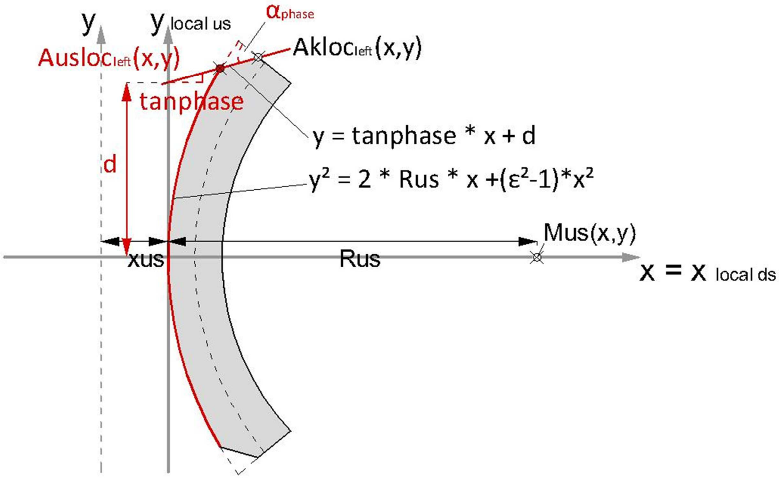
$$tanphase = \frac{norm_y + \tan(\alpha_{phase})}{1 - norm_y * \tan(\alpha_{phase})} \tag{4-49}$$

with

α_{phase} [°] ..given phase angle

Beside the gradient the y-intercept has to be re calculated too. The y-intercept d of the straight line passing through Akloc(x,y), can be written as

$$d = yAkloc - tanphase * xAkloc \tag{4-50}$$



(fig. 4.35) The calculation of the upstream abutment coordinates

As the gradient and the y-intercept are known, the intersection of the straight line and the vertex form of the equation of the conic section of the upstream arch results in

$$xAusloc = \frac{\sqrt{d^2*(\varepsilon^2-1)-2*d*tanphase*Rus+Rus^2}+d*tanphase-Rus}{(\varepsilon^2-1)-tanphase^2} \tag{4-51}$$

and

$$yAusloc = yAkloc + tanphase * (xAusloc - xAkloc) \tag{4-52}$$

It has to be mentioned that the real upstream abutment coordinates are equal to the pseudo upstream abutment coordinates, if the phase length is 1 or the phase angle is 0.

When the calculation of the left abutment coordinates is done, the calculation steps have to be redone to gain the coordinates of the right abutment, using the corresponding input data. Not until then, the calculation of the remaining section coordinates is possible.

4.3.4 The calculation of the section coordinates from the variable angle phi

To calculate the remaining section coordinates (fig.4.36) it is necessary to determine the opening angle of the downstream and upstream arch. With the arctangent of the proportion of the radius and the corresponding abutment coordinates, which can be written as

$$\alpha_{left} = \arctan\left(\frac{yAdsloc_{left}}{Rds - xAdsloc_{left}}\right) \tag{4-53}$$

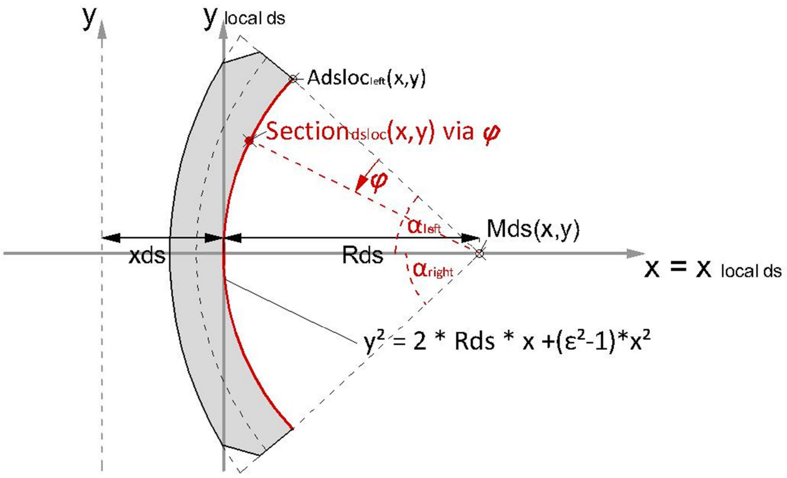
the opening angle of the left side of the downstream arch can be determined. Analogous the opening angle of the downstream arch of the right side can be calculated using the corresponding right downstream abutment coordinates. To gain the full opening angle these two partial opening angles have to be summed up.

The amount of section coordinates, which have to be calculated, depends on the number of sections chosen within the core data definition. The full opening angle from above has to be divided by the chosen number of sections to gain the corresponding angle phi for each of the section points. With the correlation of the tangent, which can be written as

$$\tan(\varphi) = \frac{y}{Rds - x} \tag{4-54}$$

and the vertex form of the equation of the conic section of the downstream arch the remaining section coordinates can be calculated, depending on the corresponding angle phi using the correlation

$$\tan(\varphi) * (Rds - x) = \sqrt{2 * Rds * x + (\varepsilon^2 - 1) * x^2} \tag{4-55}$$



(fig. 4.36) The calculation of the remaining section coordinates

Analogous to the downstream surface the section coordinates of the upstream surface can be calculated using the corresponding upstream parameters.

4.3.5 THE CALCULATION OF THE CHORD LENGTH FROM THE INCLINATION OF THE ABUTMENT

As the calculation method via k is chosen within the core data definition, the parameter B has to be calculated from this parameter k (fig.4.37), due to the fact that parameter B is required to calculate the coordinates of a horizon. To gain the parameter B of a horizon2, the intersection of the vertex form of the equation of the middle arch of horizon2 and a circle, with a radius depending on the inclination k and centre at the abutment of the middle line of the horizon1, which is lying underneath horizon2, has to be determined. Using the equation of a decentred circle from (3-43), with a radius R_{kr} and known centre $Ams_1(x,y)$, which can be written as

$$y = \sqrt{Rkr^2 * (x - xAms_1)^2} - yAms_1 \tag{4-56}$$

with

$$Rkr = k * dz \tag{4-57}$$

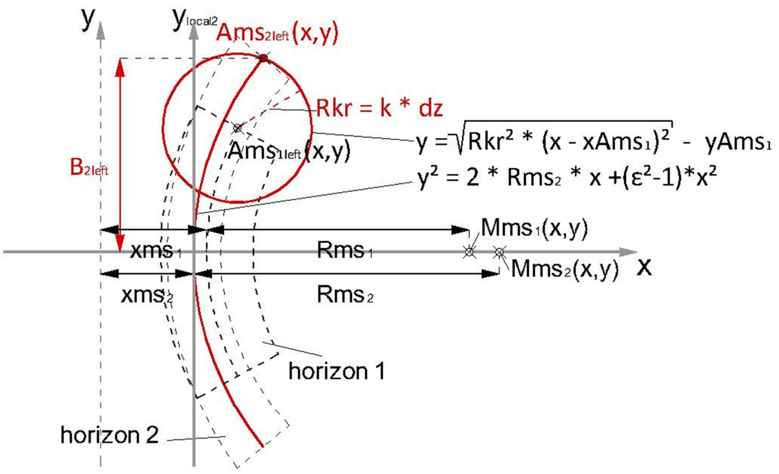
with

dz [m] .. difference in altitude between horizon1 and horizon2

and the vertex form of the equation of a conic section of the middle arch of horizon2, leads to the correlation

$$\sqrt{Rkr^2 * (x - xAms_1)^2} - yAms_1 = \sqrt{2 * Rms_2 * x + (\varepsilon^2 - 1) * x^2} \tag{4-58}$$

As x, which represents $xAms_{2,}$ is extracted and calculated from the correlation above, $yAms_2$, which is equal to $B_{horizon2}$, can be determined using the equation (4-56).



(fig. 4.37) The calculation of the parameter B via k

5 Python: The Interface between the coordinates and Abaqus

5.1 Introduction into Python ·[2]

As the calculation of the coordinates is done, the results have to be processed to create a solid 3D model within Abaqus CAE. Abaqus CAE uses Python as standard programming language, which means that Python commands are used internally to regulate the program. Therefore, the calculated coordinates have to be integrated within a Python script providing the necessary commands to create the solid model with Abaqus. The Arch Dam Design – Input Manager creates this necessary script, which supplies these Python commands automatically when clicking on the corresponding button.

Python is an object orientated programming language, comparable with VBA, which is explained in the chapter above. Within Python, the data, which is related to an object, is called *object member* and functions, which are used to manipulate the object are called *methods*.

The sub chapters below show, how to create a solid model of an arch dam within Abaqus CAE using Python *objects* and *members*.

5.2 Commands Overview·[2]

5.2.1 Import commands

The first two lines of the script, which are listed below, guarantee the access to pre-defined *objects* and *symbolic constants* used by Abaqus. These lines are fundamental to make a Python script, more precisely the used objects and methods, work.

```
from abaqus import *
from abaqusConstants import *
```

The next two lines of the script are used to gain access to objects, which are related to sketches and parts. Sketch and Part represent a *method* within Python.

```
import sketch
import part
```

Beside these import commands, Python provides other import methods, which are not used within the script for the creation of the arch dam, due to fact that they are not required. To simplify and shorten the script code a variable named *myModel* is defined at the beginning of the script. The definition of the variable contains the expression *Model*, to create a model named 'Arch Dam Design' and the expression *mdb.*, which is used to store the model in the Abaqus model database. As the new model is called within a script command later, it is adequate to use the variable name only.

```
myModel = mdb.Model(name='Arch Dam Design')
```

5.2.2 COORDINATES DEFINITION

As the import and model definition lines are done, four variables have to be defined for each horizon, providing the coordinates calculated with ADD. The variables are horizonright$_i$ and horizonleft$_i$, which contain the coordinates of the right and left abutment of each horizon and horzonUS$_i$ and horzonDS$_i$, which provide the coordinates of the upstream and downstream arch of each horizon. It has to be mentioned the three x-coordinates and three y-coordinates are stored in each abutment variable, which are the coordinates of the downstream abutment, the coordinates of the phase and the coordinates of the upstream abutment, corresponding to each side and that the amount of x and y-coordinates of each arch variable depend on the number of sections, chosen within the core data definition.

```
#set x and y coordinates for each horizon
horizonright1 = ((44.731,-62.695) ,(9.512,-93.624) ,(-10.141,-92.349) ,)
horizonleft1 = ((-29.741,22.316) ,(-11.628,30.046) ,(31.866,12.573) ,)
horizonUS1 = ((31.866,12.573) ,(30.711,10.806) ,(29.626,8.997) ,……………………………)
horizonDS1 = ((-10.141,-92.349) ,(-11.685,-89.799) ,(-13.179,-87.219)………………)
```

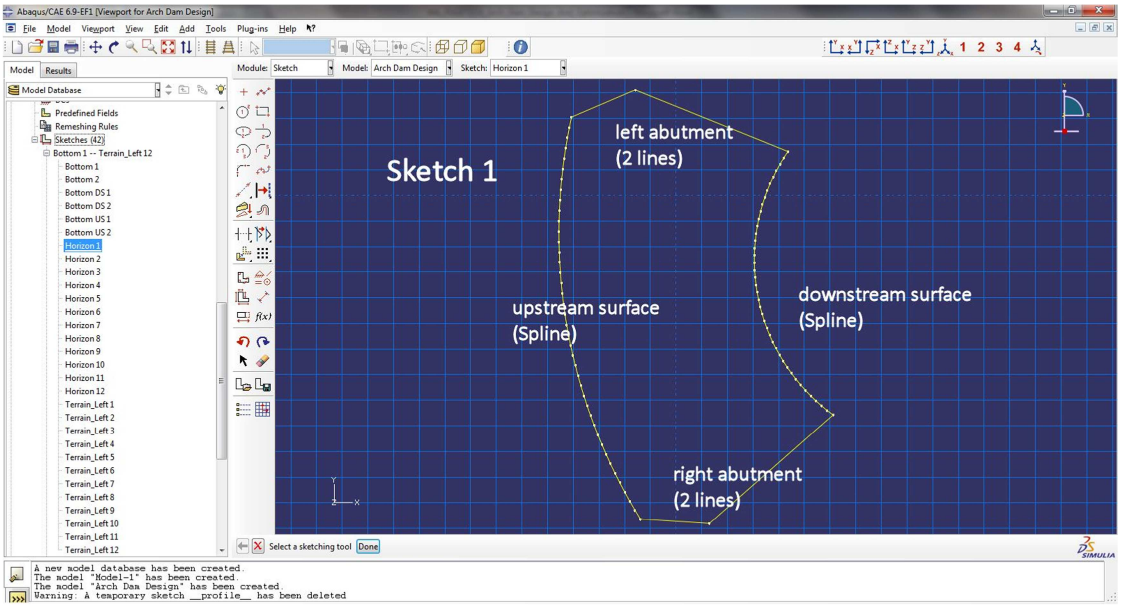
5.2.3 THE CREATION OF SKETCHES

Beside the model, sketches are the second type of objects, which are created within an arch dam script. Sketches can be used to draw cross-sections of the arch dam using different types of elements like lines, arches or splines (fig.5.1). The creation of a sketch starts with the definition of a numbered variable, which contains the variable *myModel*, the expression *ConstrainedSketch*, to create the sketch object and the name of the sketch as well as its approximate size. The use of the variable myModel is needed to refer the sketch to the corresponding model. It has to be noted that the number of cross-sections and therefore sketches of the 3D model, depends on the number of horizons chosen within the core data definition.

```
#create sketch for each horizon
mySketch1 = myModel.ConstrainedSketch(name='Horizon 1', sheetSize=500.0)
```

As the variables of the sketches are set, the drawing of lines and splines is done within the new sketch. To draw the right and left abutment the *Line* command is used within a *for* loop, which creates two lines at each abutment using the coordinate variables, pre-defined in the previous step. The drawing of the upstream and downstream arch is done by using the *Spline* command. The coordinates for the upstream and downstream arch are referred to the spline with the *points* syntax and the corresponding arch variable. Having two lines for each abutment and one spline for each arch, causes each sketch to have an amount of six elements.

```
#draw sketches
for i in range(len(horizonright1)-1):mySketch1.Line
(point1=horizonright1[i],point2=horizonright1[i+1])
for i in range(len(horizonleft1)-1):mySketch1.Line
(point1=horizonleft1[i],point2=horizonleft1[i+1])
mySketch1.Spline(points=horizonUS1)
mySketch1.Spline(points=horizonDS1)
```



Module: Sketch Model: Arch Dam Design Sketch: Horizon 1

```
A new model database has been created.
The model "Model-1" has been created.
The model "Arch Dam Design" has been created.
Warning: A temporary sketch __profile__ has been deleted
```

(fig. 5.1) The elements of a sketch

5.2.4 THE CREATION OF DATUM PLANES

After the definition of the sketches, it is necessary to create datum planes. Datum planes are used as construction aid, providing geometry data, which is required to create the 3D model. When using horizontal planes, a specific offset represents the geometry information. Therefore horizontal datum planes are used within the script to set the elevation of the different horizons of the arch dam (fig.5.2).

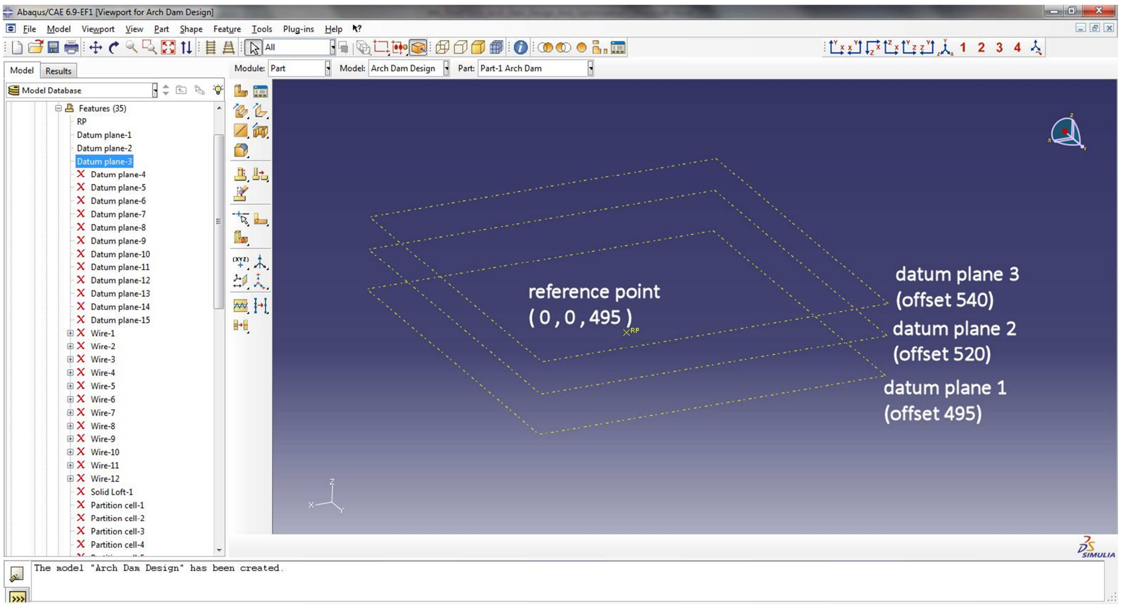
To create a datum plane it is required to create a part first. Creating a part is done by implementing the part variable *myPart* using the variable *myModel*, to refer the part to the corresponding model and the expression *Part*, to create the part. Additionally the name of the part, the dimensionality and type has to be set. It has to be noticed that the determination of the part variable is useful to shorten the script code.

```
#create Datum Planes
reference=(0,0,495)
myPart = myModel.Part(name='Part-1 Arch Dam',dimensionality=THREE_D,type=DEFORMABLE
_BODY)
```

As the creation of the part is done, the datum planes can be referred to the part, using the part variable and the expression *DatumPlaneByPrincipalPlane*. When creating horizontal planes the principal plane has to be the XY plane and the offset has to be taken from each horizon elevation, which causes the model to have the same number of datum planes as number of horizons.

```
myPart.ReferencePoint(point=(reference))
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=495)
```

Beside the datum planes, a reference point can be defined using the corresponding syntax. A reference point is useful to set the default view, when using the program and to refer parts of the model to a specific point.



Module: Part Model: Arch Dam Design Part: Part-1 Arch Dam

```
The model "Arch Dam Design" has been created.
```

(fig. 5.2) Horizontal datum planes with different offset

5.2.5 THE CREATION OF WIRES

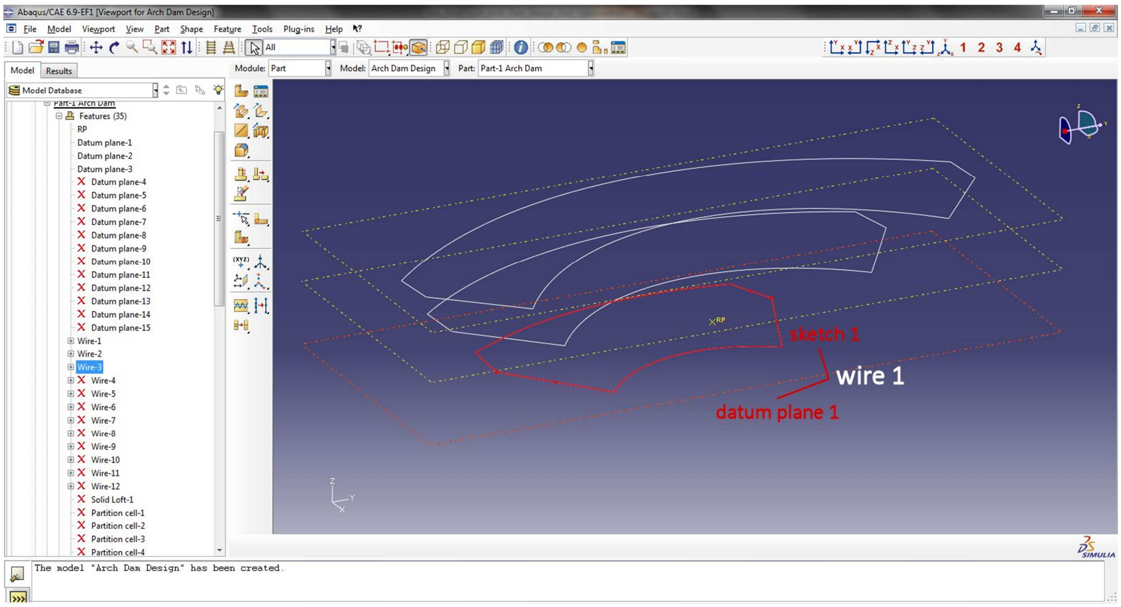
The creation of a wire is necessary to put all the previous geometry information together. Within the arch dam script, a wire represents the cross section of the arch dam at a certain elevation. Therefore it is necessary to refer a sketch, which provides the x and y coordinates and a datum plane, which provides the z coordinate, to the wire.

To determine a specific sketch with the corresponding datum plane for the creation of a wire, it is necessary to set some definitions previously. Beside the implementation of a *myPart* and a *myPlane* variable to shorten the script code, a *myTransformation* and a *myCurrentSketch* definition is needed, using the following syntax.

```
#create Wires
myPart = mdb.models['Arch Dam Design'].parts['Part-1 Arch Dam']
myPlane = myPart.datums

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[2],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))

myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)

myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam Design'].sketches['
Horizon 1'])
```

As the definitions are done the wire can be created using the *myPart* variable, to create the part and refer it to the corresponding model and the expression *Wire*, to prompt the new part to be a wire. Within the wire syntax the previous definitions are inserted for the definition of the datum plane and the specific sketch.

```
myPart.Wire(sketchPlane=myPlane[2], sketchPlaneSide=SIDE1, sketchOrientation=RIGHT,
sketch=myCurrentSketch)
```



```
The model "Arch Dam Design" has been created.
```

(fig. 5.3) A wire, created out of a datum plane and a sketch

5.2.6 Lofting wires to create a solid model

As all the wires according to the number of horizons are set a solid part can be created using the wires as known cross sections. To create a solid part the *myPart* variable as well as the expression *SolidLoft* has to be used. The solid loft syntax contains so called loft sections, which are taken from the edges of each wire, to determine the cross sections of the model. As a wire is created out of a sketch with six elements (four lines and two splines), every wire has six edges which are stored in an edge database. The following algorithm should explain the numbering of these edges, which is necessary to select the correct edges of each wire.

$$myEdge[i]_j = n_{elements} * \left(\frac{n_{horizons}}{2} - 1 \pm j\right) + i \tag{5-1}$$

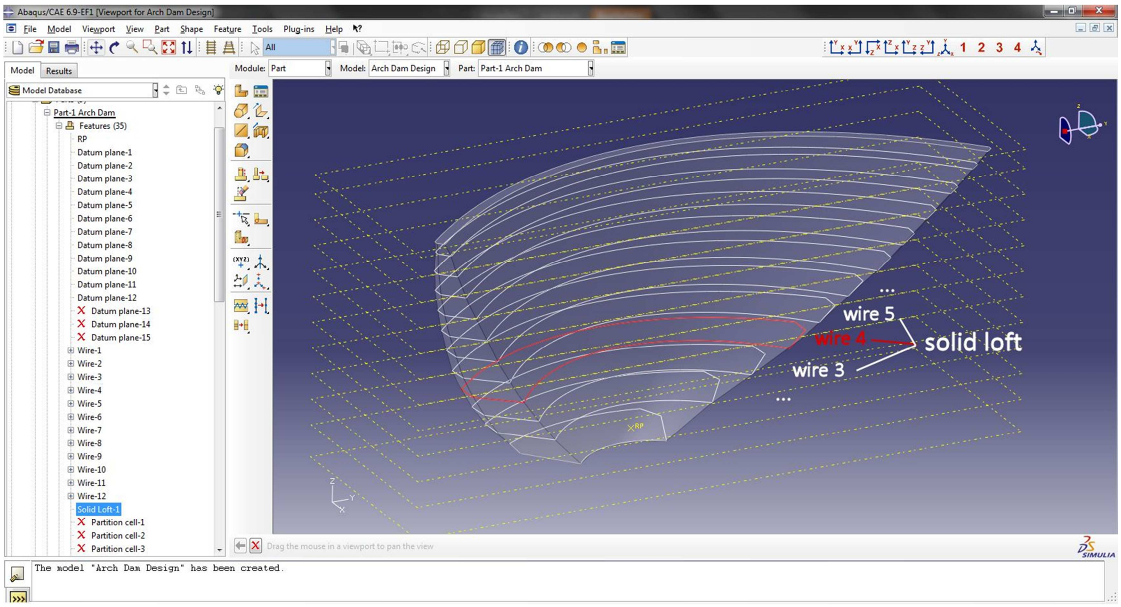
$$with\ 0 \leq i \leq n_{elements} - 1\ \ and\ 0 \leq j \leq \frac{n_{horizons}}{2}$$

with

$n_{elements}$	[-]	.. number of elements of each wire (6)
$n_{horizons}$	[-]	.. number of horizons chosen within the core data definition

For example, when having 12 horizons, the numbering of the edges of the base horizons starts with the number 30 and ends up with the number 35, using a variable i from 0 to 5 and a variable j of 0. For the next cross sections the variable i uses the same variation but the variable j varies from 0 to the number of horizons divided by 2 using an alternating sign, which means that j is +1 for the second horizon and -1 for the third and +2 for the fourth horizon and -2 for the fifth, until reaching the crest horizon using a variable j of +6.

```
#lofting arch dam
myEdge = myPart.edges
myPart.SolidLoft(loftsections=(
(myEdge[30], myEdge[31], myEdge[32], myEdge[33], myEdge[34], myEdge[35]),
(myEdge[36], myEdge[37], myEdge[38], myEdge[39], myEdge[40], myEdge[41]),
(myEdge[24], myEdge[25], myEdge[26], myEdge[27], myEdge[28], myEdge[29], ),…………
```



Module: Part Model: Arch Dam Design Part: Part-1 Arch Dam

Drag the mouse in a viewport to pan the view

The model "Arch Dam Design" has been created.

(fig. 5.4) A solid loft created out of wires

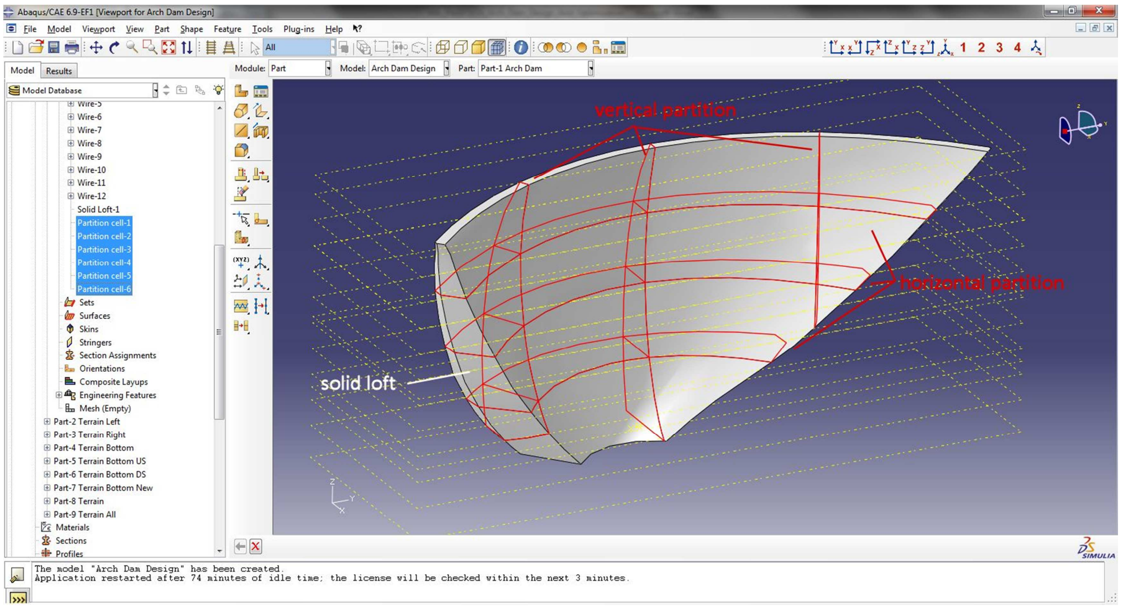
5.2.7 THE PARTITION OF THE SOLID MODEL

Partitions of a solid part are needed to investigate the stresses and displacements within the solid structure. For the analysis of the arch dam three horizontal and three vertical partitions are created to evaluate these analysis results.

The creation of a horizontal partition or cutting plane is done by using the *myPart* variable and the *PartitionCellByDatumPlane* expression, which requires a specific datum plane to be the cutting plane and the *cells* expression, which determines the position of the solid, which have to be cut.

The vertical partitions are done by using the syntax *myPart* and *PartitionCellByPlaneThreePoints*, which creates a cutting plane from three known points. Beside the three known points of the cutting plane the *cells* expression, which determines the position of the solid, is required to create vertical partitions.

```
#create partition
myPart = mdb.models['Arch Dam Design'].parts['Part-1 Arch Dam']
myCells = myPart.cells
pickedCells = myCells.findAt(((0.031,11.657,715), ))
myDatum = myPart.datums

myPart.PartitionCellByDatumPlane(datumPlane=myDatum[16], cells=pickedCells)……

myPart.PartitionCellByPlaneThreePoints(point1=(0.031,11.657,715),
point2=(8.03,11.532,715), point3=(0.031,11.657,495), cells=pickedCells)…………………
```



```
The model "Arch Dam Design" has been created.
Application restarted after 74 minutes of idle time; the license will be checked within the next 3 minutes.
```

(fig. 5.5) The partition of a solid part

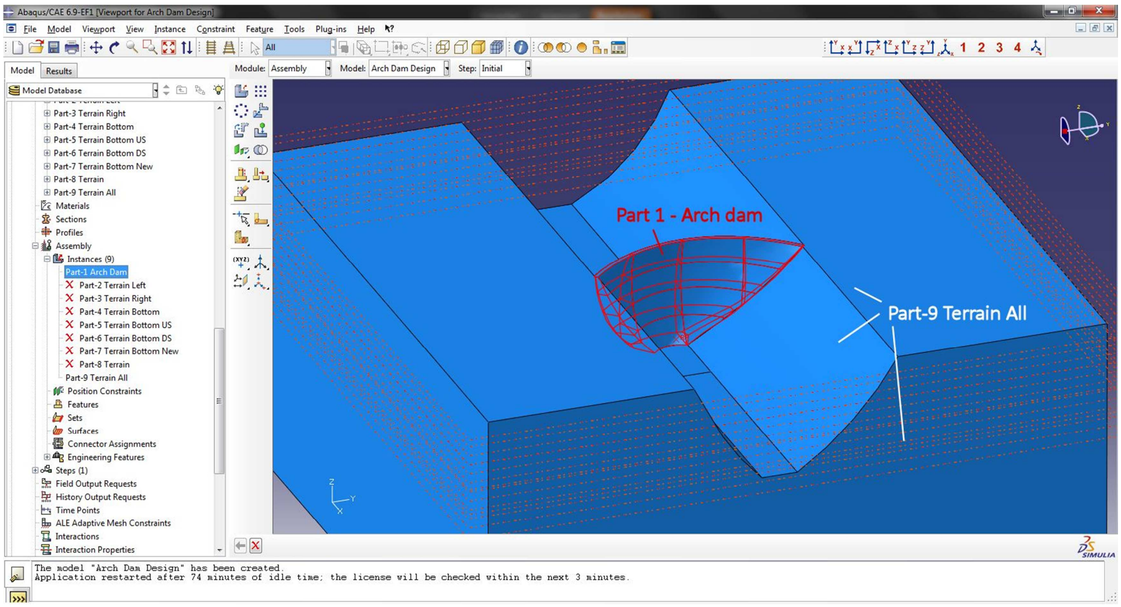
5.2.8 THE ASSEMBLY OF THE PARTS TO CREATE ENTIRE CALCULATION MODEL

The last step to create the 3D model is the assembly of the single parts. Beside the arch dam, the parts of the virtual terrain have to be added to the assembly to create the full calculation model.

To add a part to the assembly, the *MyPart* variable has to be determined using the expression *mdb.models*, which specifies the name of the model and the expression *parts*, which defines the part, that have to be added. Beside the MyPart variable, the variable *MyAssembly*, which is used to shorten the script code, has to be created, containing the model definition and the *rootAssembly* expression. Parts that are used within the assembly are called instances and therefore the *Instance* expression is used to add a part to the assembly.

In addition to the adding of a part, single parts can be merged and cut. The expression *InstanceFromBooleanCut* is used to cut parts and the expression *InstanceFromBooleanMerge* enables the merging of two parts.

```
MyAssembly = mdb.models['Arch Dam Design'].rootAssembly

MyPart = mdb.models['Arch Dam Design'].parts['Part-1 Arch Dam']
MyAssembly.Instance(name='Part-1 Arch Dam', part=MyPart, dependent=ON)


MyAssembly.InstanceFromBooleanCut(name='Part-7 Terrain Bottom New',
instanceToBeCut=mdb.models['Arch Dam Design'].rootAssembly.instances['Part-4
Terrain Bottom'], cuttingInstances=(MyAssembly.instances['Part-5 Terrain Bottom
US'], MyAssembly.instances['Part-6 Terrain Bottom DS'], ),
originalInstances=SUPPRESS)


MyAssembly.InstanceFromBooleanMerge(name='Part-8 Terrain',
instances=(MyAssembly.instances['Part-2 Terrain Left'], MyAssembly.instances['Part-
3 Terrain Right'], MyAssembly.instances['Part-7 Terrain Bottom New'], ),
keepIntersections=OFF, originalInstances=SUPPRESS, domain=GEOMETRY)
```



```
The model "Arch Dam Design" has been created.
Application restarted after 74 minutes of idle time; the license will be checked within the next 3 minutes.
```

(fig. 5.6) The assembly of the created parts

6 Modelling and calculation with Abaqus CAE

6.1 An introduction into the calculation with Abaqus CAE*[1]

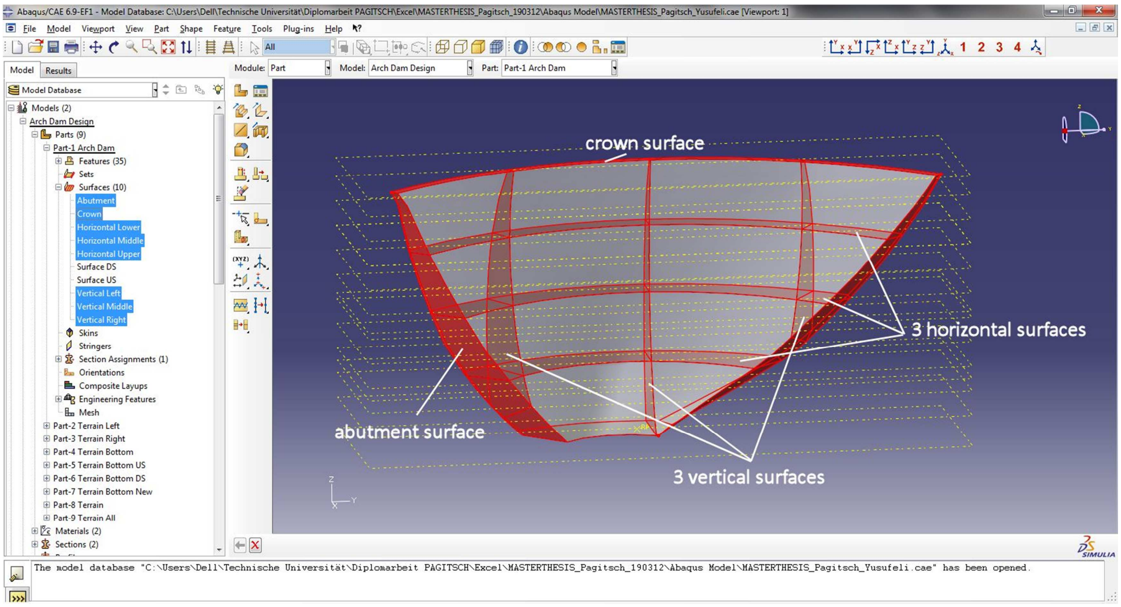
The last part of this master thesis is concerned with the calculation of the stresses and displacements of the arch dam, created with the Arch Dam Design – Input Manager. As the model is imported using the script, which is explained in the previous chapter, some steps are necessary before starting the investigation of the displacements and stresses. Beside the definition of surfaces, which simplify the evaluation of the results, the discretisation of the model, the adding of loads and materials and the setting of boundary conditions, have to be done. The sub chapters from below would detail the procedure to complete to model until starting the calculation.

6.2 Modelling Steps

6.2.1 Define surfaces

Surfaces are created to simplify the investigation of the calculation results of the structural analysis. On the one hand, the surfaces of the partitions are used to display the surface only, when evaluate the results and on the other hand, the definition of the surface of the abutment and the upstream and downstream side is used to facilitate the assigning of loads or boundary conditions.

The surfaces, which are created for the arch dam, are the three surfaces of the vertical and three surfaces of the horizontal partitions and the surface of the abutment as well as the surface of the crown and the upstream and downstream side.



Abaqus/CAE 6.9-EF1 - Model Database: C:\Users\Dell\Technische Universität\Diplomarbeit PAGITSCH\Excel\MASTERTHESIS_Pagitsch_190312\Abaqus Model\MASTERTHESIS_Pagitsch_Yusufeli.cae [Viewport: 1]

Module: Part Model: Arch Dam Design Part: Part-1 Arch Dam

```
The model database "C:\Users\Dell\Technische Universität\Diplomarbeit PAGITSCH\Excel\MASTERTHESIS_Pagitsch_190312\Abaqus Model\MASTERTHESIS_Pagitsch_Yusufeli.cae" has been opened.
```

(fig. 6.1) The surfaces of the arch dam

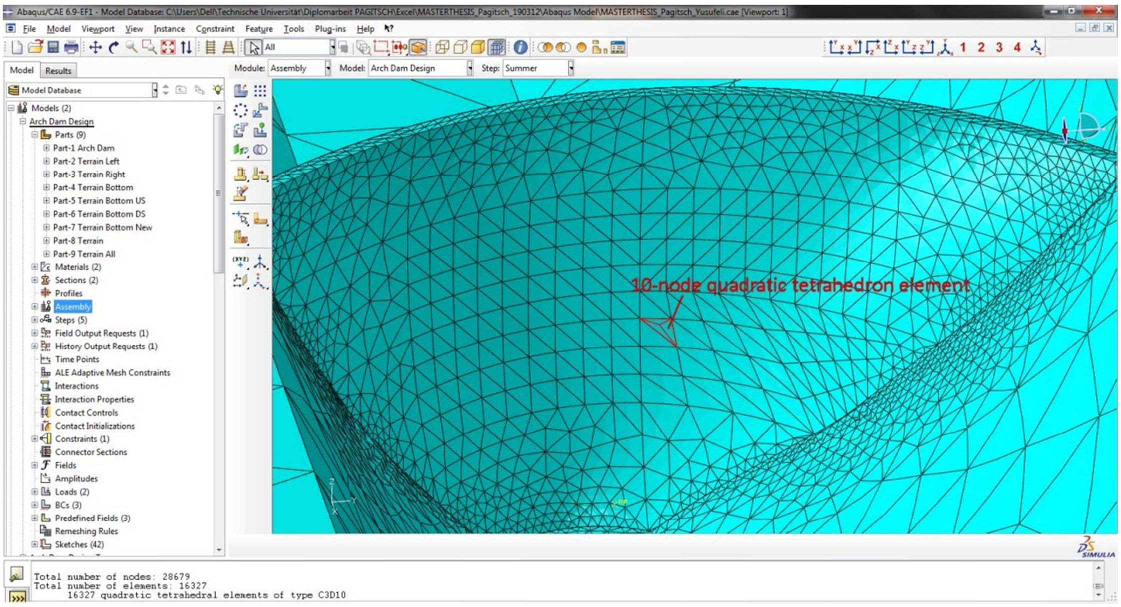
6.2.2 Create a mesh – model discretisation

As mentioned in the FEM chapter 3.3, the model has to be discretised to enable the calculation of the stresses and displacements using a numeric method. The chosen element type for the calculation of the temperature distribution, which is explained in the loads chapter 6.2.4.3, is a 10 node quadratic heat transfer tetrahedron and the chosen element type for the structural analysis is a 10 node quadratic tetrahedron. Using an approximate global size of 15 for the mesh of the arch dam, with refinements along the abutment and crown, leads to 21110 elements with 33459 nodes and using an approximate global size of 120 for the virtual terrain with refinements along the abutment too, results in 16327 elements with 28679 nodes for the virtual terrain.

Table 6-1: Mesh data

Mesh	**Arch Dam**	**Virtual Terrain**
Total number of nodes:	33459	28679
Total number of elements:	21110	16327
*1st elements type**	C3D10	C3D10
*2nd elements type**	DC3D10	DC3D10
DC3D10	A 10-node quadratic heat transfer tetrahedron.	
C3D10	A 10-node quadratic tetrahedron.	

As the structural analysis of the arch dam considers static loads only, the number of elements is appropriate regarding the calculation time. When performing an analysis considering dynamic loads like earthquake accelerations the number of elements should be reduced significantly.



Abaqus/CAE 6.9-EF1 - Model Database: C:\Users\Dell\Technische Universität\Diplomarbeit PAGITSCH\Excel\MASTERTHESIS_Pagitsch_190312\Abaqus Model\MASTERTHESIS_Pagitsch_Yusufeli.cae [Viewport: 1]

Module: Assembly Model: Arch Dam Design Step: Summer

```
Total number of nodes: 28679
Total number of elements: 16327
      16327 quadratic tetrahedral elements of type C3D10
```

(fig. 6.2) The mesh of the calculation model

6.2.3 SET MATERIAL AND SECTION PARAMETERS

After the discretisation of the model, the material and section parameters have to be set for the arch dam and the surrounding virtual terrain. Beside the density, which is necessary to calculate the dead weight of the parts, the Young's Modulus and the Poisson's Ratio have to be determined to set the structural behaviour of the different parts. Additionally the expansion coefficient α as well as the thermal conductivity and the specific heat have to be defined to handle the behaviour concerning thermal expansion. As these material parameters are chosen, the section parameters have to be assigned to the different parts.

6.2.3.1 CONCRETE

The material, which is assigned to the arch dam, is concrete. The density of the taken concrete is 0.0024kt/m³ and the Young' Modulus is 25000 MP. The chosen Poisson's Ratio is 0.2 and the Expansion coefficient is 1E-5 m/(m*K). The remaining thermal parameters are the thermal conductivity, with 2.16 W/(m*K) and the specific heat, with 879 J/(kg*°C). The chosen section category is solid and the type is homogenous.

6.2.3.2 ROCK

The material, which is assigned to the surrounding virtual terrain, is rock, with a density of 0.0026kt/m³, a Young' Modulus of 26000 MP and a chosen Poisson's Ratio of 0.2. As there is no consideration of thermal expansion or conductivity within the rock, the Expansion coefficient is 0 m/(m*K), the taken thermal conductivity, which is set to be very small compared to the concrete's conductivity, is 2,16E-3 W/(m*K) and the specific heat of the rock is 1500 J/(kg*°C). Additionally the chosen section category is solid and the type is homogenous.

Table 6-2: Material and section parameters

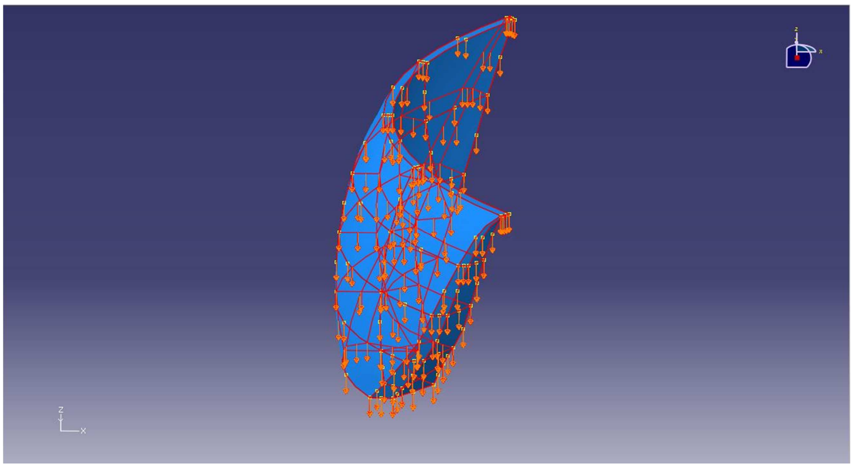
Material parameters	**Arch Dam**	**Virtual Terrain**
Density	0.0024 kt/m³	0.0026 kt/m³
Young's Modulus	25000 MPa	26000 MPa
Poisson's Ratio	0.2	0.2
Expansion coefficient	1E-5 m/(m*K)	0 m/(m*K)
Thermal conductivity	2,16 W/(m*K)	2,16E-3 W/(m*K)
Specific heat	879 J/(kg*K)	1500 J/(kg*K)
Section parameters	**Arch Dam**	**Virtual Terrain**
Category	Solid	Solid
Type	Homogenous	Homogenous
	Isotropic	Isotropic

6.2.4 CREATE LOADS

The loads, which are assigned to the arch dam to investigate the structural behaviour are the dead weight, the water pressure and two different temperature distributions.

6.2.4.1 DEAD WEIGHT

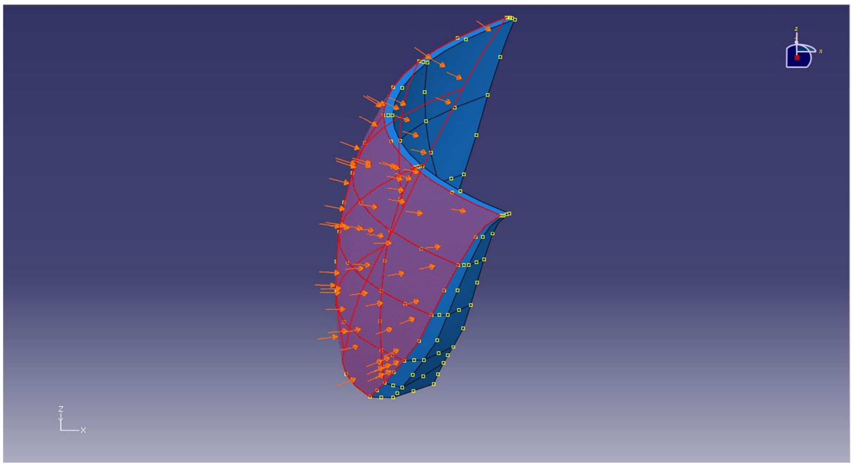
The dead weight is assigned to the whole arch dam and created by using the mass of the arch dam and the acceleration of -9,81 m/s in the z direction (fig.6.3). To simplify the modelling and calculation, the deadweight is assigned monolithic.



(fig. 6.3) The dead weight of the arch dam

6.2.4.2 WATER PRESSURE

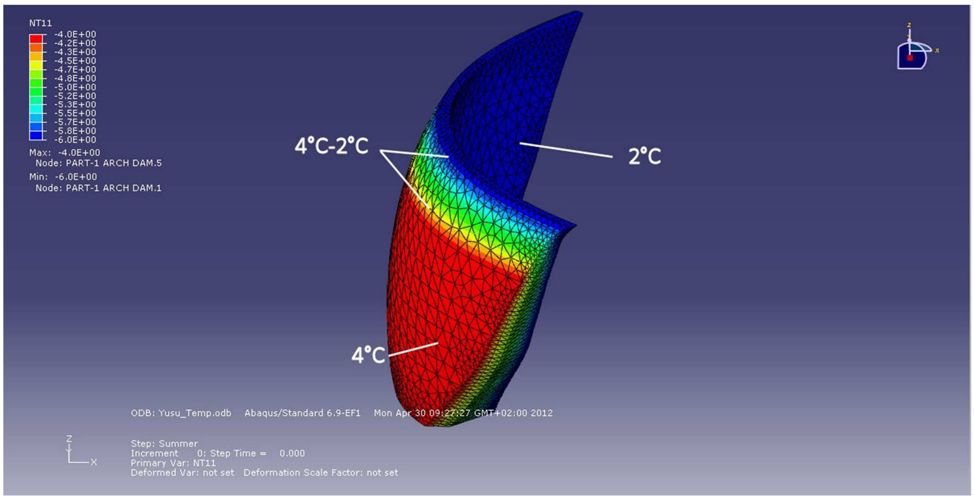
The Water pressure is assigned to the upstream surface of the arch dam only. To set the water pressure the distribution hydrostatic and the magnitude 2.2 MN, which can be calculated from the dam height and the specific weight of water, has to be chosen. Further the zero pressure height, which is the height at the crest elevation and the reference pressure height, which is the height at the base elevation of the dam, have to be determined (fig.6.4).



(fig. 6.4) The hydrostatic distribution of the water pressure

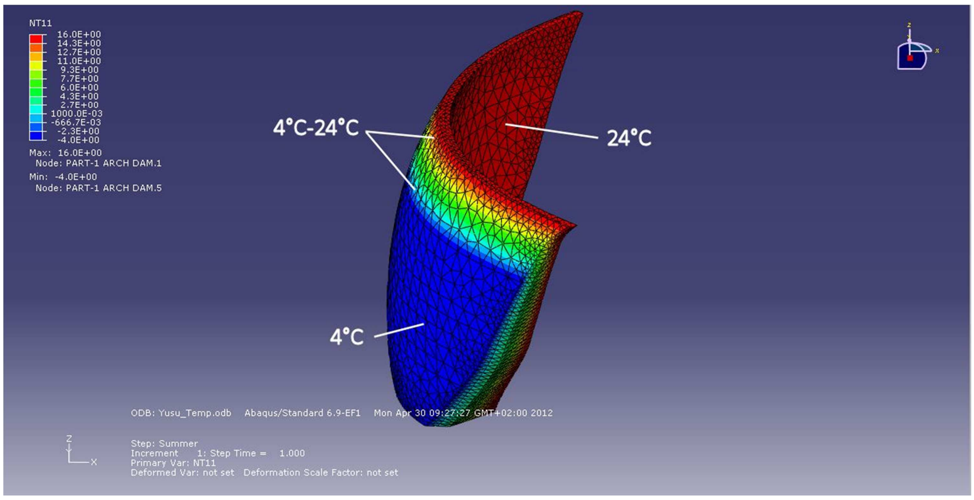
6.2.4.3 Temperature

Two different temperature distributions represent the last load cases, which are assigned to the arch dam. The mean winter temperature at the specific location of the arch dam characterizes the first distribution of the temperature (fig.6.5). To gain this winter temperature induced distribution a temperature of 2°C, which represents the air temperature, has to be assumed for the downstream surface and a temperature of 4°C, which represents the water temperature, has to be taken for the upstream surface. To take account on the transition area a subroutine has to be determined, which handles the linear distribution from 75 percent of the upstream surface height to the crest elevation. Due to the fact that the temperature has to be set relative to the grouting temperature (8°C) the temperature, which is assigned to the downstream surface, is -6°C and the temperature, which is assigned to the upstream surface, is -4°C.



(fig. 6.5) The winter temperature distribution

Analogous to the winter induced distribution, the assumed summer temperature of the downstream surface is 24°C and therefore the assigned temperature is +16°C and the summer temperature of the water, which represents the temperature of the upstream surface, is equal to the assigned winter temperature of the water.

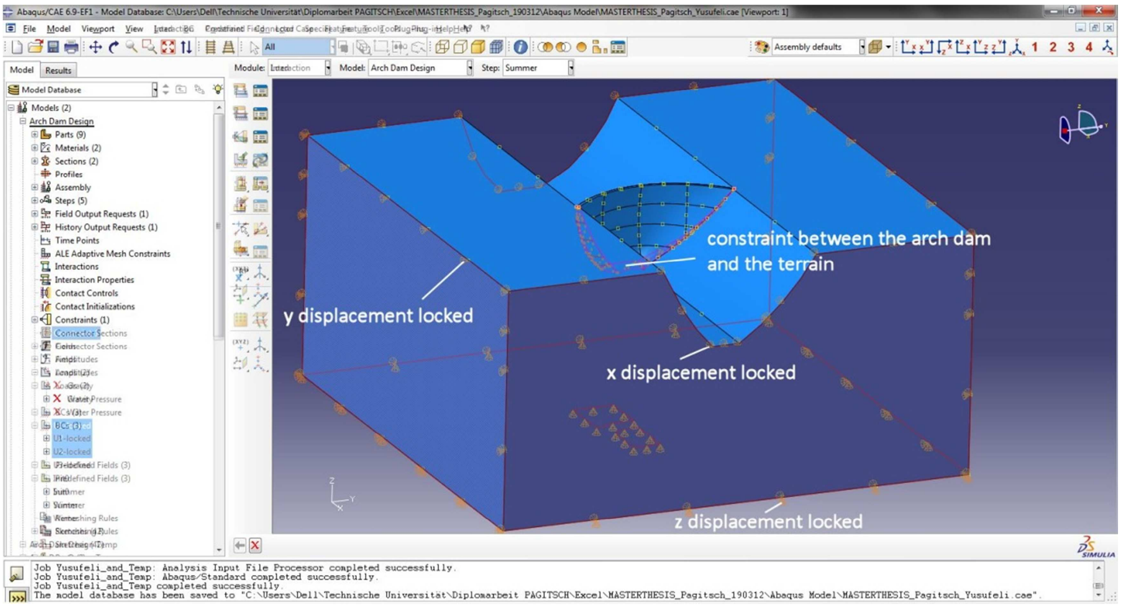


(fig. 6.6) The summer temperature distribution

6.2.5 SET BOUNDARY CONDITIONS BCS AND CONSTRAINTS

The last step before starting the calculation of the stresses and displacements is the definition of the boundary conditions and constraints (fig.6.7). The boundary conditions have to be set for the edges of the whole model, whereas the constraints describe the conditions between two parts within the model.

The boundary conditions are set by locking the displacements of the direction orthogonal to the surface of each edge and the constraints are set by determining the abutment surface between the arch dam and the virtual terrain to be tied.



```
Job Yusufeli_and_Temp: Analysis Input File Processor completed successfully.
Job Yusufeli_and_Temp: Abaqus/Standard completed successfully.
Job Yusufeli_and_Temp completed successfully.
The model database has been saved to "C:\Users\Dell\Technische Universität\Diplomarbeit PAGITSCH\Excel\MASTERTHESIS_Pagitsch_190312\Abaqus Model\MASTERTHESIS_Pagitsch_Yusufeli.cae".
```

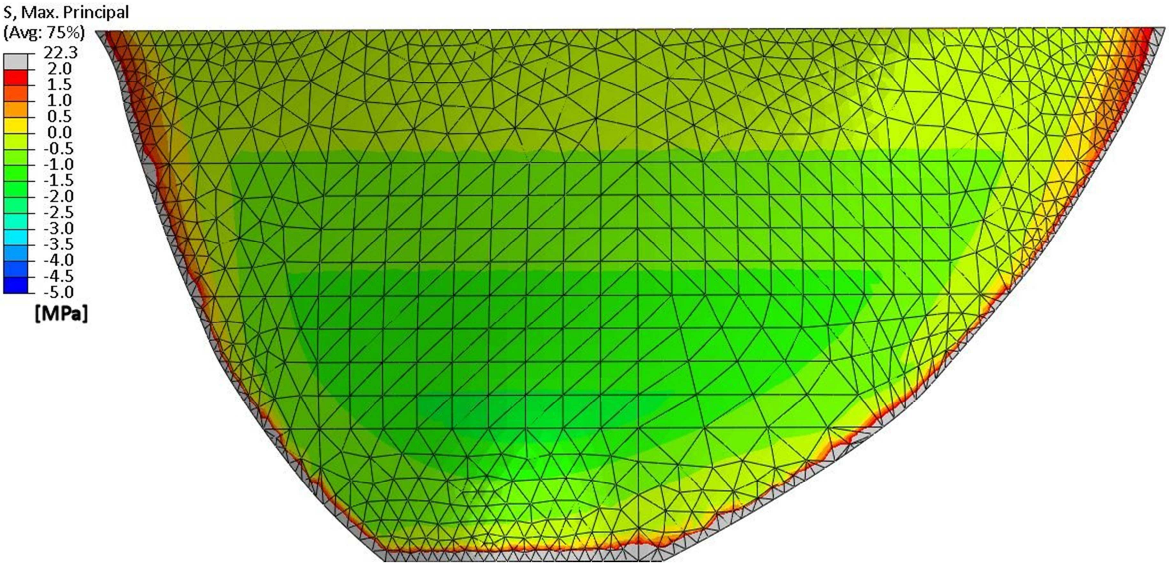
(fig. 6.7) The boundary conditions and constraints of the model

6.3 Calculation results

6.3.1 Upstream surface maximum principal stresses

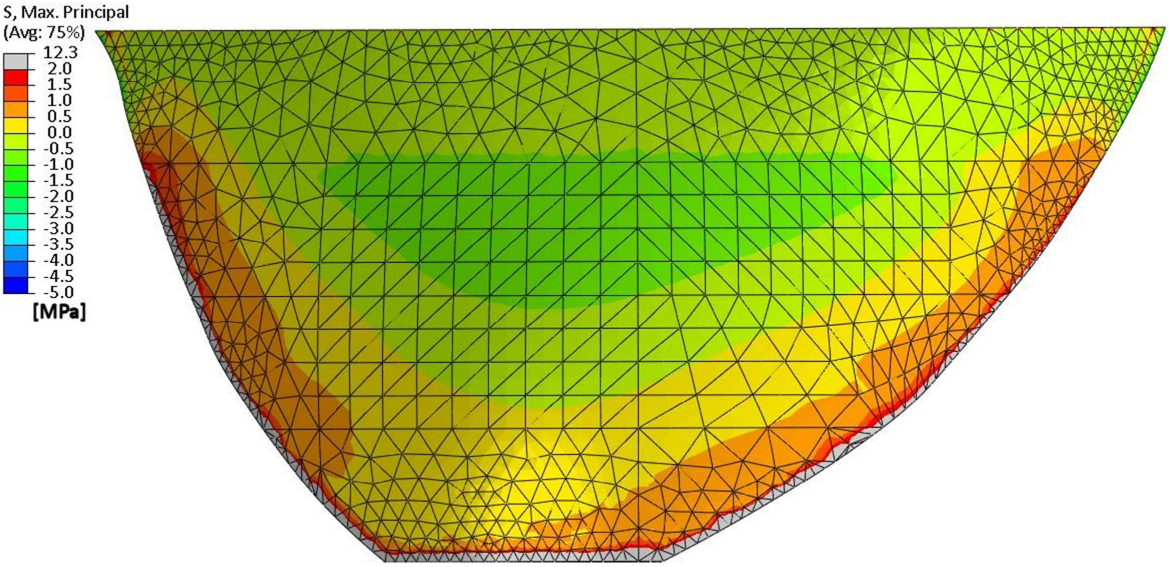
The figures below show the maximum principal stresses at the upstream surface of the arch dam. The first figure shows these stresses with assigned deadweight, hydrostatic water pressure and winter temperature distribution and the second figure shows these stresses with assigned deadweight, hydrostatic water pressure and summer temperature distribution.

6.3.1.1 Winter



(fig. 6.8) Maximum Principal Stresses US Surface (winter)

6.3.1.2 Summer

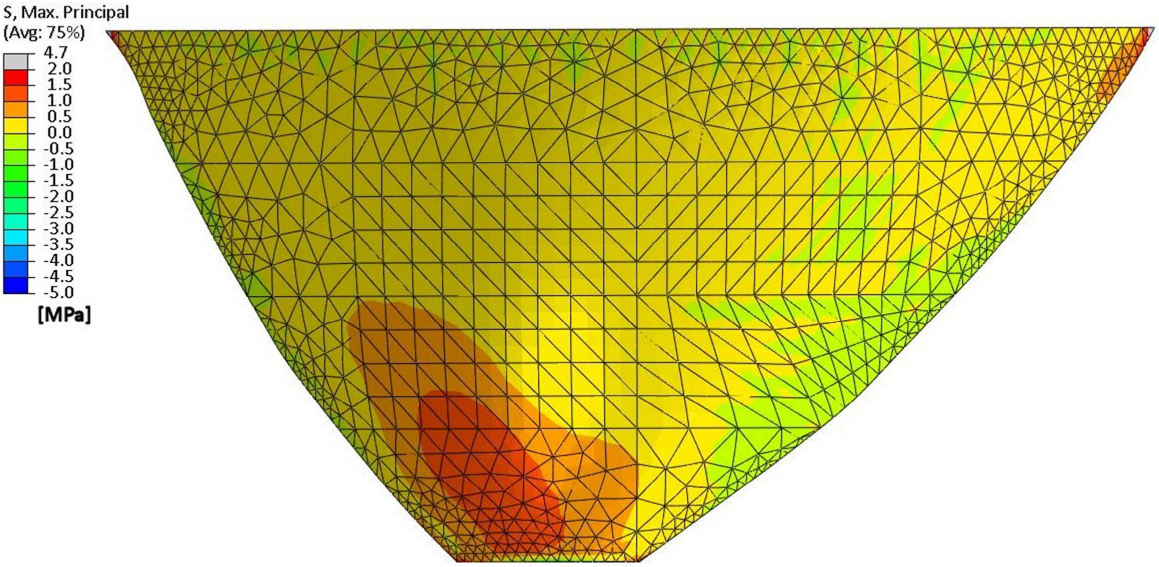


(fig. 6.9) Maximum Principal Stresses US Surface (summer)

6.3.2 Downstream surface maximum principal stresses

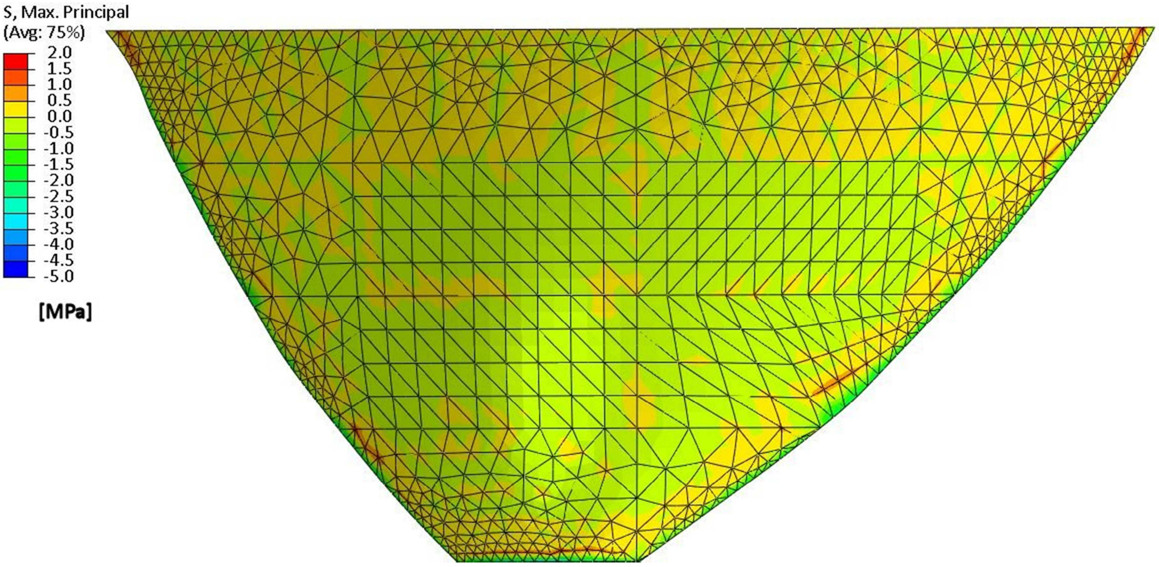
The figures below show the maximum principal stresses at the downstream surface of the arch dam. The first figure shows these stresses with assigned deadweight, hydrostatic water pressure and winter temperature distribution and the second figure shows these stresses with assigned deadweight, hydrostatic water pressure and summer temperature distribution.

6.3.2.1 Winter



(fig. 6.10) Maximum Principal Stresses DS Surface (winter)

6.3.2.2 Summer

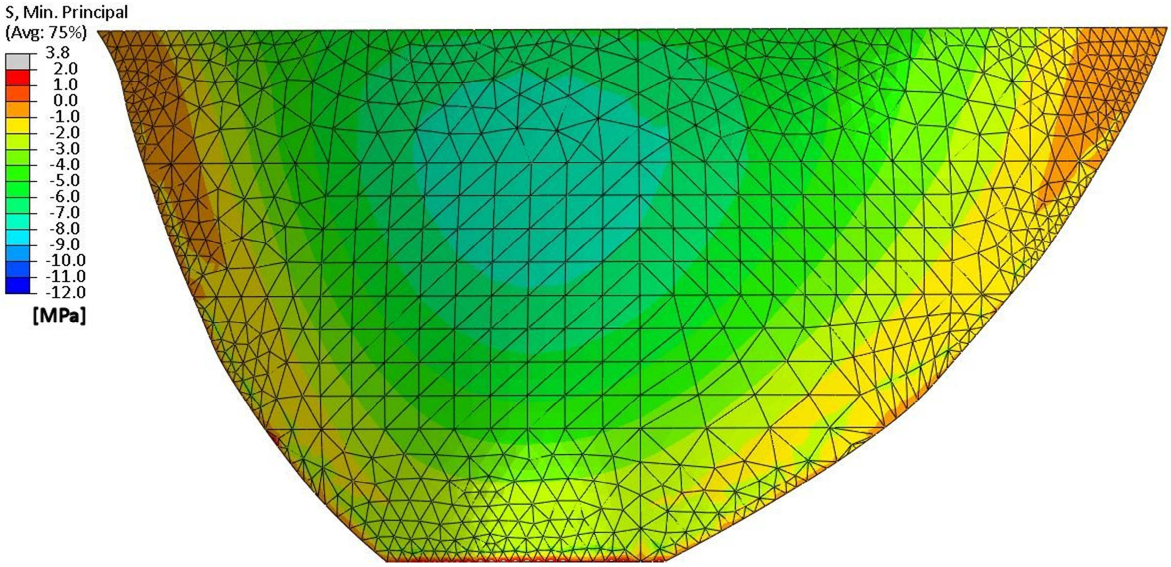


(fig. 6.11) Maximum Principal Stresses DS Surface (summer)

6.3.3 Upstream surface minimum principal stresses

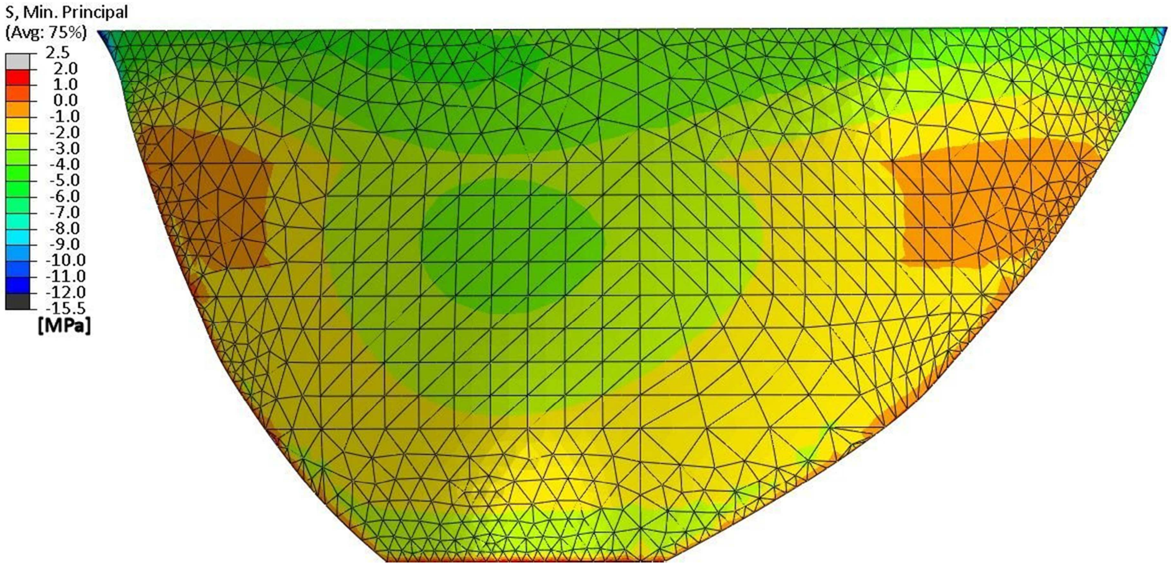
The figures below show the minimum principal stresses at the upstream surface of the arch dam. The first figure shows these stresses with assigned deadweight, hydrostatic water pressure and winter temperature distribution and the second figure shows these stresses with assigned deadweight, hydrostatic water pressure and summer temperature distribution.

6.3.3.1 Winter



(fig. 6.12) Minimum Principal Stresses US Surface (winter)

6.3.3.2 Summer

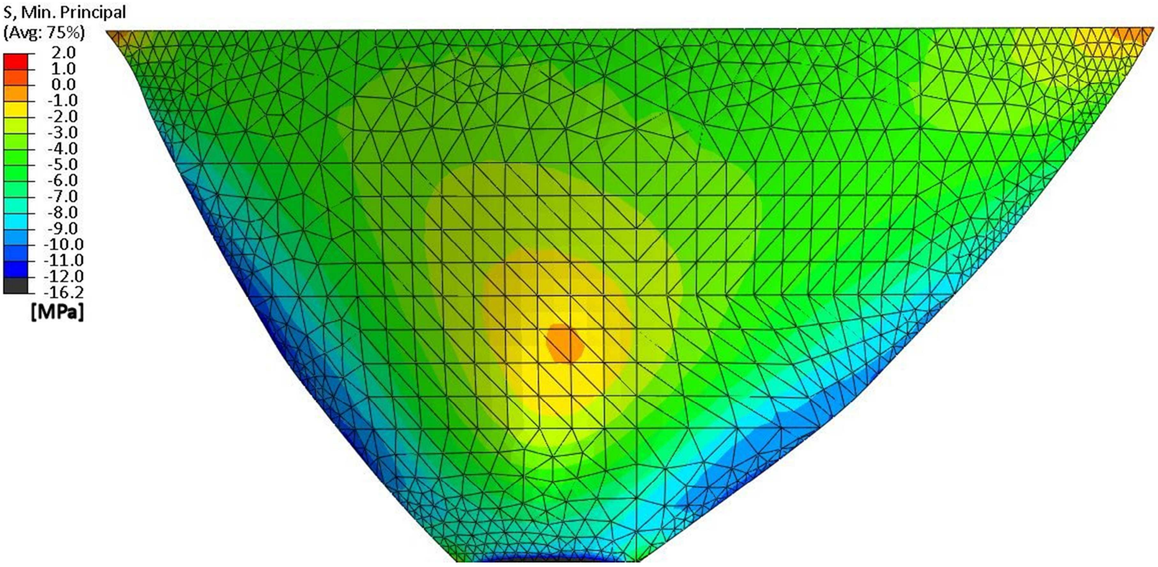


(fig. 6.13) Minimum Principal Stresses US Surface (summer)

6.3.4 DOWNSTREAM SURFACE MINIMUM PRINCIPAL STRESSES

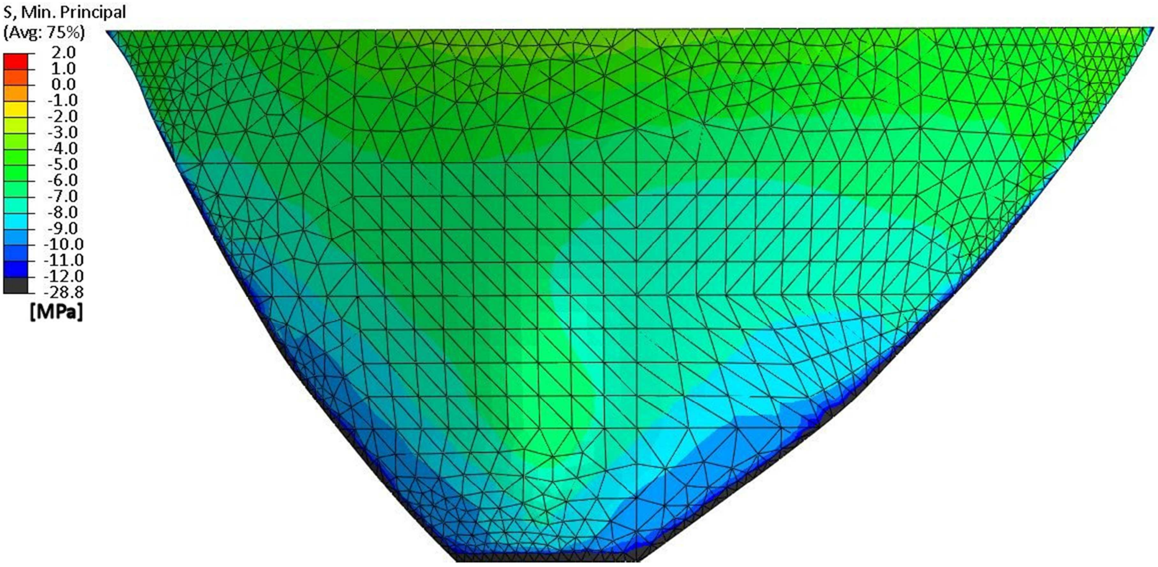
The figures below show the minimum principal stresses at the downstream surface of the arch dam. The first figure shows these stresses with assigned deadweight, hydrostatic water pressure and winter temperature distribution and the second figure shows these stresses with assigned deadweight, hydrostatic water pressure and summer temperature distribution.

6.3.4.1 WINTER



(fig. 6.14) Minimum Principal Stresses DS Surface (winter)

6.3.4.2 SUMMER

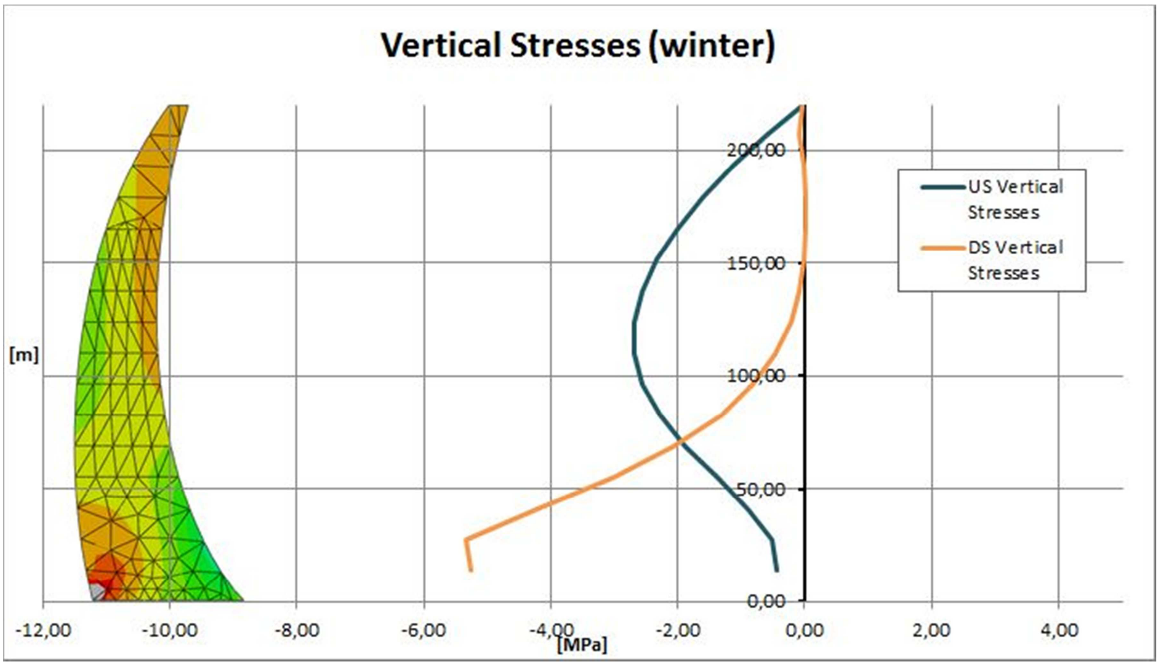


(fig. 6.15) Minimum Principal Stresses DS Surface (summer)

6.3.5 Main section vertical stresses

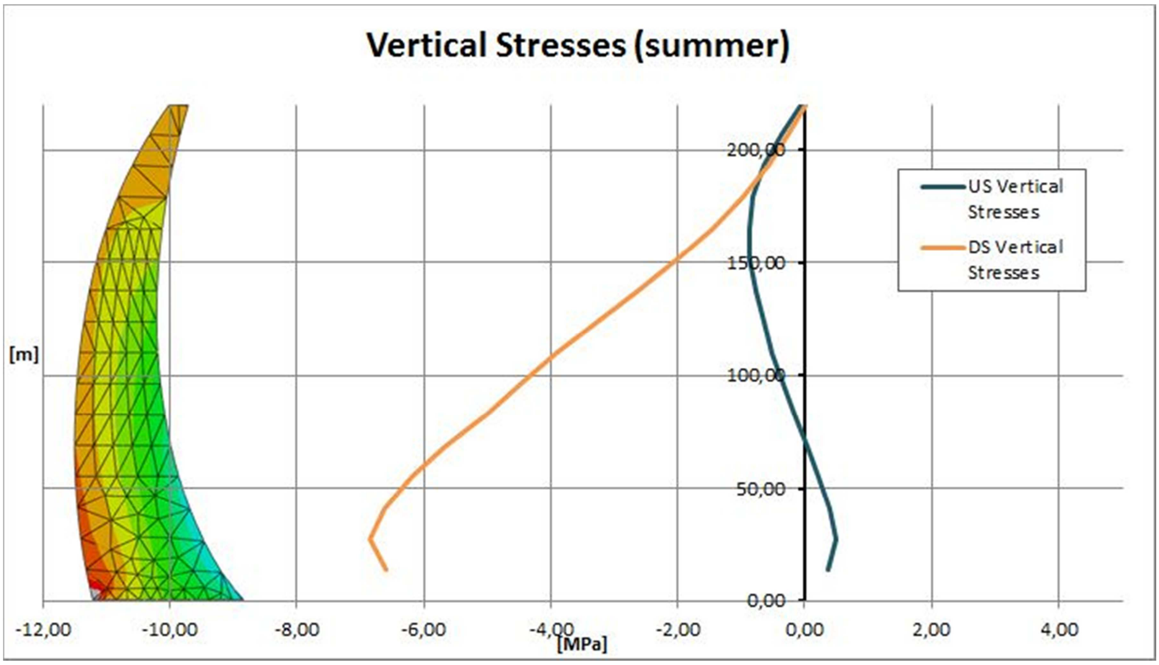
The figures below show the vertical stresses at the main section of the arch dam. The first figure shows these stresses with assigned deadweight, hydrostatic water pressure and winter temperature distribution and the second figure shows these stresses with assigned deadweight, hydrostatic water pressure and summer temperature distribution.

6.3.5.1 Winter



(fig. 6.16) Vertical Stresses (winter)

6.3.5.2 Summer

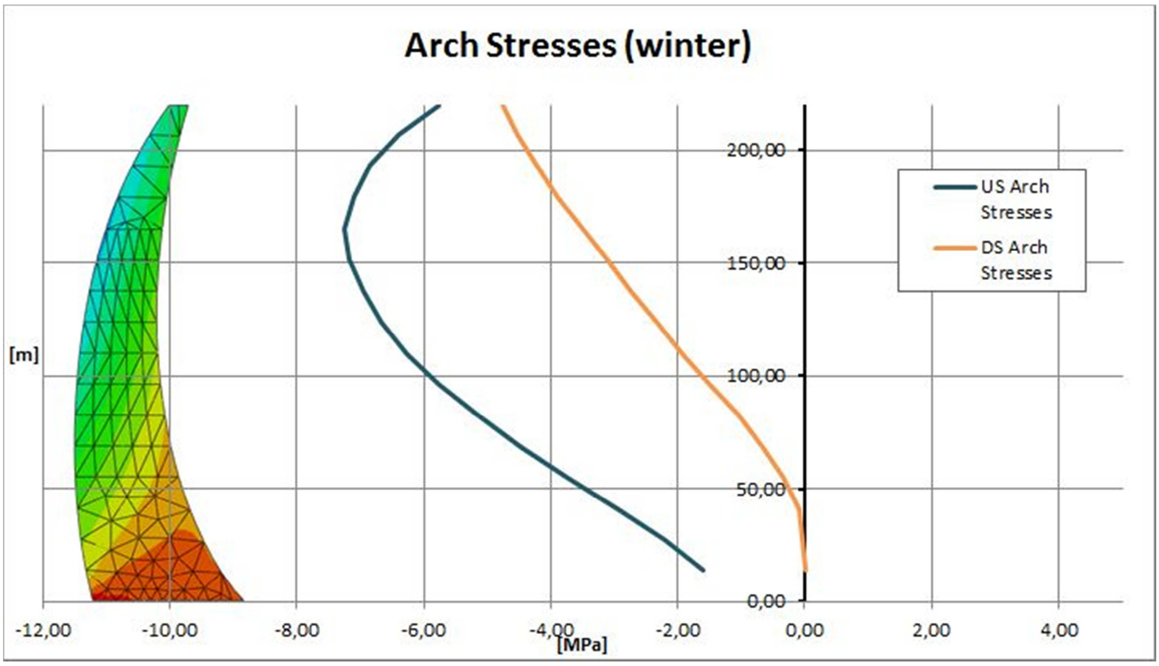


(fig. 6.17) Vertical Stresses (summer)

6.3.6 Main section arch stresses

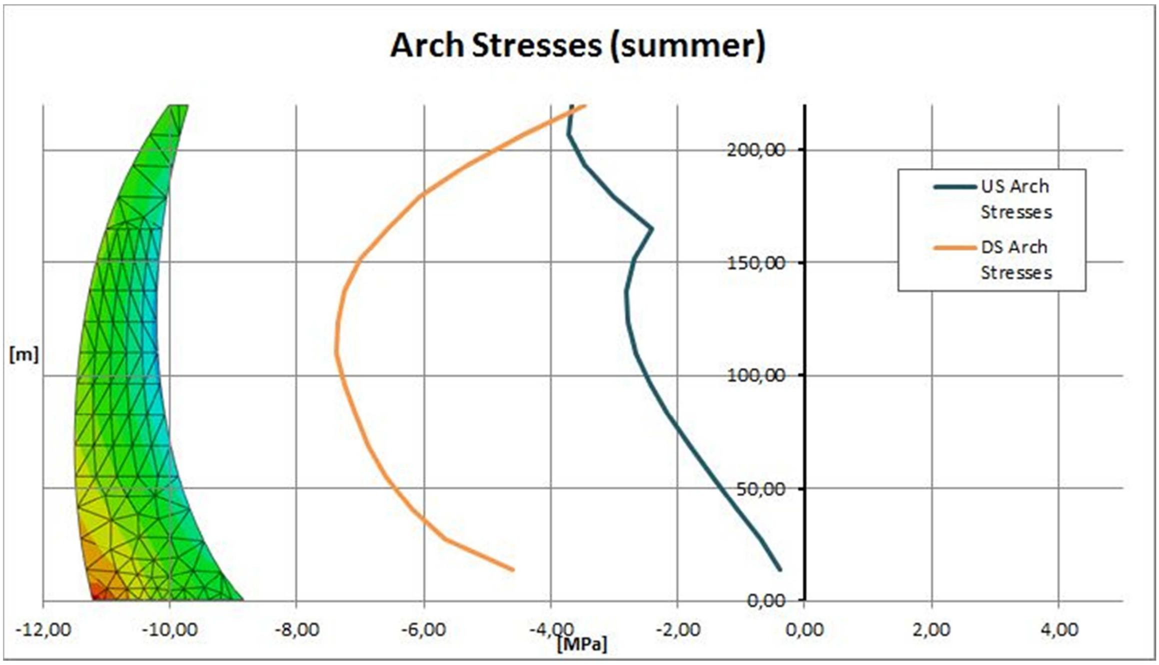
The figures below show the arch stresses at the main section of the arch dam. The first figure shows these stresses with assigned deadweight, hydrostatic water pressure and winter temperature distribution and the second figure shows these stresses with assigned deadweight, hydrostatic water pressure and summer temperature distribution.

6.3.6.1 Winter



(fig. 6.18) Arch Stresses (winter)

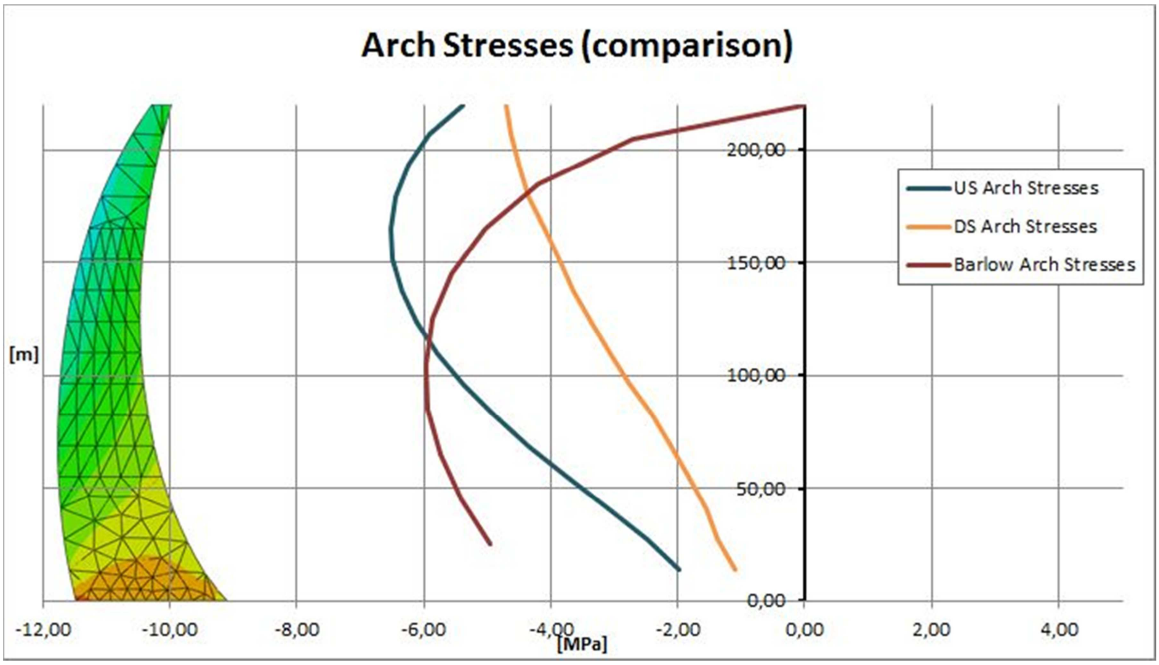
6.3.6.2 Summer



(fig. 6.19) Arch Stresses (summer)

The figure below shows the comparison of the arch stresses calculated with the Finite Element program Abaqus CAE and the arch stresses calculated with Barlow's Formula at the main section of the arch dam. The arch dam is assigned with static loads, which means deadweight and hydrostatic water pressure, only in this case.

6.3.6.3 ARCH STRESSES COMPARISON

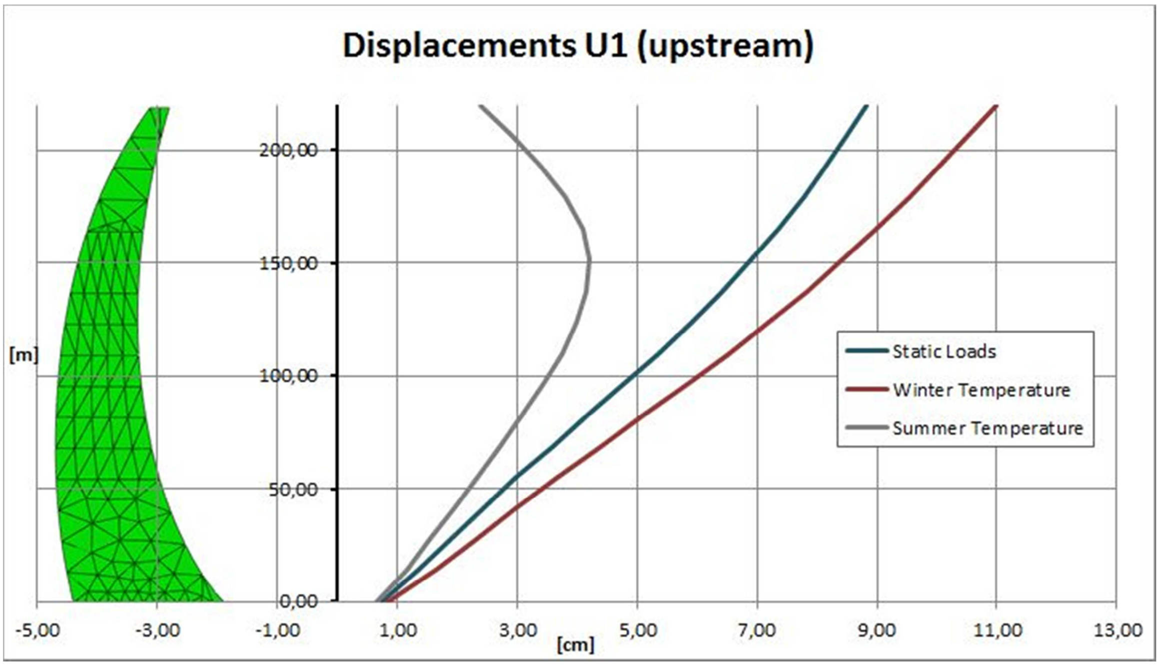


(fig. 6.20) Arch Stresses Comparison

6.3.7 MAIN SECTION DISPLACEMENTS

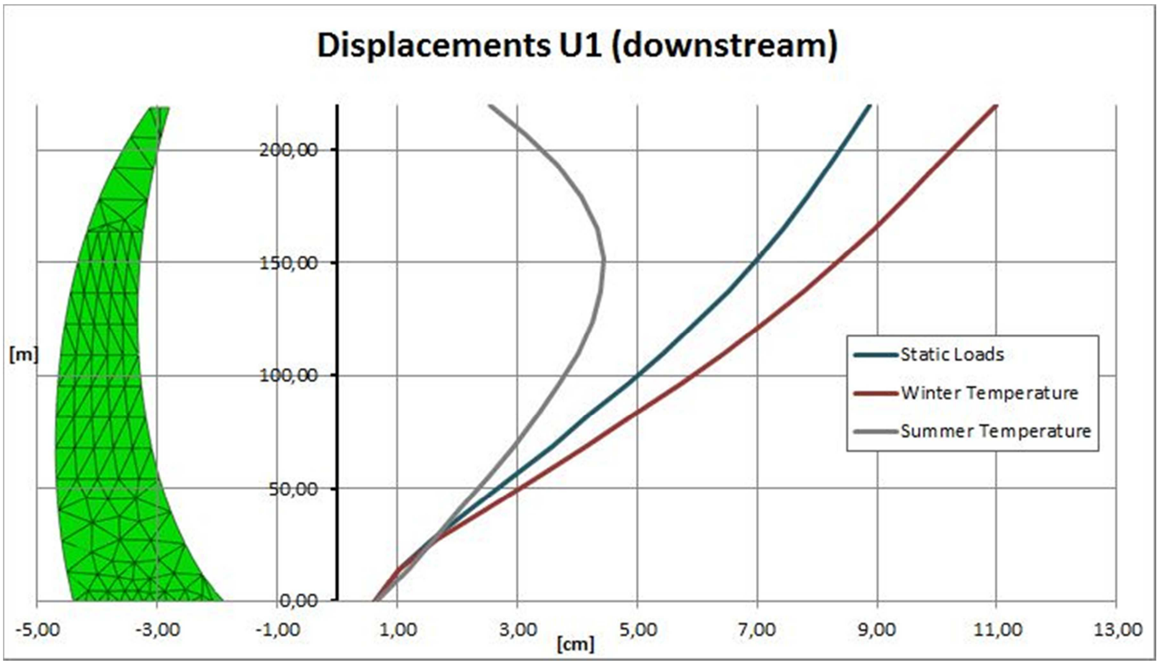
The figures below show the displacements in x direction at the main section of the arch dam. The first figure shows the displacements of the upstream surface and the second figure shows the displacements of the downstream surface with assigned static loads and additional winter temperature distribution or summer temperature distribution.

6.3.7.1 UPSTREAM SURFACE DISPLACEMENTS



(fig. 6.21) Upstream U1 displacements

6.3.7.2 DOWNSTREAM SURFACE DISPLACEMENTS



(fig. 6.22) Downstream U1 displacements

7 SUMMARY

To calculate the 3D coordinates of an arch dam, there are two essential definition steps. On the one hand, the main vertical section, with the x-coordinates and the thickness has to be determined and on the other hand, the horizontal cross-sections, also called horizons, have to be defined using the radii, arch type and chord length of each upstream and downstream arch. These input parameters are necessary when using the vertex form of the equation of a conic section to describe the shape of the horizontal arches. The coordinates of the arch dam can then be calculated by intersecting these defined arches with the abutment of arch dam, which is represented by straight lines at the left and right side.

Within the developed program Arch Dam Design – Input Manager, briefly speaking ADD, the required input steps are supported, using a Graphical User Interface GUI. Different sub forms of the program allow a step by step input procedure to simplify the input and varying of the parameters. Beside the calculated coordinates, the program provides different previews to show the geometry of the arch dam. Additionally the export of the coordinates and a Python script is enabled for further processing. It has to be mentioned that the developed program is a useful tool for the preliminary design of an arch dam however there are necessary add-ons, which have to be done, to use the program to detail an arch dam project. Supporting other arch types instead of conic sections only, calculating the coordinates of the block joints of the arch dam or adding the real topography, would be some of these necessary add-ons.

To gain an overview over the structural behaviour of the chosen arch dam geometry and to investigate the stresses and displacements, as loads are assigned to the arch dam, it is necessary to process the arch dam geometry within a Finite Element program. The Python script export, which is noted above, enables this further processing, by providing the calculated coordinates and some additional script parts to create a solid part within the Finite Element program Abaqus CAE.

When having the chosen geometry of the arch dam within Abaqus CAE some additional steps are required to do the structural analysis. To get an insight into the basic structural behaviour of the arch dam, it is necessary to discretise the solid parts, assign the dead weight, water pressure and some temperature load to the arch dam as well as setting the boundary conditions, prior to the calculation. As the calculation is done the stresses and displacements can be evaluated by investigating the maximum and minimum principal stresses of the upstream and downstream surface as well as the vertical stresses and the arch stresses of the vertical main section.

Finally it can be mentioned that the developed evaluation procedure is useful to do the preliminary design of an arch dam but should be extended to enable a detailed design.

BIBLIOGRAPHY

[1] Abaqus, Inc., 2003. *Abaqus / CAE User's Manual.* Version 6.4 Hrsg. United States of America: Abaqus, Inc..

[2] Abaqus, Inc., 2004. *Scripting User's Manual.* Abaqus Version 6.5 Hrsg. United States of America: Abaqus, Inc..

[3] Bartsch, H. J., 2004. *Taschenbuch mathematischer Formeln.* 20.Auflage Hrsg. Leipzig Germany: Fachbuchverlag Leipzig.

[4] Bathe, K. J., 2002. *Finite-Elemente-Methoden ; aus dem englischen übersetzt von Peter Zimmermann.* 2. Auflage Hrsg. Berlin, Heidelberg Germany: Springer Verlag.

[5] Beer, G., 2009. 1 Einführung in Näherungsmethoden, 2 Methode der Finiten Elemente. *Skriptum zur Vorlesung Finite Elemente Methode.*

[6] Boggs, H. L., 1977. *Guide for preliminary design of arch dams.* 3.Printing Hrsg. Denver, Colorado: United States Department of the Interior: Bureau of Reclamation.

[7] Breden, M. & Schwimmer, M., 2008. *Das Excel-VBA Codebook.* München, Germany: Addison Wesley.

[8] Gross, D., Hauger, W. & Wriggers, P., 2009. *Technische Mechanik 4 - Hydromechanik.* 7.Auflage Hrsg. Heidelberg, Germany: Springer Verlag.

[9] Kemnitz, A., 2010. *Mathematik zum Studienbeginn: Grundlagenwissen für alle technischen, mathematisch- naturwissenschaftlichen und wirtschaftlichen Studiengänge.* 9.Auflage Hrsg. Wiesbaden Germany: Vieweg+Teuber Verlag.

[10] Microsoft Dynamics GP, 2006. *VBA Developer's Guide.* Release 9.0 Hrsg. s.l.:Microsoft Corporation.

[11] Strobl, T. & Zunic, F., 2006. *Wasserbau - Aktuelle Grundlagen, neue Entwicklungen.* 1. Auflage Hrsg. Berlin, Heidelberg Germany: Springer-Verlag.

[12] United States Department of Interior, U., 1977. *Design of arch dams - design manual for concrete arch dams.* Denver, Colorado: United States Government Printing Office.

[13] US Army Corps of Engineers, 1994. *Arch dam design.* Washington DC: Department of the Army: US Army Corps of Engineers.

[14] Wikipedia, 2011. *http://de.wikipedia.org/wiki/Artvin_(Provinz).* Access at 30th of March 2012: s.n.

[15] Witt, K. J., 2008. *Grundbau Taschenbuch Teil 1 - Geotechnische Grundlagen.* 7.Auflage Hrsg. Berlin: Ernst & Sohn.

[16] Zienkiewicz, O. C., Taylor, R. L. & Zhu, J. Z., 2005. *The finite element method: its basis and fundamentals.* Auflage: 6th ed. Hrsg. s.l.:Butterworth Heinemann.

[17] Zilch, K., Diedrichs, C. & Katzenbach, R., 2002. *Handbuch für Bauingenieure.* Berlin, Heidelberg Germany: Springer-Verlag.

LIST OF FIGURES

LIST OF TABLES

List of symbols

Engineering aspects Symbols

G	[N]	dead load
m_d	[kg]	mass of the dam
g	[m/s²]	.... acceleration due to gravity
F_e	[N]	.... force caused by earthquake
m	[kg]	.... mass of the dam/water
a_e	[m/s²]	.... acceleration due to earthquake
$e_{0v,s}$	[MPa]	.... vertical earth pressure caused by silt
$e_{0h,s}$	[MPa]	.... horizontal earth pressure caused by silt
K_0	[-]	.... earth pressure coefficient $K_0 = 1 - \sin(\varphi_s)$
φ_s	[°]	.... friction angle of silt
$\gamma_{u,s}$	[MN/m³]	.... specific weight of the uplifted silt
τ_f	[MPa]	.... shear strength of the intact rock
c'	[MPa]	.... cohesion
$\sigma_n{}'$	[MPa]	.... effective normal stress
φ'	[°]	.... friction angle of the intact rock

Hydro mechanical essentials Symbols

ρ	[kg/m³]	.... density of the fluid
a_z	[m/s²]	.... acceleration in the x direction
a_y	[m/s²]	.... acceleration in the y direction
a_z	[m/s²]	.... acceleration in the z direction
p	[Pa]	.... hydrostatic pressure
z	[m]	.... vertical distance to the reference point
p_a	[Pa]	.... reference pressure
σ_φ	[MN/m²]	.... circumferential stress
p_i	[MPa]	.... internal pressure
r	[m]	.... radius of the ring element
t	[m]	.... thickness of the ring element

Finite Element Method Symbols

$x_{(\xi,\eta)}$		.... local coordinates of each node
$N_{n(\xi,\eta)}$		.... shape function
x		.... global coordinates of each node

n		.... number of nodes of each element
$u_{(\xi,\eta)}$		.... local displacement of the node
$N_{n(\xi,\eta)}$		.... shape function
u		.... global displacement of each node
E	[MPa]	.... Young's modulus
υ	[-]	.... Poisson ratio
P	[MN]	.... External forces

Arch Dam Design – Input Manager Symbols

h_{dam}	[m]	.... dam height
$elev_{crest}$	[m]	.... crest elevation
$elev_{base}$	[m]	.... base elevation
T_C	[m]	.... thickness of the dam at the crest
T_B	[m]	.... thickness of the dam at the base
$T_{0.45}$	[m]	.... thickness of the dam at 45 percent of the dam height
L_1	[m]	.... width of the valley at the crest
L_2	[m]	.... width of the valley at 15 percent of the dam height
H	[m]	.... dam height USBR
R_{axis}	[m]	.... radii of the upstream surface at the crest
L_{base}	[m]	.... width of the valley at the base of the arch dam
$\mathrm{R}_{US,crest}$	[m]	.... upstream radius at the crest
$\mathrm{R}_{US,base}$	[m]	.... upstream radius at the base
$\mathrm{R}_{DS,crest}$	[m]	.... downstream radius at the crest
$\mathrm{R}_{DS,base}$	[m]	.... downstream radius at the base
$\mathrm{R}_{DS,LC}$	[m]	.... radius of the downstream line of centres
$\mathrm{R}_{US,LC}$	[m]	.... radius of the upstream line of centres
$\Delta\mathrm{d}_{horizon}$	[m]	.... difference in altitude between each horizon
$n_{horizon}$	[-]	.... number of horizons
$\frac{R'}{R}$	[m]	.... radii ratio of the conic section
$\Delta h_{base,i}$	[m]	.... difference in altitude between the current and the base horizon
$\Delta h_{crest,i}$	[m]	.... difference in altitude between the crest and the current horizon
sd_4	$[m^{-3}]$	.... 5th x coordinate shape coefficient
sd_3	$[m^{-2}]$	.... 4th x coordinate shape coefficient
sd_2	$[m^{-1}]$	.... 3rd x coordinate shape coefficient
sd_1	$[m^{0}]$	.... 2nd x coordinate shape coefficient
sd_0	$[m^{1}]$	.... 1st x coordinate shape coefficient
st_4	$[m^{-3}]$	.... 5th thickness shape coefficient

st_3	[m^{-2}]	.... 4th thickness shape coefficient
st_2	[m^{-1}]	.... 3rd thickness shape coefficient
st_1	[m^{0}]	.... 2nd thickness shape coefficient
st_0	[m^{1}]	.... 1st thickness shape coefficient
cd_4	[m^{-3}]	.... 5th downstream radii shape coefficient
cd_3	[m^{-2}]	.... 4th downstream radii shape coefficient
cd_2	[m^{-1}]	.... 3rd downstream radii shape coefficient
cd_1	[m^{0}]	.... 2nd downstream radii shape coefficient
cd_0	[m^{1}]	.... 1st downstream radii shape coefficient
cu_4	[m^{-3}]	.... 5th upstream radii shape coefficient
cu_3	[m^{-2}]	.... 4th upstream radii shape coefficient
cu_2	[m^{-1}]	.... 3rd upstream radii shape coefficient
cu_1	[m^{0}]	.... 2nd upstream radii shape coefficient
cu_0	[m^{1}]	.... 1st upstream radii shape coefficient
A_G	[m^{2}]	.... base surface of the prism
A_D	[m^{2}]	.... top surface of the prism
A_M	[m^{2}]	.... mean surface of the prism
a	[-]	.... sin φ sinus of the angle phi
b	[-]	.... cos φ cosines of the angle phi
c	[-]	.... sin θ sinus of the angle theta
d	[-]	.... cos θ cosines us of the angle theta
e	[-]	.... sin ψ sinus of the angle psi
f	[-]	.... cos ψ cosines of the angle psi
φ	[°]	.... rotation around the x axis
θ	[°]	.... rotation around the y axis
ψ	[°]	.... rotation around the z axis
p	[-]	.... phase of the abutment
α_{phase}	[°]	.... given phase angle
dz	[m]	.... difference in altitude between horizon1 and horizon2

PYTHON SCRIPT SYMBOLS

$n_{elements}$	[-]	.... number of elements of each wire
$n_{horizons}$	[-]	.... number of horizons chosen within the core data definition

APPENDIX A – PART OF THE VBA CALCULATION CODE

```
Function xyAms(Rtype, b, axis_shift, axis_rot, xds, t, Rds, Rus, dir, coor)
As Double

'calculation of the abutment coordinates of the middle line

On Error GoTo ErrorMSG

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2
yAms = b

      If Rtype = 0 Then
      aa = b ^ 2 / (2 * Rms)
      Else
      aa = -Rms / Rtype + Sgn(Rtype) * ((Rms / Rtype) ^ 2 + b ^ 2 / Rtype)
      ^ (1 / 2)
      End If

xAms = aa
x6 = xAms
y6 = dir * yAms


'coordinate transformation from local to global system

      If coor = 1 Then
      xyAms = xMms - (Rms - x6) * Cos(alpha) + y6 * Sin(alpha)
      ElseIf coor = -1 Then
      xyAms = yMms + (Rms - x6) * Sin(alpha) + y6 * Cos(alpha)
      End If

Exit Function

ErrorMSG:
MsgBox "Radii must be greater than B!" & vbNewLine & vbNewLine &
"Calculation of xAms and yAms not possible. Please redo current Horizon.",
vbOKOnly + vbCritical

      If coor = 1 Then
      xyAms = 0
      ElseIf coor = -1 Then
      xyAms = 0
      End If

End Function
```

```
Function xyDS(Rtype, b, axis_shift, axis_rot, xds, t, Rds, Rus, dir, coor)
As Double

'calculation of the DS coordinates

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2


'calculation of the abutment coordinates for the middle line (xAms,yAms)

yAms = b

      If Rtype = 0 Then
      aa = b ^ 2 / (2 * Rms)
      Else
      aa = -Rms / Rtype + Sgn(Rtype) * ((Rms / Rtype) ^ 2 + b ^ 2 / Rtype)
      ^ (1 / 2)
      End If

xAms = aa


'calculation of the line parameters d and normy

normy = -b / (Rms + Rtype * xAms)
d = b - normy * (xAms + xms - xds)

'calculation of the intersection point of the line and the arc
(xAdsloc,yAdsloc)

      If normy ^ 2 - Rtype = 0# Then
      bb = d ^ 2 / (2# + (normy * d - Rds))
      Else
      bb = -(normy * d - Rds) / (normy ^ 2 - Rtype) + Sgn(Rtype) * (((normy
      * d - Rds) / (
      normy ^ 2 - Rtype)) ^ 2 - d ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If

xAdsloc = bb
yAdsloc = b + normy * (xAdsloc - (xAms + xms - xds))

x1 = xAdsloc + xds - xms
y1 = dir * yAdsloc

'coordinate transformation from local to global system

      If coor = 1 Then
      xyDS = xMms - (Rms - x1) * Cos(alpha) + y1 * Sin(alpha)
      ElseIf coor = -1 Then
      xyDS = yMms + (Rms - x1) * Sin(alpha) + y1 * Cos(alpha)
      End If

End Function
```

```
Function xyPhase(Rtype, b, axis_shift, axis_rot, xds, t, Rds, Rus, phase,
phase_angle, dir, coor) As Double

'calculation of the Phase coordinates

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2
yAms = b

      If Rtype = 0 Then
      aa = b ^ 2 / (2 * Rms)
      Else
      aa = -Rms / Rtype + Sgn(Rtype) * ((Rms / Rtype) ^ 2 + b ^ 2 / Rtype)
      ^ (1 / 2)
      End If

xAms = aa
normy = -b / (Rms + Rtype * xAms)
d_1 = b - normy * (xAms + xms - xds)

      If normy ^ 2 - Rtype = 0 Then
      bb = d_1 ^ 2 / (2 + (normy * d_1 - Rds))
      Else
      bb = -(normy * d_1 - Rds) / (normy ^ 2 - Rtype) + Sgn(Rtype) *
      (((normy * d_1 - Rds)
      / (normy ^ 2 - Rtype)) ^ 2 - d_1 ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If

xAdsloc = bb
yAdsloc = b + normy * (xAdsloc - (xAms + xms - xds))
d_2 = b - normy * (xAms + xms - xus)

      If normy ^ 2 - Rtype = 0 Then
      cc = d_2 ^ 2 / (2 + (normy * d_2 - Rus))
      Else
      cc = -(normy * d_2 - Rus) / (normy ^ 2 - Rtype) + Sgn(Rtype) *
      (((normy * d_2 - Rus)
      / (normy ^ 2 - Rtype)) ^ 2 - d_2 ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If

xApusloc = cc
yApusloc = b + normy * (xApusloc - (xAms + xms - xus))
xAkloc = phase * xApusloc + (1 - phase) * (xAdsloc + xds - xus)
yAkloc = phase * yApusloc + (1 - phase) * yAdsloc

x5 = xAkloc + xus - xms
y5 = dir * yAkloc

'coordinate transformation from local to global system

      If coor = 1 Then
      xyPhase = xMms - (Rms - x5) * Cos(alpha) + y5 * Sin(alpha)
      ElseIf coor = -1 Then
      xyPhase = yMms + (Rms - x5) * Sin(alpha) + y5 * Cos(alpha)
      End If

End Function
Function xyUS(Rtype, b, axis_shift, axis_rot, xds, t, Rds, Rus, phase,
phase_angle, dir, coor) As Double

'calculation of the US coordinates

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
```

```
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2
yAms = b


'calculation of the abutment coordinates for the middle line (xAms,yAms)

      If Rtype = 0 Then
      aa = b ^ 2 / (2 * Rms)
      Else
      aa = -Rms / Rtype + Sgn(Rtype) * ((Rms / Rtype) ^ 2 + b ^ 2 / Rtype)
      ^ (1 / 2)
      End If

xAms = aa


'calculation of the line parameters d and normy for the DS

normy = -b / (Rms + Rtype * xAms)
d_1 = b - normy * (xAms + xms - xds)


'calculation of the intersection point of the line and the arc
DS(xAdsloc,yAdsloc)

      If normy ^ 2 - Rtype = 0 Then
      bb = d_1 ^ 2 / (2 + (normy * d_1 - Rds))
      Else
      bb = -(normy * d_1 - Rds) / (normy ^ 2 - Rtype) + Sgn(Rtype) *
      (((normy * d_1 - Rds)
      / (normy ^ 2 - Rtype)) ^ 2 - d_1 ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If

xAdsloc = bb
yAdsloc = b + normy * (xAdsloc - (xAms + xms - xds))


'calculation of the line parameters d and normy for the Pseudo US

d_2 = b - normy * (xAms + xms - xus)


'calculation of the intersection point of the line and the arc Pseudo
US(xApusloc,yApusloc)

      If normy ^ 2 - Rtype = 0 Then
      cc = d_2 ^ 2 / (2 + (normy * d_2 - Rus))
      Else
      cc = -(normy * d_2 - Rus) / (normy ^ 2 - Rtype) + Sgn(Rtype) *
      (((normy * d_2 - Rus)
      / (normy ^ 2 - Rtype)) ^ 2 - d_2 ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If
xApusloc = cc
yApusloc = b + normy * (xApusloc - (xAms + xms - xus))


'calculation of the phase coordinates (xAkloc,yAkloc)

xAkloc = phase * xApusloc + (1 - phase) * (xAdsloc + xds - xus)
yAkloc = phase * yApusloc + (1 - phase) * yAdsloc


'calculation of the line parameters d and tanphase for the US
```

```
tanPhase = (normy + Tan(phase_angle * pi / 180)) / (1 - normy *
Tan(phase_angle * pi / 180))
d_3 = yAkloc - tanPhase * xAkloc


'calculation of the intersection point of the line and the arc
US(xAusloc,yAusloc)

      If tanPhase ^ 2 - Rtype = 0 Then
      dd = d_3 ^ 2 / (2 + (tanPhase * d_3 - Rus))
      Else
      dd = -(tanPhase * d_3 - Rus) / (tanPhase ^ 2 - Rtype) + Sgn(Rtype) *
      (((tanPhase *
      d_3 - Rus) / (tanPhase ^ 2 - Rtype)) ^ 2 - d_3 ^ 2 / (tanPhase ^ 2 -
      Rtype)) ^ (1 / 2)
      End If

xAusloc = dd
yAusloc = yAkloc + tanPhase * (xAusloc - xAkloc)

x2 = xAusloc + xus - xms
y2 = dir * yAusloc


'coordinate transformation from local to global system

      If coor = 1 Then
      xyUS = xMms - (Rms - x2) * Cos(alpha) + y2 * Sin(alpha)
      ElseIf coor = -1 Then
      xyUS = yMms + (Rms - x2) * Sin(alpha) + y2 * Cos(alpha)
      End If

End Function
```

```
Function PhiUS(Rtype, b, axis_shift, axis_rot, xds, t, Rds, Rus, phase,
phase_angle, dir) As Double

'calculation of the US Phi

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2
yAms = b

      If Rtype = 0 Then
      aa = b ^ 2 / (2 * Rms)
      Else
      aa = -Rms / Rtype + Sgn(Rtype) * ((Rms / Rtype) ^ 2 + b ^ 2 / Rtype)
      ^ (1 / 2)
      End If

xAms = aa
normy = -b / (Rms + Rtype * xAms)
d_1 = b - normy * (xAms + xms - xds)

      If normy ^ 2 - Rtype = 0 Then
      bb = d_1 ^ 2 / (2 + (normy * d_1 - Rds))
      Else
      bb = -(normy * d_1 - Rds) / (normy ^ 2 - Rtype) + Sgn(Rtype) *
      (((normy * d_1 - Rds)
      / (normy ^ 2 - Rtype)) ^ 2 - d_1 ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If

xAdsloc = bb
yAdsloc = b + normy * (xAdsloc - (xAms + xms - xds))
d_2 = b - normy * (xAms + xms - xus)

      If normy ^ 2 - Rtype = 0 Then
      cc = d_2 ^ 2 / (2 + (normy * d_2 - Rus))
      Else
      cc = -(normy * d_2 - Rus) / (normy ^ 2 - Rtype) + Sgn(Rtype) *
      (((normy * d_2 - Rus)
      / (normy ^ 2 - Rtype)) ^ 2 - d_2 ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If

xApusloc = cc
yApusloc = b + normy * (xApusloc - (xAms + xms - xus))
xAkloc = phase * xApusloc + (1 - phase) * (xAdsloc + xds - xus)
yAkloc = phase * yApusloc + (1 - phase) * yAdsloc

tanPhase = (normy + Tan(phase_angle * pi / 180)) / (1 - normy *
Tan(phase_angle * pi / 180))

d_3 = yAkloc - tanPhase * xAkloc

      If tanPhase ^ 2 - Rtype = 0 Then
      dd = d_3 ^ 2 / (2 + (tanPhase * d_3 - Rus))
      Else
      dd = -(tanPhase * d_3 - Rus) / (tanPhase ^ 2 - Rtype) + Sgn(Rtype) *
      (((tanPhase *
      d_3 - Rus) / (tanPhase ^ 2 - Rtype)) ^ 2 - d_3 ^ 2 / (tanPhase ^ 2 -
      Rtype)) ^ (1 / 2)
      End If
xAusloc = dd
yAusloc = yAkloc + tanPhase * (xAusloc - xAkloc)


'calculation of US Phi with Arctan of US coordinates
```

```
PhiUS = dir * Atn(yAusloc / (Rus - xAusloc))
End Function
```

```
Function PhiDS(Rtype, b, axis_shift, axis_rot, xds, t, Rds, Rus, dir) As
Double
'calculation of the DS Phi

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2
yAms = b

      If Rtype = 0 Then
      aa = b ^ 2 / (2 * Rms)
      Else
      aa = -Rms / Rtype + Sgn(Rtype) * ((Rms / Rtype) ^ 2 + b ^ 2 / Rtype)
      ^ (1 / 2)
      End If

xAms = aa
normy = -b / (Rms + Rtype * xAms)
d = b - normy * (xAms + xms - xds)

      If normy ^ 2 - Rtype = 0# Then
      bb = d ^ 2 / (2# + (normy * d - Rds))
      Else
      bb = -(normy * d - Rds) / (normy ^ 2 - Rtype) + Sgn(Rtype) * (((normy
      * d - Rds) / (
      normy ^ 2 - Rtype)) ^ 2 - d ^ 2 / (normy ^ 2 - Rtype)) ^ (1 / 2)
      End If

xAdsloc = bb
yAdsloc = b + normy * (xAdsloc - (xAms + xms - xds))

'calculation of DS Phi with Arctan of DS coordinates

PhiDS = dir * Atn(yAdsloc / (Rds - xAdsloc))

End Function
```

```
Function xyUS_ViaPhi(Phi, Rtype, axis_shift, axis_rot, xds, t, Rds, Rus,
coor) As Double

'calculation of the US coordinates via Phi

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2
aa = (Tan(Phi)) ^ 2 - Rtype
bb = -2 * Rus * (1 + (Tan(Phi)) ^ 2)
cc = (Tan(Phi)) ^ 2 * Rus ^ 2

      If aa = 0 Then
      dd = -bb / cc
      Else
      dd = -bb / (2 * aa) - Sgn(aa) * (bb ^ 2 / (4 * aa ^ 2) - cc / aa) ^
      (1 / 2)
      End If

X = dd
Y = Tan(Phi) * (Rus - X)
x3 = X + xus - xms
y3 = Y

'coordinate transformation from local to global system

      If coor = 1 Then
      xyUS_ViaPhi = xMms - (Rms - x3) * Cos(alpha) + y3 * Sin(alpha)
      ElseIf coor = -1 Then
      xyUS_ViaPhi = yMms + (Rms - x3) * Sin(alpha) + y3 * Cos(alpha)
      End If

End Function
```

```
Function xyDS_ViaPhi(Phi, Rtype, axis_shift, axis_rot, xds, t, Rds, Rus,
coor) As Double

'calculation of the DS coordinates via Phi

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2
aa = (Tan(Phi)) ^ 2 - Rtype
bb = -2 * Rds * (1 + (Tan(Phi)) ^ 2)
cc = (Tan(Phi)) ^ 2 * Rds ^ 2

      If aa = 0 Then
      dd = -bb / cc
      Else
      dd = -bb / (2 * aa) - Sgn(aa) * (bb ^ 2 / (4 * aa ^ 2) - cc / aa) ^
      (1 / 2)
      End If

X = dd
Y = Tan(Phi) * (Rds - X)
x4 = X + xds - xms
y4 = Y


'coordinate transformation from local to global system

      If coor = 1 Then
      xyDS_ViaPhi = xMms - (Rms - x4) * Cos(alpha) + y4 * Sin(alpha)
      ElseIf coor = -1 Then
      xyDS_ViaPhi = yMms + (Rms - x4) * Sin(alpha) + y4 * Cos(alpha)
      End If

End Function
```

```
Function BfromK(Rtype, elv1, elv2, xAms, yAms, axis_shift, axis_rot, xds,
t, Rds, Rus, k,coor) As Double

'calculation of the parameters B from k

On Error GoTo ErrorMSG

pi = 3.14159265358979
alpha = axis_rot / 2 * pi / 180
Rms = (Rds + Rus) / 2
xus = xds - t
xms = (xds + xus) / 2
xMms = (xds + xus) / 2 + Rms
yMms = axis_shift / 2


'calculate intersection point of the circle and the arc

xM = Rms - (xMms - xAms) * Cos(alpha) - (yAms - yMms) * Sin(alpha)
yM = Abs(-(xMms - xAms) * Sin(alpha) + (yAms - yMms) * Cos(alpha))
dz = elv1 - elv2
Rkr = dz * k

e = 2 * Rms
f = Rtype
g = Rkr ^ 2 - xM ^ 2

aa = 1 + f ^ 2 + 2 * f
bb = -4 * xM + 2 * e * f + 2 * e - 4 * xM * f
cc = 4 * xM ^ 2 - 2 * g + e ^ 2 + 2 * yM ^ 2 - 2 * yM ^ 2 * f - 4 * xM * e
- 2 * f * g
dd = 4 * g * xM - 4 * xM * yM ^ 2 - 2 * yM ^ 2 * e - 2 * e * g

EE = yM ^ 4 + g ^ 2 - 2 * yM ^ 2 * g

      If Rtype = -1 Then
      xAmsnew = -dd / (2 * cc) + (dd ^ 2 / (4 * cc ^ 2) - EE / cc) ^ (1 /
      2)
      Else
      bbb = bb / aa
      CCC = cc / aa
      DDD = dd / aa
      EEE = EE / aa

      p = -3 * bbb ^ 2 / 8 + CCC
      q = bbb ^ 3 / 8 - bbb * CCC / 2 + DDD

      rr = -3 * bbb ^ 4 / 256 + bbb ^ 2 * CCC / 16 - bbb * DDD / 4 + EEE

      aaaa = 8#
      bbbb = 20 * p
      cccc = 16 * p ^ 2 - 8 * rr
      dddd = 4 * p ^ 3 - 4 * p * rr - q ^ 2

      pp = cccc / aaaa - bbbb ^ 2 / (3 * aaaa ^ 2)
      qq = dddd / aaaa + 2 * bbbb ^ 3 / (27 * aaaa ^ 3) - bbbb * cccc / (3
      * aaaa ^ 2)

      q24p327 = qq ^ 2 / 4 + pp ^ 3 / 27

            If q24p327 > 0 Then
            z = (-qq / 2 + q24p327 ^ (1 / 2)) ^ (1 / 3) + (-qq / 2 -
            q24p327 ^ (1 / 2)) ^ (1/ 3) - bbbb / (3 * aaaa)
            ElseIf q24p327 = 0 Then
            z = (qq / 2) ^ (1 / 3)
            Else
            z = 2 * (-pp / 3) ^ (1 / 2) * Cos(1 / 3 * ArcCos(-qq / 2 * (-27
            / pp ^ 3) ^ (1 /2))) - bbbb / (3 * aaaa)
            End If
      xAmsnew = Sgn(-Rtype - 1) * (-(p + 2 * z) ^ (1 / 2) / 2 - Sgn(Rtype +
      1) * (((p + z)^ 2 - rr) ^ (1 / 2) - 3 * p / 4 - z / 2) ^ (1 / 2)) -
      (Rms - xM) / (1 + Rtype)
      End If

yAmsnew = (2 * Rms * xAmsnew + Rtype * xAmsnew ^ 2) ^ (1 / 2)

      If coor = 1 Then
      BfromK = xAmsnew
      ElseIf coor = -1 Then
      BfromK = yAmsnew
      End If

Exit Function

ErrorMSG:
MsgBox "Calculation of further Horizons not possible!" & vbNewLine &
vbNewLine & "Please redo previous Horizons.", vbOKOnly + vbCritical


End Function
```

APPENDIX B - PYTHON SCRIPT OF THE ARCH DAM

```
from abaqus import *
from abaqusConstants import *
import sketch
import part

myModel = mdb.Model(name='Arch Dam Design')



#create sketch for each horizon
mySketch1 = myModel.ConstrainedSketch(name='Horizon 1', sheetSize=500.0)
mySketch2 = myModel.ConstrainedSketch(name='Horizon 2', sheetSize=500.0)
mySketch3 = myModel.ConstrainedSketch(name='Horizon 3', sheetSize=500.0)
mySketch4 = myModel.ConstrainedSketch(name='Horizon 4', sheetSize=500.0)
mySketch5 = myModel.ConstrainedSketch(name='Horizon 5', sheetSize=500.0)
mySketch6 = myModel.ConstrainedSketch(name='Horizon 6', sheetSize=500.0)
mySketch7 = myModel.ConstrainedSketch(name='Horizon 7', sheetSize=500.0)
mySketch8 = myModel.ConstrainedSketch(name='Horizon 8', sheetSize=500.0)
mySketch9 = myModel.ConstrainedSketch(name='Horizon 9', sheetSize=500.0)
mySketch10 = myModel.ConstrainedSketch(name='Horizon 10', sheetSize=500.0)
mySketch11 = myModel.ConstrainedSketch(name='Horizon 11', sheetSize=500.0)
mySketch12 = myModel.ConstrainedSketch(name='Horizon 12', sheetSize=500.0)



#set x and y coordinates for each horizon
horizonright1 = ((44.731,-62.695) ,(9.512,-93.624) ,(-10.141,-92.349) ,)
horizonright2 = ((50.546,-86.64) ,(21.199,-118.286) ,(2.673,-118.984) ,)
horizonright3 = ((56.925,-103.313) ,(30.542,-133.886) ,(13.01,-135.176) ,)
horizonright4 = ((64.077,-119.221) ,(39.994,-147.885) ,(23.558,-149.312) ,)
horizonright5 = ((69.957,-134.063) ,(47.856,-160.256) ,(32.644,-161.545) ,)
horizonright6 = ((73.842,-145.859) ,(53.464,-169.097) ,(39.583,-170.007) ,)
horizonright7 = ((77.806,-157.541) ,(59.486,-177.742) ,(47.102,-178.346) ,)
horizonright8 = ((81.502,-168.177) ,(65.565,-185.196) ,(54.863,-185.547) ,)
horizonright9 = ((85.635,-178.466) ,(72.381,-192.154) ,(63.548,-192.297) ,)
horizonright10 = ((90.446,-188.102) ,(80.141,-198.369) ,(73.344,-198.357) ,)
horizonright11 = ((96.219,-196.486) ,(89.146,-203.32) ,(84.529,-203.24) ,)
horizonright12 = ((106.582,-206.994) ,(102.125,-211.349) ,(99.199,-211.315) ,)


horizonleft1 = ((-29.741,22.316) ,(-11.628,30.046) ,(31.866,12.573) ,)
horizonleft2 = ((-19.152,67.723) ,(-0.908,71.33) ,(35.081,47.225) ,)
horizonleft3 = ((-4.603,99.273) ,(12.878,100.598) ,(43.417,74.364) ,)
horizonleft4 = ((10.376,125.387) ,(26.862,125.534) ,(53.55,99.318) ,)
horizonleft5 = ((22.375,143.866) ,(37.633,143.598) ,(61.429,118.953) ,)
horizonleft6 = ((34.011,161.074) ,(47.882,160.608) ,(68.94,138.086) ,)
horizonleft7 = ((46.697,177.554) ,(59.089,176.982) ,(77.47,156.822) ,)
horizonleft8 = ((58.149,191.11) ,(68.843,190.572) ,(84.481,173.278) ,)
horizonleft9 = ((69.67,203.356) ,(78.494,202.939) ,(91.314,188.847) ,)
horizonleft10 = ((82.425,214.827) ,(89.213,214.529) ,(99.033,203.808) ,)
horizonleft11 = ((95.952,224.424) ,(100.565,224.253) ,(107.257,217.047) ,)
horizonleft12 = ((106.953,230.137) ,(109.868,230.042) ,(114.112,225.509) ,)


horizonUS1 = ((31.866,12.573) ,(30.711,10.806) ,(29.626,8.997) ,(28.611,7.148)
,(27.667,5.261) ,(26.796,3.339) ,(26,1.385) ,(25.279,-0.598) ,(24.635,-2.607)
,(24.069,-4.64) ,(23.58,-6.692) ,(23.171,-8.762) ,(22.842,-10.846) ,(22.592,-
12.941) ,(22.424,-15.044) ,(22.336,-17.152) ,(22.329,-19.262) ,(22.402,-21.371)
,(22.557,-23.475) ,(22.792,-25.572) ,(23.108,-27.658) ,(23.503,-29.73) ,(23.978,-
31.786) ,(24.531,-33.822) ,(25.162,-35.836) ,(25.869,-37.824) ,(26.652,-39.783)
,(27.51,-41.71) ,(28.441,-43.604) ,(29.444,-45.46) ,(30.517,-47.276) ,(31.66,-
49.05) ,(32.869,-50.779) ,(34.144,-52.46) ,(35.483,-54.091) ,(36.883,-55.67)
,(38.342,-57.193) ,(39.859,-58.66) ,(41.431,-60.067) ,(43.056,-61.413) ,(44.731,-
62.695) ,)

horizonUS2 = ((35.081,47.225) ,(32.599,44.231) ,(30.235,41.162) ,(27.992,38.021)
,(25.874,34.809) ,(23.883,31.529) ,(22.024,28.181) ,(20.299,24.77) ,(18.714,21.298)
,(17.271,17.769) ,(15.975,14.187) ,(14.831,10.557) ,(13.841,6.884) ,(13.01,3.173)
,(12.341,-0.57) ,(11.837,-4.338) ,(11.501,-8.124) ,(11.334,-11.922) ,(11.339,-
15.722) ,(11.515,-19.519) ,(11.864,-23.304) ,(12.385,-27.069) ,(13.076,-30.806)
,(13.935,-34.509) ,(14.961,-38.169) ,(16.15,-41.779) ,(17.498,-45.333) ,(19.002,-
48.825) ,(20.656,-52.249) ,(22.457,-55.6) ,(24.4,-58.873) ,(26.478,-62.064)
```

```
,(28.688,-65.169) ,(31.025,-68.185) ,(33.483,-71.11) ,(36.058,-73.942) ,(38.746,-
76.678) ,(41.542,-79.317) ,(44.443,-81.857) ,(47.446,-84.299) ,(50.546,-86.64) ,)

horizonUS3 = ((43.417,74.364) ,(39.656,70.573) ,(36.061,66.682) ,(32.634,62.691)
,(29.378,58.599) ,(26.296,54.407) ,(23.392,50.117) ,(20.672,45.729)
,(18.142,41.245) ,(15.807,36.67) ,(13.676,32.006) ,(11.756,27.26) ,(10.054,22.435)
,(8.577,17.541) ,(7.333,12.585) ,(6.329,7.577) ,(5.57,2.526) ,(5.061,-2.556)
,(4.807,-7.656) ,(4.809,-12.763) ,(5.069,-17.863) ,(5.587,-22.944) ,(6.361,-27.991)
,(7.387,-32.994) ,(8.661,-37.94) ,(10.178,-42.817) ,(11.93,-47.616) ,(13.91,-
52.327) ,(16.11,-56.942) ,(18.522,-61.454) ,(21.137,-65.858) ,(23.947,-70.148)
,(26.945,-74.322) ,(30.122,-78.376) ,(33.472,-82.308) ,(36.989,-86.117) ,(40.667,-
89.804) ,(44.503,-93.366) ,(48.492,-96.805) ,(52.634,-100.121) ,(56.925,-103.313)
,)

horizonUS4 = ((53.55,99.318) ,(48.533,94.73) ,(43.739,90.036) ,(39.165,85.23)
,(34.811,80.312) ,(30.677,75.277) ,(26.766,70.125) ,(23.084,64.853)
,(19.637,59.461) ,(16.432,53.95) ,(13.478,48.321) ,(10.787,42.578) ,(8.369,36.726)
,(6.236,30.772) ,(4.399,24.726) ,(2.87,18.598) ,(1.659,12.401) ,(0.775,6.15)
,(0.225,-0.139) ,(0.015,-6.449) ,(0.148,-12.76) ,(0.624,-19.055) ,(1.441,-25.314)
,(2.594,-31.521) ,(4.075,-37.658) ,(5.877,-43.71) ,(7.989,-49.663) ,(10.399,-
55.505) ,(13.094,-61.226) ,(16.063,-66.819) ,(19.294,-72.277) ,(22.773,-77.598)
,(26.493,-82.777) ,(30.442,-87.815) ,(34.613,-92.711) ,(38.999,-97.467) ,(43.597,-
102.084) ,(48.404,-106.565) ,(53.418,-110.913) ,(58.642,-115.131) ,(64.077,-
119.221) ,)

horizonUS5 = ((61.429,118.953) ,(55.461,113.63) ,(49.765,108.203) ,(44.332,102.665)
,(39.158,97.01) ,(34.243,91.233) ,(29.587,85.329),(25.194,79.295) ,(21.071,73.127)
,(17.226,66.823) ,(13.668,60.384) ,(10.411,53.81) ,(7.468,47.108) ,(4.853,40.282)
,(2.581,33.341) ,(0.667,26.299) ,(-0.875,19.169) ,(-2.034,11.968) ,(-2.798,4.715)
,(-3.162,-2.569) ,(-3.119,-9.862) ,(-2.669,-17.141) ,(-1.816,-24.383),(-0.564,-
31.568) ,(1.078,-38.675) ,(3.097,-45.685) ,(5.482,-52.582) ,(8.216,-59.354)
,(11.284,-65.988) ,(14.672,-72.478) ,(18.364,-78.818) ,(22.346,-85.004) ,(26.607,-
91.036) ,(31.136,-96.914) ,(35.924,-102.641) ,(40.966,-108.22) ,(46.259,-113.655)
,(51.801,-118.952) ,(57.595,-124.115) ,(63.644,-129.15) ,(69.957,-134.063) ,)

horizonUS6 = ((68.94,138.086) ,(62.222,132.136) ,(55.806,126.081) ,(49.684,119.913)
,(43.849,113.624) ,(38.299,107.207) ,(33.035,100.655) ,(28.059,93.962)
,(23.378,87.124) ,(19.001,80.136) ,(14.939,72.998) ,(11.206,65.71) ,(7.818,58.275)
,(4.79,50.7) ,(2.14,42.994) ,(-0.115,35.169) ,(-1.959,27.24) ,(-3.378,19.226) ,(-
4.359,11.147) ,(-4.894,3.027) ,(-4.976,-5.11) ,(-4.606,-13.24) ,(-3.784,-21.336)
,(-2.519,-29.375) ,(-0.82,-37.335) ,(1.301,-45.195) ,(3.827,-52.937) ,(6.743,-
60.548) ,(10.03,-68.015) ,(13.671,-75.331) ,(17.65,-82.491) ,(21.952,-89.492)
,(26.563,-96.333) ,(31.473,-103.018) ,(36.672,-109.549) ,(42.156,-115.932)
,(47.92,-122.173) ,(53.966,-128.278) ,(60.297,-134.256) ,(66.919,-140.114)
,(73.842,-145.859) ,)

horizonUS7 = ((77.47,156.822) ,(70.051,150.267) ,(62.962,143.604) ,(56.19,136.822)
,(49.73,129.911) ,(43.579,122.864) ,(37.736,115.672) ,(32.205,108.328)
,(26.992,100.827) ,(22.107,93.163) ,(17.563,85.335) ,(13.372,77.342)
,(9.554,69.188) ,(6.124,60.879) ,(3.102,52.423) ,(0.507,43.832) ,(-1.643,35.124)
,(-3.332,26.316) ,(-4.547,17.431) ,(-5.276,8.494) ,(-5.515,-0.47) ,(-5.259,-9.434)
,(-4.512,-18.37) ,(-3.28,-27.252) ,(-1.573,-36.056) ,(0.595,-44.76) ,(3.208,-
53.345) ,(6.247,-61.795) ,(9.695,-70.097) ,(13.531,-78.242) ,(17.739,-86.224)
,(22.302,-94.041) ,(27.204,-101.692) ,(32.434,-109.18) ,(37.982,-116.509)
,(43.841,-123.685) ,(50.008,-130.714) ,(56.483,-137.606) ,(63.269,-144.368)
,(70.373,-151.01) ,(77.806,-157.541) ,)

horizonUS8 = ((84.481,173.278) ,(76.587,166.211) ,(69.031,159.02) ,(61.804,151.696)
,(54.901,144.229) ,(48.32,136.611) ,(42.063,128.834) ,(36.134,120.892)
,(30.541,112.778) ,(25.295,104.489) ,(20.409,96.023) ,(15.899,87.382)
,(11.782,78.568) ,(8.077,69.587) ,(4.804,60.451) ,(1.98,51.172) ,(-0.374,41.766)
,(-2.243,32.254) ,(-3.613,22.658) ,(-4.474,13.003) ,(-4.818,3.316) ,(-4.643,-6.375)
,(-3.951,-16.043) ,(-2.748,-25.662) ,(-1.046,-35.205) ,(1.144,-44.65) ,(3.805,-
53.977) ,(6.919,-63.17) ,(10.467,-72.213) ,(14.429,-81.098) ,(18.788,-89.817)
,(23.525,-98.366) ,(28.626,-106.745) ,(34.076,-114.956) ,(39.866,-123) ,(45.987,-
130.885) ,(52.434,-138.617) ,(59.205,-146.203) ,(66.303,-153.652) ,(73.732,-
160.974) ,(81.502,-168.177) ,)

horizonUS9 = ((91.314,188.847) ,(83.047,181.26) ,(75.122,173.537) ,(67.534,165.668)
,(60.278,157.645) ,(53.355,149.458) ,(46.768,141.1) ,(40.523,132.566)
,(34.629,123.85) ,(29.097,114.949) ,(23.943,105.862) ,(19.183,96.592)
,(14.833,87.141) ,(10.915,77.518) ,(7.446,67.733) ,(4.447,57.8) ,(1.935,47.736) ,(-
0.074,37.561) ,(-1.565,27.299) ,(-2.528,16.973) ,(-2.956,6.612) ,(-2.847,-3.758)
,(-2.203,-14.108) ,(-1.03,-24.412) ,(0.664,-34.644) ,(2.864,-44.781) ,(5.555,-
54.802) ,(8.72,-64.689) ,(12.339,-74.429) ,(16.394,-84.01) ,(20.866,-93.423)
,(25.736,-102.665) ,(30.989,-111.734) ,(36.611,-120.629) ,(42.589,-129.353)
```

```
,(48.914,-137.912) ,(55.58,-146.31) ,(62.584,-154.556) ,(69.925,-162.658)
,(77.607,-170.625) ,(85.635,-178.466) ,)

horizonUS10 = ((99.033,203.808) ,(90.415,195.738) ,(82.144,187.517)
,(74.216,179.138) ,(66.628,170.592) ,(59.383,161.87) ,(52.485,152.966)
,(45.941,143.874) ,(39.763,134.591) ,(33.962,125.113) ,(28.554,115.441)
,(23.557,105.576) ,(18.987,95.525) ,(14.866,85.294),(11.212,74.896) ,(8.044,64.345)
,(5.381,53.659) ,(3.238,42.857) ,(1.629,31.964) ,(0.564,21.005) ,(0.051,10.006)
,(0.09,-1.005) ,(0.681,-12) ,(1.818,-22.953) ,(3.492,-33.837) ,(5.689,-44.629)
,(8.396,-55.308) ,(11.593,-65.856) ,(15.263,-76.258) ,(19.387,-86.502),(23.946,-
96.58) ,(28.922,-106.484) ,(34.298,-116.214) ,(40.059,-125.768) ,(46.193,-135.149)
,(52.689,-144.36) ,(59.54,-153.407) ,(66.741,-162.296) ,(74.291,-171.036)
,(82.191,-179.635) ,(90.446,-188.102) ,)

horizonUS11 = ((107.257,217.047) ,(98.332,208.526) ,(89.76,199.846)
,(81.538,190.999) ,(73.664,181.975) ,(66.143,172.768) ,(58.979,163.371)
,(52.181,153.778) ,(45.76,143.986) ,(39.73,133.994) ,(34.106,123.801)
,(28.906,113.41) ,(24.148,102.827) ,(19.853,92.061) ,(16.038,81.123)
,(12.724,70.028) ,(9.929,58.794) ,(7.668,47.443) ,(5.955,35.996) ,(4.8,24.48)
,(4.21,12.921) ,(4.187,1.347) ,(4.729,-10.214) ,(5.833,-21.735) ,(7.488,-33.191)
,(9.683,-44.557) ,(12.402,-55.813) ,(15.629,-66.938) ,(19.345,-77.919) ,(23.531,-
88.741) ,(28.169,-99.396) ,(33.239,-109.878) ,(38.726,-120.181) ,(44.612,-130.306)
,(50.886,-140.253) ,(57.536,-150.025) ,(64.553,-159.628) ,(71.931,-169.068)
,(79.667,-178.352) ,(87.762,-187.488) ,(96.219,-196.486) ,)

horizonUS12 = ((114.112,225.509) ,(104.83,216.59) ,(95.917,207.5) ,(87.37,198.231)
,(79.189,188.773) ,(71.377,179.12) ,(63.943,169.265) ,(56.894,159.201)
,(50.244,148.927) ,(44.007,138.441) ,(38.201,127.742) ,(32.844,116.837)
,(27.956,105.73) ,(23.557,94.432) ,(19.669,82.956) ,(16.311,71.318)
,(13.502,59.538) ,(11.258,47.639) ,(9.592,35.646) ,(8.513,23.586) ,(8.029,11.488)
,(8.14,-0.619) ,(8.844,-12.707) ,(10.135,-24.747) ,(12.001,-36.711) ,(14.429,-
48.577) ,(17.402,-60.321) ,(20.903,-71.926) ,(24.912,-83.375) ,(29.408,-
94.657),(34.372,-105.762) ,(39.785,-116.684) ,(45.63,-127.42) ,(51.891,-137.971)
,(58.554,-148.337) ,(65.609,-158.523) ,(73.048,-168.535) ,(80.865,-178.379)
,(89.058,-188.065) ,(97.629,-197.599) ,(106.582,-206.994) ,)

horizonDS1 = ((-10.141,-92.349) ,(-11.685,-89.799) ,(-13.179,-87.219) ,(-14.624,-
84.611) ,(-16.018,-81.976) ,(-17.361,-79.315) ,(-18.653,-76.628) ,(-19.893,-73.917)
,(-21.08,-71.183) ,(-22.215,-68.426) ,(-23.296,-65.648) ,(-24.324,-62.85) ,(-
25.298,-60.032) ,(-26.218,-57.196) ,(-27.082,-54.343) ,(-27.892,-51.474) ,(-
28.647,-48.59) ,(-29.345,-45.692) ,(-29.988,-42.781) ,(-30.575,-39.858) ,(-31.106,-
36.925) ,(-31.58,-33.982) ,(-31.997,-31.03) ,(-32.358,-28.071) ,(-32.661,-25.105)
,(-32.907,-22.134) ,(-33.097,-19.159) ,(-33.229,-16.181) ,(-33.303,-13.2) ,(-
33.321,-10.219) ,(-33.281,-7.239) ,(-33.183,-4.259) ,(-33.029,-1.282) ,(-
32.817,1.692) ,(-32.548,4.661) ,(-32.222,7.624) ,(-31.839,10.58) ,(-31.399,13.529)
,(-30.903,16.468) ,(-30.35,19.398) ,(-29.741,22.316) ,)

horizonDS2 = ((2.673,-118.984) ,(-0.463,-115.094) ,(-3.492,-111.125) ,(-6.411,-
107.079) ,(-9.217,-102.957) ,(-11.908,-98.762) ,(-14.48,-94.496) ,(-16.93,-90.161)
,(-19.256,-85.76) ,(-21.455,-81.294) ,(-23.525,-76.768) ,(-25.461,-72.185) ,(-
27.263,-67.547) ,(-28.927,-62.858) ,(-30.451,-58.123) ,(-31.834,-53.344) ,(-
33.073,-48.526) ,(-34.167,-43.674) ,(-35.115,-38.79) ,(-35.915,-33.88) ,(-36.566,-
28.948) ,(-37.068,-23.999) ,(-37.42,-19.037) ,(-37.623,-14.067) ,(-37.676,-9.092)
,(-37.58,-4.118) ,(-37.335,0.85) ,(-36.943,5.81) ,(-36.404,10.756) ,(-
35.721,15.684) ,(-34.893,20.591) ,(-33.924,25.472) ,(-32.814,30.324) ,(-
31.566,35.144) ,(-30.183,39.928) ,(-28.665,44.674) ,(-27.017,49.377) ,(-
25.238,54.036) ,(-23.333,58.648) ,(-21.304,63.211) ,(-19.152,67.723) ,)

horizonDS3 = ((13.01,-135.176) ,(8.684,-130.385) ,(4.51,-125.482) ,(0.493,-120.468)
,(-3.363,-115.345) ,(-7.053,-110.113) ,(-10.572,-104.776) ,(-13.915,-99.334) ,(-
17.077,-93.791) ,(-20.052,-88.151) ,(-22.834,-82.417) ,(-25.418,-76.594) ,(-
27.798,-70.686) ,(-29.969,-64.7) ,(-31.927,-58.642) ,(-33.666,-52.518) ,(-35.182,-
46.336) ,(-36.473,-40.102) ,(-37.534,-33.826) ,(-38.363,-27.515) ,(-38.958,-21.177)
,(-39.32,-14.822) ,(-39.446,-8.458) ,(-39.338,-2.094) ,(-38.996,4.263) ,(-
38.423,10.603) ,(-37.621,16.918) ,(-36.592,23.201),(-35.34,29.444) ,(-33.87,35.64)
,(-32.186,41.784) ,(-30.291,47.868) ,(-28.192,53.889) ,(-25.892,59.84) ,(-
23.398,65.718) ,(-20.713,71.519) ,(-17.843,77.24) ,(-14.793,82.878) ,(-
11.567,88.43) ,(-8.169,93.896) ,(-4.603,99.273) ,)

horizonDS4 = ((23.558,-149.312) ,(18.163,-143.707) ,(12.964,-137.973) ,(7.966,-
132.108) ,(3.171,-126.113) ,(-1.413,-119.987) ,(-5.781,-113.73) ,(-9.926,-107.342)
,(-13.841,-100.827) ,(-17.517,-94.186) ,(-20.945,-87.424) ,(-24.118,-80.545) ,(-
27.025,-73.554) ,(-29.659,-66.459) ,(-32.01,-59.268) ,(-34.072,-51.99) ,(-35.835,-
44.635) ,(-37.295,-37.214) ,(-38.446,-29.739) ,(-39.284,-22.223) ,(-39.806,-14.678)
,(-40.01,-7.118) ,(-39.897,0.444) ,(-39.469,7.995) ,(-38.727,15.521) ,(-
37.677,23.011) ,(-36.323,30.453) ,(-34.671,37.837) ,(-32.729,45.151) ,(-
30.505,52.388) ,(-28.005,59.539) ,(-25.24,66.598) ,(-22.216,73.56) ,(-
18.943,80.419) ,(-15.428,87.173) ,(-11.678,93.818) ,(-7.7,100.353) ,(-
3.501,106.778) ,(0.914,113.091) ,(5.541,119.294) ,(10.376,125.387) ,)

horizonDS5 = ((32.644,-161.545) ,(26.437,-155.269) ,(20.464,-148.859) ,(14.726,-
142.314) ,(9.227,-135.628) ,(3.972,-128.801) ,(-1.033,-121.83) ,(-5.781,-114.715)
,(-10.263,-107.455) ,(-14.468,-100.052) ,(-18.388,-92.51) ,(-22.009,-84.831) ,(-
25.321,-77.022) ,(-28.312,-69.091) ,(-30.971,-61.046) ,(-33.286,-52.897) ,(-
35.248,-44.658) ,(-36.848,-36.341) ,(-38.077,-27.963) ,(-38.932,-19.537) ,(-
39.407,-11.082) ,(-39.502,-2.614) ,(-39.216,5.849) ,(-38.553,14.292) ,(-
37.516,22.697) ,(-36.113,31.049) ,(-34.351,39.335) ,(-32.239,47.541),(-
29.788,55.655) ,(-27.007,63.669) ,(-23.91,71.573) ,(-20.506,79.361) ,(-
16.806,87.029) ,(-12.822,94.573) ,(-8.562,101.99) ,(-4.036,109.28) ,(0.749,116.444)
,(5.787,123.481) ,(11.073,130.396) ,(16.603,137.19) ,(22.375,143.866) ,)

horizonDS6 = ((39.583,-170.007) ,(32.793,-163.076) ,(26.27,-156.016) ,(20.013,-
148.821) ,(14.024,-141.487) ,(8.308,-134.01) ,(2.87,-126.385) ,(-2.282,-118.61) ,(-
7.14,-110.684) ,(-11.692,-102.608) ,(-15.927,-94.382) ,(-19.832,-86.01) ,(-23.394,-
77.498) ,(-26.597,-68.853) ,(-29.429,-60.085) ,(-31.877,-51.204) ,(-33.928,-42.226)
,(-35.57,-33.166) ,(-36.796,-24.04) ,(-37.598,-14.867) ,(-37.972,-5.667) ,(-
37.915,3.541) ,(-37.429,12.736) ,(-36.518,21.898) ,(-35.186,31.01) ,(-
33.444,40.053) ,(-31.3,49.011) ,(-28.767,57.87) ,(-25.858,66.617) ,(-22.585,75.243)
,(-18.963,83.739) ,(-15.005,92.1) ,(-10.724,100.32) ,(-6.132,108.398) ,(-
1.239,116.333) ,(3.944,124.127) ,(9.41,131.78) ,(15.153,139.297) ,(21.169,146.682)
,(27.455,153.939) ,(34.011,161.074) ,)

horizonDS7 = ((47.102,-178.346) ,(39.755,-170.794) ,(32.707,-163.114) ,(25.956,-
155.298) ,(19.502,-147.339) ,(13.35,-139.232) ,(7.505,-130.972) ,(1.973,-122.555)
,(-3.234,-113.978) ,(-8.107,-105.241) ,(-12.63,-96.345) ,(-16.79,-87.293) ,(-
20.571,-78.09) ,(-23.958,-68.744) ,(-26.935,-59.267) ,(-29.486,-49.67) ,(-31.596,-
39.969) ,(-33.255,-30.181) ,(-34.45,-20.327) ,(-35.175,-10.428) ,(-35.424,-0.505)
,(-35.195,9.419) ,(-34.492,19.32) ,(-33.317,29.176) ,(-31.68,38.967) ,(-
29.59,48.673) ,(-27.06,58.275) ,(-24.105,67.76) ,(-20.739,77.113) ,(-16.978,86.325)
,(-12.839,95.387) ,(-8.337,104.294) ,(-3.485,113.044) ,(1.702,121.634)
,(7.213,130.066) ,(13.039,138.342) ,(19.171,146.466) ,(25.605,154.443)
,(32.338,162.279) ,(39.368,169.98) ,(46.697,177.554) ,)

horizonDS8 = ((54.863,-185.547) ,(47.08,-177.49) ,(39.621,-169.299) ,(32.481,-
160.969) ,(25.663,-152.49) ,(19.168,-143.857) ,(13.002,-135.063) ,(7.173,-126.106)
,(1.691,-116.981) ,(-3.431,-107.688) ,(-8.179,-98.228) ,(-12.538,-88.605) ,(-
16.491,-78.823) ,(-20.022,-68.891) ,(-23.112,-58.822) ,(-25.746,-48.627) ,(-
27.907,-38.325) ,(-29.583,-27.934) ,(-30.763,-17.476) ,(-31.437,-6.973) ,(-
31.601,3.55) ,(-31.253,14.069) ,(-30.396,24.558) ,(-29.034,34.994) ,(-
27.177,45.354) ,(-24.836,55.617) ,(-22.025,65.764) ,(-18.759,75.778),(-
15.056,85.647) ,(-10.933,95.358) ,(-6.407,104.904) ,(-1.494,114.28)
,(3.789,123.482) ,(9.429,132.51) ,(15.413,141.364) ,(21.732,150.048)
,(28.377,158.567) ,(35.342,166.925) ,(42.626,175.13) ,(50.227,183.189)
,(58.149,191.11) ,)

horizonDS9 = ((63.548,-192.297) ,(55.403,-183.78) ,(47.6,-175.126) ,(40.136,-
166.327) ,(33.011,-157.375) ,(26.228,-148.263) ,(19.793,-138.985) ,(13.714,-
129.536) ,(8.002,-119.913) ,(2.67,-110.116) ,(-2.267,-100.145) ,(-6.794,-90.005)
,(-10.892,-79.7) ,(-14.543,-69.24),(-17.73,-58.637) ,(-20.435,-47.906) ,(-22.642,-
37.064) ,(-24.336,-26.132) ,(-25.506,-15.132) ,(-26.143,-4.088) ,(-26.243,6.973)
,(-25.802,18.026) ,(-24.824,29.045) ,(-23.315,40.004) ,(-21.284,50.878) ,(-
18.743,61.647) ,(-15.708,72.289) ,(-12.195,82.788) ,(-8.221,93.13) ,(-
3.806,103.303) ,(1.032,113.299) ,(6.276,123.111) ,(11.909,132.738)
,(17.916,142.179) ,(24.284,151.434) ,(31.004,160.508) ,(38.066,169.404)
,(45.465,178.13) ,(53.198,186.692) ,(61.265,195.098) ,(69.67,203.356) ,)

horizonDS10 = ((73.344,-198.357) ,(64.908,-189.393) ,(56.83,-180.29) ,(49.107,-
171.038) ,(41.739,-161.631) ,(34.729,-152.059) ,(28.082,-142.318) ,(21.809,-
132.402) ,(15.919,-122.309) ,(10.426,-112.037) ,(5.346,-101.588) ,(0.695,-90.964)
,(-3.508,-80.174) ,(-7.246,-69.225) ,(-10.498,-58.131) ,(-13.247,-46.907) ,(-
15.477,-35.57) ,(-17.172,-24.144) ,(-18.32,-12.65) ,(-18.913,-1.114) ,(-
18.944,10.436) ,(-18.413,21.975) ,(-17.322,33.474) ,(-15.677,44.907) ,(-
13.488,56.25) ,(-10.768,67.478) ,(-7.532,78.571) ,(-3.798,89.512) ,(0.416,100.285)
,(5.089,110.879) ,(10.204,121.284) ,(15.741,131.496) ,(21.683,141.51)
,(28.015,151.326) ,(34.723,160.946) ,(41.796,170.373),(49.224,179.612)
,(57.003,188.669) ,(65.129,197.552) ,(73.602,206.268) ,(82.425,214.827) ,)

horizonDS11 = ((84.529,-203.24) ,(75.91,-193.917) ,(67.657,-184.45) ,(59.769,-
174.831) ,(52.245,-165.051) ,(45.09,-155.104) ,(38.31,-144.984) ,(31.913,-134.685)
,(25.913,-124.205) ,(20.323,-113.544) ,(15.158,-102.702) ,(10.436,-91.684)
,(6.176,-80.497) ,(2.396,-69.15) ,(-0.885,-57.657) ,(-3.648,-46.034) ,(-5.876,-
34.3) ,(-7.555,-22.476) ,(-8.673,-10.586) ,(-9.22,1.343) ,(-9.192,13.284) ,(-
8.587,25.211) ,(-7.407,37.094) ,(-5.66,48.908) ,(-3.356,60.626) ,(-0.507,72.225)
,(2.871,83.684) ,(6.759,94.986) ,(11.138,106.114) ,(15.99,117.057)
```

```
,(21.294,127.805) ,(27.031,138.354) ,(33.185,148.698) ,(39.738,158.839)
,(46.677,168.776) ,(53.99,178.514) ,(61.668,188.057),(69.706,197.412)
,(78.099,206.586) ,(86.846,215.587) ,(95.952,224.424) ,)

horizonDS12 = ((99.199,-211.315) ,(90.186,-201.709) ,(81.555,-191.96) ,(73.303,-
182.058) ,(65.429,-171.994) ,(57.937,-161.759) ,(50.831,-151.348) ,(44.12,-140.756)
,(37.816,-129.978) ,(31.933,-119.012) ,(26.485,-107.861) ,(21.492,-96.526)
,(16.97,-85.016) ,(12.942,-73.337) ,(9.425,-61.505) ,(6.439,-49.533) ,(4.003,-
37.441) ,(2.132,-25.25) ,(0.84,-12.986) ,(0.138,-0.674) ,(0.031,11.657)
,(0.522,23.979) ,(1.61,36.263) ,(3.289,48.48) ,(5.549,60.603) ,(8.378,72.608)
,(11.758,84.472) ,(15.673,96.175) ,(20.103,107.7) ,(25.027,119.034)
,(30.424,130.168) ,(36.275,141.093) ,(42.561,151.807) ,(49.265,162.307)
,(56.372,172.594) ,(63.869,182.672) ,(71.745,192.546) ,(79.995,202.222)
,(88.613,211.706) ,(97.598,221.008) ,(106.953,230.137) ,)


#draw sketches
for i in range(len(horizonright1)-
1):mySketch1.Line(point1=horizonright1[i],point2=horizonright1[i+1])
for i in range(len(horizonright2)-
1):mySketch2.Line(point1=horizonright2[i],point2=horizonright2[i+1])
for i in range(len(horizonright3)-
1):mySketch3.Line(point1=horizonright3[i],point2=horizonright3[i+1])
for i in range(len(horizonright4)-
1):mySketch4.Line(point1=horizonright4[i],point2=horizonright4[i+1])
for i in range(len(horizonright5)-
1):mySketch5.Line(point1=horizonright5[i],point2=horizonright5[i+1])
for i in range(len(horizonright6)-
1):mySketch6.Line(point1=horizonright6[i],point2=horizonright6[i+1])
for i in range(len(horizonright7)-
1):mySketch7.Line(point1=horizonright7[i],point2=horizonright7[i+1])
for i in range(len(horizonright8)-
1):mySketch8.Line(point1=horizonright8[i],point2=horizonright8[i+1])
for i in range(len(horizonright9)-
1):mySketch9.Line(point1=horizonright9[i],point2=horizonright9[i+1])
for i in range(len(horizonright10)-
1):mySketch10.Line(point1=horizonright10[i],point2=horizonright10[i+1])
for i in range(len(horizonright11)-
1):mySketch11.Line(point1=horizonright11[i],point2=horizonright11[i+1])
for i in range(len(horizonright12)-
1):mySketch12.Line(point1=horizonright12[i],point2=horizonright12[i+1])

for i in range(len(horizonleft1)-
1):mySketch1.Line(point1=horizonleft1[i],point2=horizonleft1[i+1])
for i in range(len(horizonleft2)-
1):mySketch2.Line(point1=horizonleft2[i],point2=horizonleft2[i+1])
for i in range(len(horizonleft3)-
1):mySketch3.Line(point1=horizonleft3[i],point2=horizonleft3[i+1])
for i in range(len(horizonleft4)-
1):mySketch4.Line(point1=horizonleft4[i],point2=horizonleft4[i+1])
for i in range(len(horizonleft5)-
1):mySketch5.Line(point1=horizonleft5[i],point2=horizonleft5[i+1])
for i in range(len(horizonleft6)-
1):mySketch6.Line(point1=horizonleft6[i],point2=horizonleft6[i+1])
for i in range(len(horizonleft7)-
1):mySketch7.Line(point1=horizonleft7[i],point2=horizonleft7[i+1])
for i in range(len(horizonleft8)-
1):mySketch8.Line(point1=horizonleft8[i],point2=horizonleft8[i+1])
for i in range(len(horizonleft9)-
1):mySketch9.Line(point1=horizonleft9[i],point2=horizonleft9[i+1])
for i in range(len(horizonleft10)-
1):mySketch10.Line(point1=horizonleft10[i],point2=horizonleft10[i+1])
for i in range(len(horizonleft11)-
1):mySketch11.Line(point1=horizonleft11[i],point2=horizonleft11[i+1])
for i in range(len(horizonleft12)-
1):mySketch12.Line(point1=horizonleft12[i],point2=horizonleft12[i+1])


mySketch1.Spline(points=horizonUS1)
mySketch2.Spline(points=horizonUS2)
mySketch3.Spline(points=horizonUS3)
mySketch4.Spline(points=horizonUS4)
mySketch5.Spline(points=horizonUS5)
mySketch6.Spline(points=horizonUS6)
mySketch7.Spline(points=horizonUS7)
mySketch8.Spline(points=horizonUS8)
```

```
mySketch9.Spline(points=horizonUS9)
mySketch10.Spline(points=horizonUS10)
mySketch11.Spline(points=horizonUS11)
mySketch12.Spline(points=horizonUS12)


mySketch1.Spline(points=horizonDS1)
mySketch2.Spline(points=horizonDS2)
mySketch3.Spline(points=horizonDS3)
mySketch4.Spline(points=horizonDS4)
mySketch5.Spline(points=horizonDS5)
mySketch6.Spline(points=horizonDS6)
mySketch7.Spline(points=horizonDS7)
mySketch8.Spline(points=horizonDS8)
mySketch9.Spline(points=horizonDS9)
mySketch10.Spline(points=horizonDS10)
mySketch11.Spline(points=horizonDS11)
mySketch12.Spline(points=horizonDS12)



#create Datum Planes
reference=(0,0,495)
myPart = myModel.Part(name='Part-1 Arch Dam',
dimensionality=THREE_D,type=DEFORMABLE_BODY)
myPart.ReferencePoint(point=(reference))
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=495)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=520)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=540)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=560)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=580)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=600)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=620)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=640)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=660)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=680)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=700)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=715)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=550)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=605)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=660)



#create Wires
myPart = mdb.models['Arch Dam Design'].parts['Part-1 Arch Dam']
myPlane = myPart.datums

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[2],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 1'])
myPart.Wire(sketchPlane=myPlane[2], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[3],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 2'])
myPart.Wire(sketchPlane=myPlane[3], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[4],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 3'])
myPart.Wire(sketchPlane=myPlane[4], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[5],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
```

```
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 4'])
myPart.Wire(sketchPlane=myPlane[5], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[6],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 5'])
myPart.Wire(sketchPlane=myPlane[6], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[7],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 6'])
myPart.Wire(sketchPlane=myPlane[7], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[8],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 7'])
myPart.Wire(sketchPlane=myPlane[8], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[9],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 8'])
myPart.Wire(sketchPlane=myPlane[9], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[10],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 9'])
myPart.Wire(sketchPlane=myPlane[10], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[11],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 10'])
myPart.Wire(sketchPlane=myPlane[11], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[12],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 11'])
myPart.Wire(sketchPlane=myPlane[12], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[13],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Horizon 12'])
```

```
myPart.Wire(sketchPlane=myPlane[13], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)



#lofting arch dam
myEdge = myPart.edges
myPart.SolidLoft(loftsections=(
(myEdge[30], myEdge[31], myEdge[32], myEdge[33], myEdge[34], myEdge[35]),
(myEdge[36], myEdge[37], myEdge[38], myEdge[39], myEdge[40], myEdge[41]),
(myEdge[24], myEdge[25], myEdge[26], myEdge[27], myEdge[28], myEdge[29], ),
(myEdge[42], myEdge[43], myEdge[44], myEdge[45], myEdge[46], myEdge[47]),
(myEdge[18], myEdge[19], myEdge[20], myEdge[21], myEdge[22], myEdge[23], ),
(myEdge[48], myEdge[49], myEdge[50], myEdge[51], myEdge[52], myEdge[53]),
(myEdge[12], myEdge[13], myEdge[14], myEdge[15], myEdge[16], myEdge[17], ),
(myEdge[54], myEdge[55], myEdge[56], myEdge[57], myEdge[58], myEdge[59]),
(myEdge[6], myEdge[7], myEdge[8], myEdge[9], myEdge[10], myEdge[11], ),
(myEdge[60], myEdge[61], myEdge[62], myEdge[63], myEdge[64], myEdge[65]),
(myEdge[0], myEdge[1], myEdge[2], myEdge[3], myEdge[4], myEdge[5], ),
(myEdge[66], myEdge[67], myEdge[68], myEdge[69], myEdge[70], myEdge[71], ),
),startCondition=NONE, endCondition=NONE)


#create partition
myPart = mdb.models['Arch Dam Design'].parts['Part-1 Arch Dam']
myCells = myPart.cells
pickedCells = myCells.findAt(((0.031,11.657,715), ))
myDatum = myPart.datums

myPart.PartitionCellByDatumPlane(datumPlane=myDatum[16], cells=pickedCells)
myPart.PartitionCellByDatumPlane(datumPlane=myDatum[15], cells=pickedCells)
myPart.PartitionCellByDatumPlane(datumPlane=myDatum[14], cells=pickedCells)
pickedCells = myCells.findAt(((-13.789, 11.657, 522.5), ), ((-21.252, 11.657,
577.5), ), ((-19.21, 11.657, 632.5), ), ((-7.131, 11.657, 687.5), ))

myPart.PartitionCellByPlaneThreePoints(point1=(0.031,11.657,715),
point2=(8.03,11.532,715), point3=(0.031,11.657,495), cells=pickedCells)
pickedCells = myCells.findAt(((-13.789, 10.657, 522.5), ), ((-21.252, 10.657,
577.5), ), ((-19.21, 10.657, 632.5), ), ((-7.131, 10.657, 687.5), ))

myPart.PartitionCellByPlaneThreePoints(point1=(25.64,-106.025,715),
point2=(32.832,-102.443,715), point3=(32.832,-102.443,495), cells=pickedCells)
pickedCells = myCells.findAt(((-13.789, 12.657, 522.5), ), ((-21.252, 12.657,
577.5), ), ((-19.21, 12.657, 632.5), ), ((-7.131, 12.657, 687.5), ))

myPart.PartitionCellByPlaneThreePoints(point1=(31.348,131.964,715),
point2=(38.36,128.05,715), point3=(38.36,128.05,495), cells=pickedCells)




#create sketch for each left terrain horizon
mySketch13 = myModel.ConstrainedSketch(name='Terrain_Left 1', sheetSize=500.0)
mySketch14 = myModel.ConstrainedSketch(name='Terrain_Left 2', sheetSize=500.0)
mySketch15 = myModel.ConstrainedSketch(name='Terrain_Left 3', sheetSize=500.0)
mySketch16 = myModel.ConstrainedSketch(name='Terrain_Left 4', sheetSize=500.0)
mySketch17 = myModel.ConstrainedSketch(name='Terrain_Left 5', sheetSize=500.0)
mySketch18 = myModel.ConstrainedSketch(name='Terrain_Left 6', sheetSize=500.0)
mySketch19 = myModel.ConstrainedSketch(name='Terrain_Left 7', sheetSize=500.0)
mySketch20 = myModel.ConstrainedSketch(name='Terrain_Left 8', sheetSize=500.0)
mySketch21 = myModel.ConstrainedSketch(name='Terrain_Left 9', sheetSize=500.0)
mySketch22 = myModel.ConstrainedSketch(name='Terrain_Left 10', sheetSize=500.0)
mySketch23 = myModel.ConstrainedSketch(name='Terrain_Left 11', sheetSize=500.0)
mySketch24 = myModel.ConstrainedSketch(name='Terrain_Left 12', sheetSize=500.0)


#set x and y coordinates for each left terrain horizon
terrain_left1 = ((-29.741,22.316) ,(-11.628,30.046) ,(31.866,12.573) ,(657,12.573)
,(657,669.938) ,(-657,674.566) ,(-657,22.316) ,(-29.741,22.316) ,)
terrain_left2 = ((-19.152,67.723) ,(-0.908,71.33) ,(35.081,47.225) ,(657,47.225)
,(657,669.938) ,(-657,674.566) ,(-657,67.723) ,(-19.152,67.723) ,)
terrain_left3 = ((-4.603,99.273) ,(12.878,100.598) ,(43.417,74.364) ,(657,74.364)
,(657,669.938) ,(-657,674.566) ,(-657,99.273) ,(-4.603,99.273) ,)
terrain_left4 = ((10.376,125.387) ,(26.862,125.534) ,(53.55,99.318) ,(657,99.318)
,(657,669.938) ,(-657,674.566) ,(-657,125.387) ,(10.376,125.387) ,)
```

```
terrain_left5 = ((22.375,143.866) ,(37.633,143.598) ,(61.429,118.953)
,(657,118.953) ,(657,669.938) ,(-657,674.566) ,(-657,143.866) ,(22.375,143.866) ,)
terrain_left6 = ((34.011,161.074) ,(47.882,160.608) ,(68.94,138.086) ,(657,138.086)
,(657,669.938) ,(-657,674.566) ,(-657,161.074) ,(34.011,161.074) ,)
terrain_left7 = ((46.697,177.554) ,(59.089,176.982) ,(77.47,156.822) ,(657,156.822)
,(657,669.938) ,(-657,674.566) ,(-657,177.554) ,(46.697,177.554) ,)
terrain_left8 = ((58.149,191.11) ,(68.843,190.572) ,(84.481,173.278) ,(657,173.278)
,(657,669.938) ,(-657,674.566) ,(-657,191.11) ,(58.149,191.11) ,)
terrain_left9 = ((69.67,203.356) ,(78.494,202.939) ,(91.314,188.847) ,(657,188.847)
,(657,669.938) ,(-657,674.566) ,(-657,203.356) ,(69.67,203.356) ,)
terrain_left10 = ((82.425,214.827) ,(89.213,214.529) ,(99.033,203.808)
,(657,203.808) ,(657,669.938) ,(-657,674.566) ,(-657,214.827),(82.425,214.827) ,)
terrain_left11 = ((95.952,224.424) ,(100.565,224.253) ,(107.257,217.047)
,(657,217.047) ,(657,669.938) ,(-657,674.566) ,(-657,224.424),(95.952,224.424) ,)
terrain_left12 = ((106.953,230.137) ,(109.868,230.042) ,(114.112,225.509)
,(657,225.509) ,(657,669.938) ,(-657,674.566) ,(-657,230.137) ,(106.953,230.137) ,)
#draw left terrain sketches
for i in range(len(terrain_left1)-
1):mySketch13.Line(point1=terrain_left1[i],point2=terrain_left1[i+1])
for i in range(len(terrain_left2)-
1):mySketch14.Line(point1=terrain_left2[i],point2=terrain_left2[i+1])
for i in range(len(terrain_left3)-
1):mySketch15.Line(point1=terrain_left3[i],point2=terrain_left3[i+1])
for i in range(len(terrain_left4)-
1):mySketch16.Line(point1=terrain_left4[i],point2=terrain_left4[i+1])
for i in range(len(terrain_left5)-
1):mySketch17.Line(point1=terrain_left5[i],point2=terrain_left5[i+1])
for i in range(len(terrain_left6)-
1):mySketch18.Line(point1=terrain_left6[i],point2=terrain_left6[i+1])
for i in range(len(terrain_left7)-
1):mySketch19.Line(point1=terrain_left7[i],point2=terrain_left7[i+1])
for i in range(len(terrain_left8)-
1):mySketch20.Line(point1=terrain_left8[i],point2=terrain_left8[i+1])
for i in range(len(terrain_left9)-
1):mySketch21.Line(point1=terrain_left9[i],point2=terrain_left9[i+1])
for i in range(len(terrain_left10)-
1):mySketch22.Line(point1=terrain_left10[i],point2=terrain_left10[i+1])
for i in range(len(terrain_left11)-
1):mySketch23.Line(point1=terrain_left11[i],point2=terrain_left11[i+1])
for i in range(len(terrain_left12)-
1):mySketch24.Line(point1=terrain_left12[i],point2=terrain_left12[i+1])



#create Datum Planes for left terrain
reference=(0,0,495)
myPart = myModel.Part(name='Part-2 Terrain Left',
dimensionality=THREE_D,type=DEFORMABLE_BODY)

myPart.ReferencePoint(point=(reference))
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=495)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=520)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=540)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=560)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=580)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=600)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=620)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=640)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=660)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=680)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=700)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=715)



#create left Terrain Wires
myPart = mdb.models['Arch Dam Design'].parts['Part-2 Terrain Left']
myPlane = myPart.datums
myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[2],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 1'])
myPart.Wire(sketchPlane=myPlane[2], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)
```

```
myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[3],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 2'])
myPart.Wire(sketchPlane=myPlane[3], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)
myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[4],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 3'])
myPart.Wire(sketchPlane=myPlane[4], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[5],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 4'])
myPart.Wire(sketchPlane=myPlane[5], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[6],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 5'])
myPart.Wire(sketchPlane=myPlane[6], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[7],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 6'])
myPart.Wire(sketchPlane=myPlane[7], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[8],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 7'])
myPart.Wire(sketchPlane=myPlane[8], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[9],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 8'])
myPart.Wire(sketchPlane=myPlane[9], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[10],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 9'])
myPart.Wire(sketchPlane=myPlane[10], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[11],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 10'])
```

```
myPart.Wire(sketchPlane=myPlane[11], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[12],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 11'])
myPart.Wire(sketchPlane=myPlane[12], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[13],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Left 12'])
myPart.Wire(sketchPlane=myPlane[13], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)



#lofting terrain
myEdge = myPart.edges
myPart.SolidLoft(loftsections=(
(myEdge[35], myEdge[36], myEdge[37], myEdge[38], myEdge[39], myEdge[40],
myEdge[41]),
(myEdge[42], myEdge[43], myEdge[44], myEdge[45], myEdge[46], myEdge[47],
myEdge[48]),
(myEdge[28], myEdge[29], myEdge[30], myEdge[31], myEdge[32], myEdge[33],
myEdge[34], ),
(myEdge[49], myEdge[50], myEdge[51], myEdge[52], myEdge[53], myEdge[54],
myEdge[55]),
(myEdge[21], myEdge[22], myEdge[23], myEdge[24], myEdge[25], myEdge[26],
myEdge[27], ),
(myEdge[56], myEdge[57], myEdge[58], myEdge[59], myEdge[60], myEdge[61],
myEdge[62]),
(myEdge[14], myEdge[15], myEdge[16], myEdge[17], myEdge[18], myEdge[19],
myEdge[20], ),
(myEdge[63], myEdge[64], myEdge[65], myEdge[66], myEdge[67], myEdge[68],
myEdge[69]),
(myEdge[7], myEdge[8], myEdge[9], myEdge[10], myEdge[11], myEdge[12], myEdge[13],
),
(myEdge[70], myEdge[71], myEdge[72], myEdge[73], myEdge[74], myEdge[75],
myEdge[76]),
(myEdge[0], myEdge[1], myEdge[2], myEdge[3], myEdge[4], myEdge[5], myEdge[6], ),
(myEdge[77], myEdge[78], myEdge[79], myEdge[80], myEdge[81], myEdge[82],
myEdge[83], ), ),startCondition=NONE, endCondition=NONE)




#create sketch for each right terrain horizon

mySketch25 = myModel.ConstrainedSketch(name='Terrain_Right 1', sheetSize=500.0)
mySketch26 = myModel.ConstrainedSketch(name='Terrain_Right 2', sheetSize=500.0)
mySketch27 = myModel.ConstrainedSketch(name='Terrain_Right 3', sheetSize=500.0)
mySketch28 = myModel.ConstrainedSketch(name='Terrain_Right 4', sheetSize=500.0)
mySketch29 = myModel.ConstrainedSketch(name='Terrain_Right 5', sheetSize=500.0)
mySketch30 = myModel.ConstrainedSketch(name='Terrain_Right 6', sheetSize=500.0)
mySketch31 = myModel.ConstrainedSketch(name='Terrain_Right 7', sheetSize=500.0)
mySketch32 = myModel.ConstrainedSketch(name='Terrain_Right 8', sheetSize=500.0)
mySketch33 = myModel.ConstrainedSketch(name='Terrain_Right 9', sheetSize=500.0)
mySketch34 = myModel.ConstrainedSketch(name='Terrain_Right 10', sheetSize=500.0)
mySketch35 = myModel.ConstrainedSketch(name='Terrain_Right 11', sheetSize=500.0)
mySketch36 = myModel.ConstrainedSketch(name='Terrain_Right 12', sheetSize=500.0)


#set x and y coordinates for each right terrain horizon
terrain_right1 = ((44.731,-62.695) ,(9.512,-93.624) ,(-10.141,-92.349) ,(-657,-
92.349) ,(-657,-643.794) ,(657,-643.794) ,(657,-62.695),(44.731,-62.695) ,)
terrain_right2 = ((50.546,-86.64) ,(21.199,-118.286) ,(2.673,-118.984) ,(-657,-
118.984) ,(-657,-643.794) ,(657,-643.794) ,(657,-86.64),(50.546,-86.64) ,)
terrain_right3 = ((56.925,-103.313) ,(30.542,-133.886) ,(13.01,-135.176) ,(-657,-
135.176) ,(-657,-643.794) ,(657,-643.794) ,(657,-103.313) ,(56.925,-103.313) ,)
```

```
terrain_right4 = ((64.077,-119.221) ,(39.994,-147.885) ,(23.558,-149.312) ,(-657,-
149.312) ,(-657,-643.794) ,(657,-643.794) ,(657,-119.221) ,(64.077,-119.221) ,)
terrain_right5 = ((69.957,-134.063) ,(47.856,-160.256) ,(32.644,-161.545) ,(-657,-
161.545) ,(-657,-643.794) ,(657,-643.794) ,(657,-134.063) ,(69.957,-134.063) ,)
terrain_right6 = ((73.842,-145.859) ,(53.464,-169.097) ,(39.583,-170.007) ,(-657,-
170.007) ,(-657,-643.794) ,(657,-643.794) ,(657,-145.859) ,(73.842,-145.859) ,)
terrain_right7 = ((77.806,-157.541) ,(59.486,-177.742) ,(47.102,-178.346) ,(-657,-
178.346) ,(-657,-643.794) ,(657,-643.794) ,(657,-157.541) ,(77.806,-157.541) ,)
terrain_right8 = ((81.502,-168.177) ,(65.565,-185.196) ,(54.863,-185.547) ,(-657,-
185.547) ,(-657,-643.794) ,(657,-643.794) ,(657,-168.177) ,(81.502,-168.177) ,)
terrain_right9 = ((85.635,-178.466) ,(72.381,-192.154) ,(63.548,-192.297) ,(-657,-
192.297) ,(-657,-643.794) ,(657,-643.794) ,(657,-178.466) ,(85.635,-178.466) ,)
terrain_right10 = ((90.446,-188.102) ,(80.141,-198.369) ,(73.344,-198.357) ,(-657,-
198.357) ,(-657,-643.794) ,(657,-643.794) ,(657,-188.102) ,(90.446,-188.102) ,)
terrain_right11 = ((96.219,-196.486) ,(89.146,-203.32) ,(84.529,-203.24) ,(-657,-
203.24) ,(-657,-643.794) ,(657,-643.794) ,(657,-196.486) ,(96.219,-196.486) ,)
terrain_right12 = ((106.582,-206.994) ,(102.125,-211.349) ,(99.199,-211.315) ,(-
657,-211.315) ,(-657,-643.794) ,(657,-643.794) ,(657,-206.994) ,(106.582,-206.994)
,)


#draw right terrain sketches
for i in range(len(terrain_right1)-
1):mySketch25.Line(point1=terrain_right1[i],point2=terrain_right1[i+1])
for i in range(len(terrain_right2)-
1):mySketch26.Line(point1=terrain_right2[i],point2=terrain_right2[i+1])
for i in range(len(terrain_right3)-
1):mySketch27.Line(point1=terrain_right3[i],point2=terrain_right3[i+1])
for i in range(len(terrain_right4)-
1):mySketch28.Line(point1=terrain_right4[i],point2=terrain_right4[i+1])
for i in range(len(terrain_right5)-
1):mySketch29.Line(point1=terrain_right5[i],point2=terrain_right5[i+1])
for i in range(len(terrain_right6)-
1):mySketch30.Line(point1=terrain_right6[i],point2=terrain_right6[i+1])
for i in range(len(terrain_right7)-
1):mySketch31.Line(point1=terrain_right7[i],point2=terrain_right7[i+1])
for i in range(len(terrain_right8)-
1):mySketch32.Line(point1=terrain_right8[i],point2=terrain_right8[i+1])
for i in range(len(terrain_right9)-
1):mySketch33.Line(point1=terrain_right9[i],point2=terrain_right9[i+1])
for i in range(len(terrain_right10)-
1):mySketch34.Line(point1=terrain_right10[i],point2=terrain_right10[i+1])
for i in range(len(terrain_right11)-
1):mySketch35.Line(point1=terrain_right11[i],point2=terrain_right11[i+1])
for i in range(len(terrain_right12)-
1):mySketch36.Line(point1=terrain_right12[i],point2=terrain_right12[i+1])


#create Datum Planes for right terrain
reference=(0,0,495)
myPart = myModel.Part(name='Part-3 Terrain Right',
dimensionality=THREE_D,type=DEFORMABLE_BODY)

myPart.ReferencePoint(point=(reference))
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=495)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=520)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=540)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=560)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=580)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=600)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=620)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=640)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=660)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=680)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=700)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=715)


#create right Terrain Wires
myPart = mdb.models['Arch Dam Design'].parts['Part-3 Terrain Right']
myPlane = myPart.datums

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[2],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
```

```
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 1'])
myPart.Wire(sketchPlane=myPlane[2], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[3],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 2'])
myPart.Wire(sketchPlane=myPlane[3], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[4],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 3'])
myPart.Wire(sketchPlane=myPlane[4], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[5],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 4'])
myPart.Wire(sketchPlane=myPlane[5], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[6],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 5'])
myPart.Wire(sketchPlane=myPlane[6], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[7],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 6'])
myPart.Wire(sketchPlane=myPlane[7], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[8],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 7'])
myPart.Wire(sketchPlane=myPlane[8], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[9],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 8'])
myPart.Wire(sketchPlane=myPlane[9], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[10],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 9'])
```

```
myPart.Wire(sketchPlane=myPlane[10], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[11],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 10'])
myPart.Wire(sketchPlane=myPlane[11], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[12],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 11'])
myPart.Wire(sketchPlane=myPlane[12], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[13],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Terrain_Right 12'])
myPart.Wire(sketchPlane=myPlane[13], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)



#lofting terrain
myEdge = myPart.edges
myPart.SolidLoft(loftsections=(
(myEdge[35], myEdge[36], myEdge[37], myEdge[38], myEdge[39], myEdge[40],
myEdge[41]),
(myEdge[42], myEdge[43], myEdge[44], myEdge[45], myEdge[46], myEdge[47],
myEdge[48]),
(myEdge[28], myEdge[29], myEdge[30], myEdge[31], myEdge[32], myEdge[33],
myEdge[34], ),
(myEdge[49], myEdge[50], myEdge[51], myEdge[52], myEdge[53], myEdge[54],
myEdge[55]),
(myEdge[21], myEdge[22], myEdge[23], myEdge[24], myEdge[25], myEdge[26],
myEdge[27], ),
(myEdge[56], myEdge[57], myEdge[58], myEdge[59], myEdge[60], myEdge[61],
myEdge[62]),
(myEdge[14], myEdge[15], myEdge[16], myEdge[17], myEdge[18], myEdge[19],
myEdge[20], ),
(myEdge[63], myEdge[64], myEdge[65], myEdge[66], myEdge[67], myEdge[68],
myEdge[69]),
(myEdge[7], myEdge[8], myEdge[9], myEdge[10], myEdge[11], myEdge[12], myEdge[13],
),
(myEdge[70], myEdge[71], myEdge[72], myEdge[73], myEdge[74], myEdge[75],
myEdge[76]),
(myEdge[0], myEdge[1], myEdge[2], myEdge[3], myEdge[4], myEdge[5], myEdge[6], ),
(myEdge[77], myEdge[78], myEdge[79], myEdge[80], myEdge[81], myEdge[82],
myEdge[83], ), ),startCondition=NONE, endCondition=NONE)




#create sketch for bottom terrain horizon
mySketch37 = myModel.ConstrainedSketch(name='Bottom 1', sheetSize=500.0)
mySketch38 = myModel.ConstrainedSketch(name='Bottom 2', sheetSize=500.0)



#set x and y coordinates for bottom terrain horizon
bottom1 = ((657,669.938) ,(-657,674.566) ,(-657,-643.794) ,(657,-643.794)
,(657,669.938) ,)
bottom2 = ((657,669.938) ,(-657,674.566) ,(-657,-643.794) ,(657,-643.794)
,(657,669.938) ,)
```

```
#draw bottom terrain sketches
for i in range(len(bottom1)-
1):mySketch37.Line(point1=bottom1[i],point2=bottom1[i+1])
for i in range(len(bottom2)-
1):mySketch38.Line(point1=bottom2[i],point2=bottom2[i+1])



#create Datum Planes for bottom terrain
reference=(0,0,495)
myPart = myModel.Part(name='Part-4 Terrain Bottom',
dimensionality=THREE_D,type=DEFORMABLE_BODY)

myPart.ReferencePoint(point=(reference))
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=495)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=55)



#create bottom Terrain Wires
myPart = mdb.models['Arch Dam Design'].parts['Part-4 Terrain Bottom']
myPlane = myPart.datums

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[2],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Bottom 1'])
myPart.Wire(sketchPlane=myPlane[2], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[3],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Bottom 2'])
myPart.Wire(sketchPlane=myPlane[3], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)


#lofting terrain
myEdge = myPart.edges
myPart.SolidLoft(loftsections=((myEdge[4], myEdge[5], myEdge[6], myEdge[7]),
(myEdge[0], myEdge[1], myEdge[2], myEdge[3])),
startCondition=NONE, endCondition=NONE, keepInternalBoundaries=ON)




#create sketch for US bottom terrain horizon
mySketch39 = myModel.ConstrainedSketch(name='Bottom US 1', sheetSize=500.0)
mySketch40 = myModel.ConstrainedSketch(name='Bottom US 2', sheetSize=500.0)



#set x and y coordinates for US bottom terrain horizon
bottom_US_Line1 = ((-29.741,22.316) ,(-164.25,22.316) ,(-164.25,-92.349) ,(-
10.141,-92.349) ,)

bottom_US_Line2 = ((-29.741,22.316) ,(-164.25,22.316) ,(-164.25,-92.349) ,(-
10.141,-92.349) ,)

bottom_US_Spline1 = ((-10.141,-92.349) ,(-11.685,-89.799) ,(-13.179,-87.219) ,(-
14.624,-84.611) ,(-16.018,-81.976) ,(-17.361,-79.315),(-18.653,-76.628) ,(-19.893,-
73.917) ,(-21.08,-71.183) ,(-22.215,-68.426) ,(-23.296,-65.648) ,(-24.324,-62.85)
,(-25.298,-60.032) ,(-26.218,-57.196) ,(-27.082,-54.343) ,(-27.892,-51.474) ,(-
28.647,-48.59) ,(-29.345,-45.692) ,(-29.988,-42.781) ,(-30.575,-39.858) ,(-31.106,-
36.925) ,(-31.58,-33.982) ,(-31.997,-31.03) ,(-32.358,-28.071) ,(-32.661,-25.105)
,(-32.907,-22.134) ,(-33.097,-19.159) ,(-33.229,-16.181) ,(-33.303,-13.2) ,(-
33.321,-10.219) ,(-33.281,-7.239) ,(-33.183,-4.259) ,(-33.029,-1.282) ,(-
32.817,1.692) ,(-32.548,4.661) ,(-32.222,7.624) ,(-31.839,10.58) ,(-31.399,13.529)
,(-30.903,16.468) ,(-30.35,19.398) ,(-29.741,22.316) ,)
```

```
bottom_US_Spline2 = ((-10.141,-92.349) ,(-11.685,-89.799) ,(-13.179,-87.219) ,(-
14.624,-84.611) ,(-16.018,-81.976) ,(-17.361,-79.315),(-18.653,-76.628) ,(-19.893,-
73.917) ,(-21.08,-71.183) ,(-22.215,-68.426) ,(-23.296,-65.648) ,(-24.324,-62.85)
,(-25.298,-60.032) ,(-26.218,-57.196) ,(-27.082,-54.343) ,(-27.892,-51.474) ,(-
28.647,-48.59) ,(-29.345,-45.692) ,(-29.988,-42.781) ,(-30.575,-39.858) ,(-31.106,-
36.925) ,(-31.58,-33.982) ,(-31.997,-31.03) ,(-32.358,-28.071) ,(-32.661,-25.105)
,(-32.907,-22.134) ,(-33.097,-19.159) ,(-33.229,-16.181) ,(-33.303,-13.2) ,(-
33.321,-10.219) ,(-33.281,-7.239) ,(-33.183,-4.259) ,(-33.029,-1.282) ,(-
32.817,1.692) ,(-32.548,4.661) ,(-32.222,7.624) ,(-31.839,10.58) ,(-31.399,13.529)
,(-30.903,16.468) ,(-30.35,19.398) ,(-29.741,22.316) ,)


#draw US bottom terrain sketches
for i in range(len(bottom_US_Line1)-
1):mySketch39.Line(point1=bottom_US_Line1[i],point2=bottom_US_Line1[i+1])
for i in range(len(bottom_US_Line2)-
1):mySketch40.Line(point1=bottom_US_Line2[i],point2=bottom_US_Line2[i+1])

mySketch39.Spline(points=bottom_US_Spline1)
mySketch40.Spline(points=bottom_US_Spline2)


#create Datum Planes for US bottom terrain
reference=(0,0,495)
myPart = myModel.Part(name='Part-5 Terrain Bottom US',
dimensionality=THREE_D,type=DEFORMABLE_BODY)

myPart.ReferencePoint(point=(reference))
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=495)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=55)


#create US bottom Terrain Wires
myPart = mdb.models['Arch Dam Design'].parts['Part-5 Terrain Bottom US']
myPlane = myPart.datums

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[2],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Bottom US 1'])
myPart.Wire(sketchPlane=myPlane[2], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[3],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Bottom US 2'])
myPart.Wire(sketchPlane=myPlane[3], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)


#lofting terrain US
myEdge = myPart.edges
myPart.SolidLoft(loftsections=((myEdge[4], myEdge[5], myEdge[6], myEdge[7],
),(myEdge[0], myEdge[1], myEdge[2], myEdge[3], ),),
startCondition=NONE, endCondition=NONE, keepInternalBoundaries=ON)


#create sketch for DS bottom terrain horizon
mySketch41 = myModel.ConstrainedSketch(name='Bottom DS 1', sheetSize=500.0)
mySketch42 = myModel.ConstrainedSketch(name='Bottom DS 2', sheetSize=500.0)


#set x and y coordinates for DS bottom terrain horizon
```

```
bottom_DS_Line1 = ((44.731,-62.695) ,(164.25,-62.695) ,(164.25,12.573)
,(31.866,12.573) ,)

bottom_DS_Line2 = ((44.731,-62.695) ,(164.25,-62.695) ,(164.25,12.573)
,(31.866,12.573) ,)

bottom_DS_Spline1 = ((31.866,12.573) ,(30.711,10.806) ,(29.626,8.997)
,(28.611,7.148) ,(27.667,5.261) ,(26.796,3.339) ,(26,1.385) ,(25.279,-0.598)
,(24.635,-2.607) ,(24.069,-4.64) ,(23.58,-6.692) ,(23.171,-8.762) ,(22.842,-10.846)
,(22.592,-12.941) ,(22.424,-15.044),(22.336,-17.152) ,(22.329,-19.262) ,(22.402,-
21.371) ,(22.557,-23.475) ,(22.792,-25.572) ,(23.108,-27.658) ,(23.503,-29.73)
,(23.978,-31.786) ,(24.531,-33.822) ,(25.162,-35.836) ,(25.869,-37.824) ,(26.652,-
39.783) ,(27.51,-41.71) ,(28.441,-43.604) ,(29.444,-45.46),(30.517,-47.276)
,(31.66,-49.05) ,(32.869,-50.779) ,(34.144,-52.46) ,(35.483,-54.091) ,(36.883,-
55.67) ,(38.342,-57.193) ,(39.859,-58.66) ,(41.431,-60.067) ,(43.056,-61.413)
,(44.731,-62.695) ,)

bottom_DS_Spline2 = ((31.866,12.573) ,(30.711,10.806) ,(29.626,8.997)
,(28.611,7.148) ,(27.667,5.261) ,(26.796,3.339) ,(26,1.385) ,(25.279,-0.598)
,(24.635,-2.607) ,(24.069,-4.64) ,(23.58,-6.692) ,(23.171,-8.762) ,(22.842,-10.846)
,(22.592,-12.941) ,(22.424,-15.044),(22.336,-17.152) ,(22.329,-19.262) ,(22.402,-
21.371) ,(22.557,-23.475) ,(22.792,-25.572) ,(23.108,-27.658) ,(23.503,-29.73)
,(23.978,-31.786) ,(24.531,-33.822) ,(25.162,-35.836) ,(25.869,-37.824) ,(26.652,-
39.783) ,(27.51,-41.71) ,(28.441,-43.604) ,(29.444,-45.46),(30.517,-47.276)
,(31.66,-49.05) ,(32.869,-50.779) ,(34.144,-52.46) ,(35.483,-54.091) ,(36.883,-
55.67) ,(38.342,-57.193) ,(39.859,-58.66) ,(41.431,-60.067) ,(43.056,-61.413)
,(44.731,-62.695) ,)


#draw DS bottom terrain sketches
for i in range(len(bottom_DS_Line1)-
1):mySketch41.Line(point1=bottom_DS_Line1[i],point2=bottom_DS_Line1[i+1])
for i in range(len(bottom_DS_Line2)-
1):mySketch42.Line(point1=bottom_DS_Line2[i],point2=bottom_DS_Line2[i+1])

mySketch41.Spline(points=bottom_DS_Spline1)
mySketch42.Spline(points=bottom_DS_Spline2)



#create Datum Planes for DS bottom terrain
reference=(0,0,495)
myPart = myModel.Part(name='Part-6 Terrain Bottom DS',
dimensionality=THREE_D,type=DEFORMABLE_BODY)

myPart.ReferencePoint(point=(reference))
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=495)
myPart.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=55)



#create DS bottom Terrain Wires
myPart = mdb.models['Arch Dam Design'].parts['Part-6 Terrain Bottom DS']
myPlane = myPart.datums

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[2],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Bottom DS 1'])
myPart.Wire(sketchPlane=myPlane[2], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)

myTransformation = myPart.MakeSketchTransform(sketchPlane=myPlane[3],
sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(reference))
myCurrentSketch = myModel.ConstrainedSketch(name='__profile__',sheetSize=500.0,
gridSpacing=20, transform=myTransformation)
myCurrentSketch.retrieveSketch(sketch=mdb.models['Arch Dam
Design'].sketches['Bottom DS 2'])
myPart.Wire(sketchPlane=myPlane[3], sketchPlaneSide=SIDE1,
sketchOrientation=RIGHT,sketch=myCurrentSketch)



#lofting terrain DS
myEdge = myPart.edges
```

```
myPart.SolidLoft(loftsections=((myEdge[4], myEdge[5], myEdge[6], myEdge[7],
),(myEdge[0], myEdge[1], myEdge[2], myEdge[3], ),),
startCondition=NONE, endCondition=NONE, keepInternalBoundaries=ON)




#assembly
MyAssembly = mdb.models['Arch Dam Design'].rootAssembly

MyPart = mdb.models['Arch Dam Design'].parts['Part-1 Arch Dam']
MyAssembly.Instance(name='Part-1 Arch Dam', part=MyPart, dependent=ON)

MyPart = mdb.models['Arch Dam Design'].parts['Part-2 Terrain Left']
MyAssembly.Instance(name='Part-2 Terrain Left', part=MyPart, dependent=ON)

MyPart = mdb.models['Arch Dam Design'].parts['Part-3 Terrain Right']
MyAssembly.Instance(name='Part-3 Terrain Right', part=MyPart, dependent=ON)

MyPart = mdb.models['Arch Dam Design'].parts['Part-4 Terrain Bottom']
MyAssembly.Instance(name='Part-4 Terrain Bottom', part=MyPart, dependent=ON)
MyPart = mdb.models['Arch Dam Design'].parts['Part-5 Terrain Bottom US']
MyAssembly.Instance(name='Part-5 Terrain Bottom US', part=MyPart, dependent=ON)

MyPart = mdb.models['Arch Dam Design'].parts['Part-6 Terrain Bottom DS']
MyAssembly.Instance(name='Part-6 Terrain Bottom DS', part=MyPart, dependent=ON)

MyAssembly.InstanceFromBooleanCut(name='Part-7 Terrain Bottom New',
instanceToBeCut=mdb.models['Arch Dam Design'].rootAssembly.instances['Part-4
Terrain Bottom'], cuttingInstances=(MyAssembly.instances['Part-5 Terrain Bottom
US'], MyAssembly.instances['Part-6 Terrain Bottom DS'], ),
originalInstances=SUPPRESS)

mdb.models['Arch Dam Design'].rootAssembly.features.changeKey(fromName='Part-7
Terrain Bottom New-1',toName='Part-7 Terrain Bottom New')

MyAssembly.resumeFeatures(('Part-5 Terrain Bottom US', 'Part-6 Terrain Bottom DS',
))

MyAssembly.InstanceFromBooleanMerge(name='Part-8 Terrain',
instances=(MyAssembly.instances['Part-2 Terrain Left'], MyAssembly.instances['Part-
3 Terrain Right'], MyAssembly.instances['Part-7 Terrain Bottom New'], ),
keepIntersections=OFF, originalInstances=SUPPRESS, domain=GEOMETRY)

mdb.models['Arch Dam Design'].rootAssembly.features.changeKey(fromName='Part-8
Terrain-1',toName='Part-8 Terrain')

MyAssembly.InstanceFromBooleanMerge(name='Part-9 Terrain All',
instances=(MyAssembly.instances['Part-5 Terrain Bottom US'],
MyAssembly.instances['Part-6 Terrain Bottom DS'], MyAssembly.instances['Part-8
Terrain'], ), keepIntersections=ON, originalInstances=SUPPRESS,domain=GEOMETRY)

mdb.models['Arch Dam Design'].rootAssembly.features.changeKey(fromName='Part-9
Terrain All-1',toName='Part-9 Terrain All')




myViewport = session.Viewport(name='Viewport for Arch Dam Design')
myViewport.setValues(displayedObject=MyAssembly)
myViewport.maximize()
myViewport.view.setValues(nearPlane=700, farPlane=1300, projection=PARALLEL,
cameraPosition=(500.5, -500.5, 893.75), cameraUpVector=(-0.1, 0.1, 1))
```