NETS OF BRICKS

Cat 08 - Mathematics
Group 08-25

Or Zhi Heng Bryan (1P2)

Pan Weiqi (1P2)

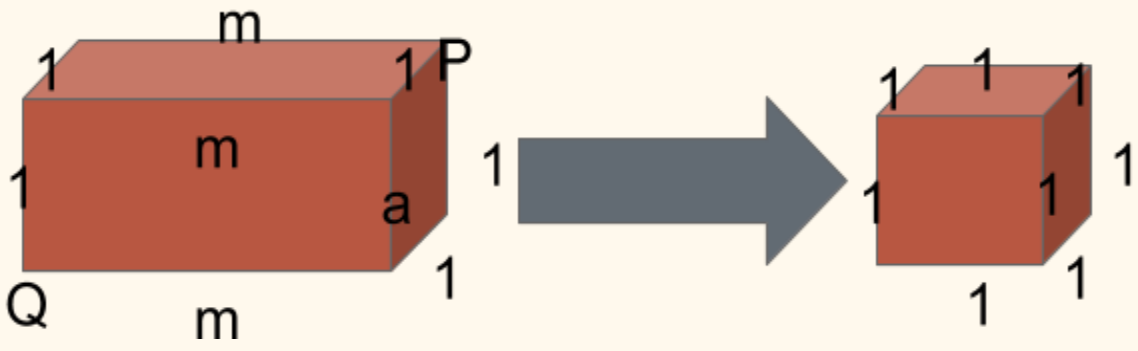
Zhang Yezhou (1P2)

Hwa Chong Institution

(High School)

1 Introduction

Consider the "brick", a rectangular solid which is $1 \times 1 \times m$ where m is a positive integer. When $m = 1$, we have a cube. a is the height of the brick. Let's say Point P and Point Q are 2 edges of the brick and they are opposite to each other. If the brick is opened, it becomes a net, and the brick becomes a surface rather than a solid.



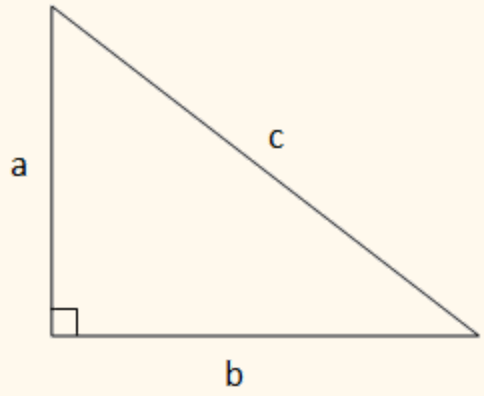
1.1Research questions

1) To find the relationship between PQ, the length of m and length of a.
2) Can the formula still be applied if m is not a integer? (e.g. a fraction)
3) Will the formula still apply if it is used on a cube?
4) How can our formula be applied in real life?

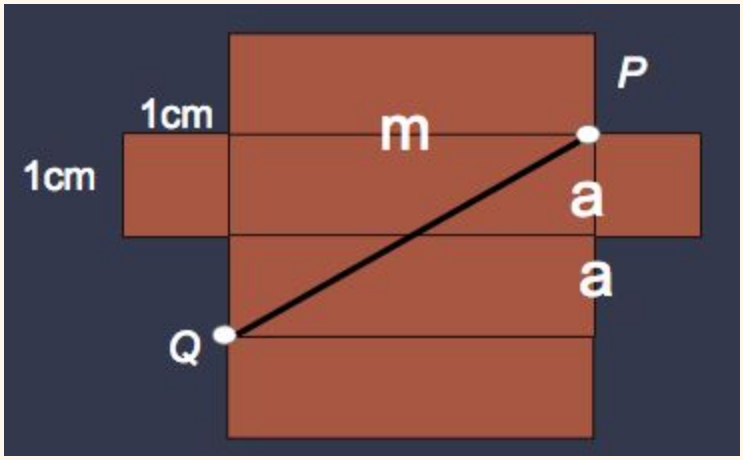
2 Literature review

Pythagoras' Theorem
It is a theorem developed by the famous Greek mathematician, Pythagoras. In a right angled triangle,the longest side $c = \sqrt{b^2 + a^2}$.

(Weisstein, Eric W. "Pythagorean Theorem." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/PythagoreanTheorem.html(Retrieved 13/4/18))



Nets
When a block is cut along the edges of a solid such that the sides of the solid can be laid flat onto a plane. The resulted shape is a net.



The picture on the left shows the layout of the brick we are examining when turned into a net.

(Shinglee New Syllabus 7th Edition Mathematics Textbook)

‘

3.0 Methodology

We intend to find out the actual length of *PQ* using the Pythagorean theorem. We will then test our formula and find out if its results are similar to those of the Pythagorean theorem.

We also created a C++ program to help us calculate the length of *PQ* more efficiently and if the value of *a* changes.

The following is the C++ program.

```
#include <bits/stdC++.h>
using namespace std;
int main() {
 long double c;
 long long choice, a, b, trial[2], into;
  trial[0] = 0;
  trial[1] = 0;
 cout << "Choice 1, find length of m with side 'a' up to..." << "\n" << "Choice 2, find length of m (longest side)..." << "\n"
<< "Enter only 1 or 2" << "\n" << "Choice 3, find length or m if its a cube" << "\n";
 cin >> choice;
  cout  << "Ans corrected to ??? decimal places: ";
  cin >> into;
 cout << fixed << setprecision(into);
 cout << "Enter m ";
  if(choice != 1){
    cout << ":";
  }
  else{
    cout << "up to... :";
  }
  cout << "\n";
  cin >> a;
  cout << "Enter 2nd side (The other side): " << "\n";
 cin >> b;
 if(choice == 1){
   for(long double i = 0; i < a + 1; i++){
```

[Correcting the numbers all to "*into*" decimal places]

```
    cout << sqrt(i * i + b * b * 4);
```
[Outputting the result of *PQ* from when "*i*" is 0 to when "*i*" is at the required number *m* (in the program it is indicated as "*a*").]

```
  }
}
else if(choice == 2){

  c = sqrt(a*a + b*b*4);
```
[Outputting the result of *PQ* from when "*i*" is *m*(in the program it is indicated as "*a*").]

```
  cout << c << "\n";
}
  else if(choice == 3){
   for(long long i = 0; i < a + 1; i++){
     trial[0] = sqrt(i * i * 5);
       cout << trial[0] - trial[1];
```
[Outputting difference in the consecutive results of *PQ* from when "*i*" is 0 to when "*i*" is at the required number *m* (in the program it is indicated as "a").]

```
     trial[1] = trial[0];
  }
}
else{
  cout << "Invalid choice.";
```
[In case the user inputs other characters e.g. alphabets]

```
}
}
```

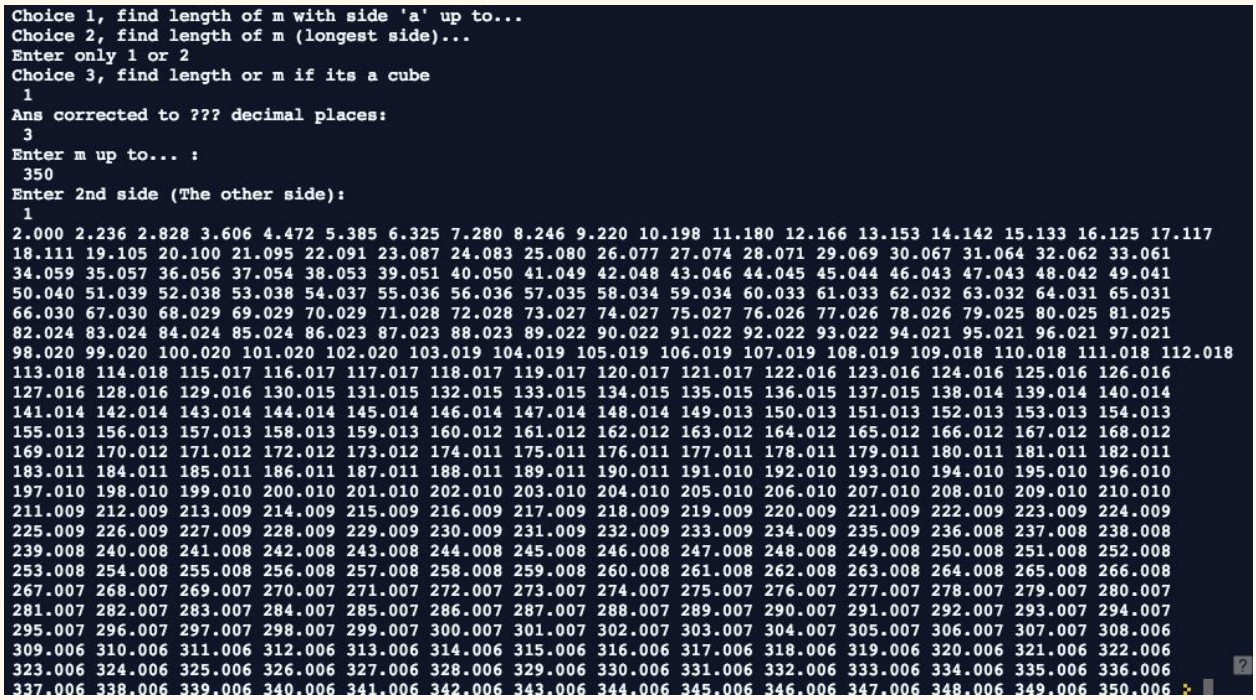
3.1 The relationship between PQ and *m* and *a*:

$$PQ = \sqrt{m^2 + (2a^2)}$$

After many researches, experiments and tests, this formula was developed. Our formula calculates the length of *PQ* on the side of the brick facing front, added to the length of *PQ* on the upwards facing side of the brick. We also compared the results against those of the Pythagorean theorem and they match.

Therefore we can conclude that our formula is correct.

The below picture shows a sample output of one of our tests. Calculated by the C++ program.



3.2 What happens if *m* is not *a* integer? (e.g. a fraction)

The formula will still work. Here are some examples. These data are gotten from a test of our C++ program.

	length of PQ	length of *a*	dimensions of the cuboid
when m is ½	2.061552813	1	½x1x1
when m is ⅓	2.02758751	1	⅓x1x1
when m is ¼	2.015564437	1	¼x1x1

3.3 What happens if the formula is used on a cube?

The formula will still work as cubes and cuboids have similar properties. For example, both cubes and cuboids have six faces, 12 edges and eight vertices, or corners, and cubes and cuboids contain exclusively right angles.. Each edge is shared by two faces. At each vertex, three faces join together.

	length of PQ	length of M
2 cm cube	2.82842712474619	2
3 cm cube	4.242640687119285	3
4 cm cube	5.65685424949238	4

Data collected from calculating the length of *PQ* in a cuboid using our formula, where its dimensions are 1x1xm:

	length of PQ	dimensions of cuboid
when m is 3 cm	3.605551275463989	1x1x3
when m is 4 cm	4.472135954999579	1x1x4
when m is 5 cm	5.385164807134504	1x1x5

4 Conclusion

We have successfully calculated a formula to find the relationship of the distance between points *P* and *Q* and the length of *m*. This is another method alternative to the Pythagorean theorem, with a C++ programming algorithm to go with it, to allow people to use it more efficiently.

A possible outcome to this project could be involved in designs of architecture e.g. Expressionist architecture as it involves brick expressionism and amount of material needed to insert or paint something on the side of the building.

5 References

Black, Kat. "Similarities & Differences of Cubes & Cuboids." Sciencing, https://sciencing.com/similarities-differences-cubes-cuboids-8639117.html. Retrieved 15/8/18.

Weisstein, Eric W. "Pythagorean Theorem." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/PythagoreanTheorem.html. Retrieved 13/4/18