Blast Design in Mines Using Python

1. Introduction

Blasting is a critical process in surface mining operations, where rock is fragmented using explosives to facilitate excavation. An optimal blast design ensures efficient rock breakage while minimizing environmental impact and operational costs. This report presents a Python-based blast design calculation tool that determines key parameters such as blast hole diameter, burden, spacing, stemming length, and expected particle size.

2. Objectives

The primary objective of this Python program is to calculate the essential parameters required for an effective blast design, including:

- Blast Hole Diameter
- Burden (distance from the blast hole to the free face)
- Spacing (distance between adjacent blast holes)
- Stemming Length (length of inert material to prevent explosive loss)
- Expected Particle Size (fragment size after blasting)

These parameters are calculated using different established methodologies, allowing for customization based on site-specific conditions.

3. Python Code for Blast Design

The following Python code calculates the blast design parameters based on user inputs:

```python
def blast_hole_dia(h, i):
    if i == 1:
        return h/40
    elif i == 2:
        return h/66

def burden(d, m):
    if m == 1:
        return 45*d
    elif m == 2:
        return 19.7 * d**0.79
```

```
    elif m == 3:
        return 30 * d

def spacing(b, m):
    if m == 1:
        return b
    elif m == 2:
        return 1.15*b
    elif m == 3:
        return 1.5*b

def stemming_length(b, d, m):
    if m == 1:
        return 1*b
    elif m == 2:
        return 20*d

def particle_size(d, m):
    if m == 1:
        return 0.05*d
    elif m == 2:
        return 0.15*d

H = int(input("Bench Height:"))
hard_or_soft = int(input("1:Hard rock 2:Soft rock\n"))
D = blast_hole_dia(H, hard_or_soft)

method = int(input("Choose the method for burden \n 1: Hoek and Bray 2: Atlas Copco 3: Dick et al\n"))
Burden = burden(D, method)

methodforspacing = int(input("Method for spacing\n 1: Square 2: Staggered 3: Common\n"))
Spacing = spacing(Burden, methodforspacing)

methodforstem = int(input("Method for stemming\n 1: Atlas 2: Common\n"))
Stemming_length = stemming_length(Burden, D, methodforstem)

methodforps = int(input("Method for particle size\n 1: Kenya 2: Atlas\n"))
Particle_size = particle_size(D, methodforps)

print(f"Blast Hole Diameter: {D}")
print(f"Burden: {Burden}")
print(f"Spacing: {Spacing}")
print(f"Stemming Length: {Stemming_length}")
print(f"Particle Size: {Particle_size}")
```

Listing 1: Blast Design Calculation Code

4. Methodology

The program takes user inputs related to bench height, rock type, and preferred calculation methods. It then applies well-known empirical formulas to compute each blast design parameter.

4.1 Blast Hole Diameter Calculation

The blast hole diameter (D) is determined based on rock hardness:

- **Hard Rock:** $D = \frac{H}{40}$
- **Soft Rock:** $D = \frac{H}{66}$

Where H is the bench height.

4.2 Burden Calculation

The burden (B) is calculated using three different methods:

- **Hoek and Bray Method:** $B = 45D$
- **Atlas Copco Method:** $B = 19.7D^{0.79}$
- **Dick et al. Method:** $B = 30D$

4.3 Spacing Calculation

The spacing (S) between blast holes is calculated based on the pattern used:

- **Square Pattern:** $S = B$
- **Staggered Pattern:** $S = 1.15B$
- **Common Pattern:** $S = 1.5B$

4.4 Stemming Length Calculation

Stemming length (T) is determined using:

- **Atlas Method:** $T = B$
- **Common Method:** $T = 20D$

4.5 Expected Particle Size Calculation

The estimated size of rock fragments after blasting is given by:

- **Kenya Method:** $P = 0.05D$
- **Atlas Method:** $P = 0.15D$

5. Results and Discussion

Upon execution, the program prompts the user for inputs and returns calculated values for blast design parameters. The flexibility in choosing different methodologies ensures adaptability across various mining conditions.

6. Conclusion

The developed Python-based tool provides a systematic approach for blast design in mining operations. By allowing users to choose different empirical formulas, the program offers flexibility and adaptability. Future improvements may include graphical visualization, database integration, and real-time analysis.

7. Recommendations

- Validate the model against field data for improved accuracy.
- Expand the tool to include cost estimation and environmental impact assessment.
- Implement graphical user interfaces (GUI) for ease of use.

8. Project Repository

You can access the complete code on the GitHub Repository.