**Modelling And Simulation of Sheet Metal Forming**

A Report of

**B. Tech Project Work**

Submitted by

**Kumara Swamy B S (200103131)**

**Neeraj Kumar (200103136)**

Under the supervision of

**Dr. Uday Shanker Dixit**



**Indian Institute of Technology**

**Guwahati-781039, India**

**30 April 2023**

**CERTIFICATE**

It is certified that the work contained in the project report entitled “**Modelling and Simulation of Sheet Metal Forming**”, by **Kumara Swamy B S** (200103131) and **Neeraj Kumar** (200103136)has been carried out under my supervision and that this work has not been submitted elsewhere for the award of a degree or diploma.

Date: 30/04/2023

**Dr. Uday Shanker Dixit**

Professor

Department of Mechanical Engineering

Indian Institute of Technology, Guwahati

**DECLARATION**

I certify that the writing I've submitted reflects my views in my own words, and when I've borrowed someone else's thoughts or words, I've properly acknowledged and referenced those sources. In addition, I affirm that I have followed all rules governing academic honesty and integrity and that I have not created or manipulated any idea, data, fact, or source in my work. I am aware that any breach of the aforementioned rules may result in Institute disciplinary action and may also result in penalties from the sources who were not properly cited or from whose proper permission was not obtained when required.

Kumara Swamy B S (200103131)

Neeraj Kumar (200103136)

Department of Mechanical Engineering,

Indian Institute of Technology, Guwahati.

## ACKNOWLEDGEMENT

We express our gratitude to Dr. Uday Shanker Dixit, our project supervisor, for entrusting us with this assignment. His amicable demeanor and encouraging disposition served as a strong impetus for us. We acknowledge that we have gained a wealth of knowledge under his guidance and we extend our heartfelt appreciation to him for consistently inspiring us to strive for excellence. As a token of our appreciation, we dedicate this work to Dr. Uday Shanker Dixit and the Department of Mechanical Engineering at IIT Guwahati.

## ABSTRACT

This project report "Modelling and Simulation of Sheet Metal Forming" presents a comprehensive study on the simulation and modeling of sheet metal forming processes. The objective of this project was to study the developed analytical models for sheet metal forming that can predict the deformation behavior, strain distribution, and formability of the material under different forming conditions. This report includes an overview of sheet metal forming processes and the fundamental concepts of material behavior during deformation. The analytical results were validated through experimental testing. The report also presents a detailed analysis of the effects of various process parameters such as tool radius, wall angle, step depth and sheet thickness on the sheet metal forming. Finally, the project report concludes with the C++ and Python codes that calculates forming forces for different forming processes and provides recommendations for future research in the field of sheet metal forming.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
|  |  |
| **CONTENTS** | **PAGE NO.** |
|  |  |
| Certificate | 2 |
|  |  |
| Declaration | 3 |
|  |  |
| Acknowledgement | 4 |
|  |  |
| Abstract | 5 |
|  |  |
| CHAPTER 1: Introduction | 7 |
|  |  |
| CHAPTER 2: Literature Review | 8 |
|  |  |
| CHAPTER 3: Objectives | 9 |
|  |  |
| CHAPTER 4: Summary of the research paper | 10 |
|  |  |
| CHAPTER 5: C++ Code | 16 |
|  |  |
| CHAPTER 6: Python Code | 22 |
|  |  |
| CHAPTER 7: Conclusion | 25 |
|  |  |
| CHAPTER 8: Upcoming / Future Work | 27 |
|  |  |
| REFERENCES | 28 |
|  |  |

**CHAPTER 1**

# **Introduction**

In today's world, solving real-world problems is of utmost importance. However, it is not always feasible to conduct experiments on actual systems, especially when there are time and cost constraints involved. This is where simulation modelling comes into play, offering an essential analytical technique that can help solve complex problems.

Simulation modelling involves creating a digital model of a system, and it allows users to experiment on a reliable replica of the system. Unlike physical modelling, simulation modelling is computer-based and uses equations and algorithms to provide accurate insights into complex systems.

Another benefit of simulation modelling is that it is highly customizable. Users can tailor simulations to fit their specific needs by adjusting parameters, adding or removing components, and changing inputs. This allows users to test different scenarios and evaluate the impact of different decisions on the system being modelled.

Simulation modelling is also highly visual, which makes it easy to understand and explain. Simulations can be displayed in 2D or 3D, allowing users to see how different components interact with each other and how changes in one part of the system can affect other parts. This makes it easier to communicate complex ideas to stakeholders and decision-makers who may not have a technical background. This report focuses on the applications of simulation modelling in business and the benefits it offers, such as creating understanding and trust by allowing users to interact with simulation models and observe processes in operation.

**CHAPTER 2**

# **Literature Review**

Sheet metal forming is a common manufacturing process used to create a wide range of products, from car body parts to household appliances. However, the process is complex and involves many variables, including the material properties of the sheet metal, the tool geometry, and the forming forces. Modelling and simulation can help improve the understanding of sheet metal forming and optimize the process parameters.

One recent research paper that explores the modelling and simulation of sheet metal forming is "Analytical modelling and experimental validation of the forming force in several typical incremental sheet forming processes" by Zhidong Chang, Ming Li and Jun Chen. The paper presents an analytical model to predict the forming force in several typical incremental sheet forming processes, including single pass incremental forming (SPIF), multi pass incremental sheet forming (MPISF), and incremental hole flanging (IHF). The model takes into account the material properties of the sheet metal, the tool geometry, and the forming parameters such as the sheet thickness, the radius of the tool, wall angle and the forming depth. The authors also conduct experiments to validate the model and compare the predicted forming forces with the measured values.

Building on this research, the project aims to develop a code in C++ and Python to calculate different forming forces for different sheet metal forming processes. The code will be based on the analytical model presented by Zhidong Chang, Ming Li, Jun Chen and will incorporate additional features such as user inputs for sheet metal properties and tool geometry, as well as a graphical interface to visualize the results.

Overall, the literature review shows that modelling and simulation are valuable tools for improving the understanding of sheet metal forming processes. The project builds on the work of Zhidong Chang, Ming Li, Jun Chen and contributes to the development of a code that can be used to calculate forming forces for different sheet metal forming processes.

**CHAPTER 3**

# **Objectives**

* To study and understand the research paper published on “Analytical modelling and experimental validation of the forming force in several typical incremental sheet forming processes” by Zhidong Chang, Ming Li, Jun Chen.
* To develop a code in C++ and Python that can calculate the forming forces for different sheet metal forming processes and incorporate user inputs for sheet metal properties and tool geometry into the code.
* To validate the code by comparing the predicted forming forces with experimental measurements or existing analytical models.
* To investigate the effects of different parameters, such as the step size and the forming depth, on the forming forces in sheet metal forming processes.
* To analyse the potential for optimization of sheet metal forming processes using the developed code and simulation results.

**CHAPTER 4**

**Summary of the research paper**

The research paper published by Zhidong Chang, Ming Li and Jun Chen on incremental sheet metal forming was studied by the authors. In the paper, a complete analytical model was presented for calculating the forming force in several typical incremental sheet metal forming processes, which was validated with experimental results. Prior to this, an analytical expression based on experimental observations had been proposed by Aeren, but Zhidong and his team's model was found to be more accurate. They focused on three different processes of ISF, namely SPIF (Single Pass Incremental Forming), MPISF (Multi-Pass Incremental Sheet Metal Forming), and IHF (Incremental Hole Flanging). The modelling of forming force for SPIF was particularly emphasized, and the researchers considered four main parameters for their analysis: wall angle (α), step depth (h), tool radius (r), and sheet thickness (to).

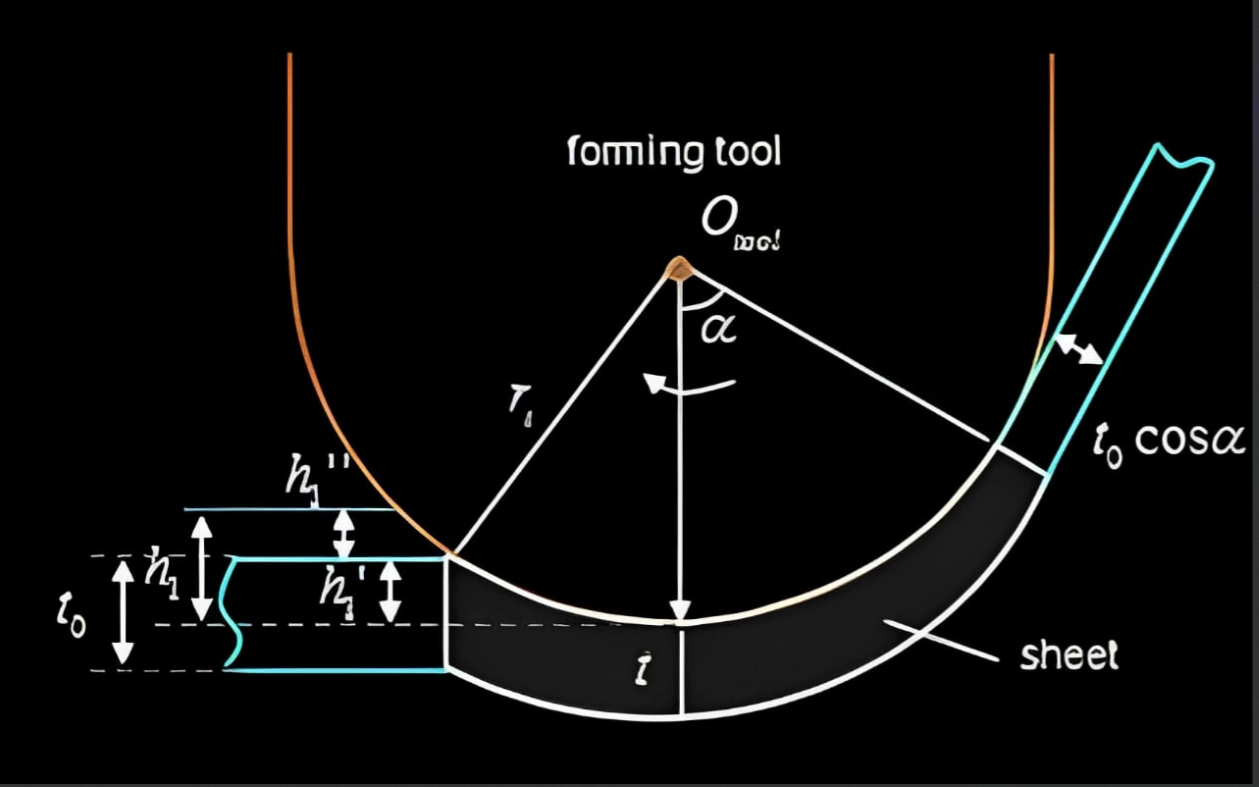
****

Fig 1. Geometric schematic

Theoretically derived expression for contact area(S) between tool and workpiece followed by contact stress in spherical coordinates. To get the contact stress some necessary assumptions are taken. They are,

1. Bending effect is neglected as the sheet thickness is much smaller than the tool radius.
2. Shear stress and are taken into account due to the shearing effect on the formability, while the other shear stress is neglected because it has no obvious effect on the plastic deformation.
3. is related to t and not related to θ: the through-thickness stress is linearly distributed along the thickness direction and evenly distributed along the circumferential direction.
4. Since h1' is much smaller than the tool radius r, h1'/r can be neglected,

h1’ = Sheet thickness thinning at the bottom of the sheet

h1” = Elastic deflection of sheet in vertical direction

h1 = Pressed amount of the forming tool

Concerning the geometric relationship, L1 can be calculated by,

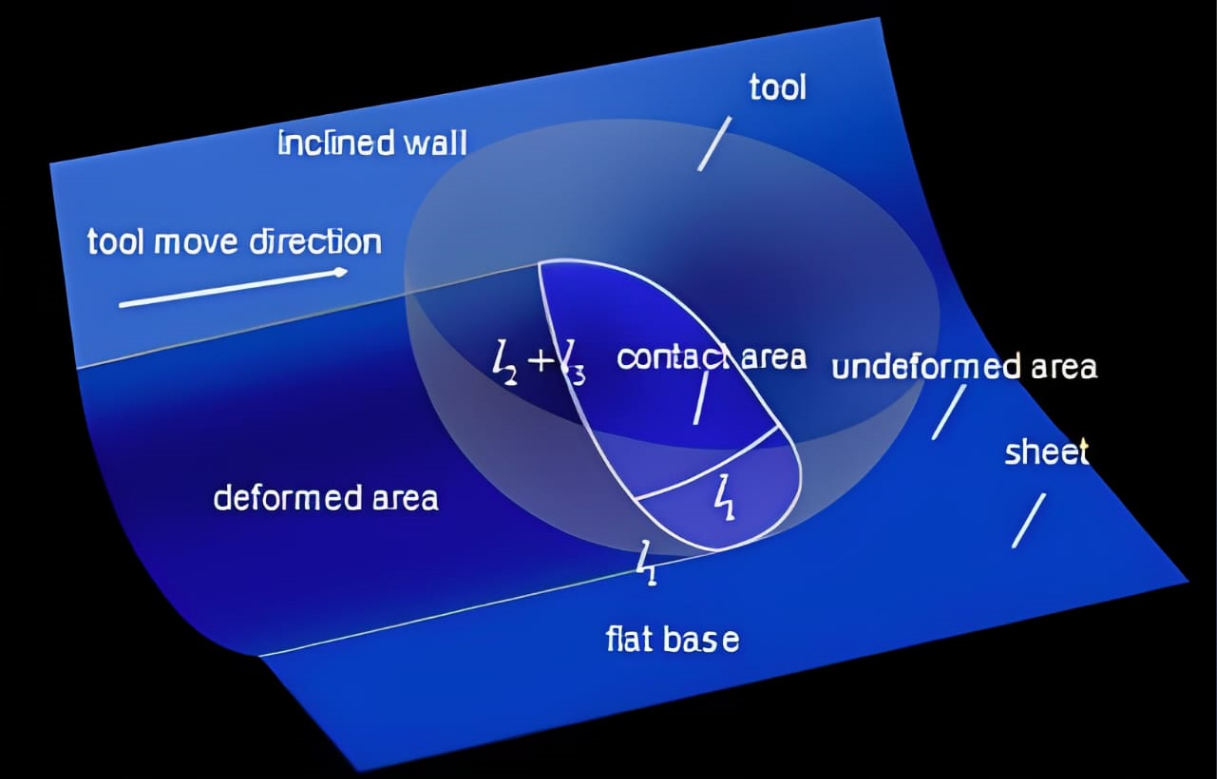
****

Fig 2. Three-dimensional perspective view

L2 and L3 can be given directly by,

The scallop height hs can be calculated by,

The formula for calculating the surface area of the ellipsoidal crown is represented as,

The equivalent stress can be calculated by,

The following equations for calculating the meridional stress and through-thickness acting on the small element can be represented as,

𝜎∅=2*3*

The cosine law is used to calculate the sheet thickness in the present work as follow

**Analytical forming force model of SPIF:**

The equivalent stress can be described by,

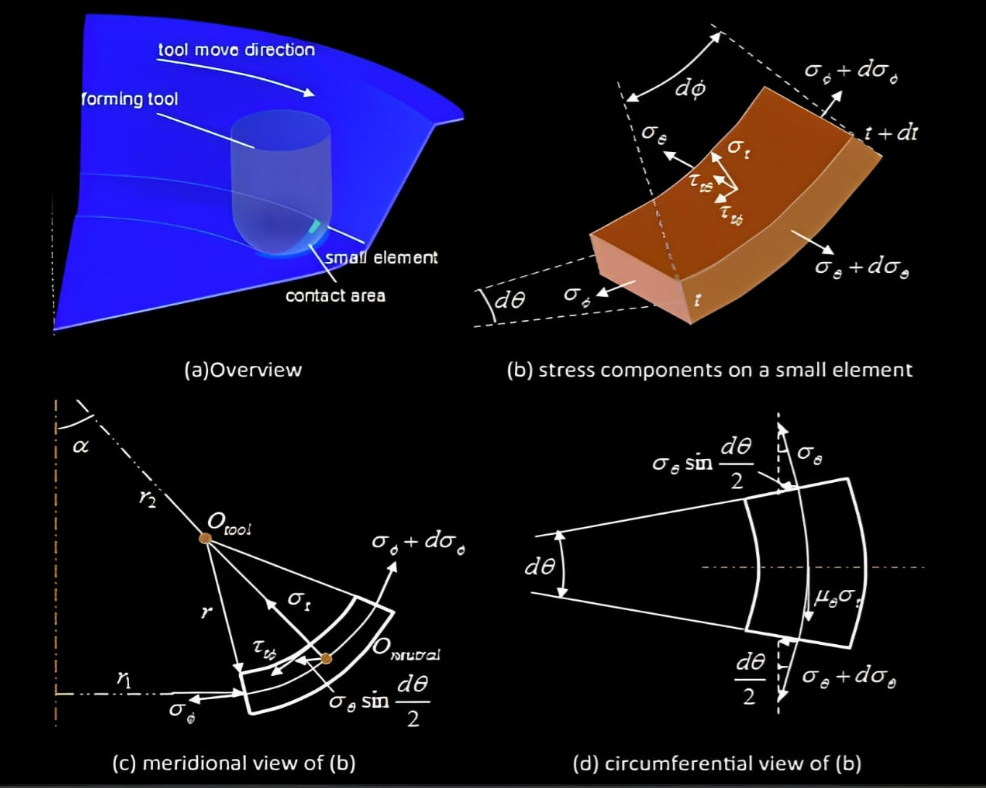


Fig 3. Schematic representation of ISF forming (inspired by Silva et al)

The circumferential strain is caused by stretching and bending, which can be derived by,

Based on volume constancy assumption and plane strain state, the thickness strain can be given by,

The equivalent plastic strain can be calculated by,

The analytical resultant force model of SPIF can be given by,

**Forming force modelling of MPISF:**

The thickness between two adjacent point P and Q can be calculated by,

In the present work, the analytical forming force model in single pass SPIF is updated to accurately predict the forming force in MPISF represented as,

The pressed amount, the equivalent plastic strain and equivalent stress of the jth pass can be calculated by,

Note that is only valid for ISF bending condition if the tool radius is much larger than sheet thickness.

**Forming force modelling of IHF:**

The force model of IHF can be given by,

* They performed experiments on sheets of two different materials (AA5052 and AA3003) and compared the results with values obtained from above expressions of forming force.
* By varying parameters individually, they plotted the graphs and concluded that analytical model is more precise and accurate.
* Analytical models for force prediction in single pass SPIF, MPISF, and IHF have been suggested and confirmed in the current work with improved accuracy compared to the Aeren’s model using a series of experimental data of various geometries for various materials with changing process parameters.
* The analytical force prediction model can be created by multiplying through-thickness stress by the contact area, which has been calculated using a novel method and an improved expression.

**CHAPTER 5**

**C++ Code**

The following C++ code uses analytical equations to determine the forming force necessary for two different sheet-forming procedures, Single Pass Incremental Forming (SPIF) and Incremental Hole Flanging (IHF). To compute equivalent stress, the user must enter the relevant data, including the tool radius, sheet thickness, step depth of the tool, wall angle of the part, yield stress of the workpiece material, and constants "K" and "n".

The forming force needed for SPIF and IHF is calculated by the code using formulas and presumptions acquired from research studies. The calculated forming force for the chosen procedure is displayed in mega newtons (MN) in the output.

The predicted forming force may not be accurate in all sheet forming scenarios because the code depends on the input values. Before using analytical models to analyze a real-world situation, it is crucial to comprehend the assumptions and constraints of those models.

#include<bits/stdc++.h>

using namespace std;

#define pi 3.14159

typedef long double ll;

// r = tool radius

// t = actual sheet thickness

// to = initial sheet thickness

// h = step depth of the tool

// alpha = wall angle of part

// h1 = pressed amount of the forming tool

// h1\_ = Sheet thickness thinning at the bottom of the sheet

// h1\_\_ = Elastic deflection of sheet in vertical direction

// hs = Scallop height on the sheet surface

ll Forming\_Force\_SPIF(ll r, ll to, ll h, ll alpha\_Degree, ll yield\_stress, ll K, ll n) {

    ll alpha = (alpha\_Degree\*pi)/180;

    ll t = to\*cos(alpha);                              // t = actual sheet thickness

    ll numerator = (t \* (1-cos(alpha)));

    ll denomenator = (alpha\*(3-(pi/2)))+1;

    ll h1\_ = numerator/denomenator;                    // h1\_ = Sheet thickness thinning at the bottom of the sheet

    ll h1\_\_ = sqrt((2\*h\*t)/r);                           // h1\_\_ = Elastic deflection of sheet in vertical direction

    ll h1 = h1\_ + h1\_\_;                                // h1 = pressed amount of the forming tool

    ll slack1 = (r\*r)-((h/(2\*sin(alpha))) \* (h/(2\*sin(alpha))));

    ll hs = r - sqrt(slack1);                           // hs = Scallop height on the sheet surface

    ll l1 = (r\*acos((r-h1)/r));

    ll l2 = r\*alpha;

    ll l3 = (r\*acos((r-hs)/r));

    ll slack2 = (r\*(l2+l3)\*(1-cos(alpha)))/l1;

    ll S = (pi\*r\*(h1 + slack2))/2;                      // S = Surface area of ellipsoidal crown

    ll equi\_strain = (2/sqrt(3))\*log((2\*r)/((2\*r\*cos(alpha))+(t\*cos(alpha))));

    ll equi\_stress = yield\_stress + (K\*(1-exp((-n)\*equi\_strain)));

    ll thickness\_stress = (4\*t\*equi\_stress)/(sqrt(3)\*(r+2.5\*t));

    ll Force\_SPIF = thickness\_stress \* S;               // Force\_SPIF = Analytical resultant force of SPIF

    return Force\_SPIF;

}

void Forming\_Force\_MPISF(ll r, ll to, ll h, ll alpha\_Degree, ll yield\_stress, ll K, ll n, ll n\_pass) {

    ll alpha = (alpha\_Degree\*pi)/180;

    ll t = to\*cos(alpha);                              // t = actual sheet thickness

    ll numerator = (t \* (1-cos(alpha)));

    ll denomenator = (alpha\*(3-(pi/2)))+1;

    ll h1\_ = numerator/denomenator;                    // h1\_ = Sheet thickness thinning at the bottom of the sheet

    ll h1\_\_ = sqrt((2\*h\*t)/r);                           // h1\_\_ = Elastic deflection of sheet in vertical direction

    ll h1 = h1\_ + h1\_\_;                                // h1 = pressed amount of the forming tool

    ll slack1 = (r\*r)-((h/(2\*sin(alpha))) \* (h/(2\*sin(alpha))));

    ll hs = r - sqrt(slack1);                           // hs = Scallop height on the sheet surface

    ll l1 = (r\*acos((r-h1)/r));

    ll l2 = r\*alpha;

    ll l3 = (r\*acos((r-hs)/r));

    ll slack2 = (r\*(l2+l3)\*(1-cos(alpha)))/l1;

    ll S = (pi\*r\*(h1 + slack2))/2;                      // S = Surface area of ellipsoidal crown

    ll equi\_strain = (2/sqrt(3))\*log((2\*r)/((2\*r\*cos(alpha))+(t\*cos(alpha))));

    ll equi\_stress = yield\_stress + (K\*(1-exp((-n)\*equi\_strain)));

    ll thickness\_stress = (4\*t\*equi\_stress)/(sqrt(3)\*(r+2.5\*t));

    ll Force\_MPISF = thickness\_stress \* S;               // Force\_MPISF = Analytical resultant force of MPISF

    if(isnan(Force\_MPISF)) {

        cout << "\n Some of the input values entered are not logically correct with the assumptions taken in this model!!";

    }

    else {

        cout << "\n Forming force for 1 pass = " << Force\_MPISF << " N";

    }

    ll alpha\_Degree\_i;

    ll t\_j;

    ll Rm = r + (t/2);

    for (int i = 2; i <= n\_pass; i++) {

        cout << "\n alpha for " << i << " pass = ";

        cin >> alpha\_Degree\_i;

        ll alpha\_i = (alpha\_Degree\_i\*pi)/180;

        ll t\_i = t\*cos(alpha\_i);                              // t = actual sheet thickness

        ll numerator = (t\_i \* (cos(alpha)-cos(alpha\_i)));

        ll denomenator = (alpha\_i\*(3-(pi/2)))+1;

        ll h1\_ = numerator/denomenator;                    // h1\_ = Sheet thickness thinning at the bottom of the sheet

        ll h1\_\_ = sqrt((2\*h\*t)/r);                           // h1\_\_ = Elastic deflection of sheet in vertical direction

        ll h1 = h1\_ + h1\_\_;                                // h1 = pressed amount of the forming tool

        ll slack1 = (r\*r)-((h/(2\*sin(alpha))) \* (h/(2\*sin(alpha))));

        ll hs = r - sqrt(slack1);                           // hs = Scallop height on the sheet surface

        ll l1 = (r\*acos((r-h1)/r));

        ll l2 = r\*alpha\_i;

        ll l3 = (r\*acos((r-hs)/r));

        ll slack2 = (r\*(l2+l3)\*(1-cos(alpha\_i)))/l1;

        ll S\_i = (pi\*r\*(h1 + slack2))/2;                      // S = Surface area of ellipsoidal crown

        ll equi\_strain\_i = (2/sqrt(3))\*log((r\*t)/(Rm\*t\_i));

        ll equi\_stress\_i = equi\_stress + (K\*(1-exp((-n)\*equi\_strain\_i)));

        ll thickness\_stress\_i = (4\*t\_i\*equi\_stress\_i)/(sqrt(3)\*(r+2.5\*t\_i));

        ll Force\_MPISF\_i = thickness\_stress\_i \* S\_i;               // Force\_SPIF = Analytical resultant force of SPIF

        if(isnan(Force\_MPISF\_i)) {

            cout << "\n Some of the input values entered are not logically correct with the assumptions taken in this model!!";

        }

        else {

            cout << "\n Forming force for " << i << " pass = " << Force\_MPISF\_i << " N";

        }

        t = t\_i;

        alpha = alpha\_i;

        equi\_stress = equi\_stress\_i;

    }

}

ll Forming\_Force\_IHF(ll r, ll to, ll h, ll alpha\_Degree, ll yield\_stress, ll K, ll n) {

    ll Force\_ISF = Forming\_Force\_SPIF(r, to, h, alpha\_Degree, yield\_stress, K, n);

    if(!isnan(Force\_ISF)) {

        return Force\_ISF/2;

    }

    else {

        return Force\_ISF;

    }

}

int main() {

    ll r, to, h, alpha\_Degree, yield\_stress, K, n, Force\_SPIF, Force\_IHF, n\_pass;

    int option = 0;

    char choice = 'Y', s;

    while(choice == 'Y' || choice == 'y') {

        cout << "\n Select a option from the below list to proceed: ";

        cout << "\n 1.To calculate forming force of SPIF(Single Pass Incremental Forming)";

        cout << "\n 2.To calculate forming force of IHF(Incremental Hole Flanging)";

        cout << "\n 3.To calculate forming force of MPISF(Multiple Pass Incremental Sheet Forming)";

        cout << "\n 4.To exit from the program\n";

        cin >> option;

        if(option > 4) {

            cout << "\n Invalid option. Program terminates!";

        }

        if(option == 4) {

            break;

        }

        cout << "\n Enter data for calculation: ";

        cout << "\n Enter tool radius(r) in mm: ";

        cin >> r;

        cout << "\n Enter initial sheet thickness(to) in mm: ";

        cin >> to;

        cout << "\n Enter step depth of the tool(h) in mm: ";

        cin >> h;

        cout << "\n Enter wall angle of part(alpha) in degree: ";

        cin >> alpha\_Degree;

        cout << "\n Enter yield stress of workpiece material in MPa: ";

        cin >> yield\_stress;

        cout << "\n Enter the value of constant 'K' to calculate equivalent stress: ";

        cin >> K;

        cout << "\n Enter the value of constant 'n' to calculate equivalent stress: ";

        cin >> n;

        line:

        switch (option) {

            case 1:

                Force\_SPIF = Forming\_Force\_SPIF(r, to, h, alpha\_Degree, yield\_stress, K, n);

                if(isnan(Force\_SPIF)) {

                    cout << "\n Some of the input values entered are not logically correct with the assumptions taken in this model!!";

                }

                else {

                    cout << "\n Analytical resultant forming force of SPIF is: " << Force\_SPIF << " N";

                }

                break;

            case 2:

                Force\_IHF = Forming\_Force\_IHF(r, to, h, alpha\_Degree, yield\_stress, K, n);

                if(isnan(Force\_IHF)) {

                    cout << "\n Some of the input values entered are not logically correct with the assumptions taken in this model!!";

                }

                else {

                    cout << "\n Analytical resultant forming force of IHF is: " << Force\_IHF << " N";

                }

                break;

            case 3:

                cout << "\n Enter the number of passes: ";

                cin >> n\_pass;

                Forming\_Force\_MPISF(r, to, h, alpha\_Degree, yield\_stress, K, n, n\_pass);

                break;

        }

        cout << "\n Do you want to calculate other forces using same input(Y/N): ";

        cin >> s;

        if(s == 'Y' || s == 'y') {

            cout << "\n Select a option from the below list to proceed: ";

            cout << "\n 1.To calculate forming force of SPIF(Single Pass Incremental Forming)";

            cout << "\n 2.To calculate forming force of IHF(Incremental Hole Flanging)";

            cout << "\n 3.To calculate forming force of MPISF(Multiple Pass Incremental Sheet Forming)";

            cout << "\n 4.To exit from the program\n";

            cin >> option;

            if(option > 4) {

                cout << "\n Invalid option. Program terminates!";

            }

            if(option == 4) {

                break;

            }

            goto line;

        }

        cout << "\n Do you want to calculate other forces using different input(Y/N): ";

        cin >> choice;

    }

    return 0;

}

**CHAPTER 6**

# **Python Code:**

# This following Python program allows the user to calculate the forming force of two metal forming processes, Single Pass Incremental Forming (SPIF) and Incremental Hole Flanging (IHF), given various input parameters.

# The input parameters required for the calculation include tool radius, initial sheet thickness, step depth of the tool, wall angle of the part, yield stress of the workpiece material, and two constants 'K' and 'n' to calculate equivalent stress.

# The program prompts the user to select which type of forming force they want to calculate and then asks for the necessary input parameters. It then calls the appropriate function (forming force spif or forming force ihf) to calculate the forming force and displays the result.

# The program also allows the user to calculate the forming force for other cases using the same input or different input parameters. If the user enters an invalid option or input, the program provides an error message and terminates.

import math

def forming\_force\_spif(r, to, h, alpha\_degree, yield\_stress, k, n):

    alpha = alpha\_degree \* math.pi / 180

    t = to \* math.cos(alpha)  # t = actual sheet thickness

    numerator = t \* (1 - math.cos(alpha))

    denomenator = alpha \* (3 - (math.pi / 2)) + 1

    h1\_ = numerator / denomenator  # h1\_ = Sheet thickness thinning at the bottom of the sheet

    h1\_\_ = math.sqrt(2 \* h \* t / r)  # h1\_\_ = Elastic deflection of sheet in vertical direction

    h1 = h1\_ + h1\_\_  # h1 = pressed amount of the forming tool

    slack1 = r\*\*2 - ((h / (2 \* math.sin(alpha)))\*\*2)

    hs = r - math.sqrt(slack1)  # hs = Scallop height on the sheet surface

    l1 = r \* math.acos((r - h1) / r)

    l2 = r \* alpha

    l3 = r \* math.acos((r - hs) / r)

    slack2 = (r \* (l2 + l3) \* (1 - math.cos(alpha))) / l1

    S = (math.pi \* r \* (h1 + slack2)) / 2  # S = Surface area of ellipsoidal crown

    equi\_strain = (2 / math.sqrt(3)) \* math.log((2 \* r) / ((2 \* r \* math.cos(alpha)) + (t \* math.cos(alpha))))

    equi\_stress = yield\_stress + (k \* (1 - math.exp((-n) \* equi\_strain)))

    thickness\_stress = (4 \* t \* equi\_stress) / (math.sqrt(3) \* (r + 2.5 \* t))

    force\_spif = thickness\_stress \* S  # Force\_SPIF = Analytical resultant force of SPIF

    return force\_spif

def forming\_force\_ihf(r, to, h, alpha\_degree, yield\_stress, k, n):

    force\_ihf = forming\_force\_spif(r, to, h, alpha\_degree, yield\_stress, k, n)

    if not math.isnan(force\_ihf):

        return force\_ihf / 2

    else:

        return force\_ihf

while True:

    print("\n Select a option from the below list to proceed:")

    print(" 1.To calculate forming force of spif(Single Pass Incremental Forming)")

    print(" 2.To calculate forming force of ihf(Incremental Hole Flanging)")

    print(" 3.To exit from the program")

    option = int(input())

    if option > 3:

        print("\n Invalid option. Program terminates!")

    if option > 2:

        break

    r = float(input("\n Enter tool radius(r) in mm: "))

    to = float(input("\n Enter initial sheet thickness(to) in mm: "))

    h = float(input("\n Enter step depth of the tool(h) in mm: "))

    alpha\_degree = float(input("\n Enter wall angle of part(alpha) in degree: "))

    yield\_stress = float(input("\n Enter yield stress of workpiece material in MPa: "))

    k = float(input("\n Enter the value of constant 'K' to calculate equivalent stress: "))

    n = float(input("\n Enter the value of constant 'n' to calculate equivalent stress: "))

    while True:

            if option == 1:

                force\_spif = forming\_force\_spif(r, to, h, alpha\_degree, yield\_stress, k, n)

                if math.isnan(force\_spif):

                    print("\n Some of the input values entered are not logically correct with the assumptions taken in this model!!")

                else:

                    print("\n Analytical resultant forming force of SPIF is:", force\_spif, " N")

                break

            elif option == 2:

                force\_ihf = forming\_force\_ihf(r, to, h, alpha\_degree, yield\_stress, k, n)

                if math.isnan(force\_ihf):

                    print("\n Some of the input values entered are not logically correct with the assumptions taken in this model!!")

                else:

                    print("\n Analytical resultant forming force of ihf is:", force\_ihf, " N")

                break

            else:

                print("\n Invalid option. Program terminates!")

                break

    s = input("\n Do you want to calculate other forces using same input(Y/N): ")

    if(s == 'Y' or s == 'y'):

        print("\n Select a option from the below list to proceed: ")

        print(" 1.To calculate forming force of spif(Single Pass Incremental Forming)")

        print(" 2.To calculate forming force of ihf(Incremental Hole Flanging)")

        print(" 3.To exit from the program")

        option = int(input())

        if(option > 3):

            print("\n Invalid option. Program terminates!")

        if(option > 2):

            break

        continue

    choice = input("\n Do you want to calculate other forces using different input(Y/N): ")

    if(choice == 'N' or choice == 'n'):

        break

**CHAPTER 7**

# **Conclusion**

**Input-**

Tool radius: 5mm

Initial sheet Thickness: 1.2 mm

Step depth of tool: 0.5 mm

Wall angle: 50 degrees

Material Property ‘K’: 84

Material Property ‘n’: 15

No. of Passes for MPISF: 2

Wall angle for 2nd Pass: 60 degrees

**Output-**

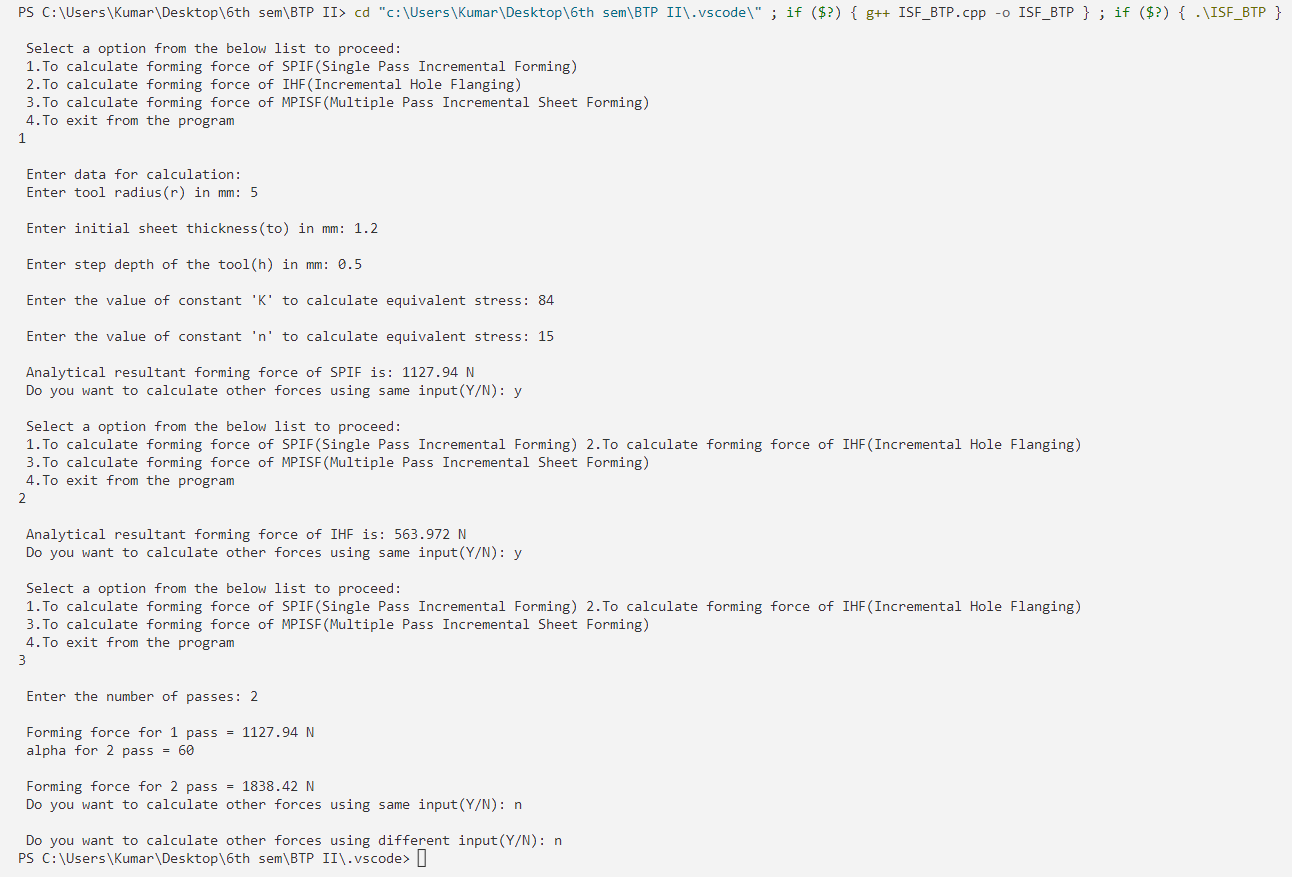
Analytical resulting forming force for SPIF: 1127.94 N

Analytical resulting forming force for IHF: 563.972 N

Analytical resulting forming force for MPISF

Forming force for Pass 1: 1127.94 N

Forming force for Pass 2: 1838.42 N



**Screenshot of output from VS Code software**

Similar output can be obtained using python code. And the outputs obtained are matching with the experimental result.

**CHAPTER 8**

# **Upcoming / Future Work**

* To study how sheet vibrates under different loads during manufacturing.
* To analyze more complicated Boolean Shapes.
* Scrap minimization and appropriate material selection from the analysis even at the initial stages of manufacturing.

**GitHub Links for codes**

* [**C++ Code**](https://github.com/K-umaraswamy/ISF_code/blob/827e43b44f819a0f2bda3705527149b5e10f4093/ISF_BTP.cpp)
* [**Python Code**](https://github.com/K-umaraswamy/ISF_code/blob/827e43b44f819a0f2bda3705527149b5e10f4093/Python_code.py)

**References**

* [**https://www.sciencedirect.com/science/article/pii/S0890695518308691**](https://www.sciencedirect.com/science/article/pii/S0890695518308691)
* **(Figure 3) M. Silva, M. Skioedt, P. Martins, N. Bay, Revisiting the fundamentals of single point incremental forming by means of membrane analysis**

**Int. J. Mach. Tool Manuf., 48 (2008), pp. 73-83**