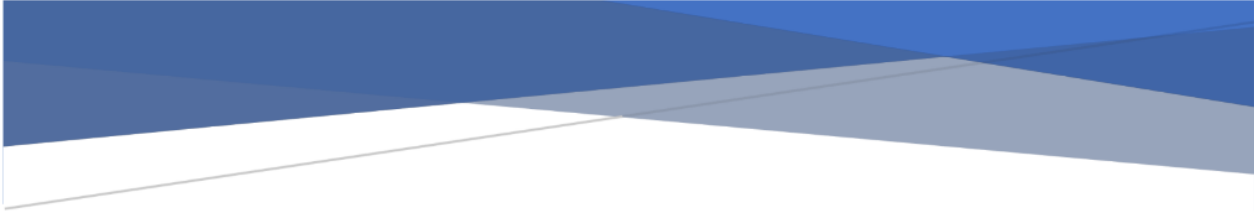


Alex_Thomas_1352162_Assignment_ACDA4003

by Alex Thomas

FILE	12461_ALEX_THOMAS_ALEX_THOMAS_1352162_ASSIGNMENT_ACDA4003_250776_573386758.DOCX (449.17K)		
TIME SUBMITTED	19-JUN-2020 10:00AM (UTC+0100)	WORD COUNT	5633
SUBMISSION ID	130895313	CHARACTER COUNT	24831



APPLICATION OF MATHS FOR SOFTWARE DEVELOPMENT

ACDA4003

Lecturer: Gordon Dickers

Alex Thomas (1352162)
1352162@student.uwtsd.ac.uk

Introduction

For the purpose of this assignment students have been asked to code a maths library using visual studio. The library will consist of several functions performing calculations based on user input. Each function will be numbered, described and tested with inputs, outputs and results displayed on screen. The language used is C++.

1. Length of a vector

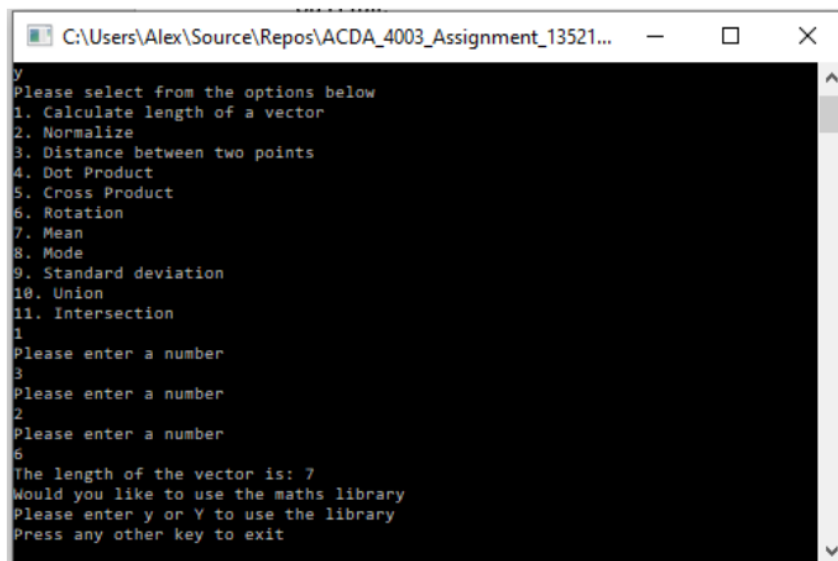
The length or magnitude of a vector is written as $|a|$ and calculated using the formula $|a| = \sqrt{x^2 + y^2 + z^2}$ for 3 dimensions. The solution provides input for 3 dimensions. The variables v1, v2 and v3 equal x, y and z. The values are set to the power of 2, the total variable is the 3 values added which has the square root calculation applied to provide the answer.

Example Length Calculation

Vector a = (3,2,6) its length is 7.

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0001	Calculate length	7	7	Pass



```
y
Please select from the options below
1. Calculate length of a vector
2. Normalize
3. Distance between two points
4. Dot Product
5. Cross Product
6. Rotation
7. Mean
8. Mode
9. Standard deviation
10. Union
11. Intersection
1
Please enter a number
3
Please enter a number
2
Please enter a number
6
The length of the vector is: 7
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 1 Length of a vector

2. Normalise Vector

A vector is normalised by dividing it by its length, the following notation is used $\hat{a} = \frac{a}{|a|}$ the variables v1, v2 and v3 are reused for x, y and z, the values are set to the power of 2, the total variable is the 3 values added which has the square root calculation applied to provide the answer.

Example Normalise

Vector a = (3,2,6) its length is 7, the normalised vector $\hat{a} = (3/7, 2/7, 6/7)$

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0002	Normalise vector	$\hat{a} = (3/7, 2/7, 6/7)$	$\hat{a} = (3/7, 2/7, 6/7)$	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
y
Please select from the options below
1. Calculate length of a vector
2. Normalize
3. Distance between two points
4. Dot Product
5. Cross Product
6. Rotation
7. Mean
8. Mode
9. Standard deviation
10. Union
11. Intersection
2
Please enter value x:
3
Please enter value y:
2
Please enter value z:
6
a=(3/7, 2/7, 6/7)
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 2 Normalise vector

3. Distance between two points

The distance between two points can be calculated to establish a relationship between two vectors. This is achieved by subtracting the points from each vector, this function uses 3 arrays arrayA, arrayB and arrayC, 2 to store the numbers entered by the user limited to 3 entries for x, y and z and the third to output the calculation after subtraction.

Example Distance between two points

Vector a =(8,5,1) vector b =(3,2,1) a-b(8-3,5-2,1-1) vector c=(5,3,0)

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0003	Distance	(5,3,0)	(5,3,0)	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
1. Calculate length of a vector
2. Normalize
3. Distance between two points
4. Dot Product
5. Cross Product
6. Rotation
7. Mean
8. Mode
9. Standard deviation
10. Union
11. Intersection
3
Please enter 3 numbers for first vector
8
5
41
Please enter 3 numbers for second vector
3
2
51
The distance between vectors is: (5,3,0,)
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 3 Distance between two points

4. Dot product of two vectors

The dot product of two vectors is written as $a \bullet b$ and the calculation produces a scalar. The angle is calculated as $= |a| \cdot |b| \cdot \cos\phi$ where ϕ is the angle. The function uses 2 arrays vectorA and vectorB to store user numbers entered by the user limited to 3 entries for x, y and z. Multiplication and addition is then calculated using array positions, the final calculation stored in the variable answer.

Example dot product of two vectors

Vector a =(6,8,0) b=(12,9,0) $a \bullet b = 144$

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0004	Dot product	144	144	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
1. Calculate length of a vector
2. Normalize
3. Distance between two points
4. Dot Product
5. Cross Product
6. Rotation
7. Mean
8. Mode
9. Standard deviation
10. Union
11. Intersection
14
Please enter 3 numbers for first vector
6
8
0
Please enter 3 numbers for second vector
12
9
0
The dot product is: 144
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 4 Dot product of two vectors

5. Cross Product of two vectors

The cross product function is written as $\mathbf{a} \times \mathbf{b} = (a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1)$ returning the vector perpendicular to a and b. The function uses 3 arrays, vectorA and vectorB store user input set to 3 entries for x y and z. The third array cP stores and outputs the calculations highlighted in the equation written above.

Example cross product of two vectors

Vector a =(2,3,4) b=(2,-1,1) cp=(7,6,-8)

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0005	Cross product	(7,6,-8)	(7,6,-8)	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
1. Calculate length of a vector
2. Normalize
3. Distance between two points
4. Dot Product
5. Cross Product
6. Rotation
7. Mean
8. Mode
9. Standard deviation
10. Union
11. Intersection
5
Please enter 3 numbers for first vector
2
3
4
Please enter 3 numbers for second vector
2
-1
1
The cross product is: (7,6,-8,)
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 5 Cross product of two vectors

6. Rotation about the y axis transformation

Rotating about the y axis will take 2 points and an angle then rotate the point to a new point anti-clockwise. To calculate the rotation involves the transformation $x_{new} = x_{old} \cos \phi - y_{old} \sin \phi$ $y_{new} = x_{old} \sin \phi + y_{old} \cos \phi$, x_{new} and y_{new} being the variables for the new point and x_{old} y_{old} the points entered by the user.

Example rotation about the y axis

$x = 3$ $y = 4$ $\text{angle} = 30$ $x_{new} = 4.41488$ $y_{new} = -2.34709$

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0006	Rotation	$x = 4.41488$	$y = -2.34709$	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
y
Please select from the options below
1. Calculate length of a vector
2. Normalize
3. Distance between two points
4. Dot Product
5. Cross Product
6. Rotation
7. Mean
8. Mode
9. Standard deviation
10. Union
11. Intersection
6
Please enter a point (x,y)
Please enter a value for x:3
Please enter a value for y:4
Please enter an angle in degrees:30
Angle of rotation is 30 degrees
0.866025
The rotated point Pnew=(4.41488, -2.34789)
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 6 Rotation about the y axis

7. Mean mode and standard deviation of a list of integers

The mean mode and standard deviation have been coded as 3 separate functions.

Mean

The mean value is calculated by adding the values of a set of numbers then dividing by the number of members in the set [1]. The numbers entered by the user are stored in an array named myArray these numbers are counted and stored in variable total which is then divided by the number variable n.

Example mean

$10+12+8+6 = 36/4$ mean = 9

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0007	Mean	9	9	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
4. Dot Product
5. Cross Product
6. Rotation
7. Mean
8. Mode
9. Standard deviation
10. Union
11. Intersection
7
Please enter numbers for mean (max 10 entries)
4
Enter value 1
10
Enter value 2
12
Enter value 3
8
Enter value 4
6
The total is: 36
The mean is: 9
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 7 Mean

Mode

The mode is the most frequently occurring value in a set of numbers [1]. The function uses an array named myArray to store data entered by the user, a nested for loop is used with conditional if statements the variable tempCounter is used to count the instances of numbers with the highest number of instances stored in the variable mode.

Example mode

4,6,8,4,8,7,8,1 the mode is 8.

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0008	Mode	8	8	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
11. Intersection
8
Please enter numbers for mode (max 10 entries)
8
Enter value 1
4
Enter value 2
6
Enter value 3
8
Enter value 4
4
Enter value 5
8
Enter value 6
7
Enter value 7
8
Enter value 8
1
The mode is: 8
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 8 Mode

Standard deviation

The standard deviation formula indicates the dispersion from members of the set against the average

[1]. Standard deviation can be written as $Std\ Dev = \sqrt{\frac{\sum(x-\bar{x})^2}{n}}$. The function stores the data entered by the user in an array named myArray, reuses the code from the mean function, then calculates the average, a for loop is used to iterate through the numbers entered then the formula is applied to calculate the standard deviation.

Example standard deviation

X = 1,2,3,4,5,6,7,8,9 standard deviation = 2.58

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0009	Standard deviation	2.58	2.58199	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
Please enter numbers for standard deviation (max 10 entries)
9
Enter value 1
1
Enter value 2
2
Enter value 3
3
Enter value 4
4
Enter value 5
5
Enter value 6
6
Enter value 7
7
Enter value 8
8
Enter value 9
9
The standard deviation is: 2.58199
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 9 Standard Deviation

8. Union and Intersection of two sets

The union of two sets consists of members that are in at least one of the sets. The formula can be written as $A \cup B$. This has been coded as two separate functions. The vector library was added for additional functions and 3 vectors used one for each data set entered and one for the union itself. A bool condition used to check for duplicate numbers.

Union

Example Union

$A = \{2,4,6,8,10,12\}$, $B = \{4,7,10,13\}$

$A \cup B = \{2,4,6,7,8,10,12,13\}$

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0010	Union	{2,4,6,7,8,10,12,13}	{2,4,6,7,8,10,12,13}	Pass

```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
9. Standard deviation
10. Union
11. Intersection
10
Please enter amount of numbers for set one:
6
Add numbers
2
4
6
8
10
12
Please enter amount of numbers for set two:
4
Add numbers
4
7
10
13
The union is: 2 4 6 7 8 10 12 13
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 10 Union

Intersection

The intersection of two sets are the members of both set A and B. The formula is written as $A \cap B$. Vectors were reused for this function the logic similar to the union function with the condition amended to true for the bool condition so if there were members of the two sets that matched these were added to the third vector which was then printed on screen.

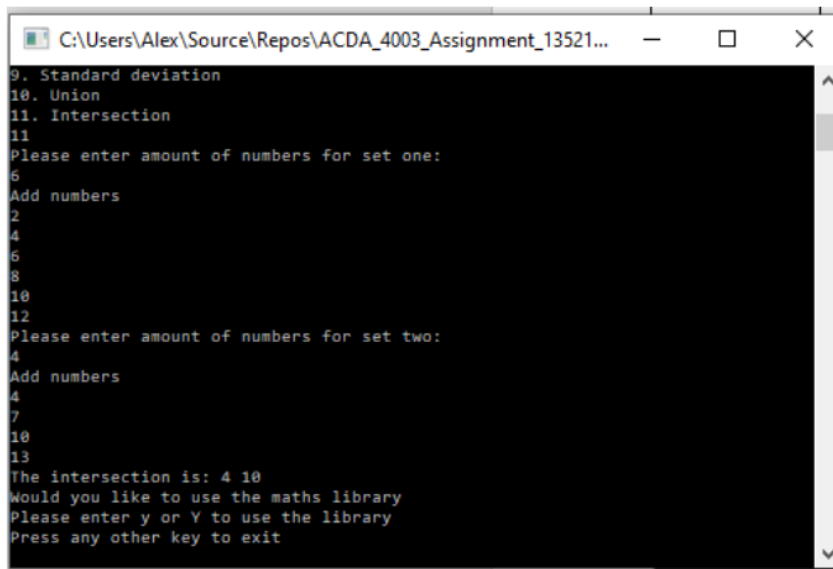
Example Intersection

$A = \{2,4,6,8,10,12\}$, $B = \{4,7,10,13\}$

$A \cap B = \{4,10\}$

Test Schedule

Test No	Test Description	Expected Result	Actual Result	Pass/Fail
0011	Intersection	{4,10}	{4,10}	Pass



```
C:\Users\Alex\Source\Repos\ACDA_4003_Assignment_13521...
9. Standard deviation
10. Union
11. Intersection
11
Please enter amount of numbers for set one:
6
Add numbers
2
4
6
8
10
12
Please enter amount of numbers for set two:
4
Add numbers
4
7
10
13
The intersection is: 4 10
Would you like to use the maths library
Please enter y or Y to use the library
Press any other key to exit
```

Figure 11 Intersection

References

[1] J. Bird, "Basic Engineering Mathematics," Routledge, Abington, 2014.

Appendices

Code

```
#include <iostream>
using namespace std;
#include <math.h>
#include <vector>
#include <string>
#define M_PI 3.14159265358979323846
//functions
char userOption();
int menu();
float enterNumber();
float length();
float normalize();
float distance();
float dotProduct();
int crossProduct();
int rotation();
float average();
int mode();
```

```

float sd();
int un();
int interSection();

int main(void)
{
    //variables returned from functions
    float answer;
    char user;
    int choice;

    user = userOption();
    while (user == 'y' || user == 'Y')//set loop for menu
    {

        choice = menu();
        switch (choice)
        {
            case 1:
                answer = length();
                break;
            case 2:
                answer = normalize();
                break;
            case 3:
                answer = distance();
                break;
            case 4:
                answer = dotProduct();
                break;
            case 5:
                answer = crossProduct();
                break;
            case 6:
                answer = rotation();
                break;
            case 7:
                answer = average();
                break;
            case 8:
                answer = mode();
                break;
            case 9:
                answer = sd();
                break;
            case 10:
                answer = un();

```

```

        break;
    case 11:
        answer = interSection();
        break;

    }
    user = userOption();
}
cout << "Please come back soon" << endl;
return 0;
}
//function for user options
char userOption()
{
    char option;
    cout << "Would you like to use the maths library" << endl;
    cout << "Please enter y or Y to use the library" << endl;
    cout << "Press any other key to exit" << endl;
    cin >> option;

    return option;
}

//menu function for user to select
int menu()
{
    int select;

    cout << "Please select from the options below" << endl;
    cout << "1. Calculate length of a vector" << endl;
    cout << "2. Normalize" << endl;
    cout << "3. Distance between two points" << endl;
    cout << "4. Dot Product" << endl;
    cout << "5. Cross Product" << endl;
    cout << "6. Rotation" << endl;
    cout << "7. Mean" << endl;
    cout << "8. Mode" << endl;
    cout << "9. Standard deviation" << endl;
    cout << "10. Union" << endl;
    cout << "11. Intersection" << endl;
    cin >> select;

    return select;
}

//number function can be used by other functions
float enterNumber()
{

```

```

float num;
cout << "Please enter a number" << endl;
cin >> num;

return num;
}
//length function
float length()
{
    float v1, v2, v3, total, answer;
    v1 = enterNumber(); //use enterNumber function to get numbers
    v2 = enterNumber();
    v3 = enterNumber();

    v1 = pow(v1, 2); //set numbers to power of 2
    v2 = pow(v2, 2);
    v3 = pow(v3, 2);

    total = v1 + v2 + v3; //add numbers
    answer = sqrt(total); // square root the total

    cout << "The length of the vector is: " << answer << endl; //display length
    return 0;
}
//normalize function
float normalize()
{
    float v1, v2, v3, total, answer = 0.0;

    cout << "Please enter value x: " << endl; //prompt user
    cin >> v1;

    cout << "Please enter value y: " << endl;
    cin >> v2;

    cout << "Please enter value z: " << endl;
    cin >> v3;

    v1 = pow(v1, 2); //set numbers to power of 2
    v2 = pow(v2, 2);
    v3 = pow(v3, 2);

    total = v1 + v2 + v3; //add numbers

    answer = sqrt(total); // square root the total

    cout << "a=(" << sqrt(v1) << "/" << answer << ", " << sqrt(v2) << "/" << answer << ", " << sqrt(v3)

```

```

    return 0;
}
//distance function
float distance()
{
    int i, j, arrayA[3], arrayB[3], arrayC[3];
    cout << "Please enter 3 numbers for first vector" << endl;//prompt user

    for (i = 0; i < 3; i++)//set limit to 3 for entries
    {
        cin >> arrayA[i];//store in array
    }

    cout << "Please enter 3 numbers for second vector" << endl;//prompt user

    for (j = 0; j < 3; j++)//set limit to 3 for entries
    {
        cin >> arrayB[j];//store in array
    }

    arrayC[0] = arrayA[0] - arrayB[0];//subtract vector a from vecor b
    arrayC[1] = arrayA[1] - arrayB[1];
    arrayC[2] = arrayA[2] - arrayB[2];

    cout << "The distance between vectors is: (" << flush;

    for (int i = 0; i < 3; i++)
    {
        cout << arrayC[i] << "," << flush;//output distance
    }
    cout << ")" << flush;

    cout << endl;

    return 0;
}
//dotproduct function
float dotProduct()
{
    int i, j, vectorA[3], vectorB[3], answer;
    cout << "Please enter 3 numbers for first vector" << endl;//prompt user

    for (i = 0; i < 3; i++)//set limit to 3 for entries
    {
        cin >> vectorA[i];//store in vector
    }

```

```

    }

    cout << "Please enter 3 numbers for second vector" << endl; //set limit to 3 for entries

    for (j = 0; j < 3; j++)
    {
        cin >> vectorB[j]; //store in array
    }

    answer = vectorA[0] * vectorB[0] + vectorA[1] * vectorB[1] + vectorA[2] * vectorB[2]; //dotproduct calculation

    cout << "The dot product is: " << answer << endl; //output to screen

    return 0;
}
//crossproduct function

int crossProduct()
{
    int i, j, vectorA[3], vectorB[3], cP[3];
    cout << "Please enter 3 numbers for first vector" << endl; //prompt user

    for (i = 0; i < 3; i++) //set limit to 3 for entries
    {
        cin >> vectorA[i]; //store in vector
    }

    cout << "Please enter 3 numbers for second vector" << endl; //prompt user

    for (j = 0; j < 3; j++) //set limit to 3 for entries
    {
        {
            cin >> vectorB[j]; //store in vector
        }
    }

    cP[0] = vectorA[1] * vectorB[2] - vectorA[2] * vectorB[1]; //calculate cross product
    cP[1] = vectorA[2] * vectorB[0] - vectorA[0] * vectorB[2];
    cP[2] = vectorA[0] * vectorB[1] - vectorA[1] * vectorB[0];

    cout << "The cross product is: (" << flush;

    for (int i = 0; i < 3; i++)

        cout << cP[i] << "," << flush; //display on screen
    cout << ")" << flush;
}

```

```

    cout << endl;

    return 0;
}

//rotation function
int rotation()
{
    //P=(x,y)
    float x = 0; //x co-ord of point
    float y = 0; //y co-ord of point
    //Pnew=(xnew,ynew)
    float xnew = 0; //x co-ord of point
    float ynew = 0; //y co-ord of point

    float angle = 0; //rotation angle

    //Input point to be Rotation
    cout << "Please enter a point (x,y)" << endl;
    cout << "Please enter a value for x:";
    cin >> x;
    cout << "Please enter a value for y:";
    cin >> y;

    //Input an angle

    cout << "Please enter an angle in degrees:";
    cin >> angle;

    cout << "Angle of rotation is " << angle << " degrees" << endl;

    //rotation calculation

    cout << sin(60.0 / 180 * M_PI) << endl; //convert degrees to radian

    //new point after anti clockwise rotation
    xnew = x * (cos(angle)) - y * (sin(angle));
    ynew = x * (sin(angle)) + y * (cos(angle));

    cout << "The rotated point Pnew=(" << xnew << ", " << ynew << ")" << endl; //display on screen

    return 0;
}

//mean function
float average()
{
    int myArray[10];
    int n = 0, i, j, total = 0;
    float answer, mean;

```

```

cout << "Please enter numbers for mean (max 10 entries) " << endl; //prompt user for numbers
cin >> n; //store numbers

for (i = 0; i < n; i++) //set loop to count numbers entered
{
    cout << "Enter value " << i + 1 << endl; //prompt user
    cin >> myArray[i]; //store users input in array
}

for (j = 0; j < n; j++) //set loop to count numbers entered
{
    total = total + myArray[j]; //add numbers and store in array
}

cout << "The total is: " << total << endl; //output total

mean = total / n;

cout << "The mean is: " << mean << endl; //output average

return 0;
}

int mode()
{
    int myArray[10];
    int i, j, k, n, mode = 0, counter = 0;

    cout << "Please enter numbers for mode (max 10 entries)" << endl; //prompt user for numbers
    cin >> n; //store numbers

    for (i = 0; i < n; i++)
    {
        cout << "Enter value " << i + 1 << endl; //prompt user
        cin >> myArray[i]; //store users input in array
    }

    for (j = 0; j < n; j++)
    {
        int tempMode = myArray[j]; //equal variable to array
        int tempCounter = 0; //set counter to zero

        for (k = 0; k < 10; k++)
        {

```

```

        if (myArray[k] == tempMode) tempCounter++; //count the number of instances of current number
        if (tempCounter > counter) { // move on to a different number
            mode = tempMode; // mode is the highest number of instances
            counter = 0; //reset counter
        }
    }

}

cout << "The mode is: " << mode << endl; //output mode

return 0;
}

float sd()
{
    int myArray[10];
    int n = 0, i, j, k, sum = 0;
    double answer = 0.0, d = 0, total = 0, mean;

    cout << "Please enter numbers for standard deviation (max 10 entries) " << endl; //prompt user for nu
    cin >> n; //store numbers

    for (i = 0; i < n; i++) //set loop to count numbers entered
    {
        cout << "Enter value " << i + 1 << endl; //prompt user
        cin >> myArray[i]; //store users input in array
    }

    for (j = 0; j < n; j++) //set loop to count numbers entered
    {
        total = total + myArray[j]; //add numbers and store in array
    }

    mean = total / n; //divide the total by the amount of numbers for mean

    for (k = 0; k < n; k++)
    {
        d += pow(myArray[k] - mean, 2); //just discovered that += adds the left variable tp the right marve
    }

    total = d / n; //divide the number of numbers into the sum
    answer = sqrt(total); //square root for the standard deviation

```

```

    cout << "The standard deviation is: " << answer << endl; //output standard deviation

    return 0;
}

int un()
{
    vector<int> setOne;
    vector<int> setTwo;
    vector<int> myUnion;
    vector<int> interSection;
    vector<int> temp;
    int n1, input1, input2, n2, i, j;

    cout << "Please enter amount of numbers for set one: " << endl; //prompt user for amount
    cin >> n1; //store numbers
    cout << "Add numbers" << endl;
    for (i = 0; i < n1; i++) //iterate through numbers set by user
    {
        cin >> input1; //store numbers
        setOne.push_back(input1); //add to vector one
    }

    cout << "Please enter amount of numbers for set two: " << endl; //prompt user for amount
    cin >> n2; //store numbers
    cout << "Add numbers" << endl;
    for (i = 0; i < n2; i++) //iterate through numbers set by user
    {
        cin >> input2; //store numbers
        setTwo.push_back(input2); //add to vector two
    }

    for (i = 0; i < setOne.size(); i++)
    {
        myUnion.push_back(setOne[i]); //add setOne numbers to union vector
    }

    for (i = 0; i < setTwo.size(); i++)
    {
        bool duplicate = false; //set bool condition

        for (j = 0; j < myUnion.size(); j++)
        {

```

```

        if (setTwo[i] == myUnion[j])//if setTwo numbers match set bool to true eradicate duplicates
        {
            duplicate = true;
        }
    }

    if (!duplicate)//if false add these numbers to union vector
    {
        myUnion.push_back(setTwo[i]);
    }

}

sort(myUnion.begin(), myUnion.end());//sort numbers

cout << "The union is: " << flush;
for (int i = 0; i < myUnion.size(); i++)
{
    cout << myUnion[i] << " " << flush;//output to screen
}
cout << endl;
return 0;
}

int interSection()
{
    vector<int> setOne;
    vector<int> setTwo;
    vector<int> myUnion;
    vector<int> interSection;
    vector<int> temp;
    int n1, input1, input2, n2, i, j;

    cout << "Please enter amount of numbers for set one: " << endl;//prompt user for amount
    cin >> n1;//store numbers
    cout << "Add numbers" << endl;
    for (i = 0; i < n1; i++)//iterate through numbers set by user
    {
        cin >> input1;//store numbers
        setOne.push_back(input1);//add to vector one
    }

    cout << "Please enter amount of numbers for set two: " << endl;//prompt user for amount
    cin >> n2;//store numbers
    cout << "Add numbers" << endl;

```

```

for (i = 0; i < n2; i++)//iterate through numbers set by user
{
    cin >> input2;//store numbers
    setTwo.push_back(input2);//add to vector two
}

for (i = 0; i < setOne.size(); i++)//iterate through setone
{

    interSection.push_back(setOne[i]);//add setone numbers to intersection vector

}

for (i = 0; i < setTwo.size(); i++)//iterate through set two
{

    bool duplicate = false;//set condition to false

    for (j = 0; j < interSection.size(); j++)//iterate through intersetion vector
    {
        if (setTwo[i] == interSection[j])//check for duplicates
        {
            duplicate = true;//if duplicate set to true
        }
    }

    if (duplicate)//if duplicate true
    {
        temp.push_back(setTwo[i]);//add duplicates to temp vector
    }

}
sort(temp.begin(), temp.end());//sort numbers for intersection
cout << "The intersection is: " << flush;
for (int i = 0; i < temp.size(); i++)

{
    cout << temp[i] << " " << flush;//output intersection to screen
}
cout << endl;

return 0;
}

```

Documents

Measures of Central Tendency

Average or mean

$$\text{Average or mean } (\bar{x}) = \frac{\text{Sum of Data } (\sum x)}{\text{Number of items } (n)}$$

e.g. Calculate mean of data $x = \{20, 5, 3, 3, 4, 1, 5, 44, 3, 3, 8\}$
number of items, $n = 11$

$$\sum_{n=1}^{11} x = 99$$

$$\bar{x} = 99/11 = 9$$

Median

This is the middle value of a sorted sequence of numbers

e.g. The median of 20, 5, 3, 3, 4, 1, 5, 44, 3, 3, 8 is obtained by first sorting the numbers:

1, 3, 3, 3, 3, 4, 5, 8, 20, 44

The middle value is 4 so the median is 4.

If the list had an even number of data then the average of the middle two numbers is taken.

e.g. if the sequence was 1, 1, 3, 3, 3, 3, 4, 5, 8, 20, 44

Then the median is $(3+4)/2 = 3.5$

Mode

The mode is the most common number in the list, i.e. the number that occurs most often. In our list: 20, 5, 3, 3, 4, 1, 5, 44, 3, 3, 8, the mode is 3.

Measures of dispersion and the standard deviation

The average value of a list of numbers doesn't give us any information regarding how the numbers are dispersed over the number line. For example the list 1, 1, 1, 99, 99, 99 has an average of $300/6=50$. The list 49, 49, 49, 51, 51, 51 also has an average of $300/6=50$. Notice that the numbers in the first list are widely distributed around the average, whilst the second list are closely distributed about the average. A figure that quantified this dispersal would be useful. Several exist, a popular one is called the "standard deviation". Several formulae for this exist but we will use the following:

$$\text{Std Dev} = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

Thus:

i) Calculate the mean $(\bar{x})$

ii) find the deviation of each value from the mean $(x - \bar{x})$

iii) square these deviations $(x - \bar{x})^2$

- iv) sum the squares ($\sum(x - \bar{x})^2$)
- v) Divide by the number of items (n)
- vi) take the square root.

E.g. In the following list the mean is 5

x: 1 2 3 4 5 6 7 8 9

$(x - \bar{x})$: -4 -3 -2 -1 0 1 2 3 4

$(x - \bar{x})^2$: 16 9 4 1 0 1 4 9 16

$\sum(x - \bar{x})^2 = 60$

s.d. = $\sqrt{\frac{60}{9}} = 2.58$.

Sets

A set is a collection of **members** (or **elements**).

For example:

- i) $V = \{\text{vowels of the English language}\}$ or $V = \{a, e, i, o, u\}$ - note the use of the {...} braces.
- ii) $A = \{\text{the positive odd numbers} < 10\}$ or $A = \{1, 3, 5, 7, 9\}$
- iii) the whole numbers between 1 and 20 which are multiples of 3 is $\{3, 6, 9, 12, 15, 18\}$
- iv) whole numbers > 10 ,
 $\{11, 12, 13, 14, 15, \dots\}$ or $\{\text{whole numbers} > 10\}$
- iv) The **null** or **empty** set, eg months with more than 32 days. $\emptyset = \{ \}$ -pronounces 'Phi'
- v) **Subsets** are sets that are part of a larger set, eg:
 $A = \{\text{JAN, FEB, MAR, APR, MAY, ... DEC}\}$; $B = \{\text{JAN, APR, JULY, OCT}\}$
B is a subset of A because every member of set B is also a member of set A.
If $A = \{3, 4, 5, 6\}$ then what are the possible subsets of A?
They are $\{3, 4, 5\}$, $\{3, 4, 6\}$, $\{4, 5, 6\}$, $\{3, 4\}$, $\{4, 5\}$, $\{5, 6\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{6\}$, ... and so on.
They are known as **proper** subsets.
- vi) The empty set, $\emptyset$ and the set itself $\{3, 4, 5, 6\}$ are also subsets but not proper subsets.
- vii) The order of listing the elements is not important, e.g. $\{1, 2, 3, 4, 5\} = \{5, 4, 1, 2, 3\}$

Some notation:

$A \subset B$. B is a subset of A (hint B is contained in A)

$x \in A$ x is an element of A

$D \not\subset B$ not contained in.

$y \notin B$ y is not an element of B .

The Universal Set is the set of all the elements of the problem context. Eg the set of even numbers and the set of odd numbers are both subsets of the universal set of whole numbers.

Union

Symbol:- $\cup$ (hint: think 'United' with)

$A = \{1,3,5,7,9,11\}$, $B = \{3,6,9,12\}$

$A \cup B = \{1,3,5,6,7,9,11,12\}$

Note: 3 and 9 occur just once in $A \cup B$.

Union can be likened to an OR operation.

Intersection:

Symbol:- $\cap$

$A \cap B$.

The set of all elements which are members of both A and B

If $A = \{1,3,5,7,9,11\}$ and $B = \{3,6,9,12\}$ then

$A \cap B = \{3,9\}$

Note: Intersection can be likened to the AND operation.

Complement

The set of elements which are not members of A .

Symbol: A'

Eg If the Universal set for a problem is $\{1,2,3,4,5,6\}$. If $A = \{1,3,5\}$ then A' is $\{2,4,6\}$.

Difference

$A - B$

$A = \{1,2,3,4,5,6\}$; $B = \{3,5,7,9\}$

$A - B = \{1,2,4,6\}$

Venn Diagrams

Venn diagrams illustrate set operations.

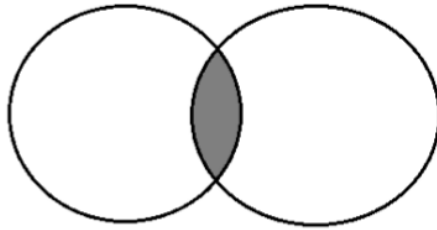


Figure 1: Intersection [1]

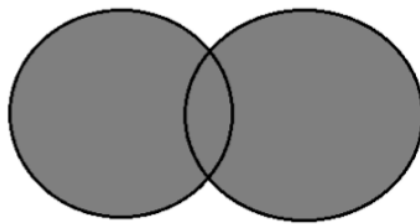


Figure 2: Union

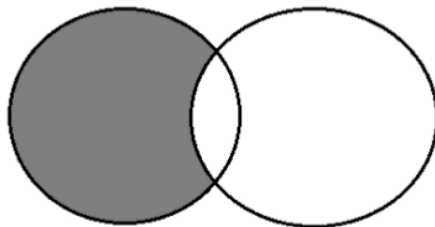


Figure 3: Difference

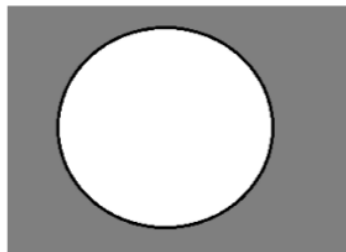


Figure 4: Complement

Matrix Operations

The determinant of a matrix A is given by $|A| = x(a_2b_3 - a_3b_2) - y(a_1b_3 - a_3b_1) + z(a_1b_2 - a_2b_1)$,

Where the matrix A =

$$\begin{pmatrix} x & y & z \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix}$$

Multiplication of a (4x4) matrix by another (4x4) is:-

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{pmatrix} =$$

$$\begin{bmatrix} (a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} + a_{14} * b_{41}) & (a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32} + a_{14} * b_{42}) & (a_{11} * b_{13} + a_{12} * b_{23} + a_{13} * b_{33} + a_{14} * b_{43}) & (a_{11} * b_{14} + a_{12} * b_{24} + a_{13} * b_{34} + a_{14} * b_{44}) \\ (a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} + a_{24} * b_{41}) & (a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32} + a_{24} * b_{42}) & (a_{21} * b_{13} + a_{22} * b_{23} + a_{23} * b_{33} + a_{24} * b_{43}) & (a_{21} * b_{14} + a_{22} * b_{24} + a_{23} * b_{34} + a_{24} * b_{44}) \\ (a_{31} * b_{11} + a_{32} * b_{21} + a_{33} * b_{31} + a_{34} * b_{41}) & (a_{31} * b_{12} + a_{32} * b_{22} + a_{33} * b_{32} + a_{34} * b_{42}) & (a_{31} * b_{13} + a_{32} * b_{23} + a_{33} * b_{33} + a_{34} * b_{43}) & (a_{31} * b_{14} + a_{32} * b_{24} + a_{33} * b_{34} + a_{34} * b_{44}) \\ (a_{41} * b_{11} + a_{42} * b_{21} + a_{43} * b_{31} + a_{44} * b_{41}) & (a_{41} * b_{12} + a_{42} * b_{22} + a_{43} * b_{32} + a_{44} * b_{42}) & (a_{41} * b_{13} + a_{42} * b_{23} + a_{43} * b_{33} + a_{44} * b_{43}) & (a_{41} * b_{14} + a_{42} * b_{24} + a_{43} * b_{34} + a_{44} * b_{44}) \end{bmatrix}$$

Multiplication of a (4x4) matrix by a (4x1) matrix is:

$$\begin{array}{cccccc} a_{11} & a_{12} & a_{13} & a_{14} & b_{11} & (a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} + a_{14} * b_{41}) \\ a_{21} & a_{22} & a_{23} & a_{24} & b_{21} & = (a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} + a_{24} * b_{41}) \\ a_{31} & a_{32} & a_{33} & a_{34} & b_{31} & (a_{31} * b_{11} + a_{32} * b_{21} + a_{33} * b_{31} + a_{34} * b_{41}) \\ a_{41} & a_{42} & a_{43} & a_{44} & b_{41} & (a_{41} * b_{11} + a_{42} * b_{21} + a_{43} * b_{31} + a_{44} * b_{41}) \end{array}$$

Transformations (right handed system and points represented by a column vector):

Transformation about the x-axis:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformation about the y-axis:

$$\begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformation about the z-axis:

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Translation:

$$\begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scaling:

$$\begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Points and Vectors

Recall we have represented a vector, $\mathbf{v}$ as a linear combination of three perpendicular unit vectors $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ where $\mathbf{i} = (1,0,0)$, $\mathbf{j} = (0,1,0)$ and $\mathbf{k} = (0,0,1)$ such that

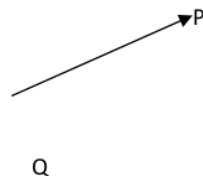
$\mathbf{v} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ where x , y and z are general coefficients ($\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are called basis vectors). Since $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are fixed and do not change, we can write this combination in a short hand way by representing $\mathbf{v}$ as an ordered triple, i.e. (x,y,z) . We can think of x , y and z as the amount we move in the $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ directions from the tail of the vector to its tip and assume the tail of the vector is at position $(0,0,0)$ in our 3D vector space.

Points

We can also represent points as a real number triple (x,y,z) representing a position in Cartesian space. Note that whilst vectors have no position but only length and direction, a point represents a position, has zero length and direction is not defined.

Recall, we can create a relationship between vectors and points by specifying a vector, $\mathbf{v}$ as the difference between a pair of two points. For example the points $P = (8,5,1)$ and $Q = (3,2,1)$ can be used to represent the vector $\mathbf{v}$ as $\mathbf{v} = P - Q$, i.e.

$\mathbf{v} = (8-3, 5-2, 1-1) = (5,3,0)$. The direction is indicated in the diagram below. Equally we can see that $P = \mathbf{v} + Q$. We can think of this as a displacement of point Q by a vector $\mathbf{v}$. So, to summarize, we can displace a point by adding a vector to it and the result is another point; also the difference of two points results in a vector.



Vectors are often represented using homogenous co-ordinates (x,y,z,w) with $w = 0$ and points are represented again using homogenous co-ordinates but this time w is set to 1. This is handy because it allows us to add and subtract vectors and points and obtain consistent results, for example:

If $P = (8,5,1,1)$ and $Q = (3,2,1,1)$

$P - Q = (8-3, 5-2, 1-1, 1-1) = (5,3,0,0)$ $\mathbf{v}$ (a vector. Note $w = 0$)

Also $\mathbf{v} + Q = (5+3, 3+2, 0+1, 0+1) = (8,5,1,1) = P$ (a point. Note $w = 1$)

Polar and spherical co-ordinates.

There are many ways to represent points. We met Cartesian co-ordinates but we can also use polar co-ordinates in a two dimensional framework. A point is now represented by a pair of values (r, θ) . The relationship between polar and rectangular co-ordinates is now:

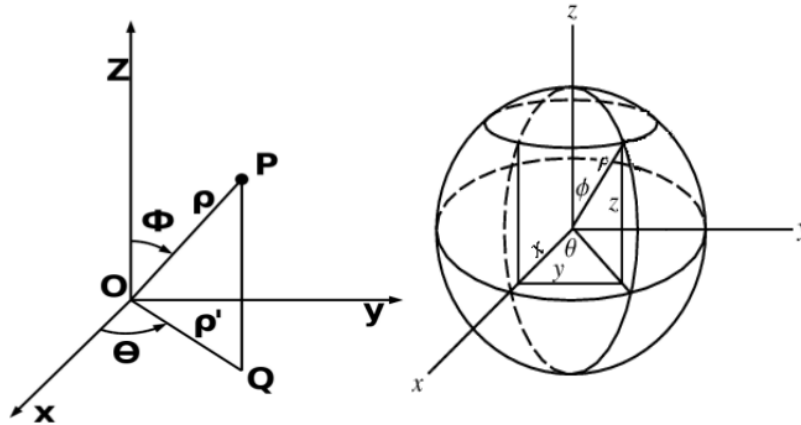
$$x = r \cos \theta$$

$$y = r \sin \theta$$

The following diagram illustrates the idea.

This idea can be extended to three dimensions and is referred to as spherical co-ordinates.

We create a sphere of radius ρ centred on the origin and define where the point lies on the sphere by using two angles θ and ϕ as shown in the diagram below.



To convert from spherical (ρ, θ, ϕ) to Cartesian (x, y, z) co-ordinates we use the relationships:

$$x = \rho \sin \phi \cos \theta$$

$$y = \rho \sin \phi \sin \theta$$

$$z = \rho \cos \phi$$

Conversely:

$$\rho = \sqrt{x^2 + y^2 + z^2}$$

$$\phi = \cos^{-1}(z/\rho)$$

$$\theta = \arctan(y/x)$$

Lines

Lines have infinite length. If a line has a start point and an end point it is called a line segment; a start, a direction but no end then it's called a ray. We know a line can be represented by the formula

$$y = mx + c$$

where m is the gradient and c is the y-intercept. This representation has problems for us (e.g how do we represent a vertical line since the gradient is infinite?)

Parametric representation of lines can solve some problems for us. Instead of using a single formula to explicitly represent the relationship between x and y , we represent the line using two separate functions, one for x and one for y . We use a parameter, t to represent a common variable and usually restrict it to a range from which we can calculate appropriate values for x and y .

For example the parametric equation for a line is given by one of the following forms:

$$\begin{aligned} L(t) &= P_0 + t(P_1 - P_0) \\ &= (1-t)P_0 + tP_1 \\ &= P_0 + t\mathbf{v} \quad (\mathbf{v} \text{ is a vector}) \end{aligned}$$

t is the parameter. We set t to be a series of values and calculate x and y for each value of t .

For example:

$P_0 = (0,1,1)$ and $\mathbf{v} = (1,2,3,0)$

t	$t\mathbf{v}_x$	$t\mathbf{v}_y$	$t\mathbf{v}_z$	L_x	L_y	L_z
0	0	0	0	0	1	1
0.2	0.2	0.4	0.6	0.2	1.4	1.6
0.4	0.4	0.8	1.2	0.4	1.8	2.2
0.6	0.6	1.2	1.8	0.6	2.2	2.8
0.8	0.8	1.6	2.4	0.8	2.6	3.4
1.0	1.0	2.0	3.0	1.0	3.0	4.0

Planes

A plane is a two dimensional, flat infinitely thin surface. Parametric representation of a plane might look like:

$$P(s,t) = (1-s-t)P_0 + sP_1 + tP_2$$

Or we could represent it as

$$P(s,t) = P_0 + s(P_1 - P_0) + t(P_2 - P_0)$$

which is an origin point plus the linear combination of two vectors, i.e:

$$= P_0 + s\mathbf{u} + t\mathbf{v}$$

Various other forms exist, e.g.

$$a(x-x_0) + b(y-y_0) + c(z-z_0) = 0$$

which can be rewritten as:

$$0 = ax + by + cz + d \quad \text{where } d = -(ax_0 + by_0 + cz_0)$$

Dot Product and Normals to Lines and Planes

The equation of a line in the form $ax + by = c$ can be written as a dot product:

$$(a,b) \cdot (x,y) = c, \text{ or } \mathbf{a} \cdot \mathbf{x} = c,$$

where $\mathbf{a} = (a, b)$ and $\mathbf{x} = (x,y)$.

The equation of a plane the form $ax + by + cz = d$ can be written as a dot product:

$$(a,b,c) \cdot (x,y,z) = d, \text{ or } \mathbf{a} \cdot \mathbf{x} = d,$$

where $\mathbf{a} = (a, b, c)$ and $\mathbf{x} = (x,y, z)$.

Normal Vector $\mathbf{a}$

If points P and Q are in the plane with equation $\mathbf{a} \cdot \mathbf{x} = d$, then $\mathbf{a} \cdot \mathbf{p} = d$ and $\mathbf{a} \cdot \mathbf{q} = d$, so

$$\mathbf{a} \cdot (\mathbf{q}-\mathbf{p}) = d - d = 0.$$

This means that the vector $\mathbf{a}$ is orthogonal to any vector PQ between points P and Q of the plane. If $\mathbf{a}$ is normalised then the result of the dot product gives the shortest distance from the point to the plane (positive then “above” the plane, negative then “below” the plane). See Van Verth and Bishop for more details.

Vector Operations

The length of a vector is found by using Pythagoras' theorem. The length of a vector **a** is written using the notation $|\mathbf{a}|$. The formula for the length of a 2 dimensional vector is $|\mathbf{a}| = \sqrt{x^2 + y^2}$ and in three dimensions it is $|\mathbf{a}| = \sqrt{x^2 + y^2 + z^2}$.

A vector can be normalised (its length made equal to one) by dividing it by its length. The direction remains the same, of course. The notation is $\hat{a} = \frac{\mathbf{a}}{|\mathbf{a}|}$. For example, if vector **a** is (0,3,4) it's length is 5 and the normalised vector, $\hat{a} = (0/5, 3/5, 4/5)$. Now check the length is really equal to 1.

Dot product

Another common vector operation is the scalar product, usually called the dot product and written $\mathbf{a} \bullet \mathbf{b}$.

It is defined thus:

If **a** = (x1, y1, z1) and **b** = (x2, y2, z2) then $\mathbf{a} \bullet \mathbf{b} = (x1 \cdot x2 + y1 \cdot y2 + z1 \cdot z2)$.

It is also equivalent to

$\mathbf{a} \bullet \mathbf{b} = |\mathbf{a}| \cdot |\mathbf{b}| \cdot \cos\phi$ where ϕ is the angle between the vectors.

$\mathbf{a} \bullet \mathbf{b}$ is a **scalar**.

If **a** and **b** are normalised then $\mathbf{a} \bullet \mathbf{b} = \cos\phi$. Therefore we can use the dot product to find the angle between two vectors.

If $\mathbf{a} \bullet \mathbf{b} = 0$ then $\phi = 90$ degrees

If $\mathbf{a} \bullet \mathbf{b} = 1$ then $\phi = 0$ degrees

If $\mathbf{a} \bullet \mathbf{b}$ is +ve then $-90 < \phi \text{ degrees} < +90$ degrees.

Cross product

The cross product of two vectors is written

$$\mathbf{a} \times \mathbf{b} = (a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1)$$

- returns vector perpendicular to **a** and **b**
- use right hand rule to determine direction of vector (if using a right handed system – left otherwise).
- length = area of parallelogram mapped out by the two vectors **a** and **b**.

A handy way of remembering the cross product is to write it as a matrix and to see it as a determinant (don't forget to negate the second term!), that is:

i	j	k
a_1	a_2	a_3
b_1	b_2	b_3

Remember, i , j and k are the three orthogonal unit vectors in the x , y and z directions.

The result of a cross product is a vector.

The length of the vector cross product can be determined by:

$$|a \times b| = |a| \cdot |b| \cdot \sin \phi \text{ where } \phi \text{ is the angle between } \mathbf{a} \text{ and } \mathbf{b}.$$

The cross product of two vectors results in another vector which is perpendicular to the two vectors.
The length of this vector equals the area of the parallelogram formed from the two vectors.

FINAL GRADE

GENERAL COMMENTS

Instructor

81/100

PAGE 1

PAGE 2

PAGE 3



Comment 1

The distance is the length of this vector - pathag will give you it $\sqrt{25+9}$

PAGE 4

PAGE 5

PAGE 6



Comment 2

A rotation around the y axis will not change the value of y. Error in choice of rotation here.

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

GRADING FORM: SMU GENERIC_GJ

ALEX
THOMAS

81

RESEARCH-INFORMED LITERATURE

 You could have included one or two more refs and avoid placing class notes at the end of the report.

3

KNOWLEDGE AND UNDERSTANDING

 Good understanding demonstrated but rotation is .faulty. Wrong rotation used.

16

ANALYSIS

 Maths examples - methods and calculations provided. Error in rotation formulae. Test design appropriate. could be more detailed in some places.

17

PRACTICAL APPLICATION AND DEP.

 Rotation incorrect formula. Testing provided. some tests could be more thorough.

42

SKILLS FOR PROFESSIONAL PRAC.



No need to include class notes at the end. Just ref them. Structured report.