DETECTION OF COLLISIONS OF POLYGONS BY USING A TRIANGULATION

D. Müller, Th. M. Liebling

Département de Mathématiques
École Polytechnique Fédérale de Lausanne
CH-1015 Lausanne (Switzerland)

INTRODUCTION

This paper presents a sketch of a method useful in numerical simulations to efficiently manage motions and collisions of polygonal grains in the plane. This method uses a triangulation of the space between the polygons and it efficiently maintains its coherence in the course of time. This triangulation allows the calculation of exact times of collision and defines a neighborhood that improves simulation time. We use it to study flows of polygonal grains in a hopper and the phenomenon of segregation in granular media.

TRIANGULATION CONSTRUCTION

The triangulation is built in two stages: we first make a *left to right triangulation* that we then modify to obtain a *constrained triangulation*. For a while just keep vertices of polygons and forget polygons themselves. A simple way to triangulate this cluster of points is the following:

1. Sort points lexicographically and number them from 1 to n.
2. **For** i=2 **to** n **do**
   connect the point i with all points j<i if the segment ij does not cross any already constructed segment.

In this way we obtain a left to right triangulation. We want now to build a constrained triangulation where we impose that edges of polygons (which will from now on be called *mandatory edges*) be edges of the triangulation.

Starting from a left to right triangulation, we repeat the following operations for all mandatory edges a_k which are not in the triangulation:

a. Eliminate the edges crossed by a_k
b. Put a_k in the triangulation
c. Triangulate the inner of both induced polygons which adjoin a_k.

The following is a simple method for c :

1. Examine the vertices of the polygon counterclockwise and put them into a circular list L
2. Let: a:=beginning of the list
   b:=following(a)
   c:=following(b)
3. **If** the triangle abc is positively oriented **and if** ac is entirely in the polygon **then**
   - add segment ac
   - remove point b from list L
   - b:=c; c:=following(b)

   else
   - a:=b; b:=c; c:=following(b)
4. **While** following(c)≠a **go to** 3.

After removing edges inside grains, we obtain a triangulation a the space between the polygonal grains similar to the fig. 1.

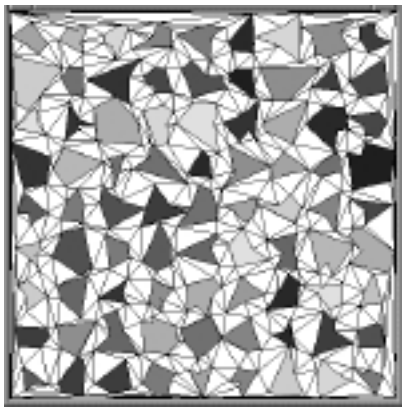

Figure 1. A triangulation of the space between the polygons

TRIANGULATION MAINTENANCE IN THE COURSE OF TIME

To illustrate this paragraph in concrete terms we will imagine that polygons are set on a vertical plane and subject to gravitational attraction. Because of the motion of polygons, the triangulation degenerates quickly if it is not maintained by topological operations. We show how to calculate when a triangle becomes flat (hence the time when the triangulation will degenerate) and how to maintain this triangulation coherent by local operations.

In principle, the motion without friction of an object has two aspects: the centre of gravity follows the law of a material point and thus describes a parable, while the objet itself rotates at a constant angular velocity about its centre of gravity. Formulas (1) and (2) are the equations of motion of the polygon centre of gravity.

$$x(t) = x(0) + \dot{x}(0)\ t + \frac{1}{2}\ddot{x}\ t^2 \qquad (1\ a)$$

$$y(t) = y(0) + \dot{y}(0)\ t + \frac{1}{2}\ddot{y}\ t^2 \qquad (1\ b)$$

$$\dot{x}(t) = \dot{x}(0)\ t + \ddot{x}\ t \qquad (2\ a)$$

$$\dot{y}(t) = \dot{y}(0)\ \ t + \ddot{y}\ \ t \qquad (2\ b)$$

Herein (x(t),y(t)) are the coordinates at time t, $(\dot{x}(t),\dot{y}(t))$ is the velocity and $(\ddot{x},\ddot{y})$ is the gravitational attraction. The formulas describing the position of a vertex v are the following:

$$x^v(t) = x(t) + r\ \ \cos(\omega t + \alpha) = x(t) + r_x\cos(\omega t) - r_y\sin(\omega t) \qquad (3\ a)$$

$$y^v(t) = y(t) + r\ \ sin(\omega t + \alpha) = y(t) + r_y\cos(\omega t) + r_x\sin(\omega t) \qquad (3\ b)$$

where ω is the rotational velocity, α the angle at time 0, r the distance between the centre of gravity and the vertex v, $r_x = r\ \cos(\alpha) = x - x^v$ and $r_y = r\ \sin(\alpha) = y - y^v$ (see fig. 2).

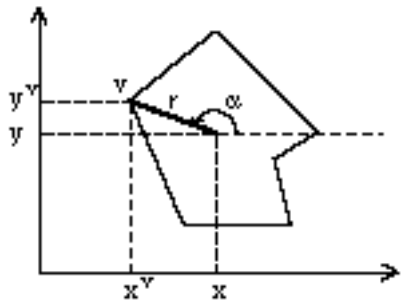

Figure 2. Nomenclature used for a polygon

Trigonometric functions have two drawbacks: computation is slow and relatively inaccurate; moreover finding the roots of a function including trigonometrical functions can be difficult. Instead we develop $\cos(\omega t+\alpha)$ and $\sin(\omega t+\alpha)$ into series of a few power terms by using Mac-Laurin formulas. The number of terms used (two terms for sinus and three for cosinus) allows a good approximation (the maximum error is 10^{-7}) for angles ranging from -0.1 to 0.1 radians. With the help of the Mac-Laurin formulas, we can rewrite equations (3 a) and (3 b) as follows:

$$x^v(t) = x(0) + r_x + (\dot{x}(0) - r_y\omega)t + \frac{1}{2}(\ddot{x} + r_x\omega^2)t^2 + \frac{1}{6}r_y\omega^3t^3 + \frac{1}{24}r_x\omega^4t^4 \qquad (4\ a)$$

$$y^v(t) = y(0) + r_y + (\dot{y}(0) - r_x\omega)t + \frac{1}{2}(\ddot{y} - r_y\omega^2)t^2 - \frac{1}{6}r_x\omega^3t^3 + \frac{1}{24}r_y\omega^4t^4 \qquad (4\ b)$$

The aspect of triangles evolves with time, since their vertices move according to (4). Now and then some triangles become flat (area = 0). Two cases are possible:

1. The longest edge of this triangle is not a mandatory edge. In this case there is no collision. We just need to flip the edge in order to still have a coherent triangulation (see fig. 3). This operation consists in exchanging both diagonals of a convex quadrilateral.
2. The longest edge of this triangle is a mandatory edge. There is a collision. We use formulas given by Wang & Mason (1992) to determine the new velocities of the grains.

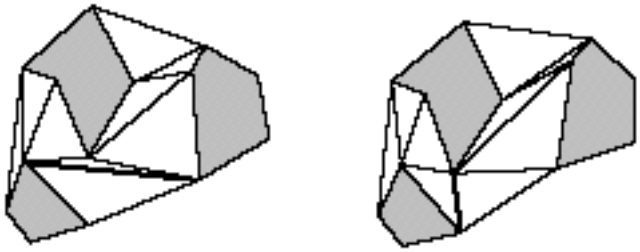

Figure 3. Triangulation maintenance when a polygon goes through a not mandatory edge

We can compute the time where a triangle with vertices **u**(t), **v**(t), **w**(t) will become flat by calculating the 2 by 2 determinant below (**u**(t), **v**(t), **w**(t) are bidimensional vectors):

$$\det(t) = \left| \mathbf{u}(t) - \mathbf{v}(t) \quad \mathbf{u}(t) - \mathbf{w}(t) \right| \tag{4}$$

and by solving the equation det(t)=0.

This determinant is a 8th degree polynomial whose roots we must extract. We have to cut time up into small *periods* of one thousandth of second in order to attain sufficient precision because of Mac-Laurin approximations. Actually only the smallest root in the interval $[t_1,t_2]$ interests us, where t_1 is the current time and t_2 the time of the period end. We find this root quickly by using *Sturm sequences* described by Ralston (1978) and a mixture between the Newton method (for rapidity) and the classical bisection method.

EXAMPLE OF APPLICATION

As an example of application, fig 4. presents a flow of polygonal grains in a hopper. We can study the influence of the shape and the matter of grains (characterized by coefficient of friction and restitution) on the flow. We can also modify the slope and the neck of the hopper. The computation time to empty the hopper is very depending on the parameters. It is contained between one and five hours. We work on a Silicon Graphic station with a 100 MHz processor.

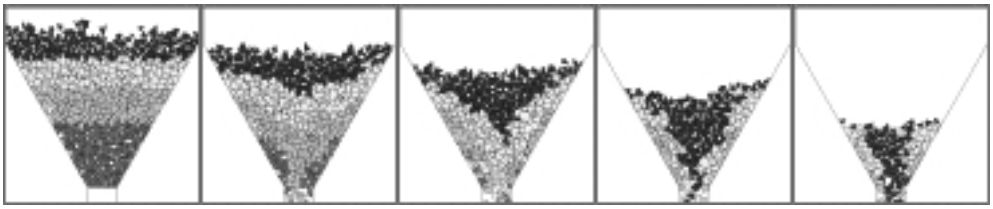

Figure 4. Flow in a hopper

REFERENCES

Ralston A., 1978, “A first course in numerical analysis”, McGraw-Hill, New York

Wang Y.,Mason M. T., 1967, Two-dimensional rigid-body collisions with friction, *Journal of Applied Mechanics* 59:635.