Decrease & Conquer Algorithm to Determine the Center of Mass of a Rigid Body Composition

Destra Bintang Perkasa 13511057
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13511057@std.stei.itb.ac.id

***Abstract*—This manuscript contains a research in applying Decrease and Conquer Algorithm for determining the center of mass of a system of rigid body. This includes many theories of center of mass and the Decrease and Conquer Algorithm which is described in simple words. This also contains an implementation in a simple program which results a true value. In fact this new, yet simple method also finding a solution in a pretty quick time and defining a solution in a way human thinks regularly.**

***Index Terms*—Algorithm, rigid body, Decrease and Conquer, Center of mass.**

I. INTRODUCTION

Nowadays, informatics has been applied in many disciplines. Bioinformatics is one example of a branch of science that combines informatics with other disciplines, namely biology. This means that it is not impossible to develop informatics with many other disciplines in the world. One that will be carried in this manuscript is the application of Decrease and Conquer algorithm (one of the existing algorithms in computer science) in physics, namely the determination of the center of mass of a system consisting of particles rigid body.

The center of mass of the object is used in some tools to make an object balanced if put / placed on a shaft. For instance, the orbit of the satellite should be positioned at the center of mass of the satellite so that the torque and force on the satellite is not made satellite deviated from its orbit.

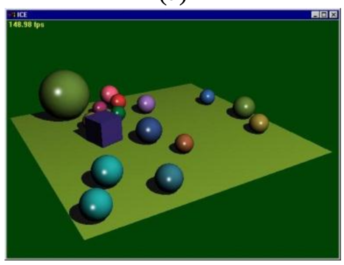
There are so many formulas to calculate of the center of mass. In calculus, the center of mass calculation is done based on the density of objects in two dimensions and position. In physics, the center of mass calculation is done based on the mass and position. It can be seen that the existing calculations always take into account the position of the object. In fact it is very difficult to determine the position of the plane so that objects get this "position". The position must have a reference, and this reference must be absolute. There must be a method to make this system has the center of mass relative to itself. Therefore, this paper is expected to be a new way to determine the center of mass is more applicable to the real world.

II. BASIC THEORIES

A. System of Rigid Body

Rigid Body is a phrase to describe an object that can be represented by a single point. In other word , the distance between any two given points of a rigid body remains constant in time Regardless of external forces exerted on it. Even though such an object cannot physically exist due to relativity, objects can normally be assumed to be perfectly rigid if they are not moving near the speed of light. For example, imagine a GPS. Whatever the actual shape - could be people, cars , or even a very large aircraft - the GPS will always represent it as a point on the screen. The point is basically the position where the GPS device is located, not the car and not the aircraft is located. The same thing applies to the rigid body. An object, the object of any kind , if it is a rigid body , then the object can be represented by a single point . The difference, a GPS device on a rigid body is the center of mass. Funny thing is, with this kind of theory , there could be an object that is outside the center of mass of the object itself .

(a)



(b)

Fig. 1 Analogy of GPS and rigid Body

B. Decrease and Conquer

Decrease and Conquer is an algorithm derived from Divide and Conquer. In computer science, divide and conquer (D&C) is an important algorithm design paradigm based on multi-branched recursion. A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem. The most fundamental difference between the Divide and Conquer and Decrease and Conquer is that the phase “divide” is not only divides the problem, but also eliminates candidate solutions that may not be the solution.

C. Center of Mass

Center of mass is the average location of all the mass in the system. In the case of a rigid body, the location of the center of mass is fixed in relation to the body object. In the case of a loose distribution of masses in free space, such as a bullet shot from a rifle or the planets in the solar system, the location of the center of mass is a point in space among them that may not be related to any position on the mass of the object. The use of the mass center often allows the use of simplified equations of motion, and it is a convenient reference for many other calculations in physics, such as angular momentum or moment of inertia. In many applications, such as in orbital mechanics, objects can be replaced by a mass of dots located in the center of their mass in order to facilitate the analysis.

The term center of mass is often equated with the term center of gravity; however, they are physically different concepts. The layout is coincident in both cases the same gravitational field, but when gravity is not the same, the center of gravity refers to the average location of the gravitational force acting on an object . This produces a gravitational torque, which is a small but measurable and should be taken into account in the operation of artificial satellites.

As we discussed in the previous section, the position and the property of the rigid body is determined by its center of mass. From several disciplines, we have obtained the formula for calculating the center of mass. For instance in the calculus, the center of mass of an object can be calculated by the system as seen from the one-dimensional, two-dimensional, and even 3-dimensional. In this paper, we will focus on the center of mass of the system as seen from the continuous mass distribution along the line.

Consider now a straight segment of thin wire of varying density (mass per unit length) for which we desire to find the balance point. We impose a coordinate line along the wire and follow our usual procedure of slice, approximate, and integrate. Supposing that the density at x is $\delta(x)$, we first obtain the total mass m and then the total moment M with respect to the origin like in Fig.2 below.

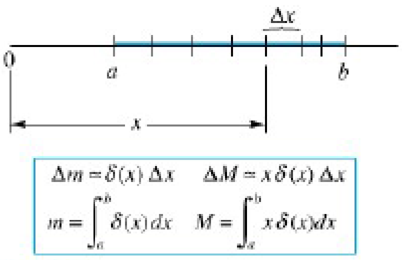


Fig.2 Illustration of continuous mass distribution along a line

If we combine this with a fact that the center of mass is total moment per mass, this leads to the formula

$$x = \frac{M}{m} = \frac{\int_a^b x\delta(x)\,dx}{\int_a^b \delta(x)\,dx}$$

Mass Distribution in The Plane.

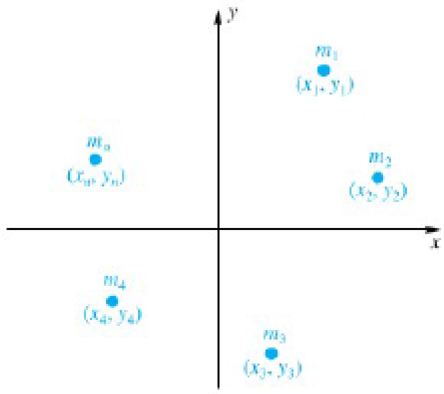


Fig.3 Center of mass in plane

Consider n point masses of sizes m_1, m_2, m_3, ..., m_n situated at points (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) in the coordinate plane like in the Fig.3. Then the total moments M_y and M_x with respect to the y-axis and x-axis, respectively are given by:

$$M_y = \sum_{i=1}^{n} x_i m_i \qquad M_x = \sum_{i=1}^{n} y_i m_i$$

The coordinates (x, y) of the center of mass (balances point) are:

$$\bar{x} = \frac{M_y}{m} = \frac{\sum_{i=1}^{n} x_i m_i}{\sum_{i=1}^{n} m_i} \qquad \bar{y} = \frac{M_x}{m} = \frac{\sum_{i=1}^{n} y_i m_i}{\sum_{i=1}^{n} m_i}$$

Not much different from the calculus, physics also defines the center of mass of the object based on homogeneity. The difference, homogeneity objects described in physics refers to the homogeneity of discrete while calculus looking at it as something more general, namely a continuous object with a mass of different types for each point x in the object.

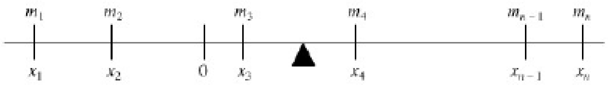


Fig.4 Illustration of discrete mass distribution

In this manuscript, we will review the more general case, with a system or object that is composed of many particles (point particles) as well as objects composed of particles that are considered dispersed continuously on objects or often known by the name of the *particle system*. Particle system can be illustrated as Fig.4 above. In a system of particles of known mass center point is defined as:

$$r_{pm} = \frac{\sum_i (m_i r_i)}{M}$$

with r_i is the position of the i-th particle in the system, m_i is the mass of the i-th particle, and M is the total mass of the system.

III. THE NEW METHOD

As already explained, the current method requires us to set an origin as a benchmark. In fact, it would be difficult to determine the origin of an object when the object is too big or too small. By combining all the theories that have been discussed in the previous section, we can create a new method for determining the center of mass by using the Decrease and Conquer.

Imagine you have a system consisting of a rigid body. Then hang the object on a thread right in the middle.

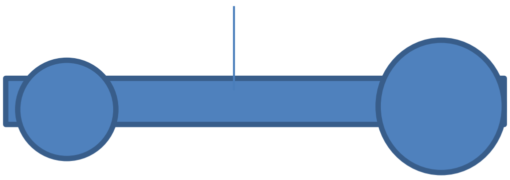


Fig.5 Initial condition

See if the object position leaning to the right or left-leaning. If inclines to the right, meaning that the heavier the object to the right. In other words, the center of mass of the object is located on the right of the thread.

Conversely, if the object is skewed to the left, meaning that the object is heavier on the left. In other words, its center of mass located on the left of the thread. Elimination lighter parts, the center of mass is definitely not there.

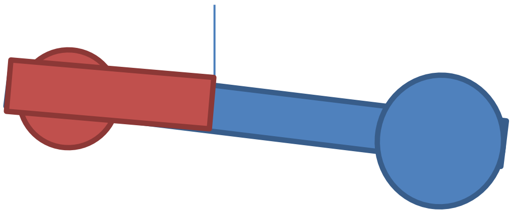


Fig.6 The center of mass is impossible to be in red region

Note Fig.6 for clearer illustration. Next, slide the thread direction to the center of mass available position. Slide the thread up to the middle of the blue area.

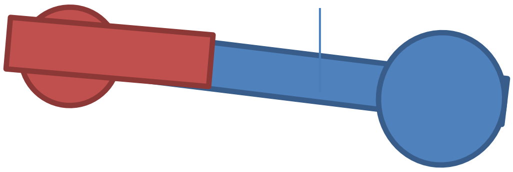


Fig.7 Second iteration. Move the thread to the center of the blue region

Check whether the rigid body is leaning to the right or to the left. Do it this way until the moment at which the thread will make the position of the rigid body equilibrium. If it meets these conditions, meaning that the position of the thread are now the center of mass of the rigid body.

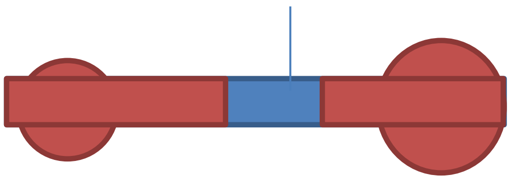


Fig.8 Balanced rigid body means the thread right on the center of mass.

IV. IMPLEMENTATION

From the theory already mentioned, we can implement it in a program code. But to make it all can be implemented; of course, there are some assumptions that must be met.

A. Critical Assumption

The program certainly cannot "see" if the rigid body is now in a state of left-leaning, right-leaning, or already balanced. Therefore, the fact of which was changed to the heavier torque. Torque is defined as the product of the mass with the distance from the thread. If the torque of the right yarn at the yarn from the left, meaning that the position of a rigid body are balanced.

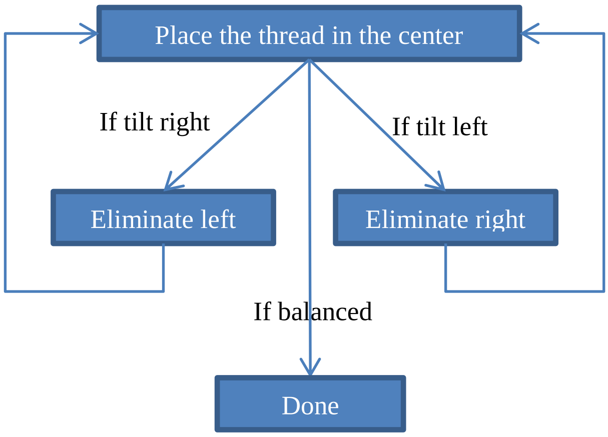
B. Specification

Implementation of programs created to do the following:

- Inputting particle mass
- Inputting particle positions
- Insert any number of existing particle
- Calculate the left and right torque for each stage
- Determine the center of mass of a rigid body.

Implementation of the program was made in a Java environment and can be run on a desktop.

C. Flow Diagram



D. Pseudo-code

The following pseudo-code assumes all particles have been inserted on the system. `Procedure DecreaseAndConquer ()` has the initial state that particles has been inserted and yet there is no thread, and the final state when the thread balance the left and right.

```
Procedure DecreaseAndConquer()
     Place thread in the center
     If (rightTorque = lefttorque)
            {done}
     Else
          If (rightTorque>leftTorque)
              Eliminate left
          Else
              Eliminate right
         Place thread in the center of
         the new area
```

E. Source Code

Source code ditulis dalam Java environtment. Berikut ini adalah header file yang digunakan pada source program.

File: RigidBody.java

```
public class RigidBody {
     private int mass;
     private int pos;
     public RigidBody(){
          mass = 0;
          pos = 0;
     }
     public RigidBody(int m, int p){
          mass = m;
          pos = p;
     }
     public int getMass(){
          return mass;
     }
     public int getPos(){
          return pos;
     }
     public void setMass(int m){
          mass = m;
     }
     public void setPos(int p){
          pos = p;
     }
}
```

File: Centerofmass.java

```
public class Centerofmass {
public static Vector<RigidBody>
sistem;
public static float Left;
public static float Right;
public static float Center;
public static float BatasRight;
public static float BatasLeft;

public static float RightTorque();
//RightTorque defines weight to the
right side of the thread

public static float LeftTorque();
//LeftTorque defines weight to the
left side of the thread

public static void
DecreaseAndConquer(){
  float r = RightTorque();
  float l = LeftTorque();
  if(r==l){
  //done
  }
  else{
    if(r > l){
      BatasLeft = Center;
      Center = (BatasLeft +
      BatasRight)/2;
    }
    else if (r < l){
      BatasRight = Center;
```

```
      Center = (BatasLeft +
       BatasRight)/2;
     }
    DecreaseAndConquer();
  }
}
public static void main(String[] args)
{
int inputMass, inputPos;
Left = 0;
Right = 9999;
Scanner sc = new Scanner(System.in);
inputMass = sc.nextInt();
while(inputMass != 9999){
   inputPos = sc.nextInt();
   if (inputPos>Left){
     Left = inputPos;
   }
   if (inputPos<Right){
     Right = inputPos;
   }
   sistem.add(new  RigidBody(inputMass,
inputPos));
  inputMass = sc.nextInt();
}
if(sistem.size()==0){
   Center = 0;
}
else{
   BatasLeft = Left;
   BatasRight = Right;
   Center = (Left+Right)/2;
   DecreaseAndConquer();
}
System.out.println(Center);
  }
}
```

V. RESULTS

A. *Inputs and Outputs*

The inputs of test cases are in the following format:

Each line represents a particle in a system particle. Each line consists of two non-negative integers: the first one is for the particle’s mass and the second one is for the particle’s position referring to the origin. The input was terminated by typing 9999 into the program.

For example:

2 0
4 100
9999

This example represents a system with 2 particles. One with mass 2 units in position 0, and another one with mass 4 units in position 100.

The output of the program is a floating number represents the distance of the center of mass from the origin.

Here are the results of the test cases run on the implemented program.

Table 1. Results of the test cases.

TEST CASE	**INPUTS**	**OUTPUT**
1	2 0 4 100 9999	66.66565
2	2 0 6 100 9999	75
3	8 5 7 10 4 25 6 90 9999	29.999619

As shown in the table above, the results are the same as we compute with the existing formulas. Therefore, this can be a new method to determine the center of mass of a system particle.

Author names and affiliations are to be centered beneath the title and printed in Times 11-point, non-boldface type. Multiple authors may be shown in a two- or three-column format, with their affiliations below their respective names. Affiliations are centered below each author name, italicized, not bold. Include e-mail addresses if possible. Follow the author information by three blank lines before main text.

The second and following pages should begin 2 cm from the top edge. On all pages, the bottom margin should be 2.7 cm from the bottom edge of the page for A4 paper.

Wherever Times is specified, Times Roman, or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times that you have access to. Please avoid using bit-mapped fonts if possible. True-Type 1 fonts are preferred. Type your main text in 10-point Times. Be sure your text is fully justified—that is, flush left and flush right. Please do not place any additional blank lines between paragraphs.

VI. CONCLUSION

Although it is pretty hard to implement it to the program, it is worth trying this new method as it returns a true value. Therefore, we can conclude that the center of mass can be calculated using the Decrease and Conquer Algorithm. Multipliers can be especially confusing. Write “Magnetization (kA/m)” or “Magnetization (10^3 A/m).” Do not write “Magnetization (A/m) × 1000” because the reader would not know whether the top axis label in Fig. 1 meant 16000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type.

REFERENCES

M. Harmon, et al., "Modul praktikum fisika dasar I," Laboratorium fisika Dasar ITB, 2011, pp. 18.

D. Halliday, R. Resnick, *Fundamental of Physics.* John Willey & Sons, 1997, pp. 123–135.

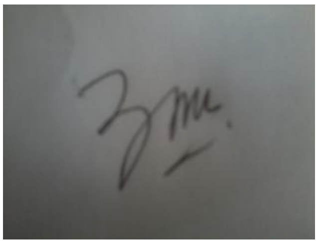
Purcell, *Calcukus 9th Edition*. New York: Springer-Verlag, 1985, ch. 4.

Goldstein, Herbert; Charles Poole, John Safko (2002). Classical Mechanics (ed. 3e). Addison Wesley.

STATEMENT

I hereby declare that this manuscript I wrote is my own writing, not adaptation, or translation of a paper someone else, and not plagiarism.

Bandung, December 20, 2013



Destra Bintang Perkasa 13511057