ENME 337 – Computing Tools for Engineering Design

Fall 2020

Project

Project summary:

Create a Python program called "PipeFatigue.py" that provides an analysis of the fatigue life of pipelines with manufacturing defects (cracks). This program should be user-friendly and accessible to a general user (client) interested in predicting pipeline fatigue properties.

Group size:

3 – 4 students

Due date:

9th Dec 2020 at 11:59 pm.

Background:

The predominant cause for pipeline disasters in the petroleum industry is fluctuations in pressure from pumps and valves. These pressure transients can lead to fatigue cracks and failure of pipelines with pre-existing manufacturing defects. Fracture mechanics can be applied to pipe fatigue analyses to analyse the growth of a crack from when it is detected to final fracture [1]. The most common graph used to depict the crack growth in a material is shown in Figure 1 and is plotted on a log-log scale. On this log-log plot, there is typically a straight-line region (Region II) which is defined by the Paris equation:

$$\frac{da}{dN} = A(\Delta K)^n \qquad \textit{Eq. 1}$$

where n is the slope of the line and A is the coefficient found by extending the straight line to $\Delta K = 1\ \mathrm{MPa(m)}^{0.5}$. A and n are material properties and K is the stress intensity factor which is used to characterize the stress field near the crack tip. The fracture mechanics in the mathematical models formulated for this Python-based project use the Paris equation.

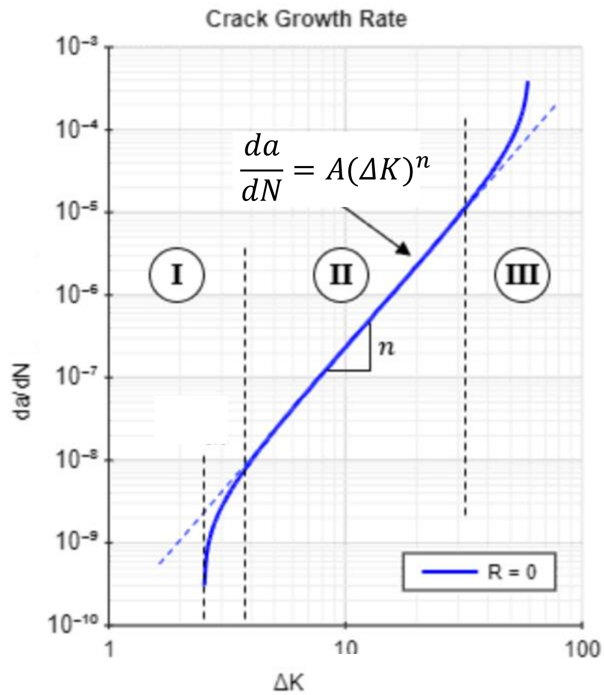


Figure 1: The Crack Growth Rate da/dN vs the Stress Intensity Range ΔK for a Material [2]

Project description:

Your PipeFatigue program should present a main menu, from which the user can select which type of pipe fatigue analysis is required. There should be 3 options: Pressure Surge Analysis (Problem A), Determination of Crack Growth Rate Material Properties (Problem B) and Failure Probability Analysis (Problem C).

Pressure Surge Analysis (Problem A) calculates the magnitude of the pressure surge that would cause complete failure of a pipeline with an initial crack. The number of cycles to failure (fatigue life) is determined and compared with the actual desired lifetime of the pipeline.

Determination of Crack Growth Rate Material Properties (Problem B) analyses data from fatigue experiments using a compact tension specimen to obtain fatigue crack growth rate properties for the Paris equation (*Eq. 1*). The standard deviation and mean are calculated considering geometry variations in the compact tension specimen.

Failure Probability Analysis (Problem C) uses a simple Monte Carlo simulation of pipeline fatigue failure to estimate the failure probability of a steel pipeline.

The detailed problem requirements, inputs, mathematical models, and required outputs are provided in the following sections. Be consistent with units. It is advised to work in the following units: m, N, MPa.

You must implement each problem (A, B, and C) in a separate file that is accessed through PipeFatigue.py, not directly executed by the user (i.e. when grading this project, we will only run your PipeFatigue.py file). There are many possible implementations to successfully complete these problems so choose structures and program flows that are clear and efficient, along with variable names and comments that make your code clear and readable. It is highly recommended that you use subprograms (user-defined functions) throughout to modularise your code flow and make debugging/grading easier.

Please read this entire document carefully. Plan out the structure and functions required for all problems before coding.

Problem A - Pressure Surge Analysis

Pressure surges caused by valve failures can burst a pipeline and cause adverse environmental and economic effects. This file/module will take inputs from the user about a metallurgical defect (a semi-elliptical surface crack in tension) at the inner surface of a long, pressurized gas pipeline as shown in Figure 2. The **magnitude of the pressure surge** that would cause complete pipe failure should be calculated.

To prevent fatigue failure, the fatigue crack growth rate can be analysed. Calculate the number of cycles to reach the critical crack depth; this is also known as the **number of cycles to failure (N_f).**

Determine if the fatigue life (N_f) is within the desired lifetime of the pipeline (L_f).

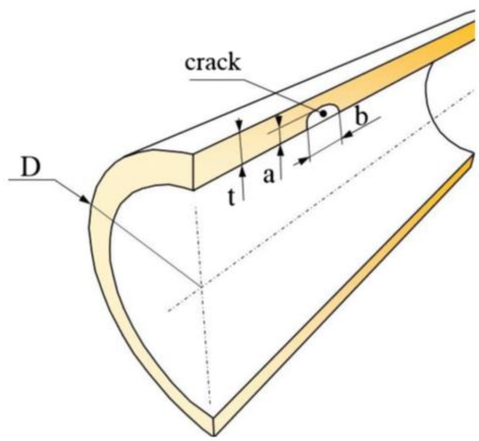


Figure 2: Crack on the Inner Surface of Pipeline

Inputs / Parameters

Table 1: Inputs and Parameters for Problem A

Parameter	Parameter Source	Units
Pipe Outer Diameter (D_o)	from Database	m
Pipe Inner Diameter (D_i)	to be Calculated	m
Pipe Wall Thickness (t)	from Database	m
Initial Depth of Crack (a)	User Input	m
Length of Crack (b)	User Input	m
Internal Pipeline Pressure (P)	User Input	MPa
Plane Strain Fracture Toughness (K_{IC})	from Database	MPa $m^{1/2}$
Geometry Correction Factor (Y)	to be Calculated	-
Desired Lifetime of Pipeline (L_f)	User Input	years
Paris Equation Coefficient (A)	from Database	m/cycle
Paris Equation Exponent (n)	from Database	-
Incremental Crack Growth (da)	User Input	m

Mathematical Model

1) Prompt the user for the nominal pipe size (NPS), pipe schedule and pipe material. Look up the actual pipe outer diameter (D_o), pipe schedule, and wall thickness (t) in the "PipeData.csv" file provided and obtain the plane strain fracture toughness (K_{IC}), Paris equation coefficient (A) and Paris equation exponent (n) from the "MaterialData.csv" file provided.

   Note: Some pipe schedules are not assigned in the database; your program should allow the user to manually input the wall thickness of the pipeline in this instance.
   Note: For instances where a minimum and maximum K_{IC} value is given, use the average value.

2) Allow the user to input the initial depth of the crack (a), length of the crack (b), internal pressure (P), desired lifetime of the pipeline (L_f) and the incremental crack growth (da).

3) Determine if the pipe is classified as a thick-wall or thin-wall pipe. Pipes with $D_o/t > 20$ are referred to as thin-wall pipes and pipes with $D_o/t < 20$ are thick-wall pipes.

4) The geometry correction factor (Y) can be calculated using the graph shown below [3]:

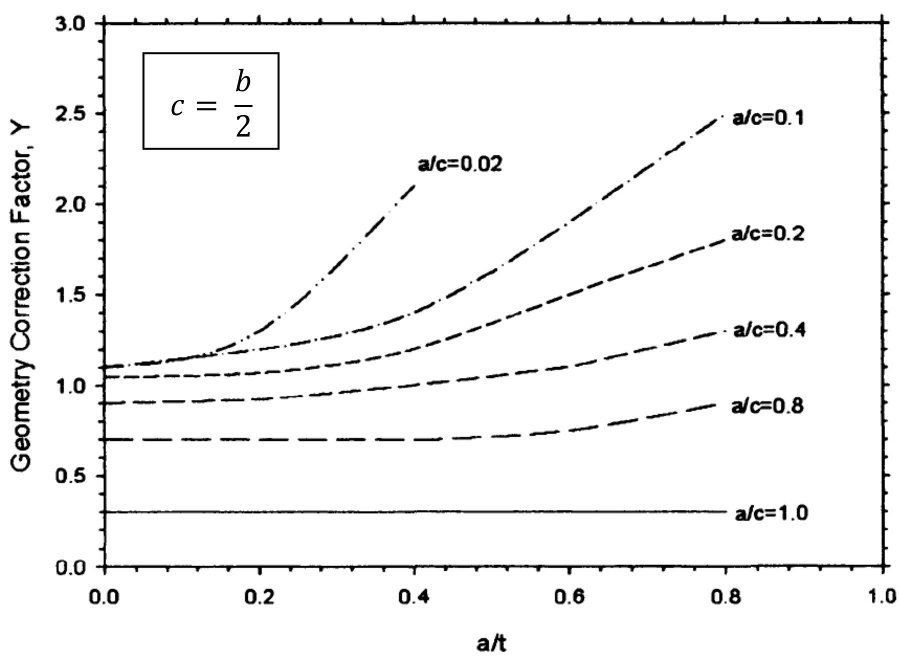


Figure 3: Geometry Correction Factor (Y) for a Semi-Elliptical Surface Crack

Calculate a/c based on the user inputs and determine which line your calculated a/c value is closest to in Figure 3. Read data from the “YGraphData.csv” file provided and fit the data for the appropriate curve in Figure 3 to determine Y.

- For a/c = 1.0, a linear equation should be used
- For a/c = 0.8, fit a polynomial of degree 4
- For a/c = 0.4, fit a polynomial of degree 3
- For a/c = 0.2, fit a polynomial of degree 4
- For a/c = 0.1, fit a polynomial of degree 4
- For a/c = 0.02, fit a polynomial of degree 4

5) At fracture (complete pipe failure), the stress intensity factor (K_I) is equal to the plane strain fracture toughness of the pipe material (K_{IC}). The surge pressure (P_s) can be calculated as follows:

For thin-wall pipes:

$$K_I = YP_s\left(\left[\frac{D_i}{2t}\right] + 1\right)\sqrt{\pi a} = K_{IC} \qquad \textit{Eq. 2}$$

For thick-wall pipes:

$$K_I = YP_s\left(\left[\frac{L^2 + 1}{L^2 - 1}\right] + 1\right)\sqrt{\pi a} = K_{IC} \qquad \textit{Eq. 3}$$

$$\text{where } L = \frac{D_o}{D_i}$$

6) Calculate the total stress that is applied (sum of the hoop stress and internal pipeline pressure):

$$S_T = S_h + \mathrm{P} \qquad \textit{Eq. 4}$$

For thin-wall pipes:

$$S_h = P\left[\frac{D_i}{2t}\right] \qquad \textit{Eq. 5}$$

For thick-wall pipes:

$$S_h = P\left[\frac{L^2+1}{L^2-1}\right] \qquad \textit{Eq. 6}$$

where $L = \frac{D_o}{D_i}$

7) Assume that Y remains constant as calculated in Step 4. Calculate the critical crack depth (a_{cr}) from:

$$K_{IC} = YS_T\sqrt{\pi a_{cr}} \qquad \textit{Eq. 7}$$

8) Calculate the sum of the number of cycles to failure (N_f) rounded to the nearest 50 cycles. For each incremental crack growth (da), from the initial depth of the crack (a) to the critical crack depth (a_{cr}), calculate and store the following in 4 different arrays:

$$a\_f = a\_o + da \qquad \textit{Eq. 8}$$

$$a_{av} = \frac{a\_o + a\_f}{2} \qquad \textit{Eq. 9}$$

$$K = YS_T\sqrt{\pi(a_{av})} \qquad \textit{Eq. 10}$$

dN which is calculated from the Paris equation: $\frac{da}{dN} = AK^n$ *Eq. 11*

In a fifth array, store the sum of dN. This is the sum of the cycles from the previous increments and the current incremental crack growth. The final sum is the total number of cycles to failure (N_f).

A sample of the expected data that the code should generate and store in arrays at this point is shown in Table 2. Separate arrays containing the original crack depth (a_o), the final crack depth (a_f), the average crack depth (a_{av}), K, dN and ∑N should be created to store these values until failure at the critical crack depth (a_cr) is reached. The last row in Table 2 represents the values at failure.

<u>Note: the values shown are for visual purposes only. The actual values will be different based on the user inputs.</u>

Table 2: Summary of the Data that the Code should Calculate (values will vary depending on user inputs)

a_o (mm)	a_f (mm)	a_{av} (mm)	K $(MPa\sqrt{m})$	dN (cycles)	∑N
a = 0.3	0.4	0.35	23.49	4600	4600
0.4	0.5	0.45	26.63	3450	8050
0.5	0.6	0.55	29.44	2800	10800
…	…	…	…	…	…
…	…	…	…	…	…
2.8	**a_cr** = 2.81	2.80	66.47	450	**N_f = 32200**

9) Assume that 1 cycle = 1 day. Determine if the fatigue life (N_f) is within the desired lifetime of the pipeline (L_f).

Expected Output

In addition to prompts for user input, program/calculation status updates (optional), error messages and re-prompts, etc, you must print the following to the Python interpreter/figure window as final outputs in user-friendly messages:

- A list of all of the user-provided inputs.
- Statement telling the user (client) the surge pressure value that will cause fracture.
- Statement telling the client the number of cycles to failure.
- Statement telling the client if the fatigue life (N_f) is within the desired lifetime of the pipeline (L_f).
- Plot of crack depth (a_{av}) vs total cycles to failure on a log-log scale.

Additional Requirements/Notes

- Create and use a function to convert units (mm to m and kN to N) to be used in calculations after reading from data file and user inputs.
- Pipeline outer diameter and wall thickness should be standard values obtained from the database files provided. Wall thickness values provided by the user (if required) can vary from the standard values however a warning message should be output.
- Check that the initial crack depth and critical crack depth are not greater than the wall thickness of the pipeline. If the critical crack depth is greater than the wall thickness, the calculated value should be overwritten with the pipe wall thickness.

- The following inputs from the user should be limited to these ranges (re-prompt the user if outside these bounds):
  - The incremental crack growth (da) input from the user can vary from 0.5 – 1.0 mm.
  - The internal pipeline pressure (P) input from the user can vary from 40 – 2000 MPa.
  - The desired lifetime of the pipeline (L_f) input from the user can vary from 10 – 50 years.
- Utilize exception handling and validation (such as length, range, and type checks, missing values in database within files) for data obtained from user inputs and data read from the provided files.

Problem B - Determination of Crack Growth Rate Material Properties

Material data for fatigue crack growth in pipeline steels are not readily available for use in fatigue analyses. Fatigue experiments as set out in ASTM E467 [4] have to be conducted to obtain the fatigue crack growth rate properties for the Paris equation (*Eq. 1*). Assume that the data provided in "avsN.csv" is for a particular pipeline steel and is obtained using compact tension specimens as shown in Figure 4. In this csv file, a is the crack depth and N is the cycles to failure. Use the data to **obtain the Paris equation coefficient A and exponent n**.

Repeat the calculations 4 times for values within the ranges given for W and B to evaluate how much the Paris coefficient (A) and Paris exponent (n) vary depending on the sample geometry. Output the standard deviation and mean values for A and n. Use equally spaced increments that cover the entire range specified for W and B.

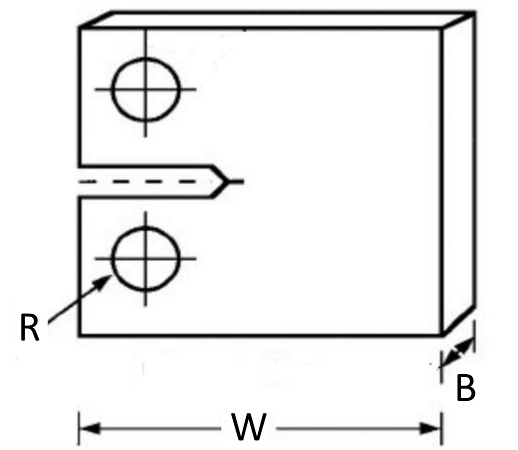


Figure 4: Geometry of Compact Tension Specimen used in Fatigue Test

Inputs/Parameters

Table 3: Inputs and Parameters for Problem B

Parameter	Value	Units
W	25 – 80	mm
Specimen Thickness (B)	$2 \leq B \leq 20$	mm
Maximum Force (P_{max})	6.14	kN
Minimum Force (P_{min})	0.089	kN
No. of Cycles (N)	from Database	-
Crack Depth (a)	from Database	mm

Mathematical Model

1) Plot the crack depth (a) vs the no. of cycles (N)

2) Use the secant method to determine the fatigue crack growth rate (da/dN). The secant method (or point-to-point technique) applied to computing the crack growth rate involves calculating the slope of the straight line connecting two adjacent data points on the a vs N curve. It is formally expressed as follows:

$$\frac{da}{dN} = \frac{a_{i+1} - a_i}{N_{i+1} - N_i} \qquad \textit{Eq. 12}$$

3) Calculate the average crack size:

$$a_{av} = \frac{a_{i+1} + a_i}{2} \qquad \textit{Eq. 13}$$

4) Calculate ΔK in $MPa\sqrt{m}$ using the equation below:

$$\Delta K = \frac{(10^{-6})(P_{max} - P_{min})\sqrt{a_{av}}}{BW}\left[30.96 - 195.8\left(\frac{a_{av}}{W}\right) + 730.6\left(\frac{a_{av}}{W}\right)^2 - 1186.3\left(\frac{a_{av}}{W}\right)^3 + 754.6\left(\frac{a_{av}}{W}\right)^4\right] \qquad \textit{Eq. 14}$$

5) Plot the values of $\frac{da}{dN}$ vs ΔK on a log-log scale. Perform a linear regression analysis. The intercept at ΔK = 1 is the Paris coefficient (A) and the slope of the line is the Paris exponent (n). Note that the Paris equation (*Eq. 1*) can also be written in $\log_{10}$ notation.

6) Calculate the standard deviation and mean of the Paris coefficient (A) and the Paris exponent (n) based on the values obtained from the 4 plots obtained from different B and W values.

Expected Output

Print the following to the Python interpreter/figure window as final outputs in user-friendly messages:

- Values of W and B used in calculations.
- Plot of the crack depth (a) vs the no. of cycles (N).
- Plot of $\frac{da}{dN}$ vs ΔK showing data points and linear regression line. Show a single figure containing 4 subplots, one for each geometry evaluated. On each subplot, include text with the Paris coefficient (A) and the Paris exponent (n).
- Statements with the standard deviation and mean of the Paris coefficient (A) and the Paris exponent (n) based on the values obtained from the 4 plots.

<u>Additional Requirements/Notes</u>

- Use exception handling for a case where the required input data file cannot be found.
- This problem does not require any user inputs.

Problem C - Failure Probability Analysis

Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique often used to understand the impact of risk and uncertainty in prediction models [5]. The following link provides a simple example of a Monte Carlo simulation: https://www.goldsim.com/Web/Introduction/MonteCarlo/.

For this problem, a Monte Carlo simulation of pipeline fatigue failure will be performed to **estimate the failure probability** of a steel pipeline. The results of this analysis can be used by the client to determine inspection intervals, if repairs are required, or if the pipeline should be shut down. This requires your code from Problem A to calculate the number of cycles to failure.

The user should be able to choose either of the following options:

1) Keep the initial length of the crack as a fixed value (standard deviation = 0) while changing the initial depth of the crack (normal distribution with a mean of 8.6 mm and a standard deviation of 1.5 mm). The desired lifetime of the pipeline will have a standard deviation of 10 years with a normal distribution.

2) Keep the initial depth of the crack as a fixed value (standard deviation = 0) while changing the initial length of the crack (normal distribution with a mean of 4.5 mm and a standard deviation of 1.6 mm) The desired lifetime of the pipeline will have a standard deviation of 10 years with a normal distribution.

Inputs

Table 4: Inputs and Parameters for Problem C

Parameter	Mean Value	Standard Deviation	Distribution Type	Units
Pipe Outer Diameter (D_o)	88.9	0	Fixed	mm
Pipe Thickness (t)	11.1252	0	Fixed	mm
Initial Depth of Crack (a)	8.6	1.5	Normal/Fixed	mm
Initial Length of Crack (b)	14.5	1.6	Normal/Fixed	mm
Internal Pipeline Pressure (P)	510	0	Fixed	MPa
Plane Strain Fracture Toughness (K_{IC})	65	0	Fixed	MPa $m^{1/2}$
Geometry Correction Factor (Y)	from Problem A	0	Fixed	-
Desired Lifetime of Pipeline (L_f)	40	10	Normal	years
Paris Equation Coefficient (A)	6.25×10^{-12}	0	Fixed	m/cycle
Paris Equation Exponent (n)	4	0	Fixed	-
Incremental Crack Growth (da)	0.5	0	Fixed	mm

Mathematical Model

1) Generate (pseudo)random values for the variable parameters being sampled (the desired lifetime of the pipeline and the initial depth of the crack / initial length of the crack) following normal distributions.

2) Run a Monte Carlo simulation: Determine how many times (NumFail) that the number of cycles to failure (N_f) exceeds the desired lifetime of the pipeline (L_f) for 10,000 (N) random values generated. Calculate the number of cycles to failure (N_f) using the code from Problem A and assume that 1 cycle = 1 day.

3) Calculate the probability of failure (P_f):

$$P_f = \frac{NumFail}{N}$$

4) Create one plot containing two overlapping histograms that show the probability distributions of the total number of cycles to failure (N_f) and the desired lifetime of the pipeline (L_f). The points where failure occur are at the intersection of the two histograms. Use semi-transparent colours for the histogram (alpha = 0.5) so that the intersection is clearly shown.

Expected Output

In addition to prompts for user input, program/calculation status updates (optional), error messages and re-prompts, etc, you must print the following to the Python interpreter/figure window as final outputs in user-friendly messages:

- Statement of which user option was selected: state which variables were fixed and which were randomly sampled from the given normal distribution.
- Statement providing the calculated failure probability value.
- Plot of the probability distributions.

Project Rubric

5 files should be submitted: main executable program “PipeFatigue.py”, code for Problem A, code for Problem B, code for Problem C and a project report (pdf document described in Table 5). These files must be submitted to the appropriate group dropbox folder in D2L.

All code should run in less than 3-4 minutes. 2 marks will be deducted from the overall project grade if the run-time exceeds this limit.

Table 5: Project Rubric

Topic	Break-down/Description	Marks [Total = 100]
Problem A [35 pts]	Read data from files and request user input	8
	Calculations are correct	8
	Overall structure of code and use of user-defined functions	4
	Outputs are correct	2
	Quality of plot	5
	Exception handling	8
Problem B [20 pts]	Read data from file	2
	Calculations and linear regression are correct	6
	Overall structure of code and use of user-defined functions	3
	Outputs are correct	2
	Quality of plots	5
	Exception handling	2
Problem C [15pts]	Request user input and import modules/functions from Problem A	2
	Calculations are correct	4
	Overall structure of code and use of user-defined functions	3
	Outputs are correct	2
	Quality of plot	4
Main Menu	Obtain user input and call functions/modules appropriately	5
Commenting and Organization	Commenting is well done (not too much or too little) and program is properly thought out and organized. Code is readable and easy to use.	10
Report	Up to 3 pages providing documentation to accompany your code. Describe your program and how it should be used. What are the limitations of the code for each problem? How can your code be improved for future use?	5
Individual Contribution	Individual grade based on evaluations of your other group members concerning your contribution to the project. Further information on how these will be submitted and calculated to follow.	10
Bonus	Write and save the outputs from Problems A, B, and C to files in an appropriate format (pdf/txt/png/etc).	2

References

[1] “Ralph I. Stephens, Ali Fatemi, Robert R. Stephens, Henry O. Fuchs, Ali Faterni, - Metal Fatigue in Engineering, 2nd Edition (2000).”

[2] “Fatigue Crack Growth.” https://mechanicalc.com/reference/fracture-mechanics#stress-intensity-factor (accessed Nov. 03, 2020).

[3] *Problems of Fracture Mechanics and Fatigue*. Springer Netherlands, 2003.

[4] “Standard Test Method for Measurement of Fatigue Crack Growth Rates,” doi: 10.1520/E0647.

[5] W. Kenton, “Monte Carlo Simulation,” Aug. 25, 2020. https://www.investopedia.com/terms/m/montecarlosimulation.asp (accessed Nov. 06, 2020).

Appendix I – Sample expected outputs for Problem A

The expected outputs for specific inputs are given to help validate your code and calculations as progress is made throughout the project. Note: Inputs and outputs should not be limited to only those shown here.

Selected inputs:

```
Enter Nominal Pipe Size (NPS): 3
Enter Pipe Schedule: 160
Enter Pipe Material: X1
Enter crack depth in mm: 1.6
Enter crack length in mm: 4.5
Enter internal pressure in MPa: 40
Enter desired pipe life in years: 30
Enter incremental crack growth in mm: 0.5
```

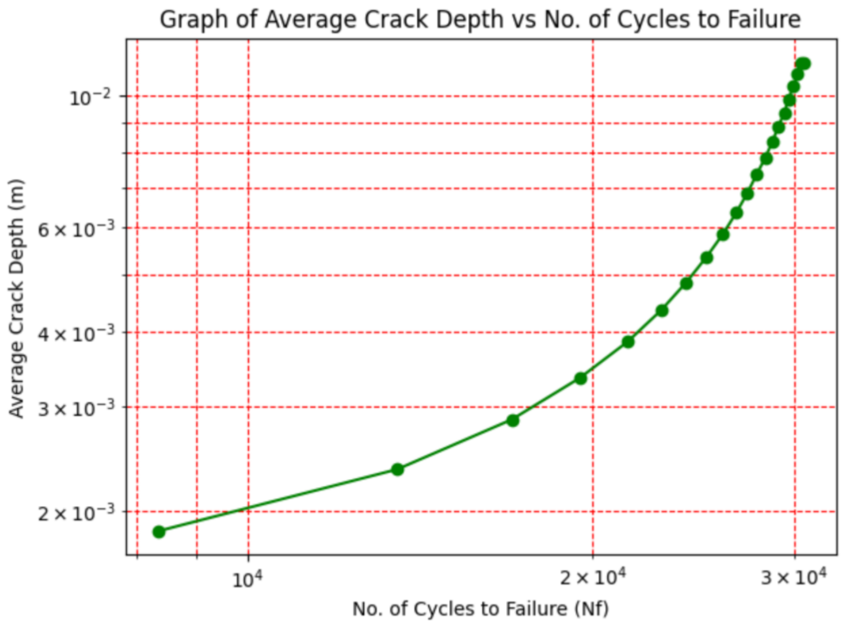
Selected intermediate values at the critical point (useful for debugging): [added 23 Nov 2020]

```
Y is: 0.7112125120055537.
total stress is: 182.67822684311142 MPa.
a_o is: 0.011600000000000006 m.
a_cr is: 0.0111252 m.
a_av is: 0.011362600000000004 m.
K is: 24.54708753995391 MPa m^(0.5).
dN is: 220.3383543210659.
```

Note that these values for a_o, a_cr (=a_f), a_av, K, and dN are the last elements of their respective arrays (i.e. the values that would be in the bottom row of Table 2 for this system).

Selected outputs:

```
The surge pressure that can cause fracture is 108.56 MPa.
The number of cycles to failure (Nf) is 30600.
The fatigue life is within the desired lifetime of the pipe.
```


Graph of Average Crack Depth vs No. of Cycles to Failure
Average Crack Depth (m)
10^{-2}
6×10^{-3}
4×10^{-3}
3×10^{-3}
2×10^{-3}
10^4
2×10^4
3×10^4
No. of Cycles to Failure (Nf)