

# VBA fatigue analysis program for metallic structural components preliminary design

Raul CORMOS\*<sup>1,2</sup>, Catalin Andrei NEAGOE<sup>2,3</sup>, Miruna CIOLCA<sup>2</sup>,  
Anton HADAR<sup>2,4,5</sup>

\*Corresponding author

<sup>1</sup>INCAS – National Institute for Aerospace Research “Elie Carafoli”,  
Iuliu Maniu Blvd. 220, 061136, Bucharest, Romania,  
cormosr@yahoo.com\*

<sup>2</sup>National University of Science and Technology POLITEHNICA Bucharest,  
Splaiul Independentei 313, 060042, Bucharest, Romania

<sup>3</sup>The Institute of Solid Mechanics of the Romanian Academy,  
Constantin Mille 15, 030167, Bucharest, Romania

<sup>4</sup>Technical Science Academy of Romania,  
Dacia Blvd. 26, 030167, Bucharest, Romania

<sup>5</sup>Academy of Romanian Scientists,  
Ilfov Street 3, 050045, Bucharest, Romania

DOI: 10.13111/2066-8201.2023.15.4.8

Received: 22 October 2023/ Accepted: 16 November 2023/ Published: December 2023

Copyright © 2023. Published by INCAS. This is an “open access” article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

**Abstract:** *The main objective of this paper is to describe the creation and use of a fatigue analysis program written in VBA, designed for the preliminary sizing of metallic structural components used in aerospace applications, subjected to one or multiple fatigue loading cases. The VBA programming language was chosen because of its direct control over the most common spreadsheet computational program, Microsoft Excel. Metal fatigue analysis is an important type of analyses for modern structures. Fatigue failure accounts for around 80-90 percent of common structural failures, and therefore, a quick and reliable analysis is necessary so as to evaluate the structure’s bearing capacity to fatigue load. Due to the nature of the fatigue load and the importance of the structural component, such an analysis can be very time consuming, starting from the finite element model preparation and going through the actual analysis; thus, there is a need for a tool that can evaluate the stress data from the numerical simulation and give reliable information about the behavior of the structural component.*

**Key Words:** *Fatigue analysis, VBA, Microsoft Excel, finite element method, Nastran*

## 1. INTRODUCTION

A fatigue analysis represents the evaluation of the structural behavior of a component under non-stationary loading. This type of analysis is thereby of special interest for primary structural components. One of the most important domains where fatigue analysis is necessary is in the aerospace industry, due to the nature of the loadings and the very strict requirements an aircraft structure must meet. Fatigue loads can be assimilated with equivalent static loads, considering that the static load is repeated by the number of times the part is in service [1-9].

The common fatigue analyses, so far encountered, are basically a specific variation of the classic fatigue calculation theory. Usually, these methods are implemented using finite element modeling, by considering a very small element size, for a faster result convergence. The classic fatigue methods are mainly used and implemented with Microsoft Office Excel. In fatigue analysis, the maximum principal stress, obtained by finite element simulation, is of interest, but depending on the hypotheses considered, the minimum principal stress may be also of interest. Microsoft Office has an implemented programming language which has many capabilities beyond classical Microsoft Office applications.

To this end, a VBA (Visual Basic for Applications) code was written for Microsoft Excel, for a general fatigue computational algorithm. The computation was done using case specific hypotheses for fatigue analysis. Among the advantages of using a VBA code are [10-12]:

- direct interface with any Microsoft Office application, without installing any other software package, or any safety requirements;
- code modification for any specific hypothesis consideration;
- easy to understand programming lines;
- automation of the analysis.

The proposed code allows for a fast and reliable fatigue evaluation of a mechanical structure.

## 2. FATIGUE COMPUTATION

### 2.1 Methodology and main computational hypotheses

Fatigue failure represents the failure of a structural component subjected to repeated variable loading. At the beginning, this recurrent deformation leads to the occurrence of a crack, that means also a stress concentrator. At this point, the local stresses are increasing significantly until rupture occurs. Hence, the fatigue failure represents the amplification of a local defect in the microstructure of a structural component.

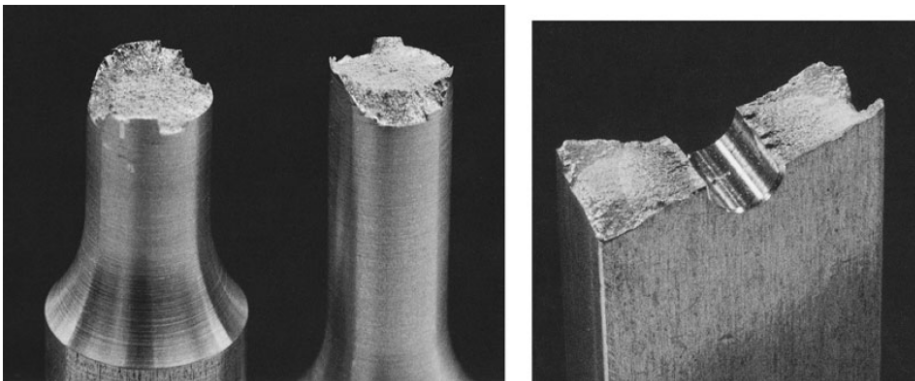


Fig. 1 – Metallic material that has failed after repeated cyclic loading [6]

The fatigue loading is described by the following mechanical components:

$\sigma_{Max}$  – maximum principal stress, [MPa]

$\sigma_{Min}$  – minimum principal stress, [MPa]

$\sigma_a$  – stress amplitude, [MPa]

$\sigma_m$  – mean stress, [MPa]

where:

$$\sigma_a = \frac{\sigma_{Max} - \sigma_{Min}}{2} \quad (1)$$

$$\sigma_m = \frac{\sigma_{Max} + \sigma_{Min}}{2} \quad (2)$$

To evaluate the fatigue behavior of different structural components, the Wohler curve may be used, which represents the correspondence between the stress amplitude and the allowable number of cycles, for each material type. For such curves, their equation can be computed. This equation is obtained by determining two material components, considering two points from the Wohler curve, and solving the curve equation. A typical form of the Wohler curve is presented in Figure 2.

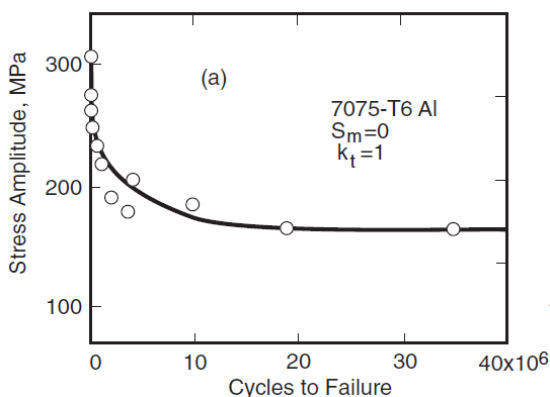


Fig. 2 – General form of the Wohler curve [6]

The general form of the curve equation can be written in the following form:

$$\sigma_a = \sigma_f' \cdot (2 \cdot N_f)^{1/b} \quad (3)$$

where:  $\sigma_f'$  and  $b$  are two material constants specific for every curve.

As a hypothesis for the computation used in this study, a zero-tension stressing has been considered, that means that the minimum principal stress value is zero. Stress amplitude was considered as half of the maximum stress and no mean stress correction was used, as per the Gerber equation. These were the premises for a preliminary fatigue analysis.

## 2.2 Fatigue damage evaluation, Miner's rule

To evaluate the damage obtained by the considered load case, Miner's rule was used. We can consider a mechanical structure subjected to a cyclic loading for a  $C_{use}$  number of cycles accumulated at stress. As a result of the given load, a maximum stress amplitude is obtained from the maximum principal and/or minimum principal load. For this stress amplitude an allowable number of cycles ( $C_{All}$ ) can be obtained from the Wohler curve. The resulting damage can be computed by dividing the number of cycles the part has been used to the allowable corresponding cycles for the known stress amplitude.

$$Damage = \frac{C_{use}}{C_{All}} \quad (4)$$

Miner's rule states that the part can be considered safe in use if the value of the computed damage is less than 1.

### 3. FATIGUE ANALYSIS PROGRAM

#### 3.1 Introduction

The fatigue analysis program was written in VBA, using several general hypotheses presented below. The program was structured in four parts and each part was given a short description. The program implements the classical fatigue sizing theory by considering that the stress amplitude is half of maximum principal stress multiplied with a correction coefficient. Thus, the program takes the following inputs [13-15]:

- Patran generated CSV files with the maximum principal stress values on elements, with no nodal averaging and extrapolation on element centroid;
- the correction coefficient was considered equal to the stress concentrator factor (general or for fatigue) multiplied with the safety factor:

$$C = k_{t/ff} \cdot SF \quad (5)$$

- the coefficients  $\sigma'_f$ ,  $b$  were used from the fatigue analysis.

Moreover, for the assessment, no fatigue curve normalization was considered. The fatigue analysis has been made taking into account the static analysis for the given load repeated by  $C_{use}$  times.

#### 3.2 Program description

The four parts of the VBA program are presented as follows:

##### *Part 1. Data import*

In the first part, the data import was handled. There were two workbooks defined. The first one was named WkbTemp, where the maximum stress values were imported one by one. After each import, the data was copied in the second workbook, for summarization, and the temporary workbook for each data was deleted. In addition, a data preparation had been done using specific text separators.

##### *'Parte 1 Import date*

*Application.ScreenUpdating = False*

*Dim PI\_NumeFisiere As Variant 'Lista nume fisiere – result file name*

*Dim PI\_WkbTemp As Workbook 'Workbook import – temporary workbook for primary import*

*Dim PI\_Wkb\_All As Workbook 'workbook import – workbook for entire data gathering*

*Dim PI\_i As Double – double precision variable for counting file load cases*

*PI\_NumeFisiere = Application.GetOpenFilename(, "Import date oboseala", "Import MaxPrincipal", MultiSelect:=True)*

*If TypeName(PI\_NumeFisiere) = "Boolean" Then*

*Exit Sub*

*End If*

(Selection of files which contain data for static finite element analysis)

*PI\_i = 1*

*Set PI\_WkbTemp = Application.Workbooks.Open(PI\_NumeFisiere(PI\_i))*

*PI\_WkbTemp.Worksheets(1).UsedRange.TextToColumns \_*

*Destination:=Range("A1"), \_*

*Data Type: =xlDelimited, \_*

*textqualifier:=xlDoubleQuote, \_*

```

consecutivedelimiter:=True, _
Tab:=True, semicolon:=True, comma:=True, Space:=True, other:=True, otherchar:=""

P1_WkbTemp.Worksheets(1).Range("A1").EntireColumn.Delete
Set P1_Wkb_All = Application.Workbooks.Add
P1_WkbTemp.Worksheets(1).Copy after:=P1_Wkb_All.Worksheets(P1_Wkb_All.Worksheets.Count)
P1_WkbTemp.Close (False)

Application.DisplayAlerts = False
P1_Wkb_All.Worksheets(1).Delete
Application.DisplayAlerts = True

For P1_i = 2 To UBound(P1_NumeFisiere)
    If UBound(P1_NumeFisiere) < 1 Then
        Exit For
    End If

    Set P1_WkbTemp = Application.Workbooks.Open(P1_NumeFisiere(P1_i))
    P1_WkbTemp.Worksheets(1).UsedRange.TextToColumns _
    Destination:=Range("A1"), _
    DataType:=xlDelimited, _
    textqualifier:=xlDoubleQuote, _
    consecutivedelimiter:=True, _
    Tab:=True, semicolon:=True, comma:=True, Space:=True, other:=True, otherchar:=""

    P1_WkbTemp.Worksheets(1).Range("A1").EntireColumn.Delete
    P1_WkbTemp.Worksheets(1).Copy after:=P1_Wkb_All.Worksheets(P1_Wkb_All.Worksheets.Count)
    P1_WkbTemp.Close (False)

Next P1_i

```

(Separation of data, definition of data separators)

### Part 2. Data preparation

In the second part of the program, a further data preparation was made. The IDs of the elements were combined with the maximum principal value of the stress. For each load case, a sub case ID and the load extraction side of the element were defined.

#### 'Partea a doua Pregatire date

```

Dim P2_CuvantCheie As String – key word specification for main data location
Dim P2_PrimaAparitie As String – first location of key word
Dim P2_PortiuneCuvantCheie As Range – active portion of key word
Dim P2_i As Double – counter for number of worksheets
Dim P2_PortiuneDate As Range – data portion definition for element id and maximum stress
Dim P2_NumarRanduri_PortiuneDate As Double – number of rows for data portion
Dim P2_j As Double – redefinition of number of worksheets

```

```
P2_CuvantCheie = "ID--Max"
```

```
For P2_i = 1 To P1_Wkb_All.Sheets.Count
```

```
P1_Wkb_All.Worksheets(P2_i).Activate
```

```
Set P2_PortiuneCuvantCheie =
```

```
P1_Wkb_All.Worksheets(P2_i).Range("A:A").Find(What:=P2_CuvantCheie, _
    LookIn:=xlValues, LookAt:=xlWhole, MatchCase:=True)
P2_PrimaAparitie = P2_PortiuneCuvantCheie.Address
```

(Defining key words and generating data selection area for each worksheet, corresponding to each load case taken into consideration)

**Do**

```
Set P2_PortiuneDate = P2_PortiuneCuvantCheie.CurrentRegion
P2_NumarRanduri_PortiuneDate = P2_PortiuneDate.Rows.Count
```

```
If P2_PortiuneDate.Cells(P2_NumarRanduri_PortiuneDate, 1).Value = "MSC.Patran" Then
P2_PortiuneDate.Cells(P2_NumarRanduri_PortiuneDate, 1).EntireRow.Insert shift:=xlDown
P2_PortiuneDate.Cells(P2_NumarRanduri_PortiuneDate, 1).EntireRow.Insert shift:=xlDown
End If
```

(Separation of  $Z_1$  and  $Z_2$  data)

```
For P2_j = 1 To P2_NumarRanduri_PortiuneDate
If P2_PortiuneDate.Cells(P2_j, 1).Value = P2_CuvantCheie Then
P2_PortiuneDate.Cells(P2_j, 1).Offset(0, 3).Value = "Id Element"
P2_PortiuneDate.Cells(P2_j, 1).Offset(0, 4).Value = "Tensiune Maxima [MPa]" & " Case " &
P2_PortiuneDate.Cells(P2_j, 1).Offset(-7, 2) & " At " & P2_PortiuneDate.Cells(P2_j, 1).Offset(-5, 6)
P2_PortiuneDate.Cells(P2_j, 1).Offset(0, 5).Value = "Case" & " " &
P2_PortiuneDate.Cells(P2_j, 1).Offset(-5, 6)
Else
P2_PortiuneDate.Cells(P2_j, 1).Offset(0, 3).Value = "Element_" & P2_PortiuneDate.Cells(P2_j,
1).Value
P2_PortiuneDate.Cells(P2_j, 2).Offset(0, 3).Value = P2_PortiuneDate.Cells(P2_j, 2).Value
End If
Next 'P2_j
```

(Creation for each worksheet of Element ID and corresponding maximum stress for each element side and load case scenario)

```
Set P2_PortiuneCuvantCheie =
P1_Wkb_All.Worksheets(P2_i).Range("A:A").FindNext(P2_PortiuneCuvantCheie)
If P2_PrimaAparitie = P2_PortiuneCuvantCheie.Address Then
Exit Do
End If
Loop
```

```
P1_Wkb_All.Worksheets(P2_i).Columns.AutoFit
Next 'P2_i
```

(Closing of loop which executes for each load case)

### Part 3. Summarization of the data

In this part, a summary was created of the Element IDs and maximum principal stresses for all worksheets, corresponding to every load case scenario.

#### 'P3 Sumarizare date

```
Dim P3_ShSumarizare As Worksheet – worksheet for data summary
Dim P3_ListaElm As Range – list of element ID
Dim P3_i As Double – number of worksheets
Dim P3_j As Double – number of rows equal with number of elements on which computation is being made
Dim P3_NumarRanduri As Double – number of rows
Dim P3_CuvantChZ1 As String – key word for Z1
Dim P3_CuvantChZ2 As String – key word for Z2
Dim P3_PortiuneZ1 As Range – selection of portion for Z1 data on each sheet
```

*Dim P3\_PortiuneDateZ1 As Range* - selection of main data for Z1 on each sheet  
*Dim P3\_PortiuneZ2 As Range* – selection of portion for Z2 data on each sheet  
*Dim P3\_PortiuneDateZ2 As Range*- selection of main data for Z2 on each sheet

*P3\_CuvantChZ1 = "Case\_Z1"*  
*P3\_CuvantChZ2 = "Case\_Z2"*

(Key words definitions)

*Set P3\_ShSumarizare =*  
*P1\_Wkb\_All.Worksheets.Add(after:=P1\_Wkb\_All.Worksheets(P1\_Wkb\_All.Worksheets.Count))*  
*P3\_ShSumarizare.Name = "Sumarizare"*

*Set P3\_ListaElm = P1\_Wkb\_All.Worksheets(1).UsedRange.Find(What:="Id Element", \_*  
*LookIn:=xlValues, LookAt:=xlWhole, MatchCase:=True)*

*If P3\_ListaElm.CurrentRegion.Cells(P3\_ListaElm.CurrentRegion.Rows.Count, 1).Value = "Element "*  
*Then*  
*P3\_ListaElm.CurrentRegion.Cells(P3\_ListaElm.CurrentRegion.Rows.Count, 1).Value = ""*  
*End If*

*P3\_ListaElm.CurrentRegion.Resize(P3\_ListaElm.CurrentRegion.Rows.Count, 1).Copy*  
*P3\_ShSumarizare.Range("A4")*  
*P3\_NumarRanduri = P3\_ListaElm.CurrentRegion.Rows.Count*

(Creation of first column for elements)

*For P3\_i = 1 To UBound(P1\_NumeFisiere)*

*Set P3\_PortiuneZ1 = P1\_Wkb\_All.Worksheets(P3\_i).UsedRange.Find(What:=P3\_CuvantChZ1, \_*  
*LookIn:=xlValues, LookAt:=xlWhole)*  
*Set P3\_PortiuneDateZ1 = P3\_PortiuneZ1.Offset(0, -2).CurrentRegion*

*Set P3\_PortiuneZ2 = P1\_Wkb\_All.Worksheets(P3\_i).UsedRange.Find(What:=P3\_CuvantChZ2, \_*  
*LookIn:=xlValues, LookAt:=xlWhole)*  
*Set P3\_PortiuneDateZ2 = P3\_PortiuneZ2.Offset(0, -2).CurrentRegion*

(Creation of data sets for Z1 and Z2 portion maximum stress)

*For P3\_j = 1 To P3\_NumarRanduri*  
*P3\_ShSumarizare.Cells(2 + 1, 1).Offset(P3\_j, 2 \* P3\_i - 1) =*  
*Application.WorksheetFunction.VLookup(P3\_ShSumarizare.Range("A" & 3 + P3\_j),*  
*P3\_PortiuneZ1.CurrentRegion, 2, 0)*  
*P3\_ShSumarizare.Cells(2 + 1, 1).Offset(P3\_j, 2 \* P3\_i) =*  
*Application.WorksheetFunction.VLookup(P3\_ShSumarizare.Range("A" & 3 + P3\_j),*  
*P3\_PortiuneZ2.CurrentRegion, 2, 0)*  
*Next 'P3\_j*

(Allocation of maximum principal stress at elements for each load case)

(Comment: for purposes of simplification, the general Excel VLOOKUP function was used to guarantee that even if element ID mismatch was done, the correct values were given for each element, a slower but equally good choice would have been for each next loop).

*Next '*

*Part 4. Fatigue analysis*

In the last part of the program, the effective fatigue calculation was made, according to the hypotheses presented above.

*'Program de calcul*

*Dim P4\_CoefCorectie As Double* – correction coefficient  
*Dim P4\_Cicluri As Double* – number of cycles of use  
*Dim P4\_SPrim As Double* – sigma prim  
*Dim P4\_b As Double* – b  
*Dim P4\_PortiuneDate As Range* – data portion for summary data  
*Dim P4\_PrimulRand As Double* – first row ID  
*Dim P4\_NumarRanduri As Double* – number of rows in summary data table  
*Dim P4\_IdRand As Double* – row ID  
*Dim P4\_i As Double* – counter for each load case computation  
*Dim P4\_NumarColoane As Double* – number of columns for summary table  
*Dim P4\_j As Double* 'numar de fisiere – counter for computation cases  
*Dim P4\_r As Double*– counter for row numbers

*P4\_PrimulRand = Range("A" & 1).End(xlDown).Row*  
*Set P4\_PortiuneDate = Range("A" & P4\_PrimulRand).CurrentRegion*  
*P4\_NumarRanduri = P4\_PortiuneDate.Rows.Count*  
*P4\_NumarColoane = P4\_PortiuneDate.Columns.Count*

(Data summarization range and its row and column numbers)

*P4\_CoefCorectie = Application.InputBox("Valoare coeficient de corectie", "Coeficient de corectie", , , , , 1)*  
*P3\_ShSumarizare.Range("A1") = "Coeficient de corectie"*  
*P3\_ShSumarizare.Range("A2") = P4\_CoefCorectie*

*P3\_ShSumarizare.Range("B1") = "Numar cicluri in exploatare [Cicluri]"*  
*P4\_Cicluri = Application.InputBox("Numar cicluri in expoatare [Cicluri]", "cicluri in lucru", , , , , 1)*  
*P3\_ShSumarizare.Range("B2") = P4\_Cicluri*

*P3\_ShSumarizare.Range("C1") = "SigmaPrim"*  
*P4\_SPrim = Application.InputBox("SigmaPrim", "SPrim", , , , , 1)*  
*P3\_ShSumarizare.Range("C2") = P4\_SPrim*

*P3\_ShSumarizare.Range("D1") = "b"*  
*P4\_b = Application.InputBox("Exponent", "b", , , , , 1)*  
*P3\_ShSumarizare.Range("D2") = P4\_b*

(The definition of correction coefficient, in used cycles for element,  $\sigma_f'$  and b)

*For P4\_j = 1 To UBound(P1\_NumeFisiere)*  
*P4\_PortiuneDate.Cells(1, P4\_NumarColoane).Offset(0, P4\_j \* 4).Value = "Valoare Amplitudine [MPa] Case SC" & P4\_j*  
*P4\_PortiuneDate.Cells(1, P4\_NumarColoane).Offset(0, P4\_j \* 4 + 1).Value = "Cicluri admisibile [Cicluri] Case SC" & P4\_j*  
*P4\_PortiuneDate.Cells(1, P4\_NumarColoane).Offset(0, P4\_j \* 4 + 2).Value = "Damage Case SC" & P4\_j*  
*P4\_PortiuneDate.Cells(1, P4\_NumarColoane).Offset(0, P4\_j \* 4 + 3).Value = "MaxDamage Case SC" & P4\_j*

(Definition of column names)



```

For P4_r = 1 To P4_NumarRanduri - 1
    P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4).Value =
P3_ShSumarizare.Range("A2").Value * Application.WorksheetFunction.Max(P4_PortiuneDate.Cells(1,
1).Offset(P4_r, 2 * P4_j - 1), P4_PortiuneDate.Cells(1, 1).Offset(P4_r, 2 * P4_j)) / 2

```

(Computation for stress amplitude = Correction coefficient \* 0.5 \* max( $Z_1, Z_2$ ))

```

If P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4).Value = 0 Then
P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4 + 1).Value = "Infinit"
P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4 + 2).Value = 0
Else
P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4 + 1).Value = 0.5 *
((P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4) / P4_SPrim) ^ (1 / P4_b))
P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4 + 2).Value = P4_Cicluri /
P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(P4_r, P4_j * 4 + 1).Value
End If

```

(Calculation of damage)

```

P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(1, P4_j * 4 + 3).Value =
Application.WorksheetFunction.Max(P4_PortiuneDate.Cells(1, P4_NumarColoane).Offset(1, P4_j * 4 +
2).EntireColumn) (maximum damage for each load case)
Next P4_r
Next P4_j

```

```

Application.ScreenUpdating = True
P3_ShSumarizare.Columns.AutoFit
End Sub

```

(End of program)

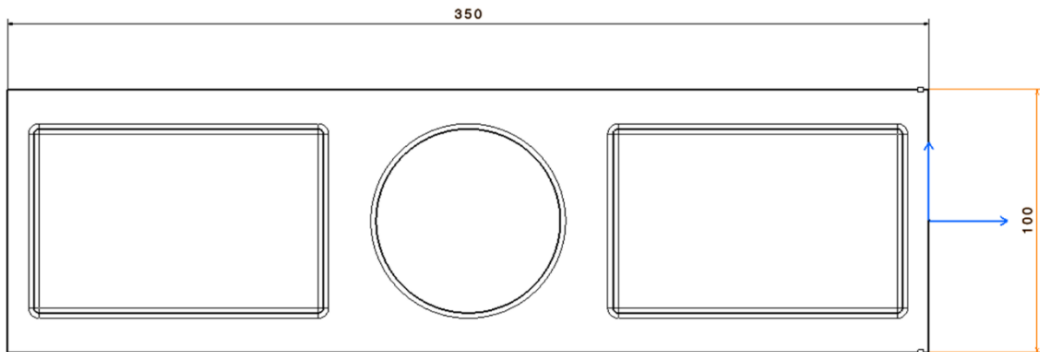
## 4. PROGRAM WORKING DESCRIPTION

This chapter describes how the program is actually working, by considering two loading cases.

### 4.1 Geometry and finite element model description

The geometry of a metallic part was considered as an example for presenting the method. The part does not serve a practical use in a project and it is used herein only as a demonstration model.

The part's dimensions are de 300x100x50 mm, and its geometry is presented in Figure 3, as modeled with the Catia software package.



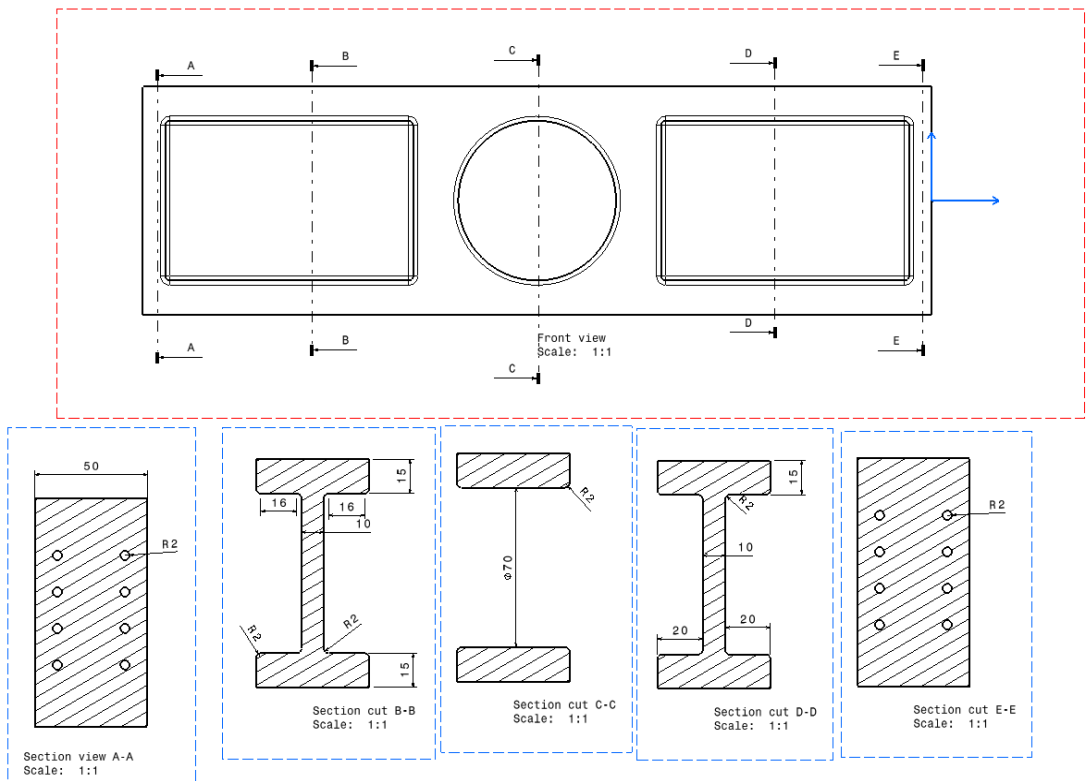


Fig 3 – Geometry of the trial finite element model

To demonstrate the use of the program a finite element model of the part was considered. The analyzed part was considered to be fixed to a wall with M4 metric bolts on both lateral and lower areas. The component was modeled in MSC Nastran with tetrahedral Tet10 elements, while for the outer skin, Tria elements were chosen, type Tria3, with a thickness of 0.001 mm. The resulting finite element model is presented in Figure 4. The part has a steel support, considered very thick, on which single point constraints (SPCs) are defined, to eliminate, as much as possible, the contributions of the reaction loads.

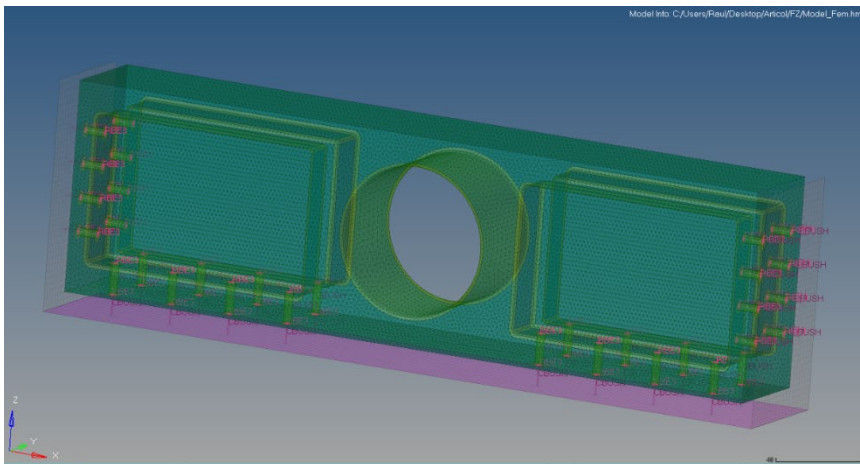


Fig. 4 – Finite element analysis model

### 4.2 Material data

The part’s material was considered standard Aluminum, with:

$$E = 71,000 \text{ MPa}$$

$$\nu = 0.33$$

In order to be able to characterize the fatigue behavior of the employed material, a Wohler curve was considered, as presented in Figure 5.

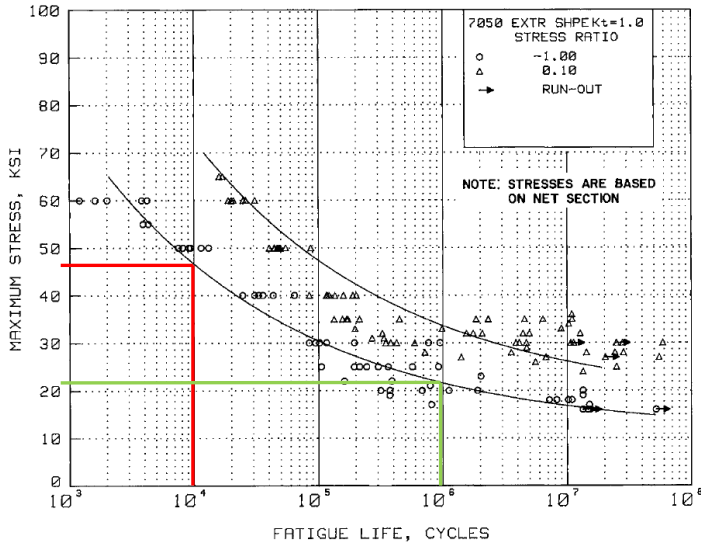


Fig. 5 – Wohler Curve for Al 7050 [16]

Two points were chosen to compute the two terms for the stress amplitude – fatigue life cycles. These points were marked on the figure  $P_1$  (48 KSi/ 330.94 MPa, 10,000 Cycles) and  $P_2$  (20.5 KSi/ 141.34 MPa, 1,000,000 cycles). The calculated values are:  $b = -0.185$  and  $\sigma_f' = 2,072.9 \text{ MPa}$ .

### 4.3 Load case description

In order to study the fatigue behavior of the part, two loading conditions were considered. In the first, a 75,000 N load was applied on the global longitudinal, X axis (Figure 6), and in the second one, the same load was applied on the vertical, Z axis (Figure 7).

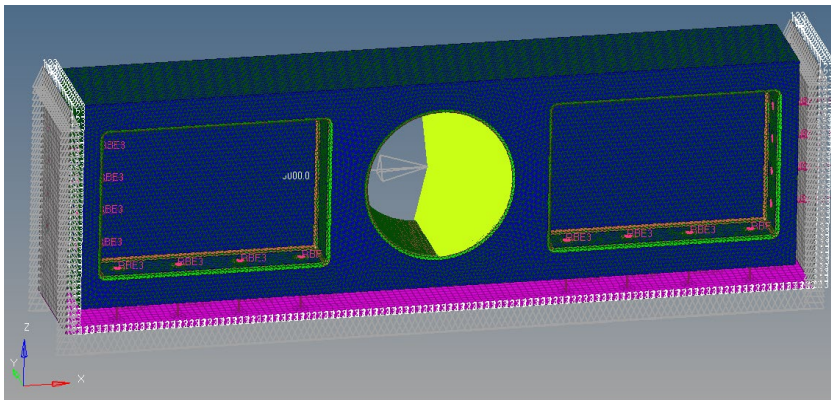


Fig. 6 – Finite element model and load set up for Fx load case scenario

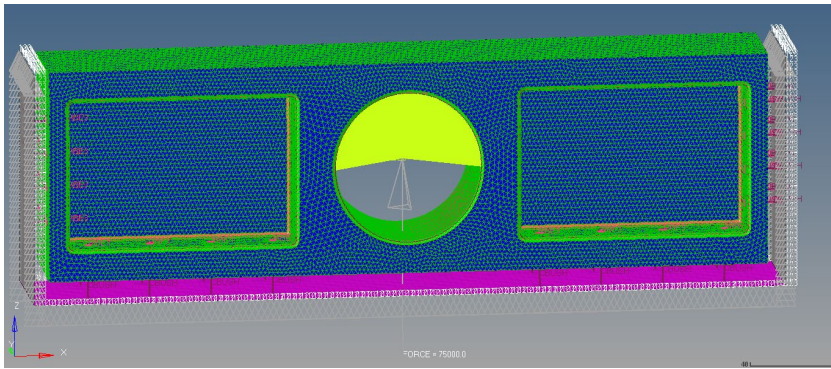


Fig. 7 – Finite element model and load set up for Fz load case scenario

The loads were applied using a rigid beam element, type RBE3, linked only on force normal contact zone for the Fx and Fz load case scenarios. The external area of the support surface was fixed using SPCs on translational degrees.

#### 4.4 Finite element simulation results

The results were extracted using Patran with no averaging and extrapolation on element centroid.

Figure 8 shows the maximum principal stress values for Fx case and Figure 9 for Fz case.

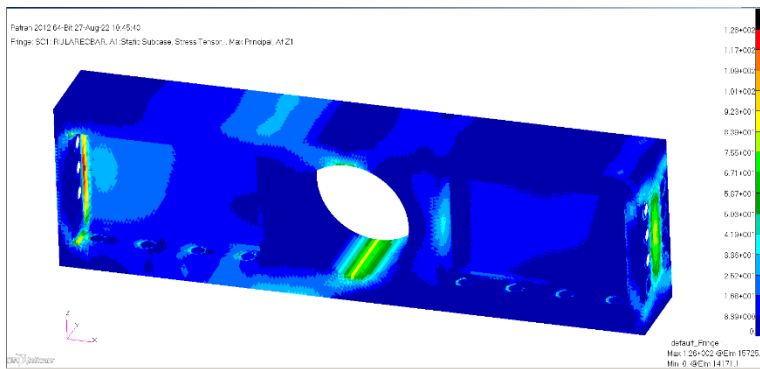


Fig. 8 – Maximum principal stress values for Fx case

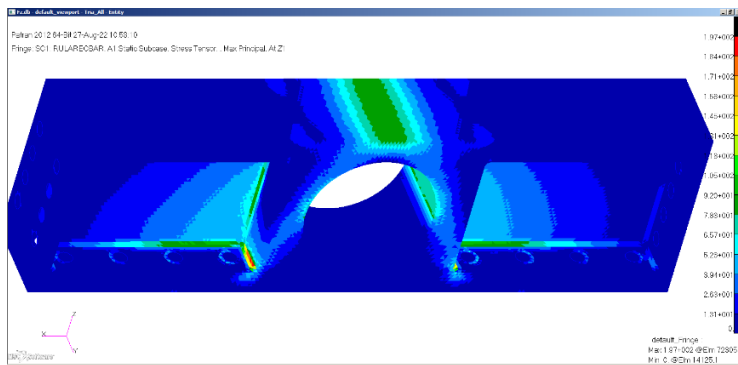


Fig. 9 – Maximum principal stress values for Fz case

After running the program, the following results were obtained, considering 3,000 cycles of use, and a correction factor of 3.

In view of the described hypotheses used in the analytical calculations, the following data table can be written:

Table 1 – Fatigue analysis results:

Case	Maximum stress [MPa]	Stress amplitude [MPa]	Allowable number of cycles	Damage
Fx	126.0	189.0	209518.1312	0.0143
Fz	197.0	295.5	18709.2237	0.1603

#### 4.5 Future program capabilities

The developed VBA fatigue analysis program can be further updated, by expanding its capabilities in the following ways:

- implement fatigue normalization using the Gerber method;
- perform direct comparisons with other fatigue methods;
- improve the user interface;
- extend the file format types for data imports;
- perform computation of fatigue curves by considering directly two graphical points.

## 5. CONCLUSIONS

The present article has described and demonstrated the potential use of a VBA fatigue analysis program for the preliminary sizing of metallic structural components found in aerospace applications. After considering the results and discussion mentioned in the article, the following conclusions can be stated:

1. Due to the prevalent nature of fatigue failure, it is important to evaluate the behavior of primary structural components for fatigue failure.
2. For large data analyses of finite element models for fatigue loading, the use of specialized programs is necessary, that can handle large amounts of data processing, thereby the main goal of this article was the elaboration of a VBA code for fast and efficient evaluation of such data.
3. The program which was made in four parts: data import, data preparation, summarization of data, and fatigue analysis is capable of predicting fatigue damage, based on a repeated equivalent static load case.

Enhancements to the design and user interface of the program can be further studied, in order to facilitate the evaluation of fatigue parameters for structural components.

## ACKNOWLEDGEMENTS

This work was supported by the POCU/993/6/13-153178 grant, co-financed by the European Social Fund through the Sectorial Operational Programme Human Capital 2014-2020.

## REFERENCES

- [1]. N. S. Gokhale, S. S. Deshpande, S. V. Bedekar, A. N. Thite, *Practical finite element analysis*, Finite to Infinite, 2008.
- [2]. A. A. Shanyavskiy, A. P. Soldatenkov, *Metallic materials fatigue behavior: Scale levels and ranges of transition between them*, *International Journal of Fatigue*, vol. **158**, 106773, ISSN 0142-1123, 2022.
- [3]. A. Castriota, V. Dattoma, B. Gambino, R. Nobile, F. Panella, A. Pirinu, A. Saponaro, Numerical and experimental analysis of a composite rear spar subjected to random fatigue loading conditions, *Procedia Structural Integrity*, vol. **12**, pp. 71-81, ISSN 2452-3216, 2018.
- [4]. H. Wang, B. Li, J. Gong, F.-Z. Xuan, Machine learning-based fatigue life prediction of metal materials: Perspectives of physics-informed and data-driven hybrid methods, *Engineering Fracture Mechanics*, vol. **284**, 109242, ISSN 0013-7944, 2023.
- [5]. E. Santecchia, A. M. S. Hamouda, F. Musharavati, E. Zalnezhad, M. Cabibbo, M. El Mehtedi, S. Spigarelli, A Review on Fatigue Life Prediction Methods for Metals, *Advances in Materials Science and Engineering*, vol. **2016**, 9573524, 2016.
- [6]. R. I. Stephens, A. Fatemi, R. R. Stephens, H. O. Fuchs, *Metal Fatigue in Engineering, Second Edition*, John Wiley & Sons, 2001.
- [7]. F.C. Campbell, *Fatigue and Fracture: Understanding the Basics*, ASM International, 2012.
- [8]. I. Deutsch, *Rezistentă Materialelor*, Ed. Didactica si Pedagogica, 1979.
- [9]. V. Ilie, C. Bia, M. V. Soare, *Rezistentă Materialelor si Teoria Elasticitatii*, Ed. Didactica si Pedagogica, 1983.
- [10]. L. Gharani, *Unlock Excel VBA and Excel Macros*, Udemy.
- [11]. G. Vaccaro, *Learning Visual Basic for Microsoft Excel*, InfiniteSkills, 2012.
- [12]. M. A. Poatsy, J. Davidson, K. Mulbery, *Exploring Microsoft 365: Excel 2021*, Pearson, 2022.
- [13]. \* \* \* MSC Nastran, *Quick Reference Guide*, version 2018.
- [14]. \* \* \* MSC Patran, *Post Processing Guide*, version 2012.
- [15]. \* \* \* *Altair HyperMesh help file*, version 2017.
- [16]. \* \* \* U.S. Department of Defense, *Military Handbook - MIL-HDBK-5J: Metallic Materials and Elements for Aerospace Vehicle Structures*, 2003.