Quaternions to describe body motion

Berend van Wachem

Thermofluids Division, Department of Mechanical Engineering
Imperial College London
Exhibition Road, London, SW7 2AZ, United Kingdom
b.van-wachem@imperial.ac.uk

Training School "*Behaviour of Non-Spherical Particles in Flows*"
COST FP1005

Imperial College London

Outline

Imperial College London

Introduction

For spherical particles we can "simply" solve for each particle

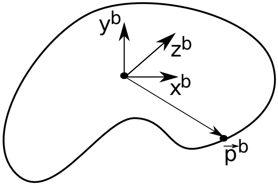
$$\boldsymbol{Y}(t) = \begin{pmatrix} \boldsymbol{x}(t) \\ \boldsymbol{v}(t) \end{pmatrix}$$

and solve for

$$\frac{d\boldsymbol{Y}(t)}{dt} = \begin{pmatrix} \boldsymbol{v}(t) \\ \frac{\boldsymbol{F}}{m}(t) \end{pmatrix}$$

But for a non-spherical particle, this does not suffice.

Imperial College London



Next to the position and velocity we also need to keep track of the orientation.

Rigid body dynamics

Angular momentum

$$\boldsymbol{L} = \bar{\bar{I}}\omega$$

External torque

$$\tau = \dot{\boldsymbol{L}}$$

Imperial College London

Derivative of angular momentum

$$\dot{\boldsymbol{L}} = \dot{\bar{\bar{I}}}\omega + \bar{\bar{I}}\dot{\omega}$$

Angular acceleration

$$\dot{\omega} = \bar{\bar{I}}^{-1}(\tau - \omega \times \bar{\bar{I}}\omega)$$

Imperial College London

Summarizing so-far:

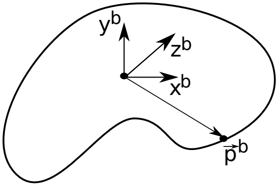
1. Find the moment of inertia, $\bar{\bar{I}}$ and its change, $\dot{\bar{\bar{I}}}$
2. Find the "external" torque, τ
    - Fluid interactions.
    - Particle-particle interactions.
3. Determine the angular acceleration, $\dot{\omega}$.
4. Determine the angular momentum, $\boldsymbol{L}$.
5. We can then accurately determine the angular velocity, ω and use this to determine the "orientation".

How can we "map" point p?

$$\boldsymbol{p}(t) = \overline{\overline{R}}(t)\boldsymbol{p}_0 + \boldsymbol{x}(t)$$

where $\overline{\overline{R}}$ is the *rotation matrix*,

$$\overline{\overline{R}} = \begin{pmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{pmatrix}$$



If we just consider the rotation of the x-axis,

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot \overline{\overline{R}} = \begin{pmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{pmatrix} = \boldsymbol{r}$$

Imperial College London

ORIENTATION: THE ROTATION MATRIX

Then the *change* of this axis is given by the angular velocity,

$$\dot{\boldsymbol{r}} = \omega(t) \times \boldsymbol{r}(t)$$

and this can be done for the rotation of every axis,

$$\dot{\overline{\overline{R}}} = \left(\omega(t) \times \begin{pmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{pmatrix} \ \omega(t) \times \begin{pmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{pmatrix} \ \omega(t) \times \begin{pmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{pmatrix} \right)$$

Imperial College London

Orientation: the rotation matrix

And after some steps we get

$$\dot{\overline{\overline{R}}} = \omega(t) \star \overline{\overline{R}}$$

$\longrightarrow$ so the *change* in *rotation matrix* is directly related to the angular velocity!

Imperial College London

So we can now define a complete state vector

$$\overline{\overline{Y}}(t) = \begin{pmatrix} \boldsymbol{x}(t) \\ \overline{\overline{R}}(t) \\ m\boldsymbol{v} \\ \boldsymbol{L}(t) \end{pmatrix}$$

and solve for

$$\frac{d\overline{\overline{Y}}(t)}{dt} = \begin{pmatrix} \boldsymbol{v}(t) \\ \omega(t) \star \overline{\overline{R}}(t) \\ \boldsymbol{F}(t) \\ \tau(t) \end{pmatrix}$$

Imperial College London

Hence, we need to determine

1. The mass of the body and the center of mass
2. The inertia tensor of the body *and its change in time*
3. The flow induced forces and torques on the body.
4. The collisional torques and forces on the body.

and then solve for Y.

Center of mass

Mass of a body

$$M = \sum_{i=1}^{N} m_i$$

Center of mass of a body

$$\boldsymbol{x}_c = \sum_{i=1}^{N} \frac{m_i \boldsymbol{r}_i}{M}$$

Imperial College London

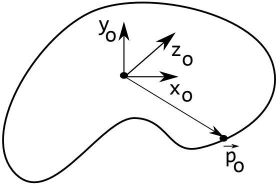
Determining the inertia tensor

Inertia tensor

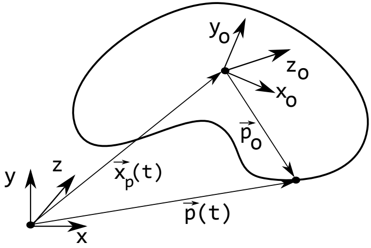
$$\bar{\bar{I}} = \sum_{i=1}^{N} \begin{pmatrix} m_i \left(r_{iy}^{\prime 2} + r_{iz}^{\prime 2} \right) & -m_i r_{ix}^{\prime} r_{iy}^{\prime} & -m_i r_{ix}^{\prime} r_{iz}^{\prime} \\ -m_i r_{iy}^{\prime} r_{ix}^{\prime} & m_i \left(r_{ix}^{\prime 2} + r_{iz}^{\prime 2} \right) & -m_i r_{iy}^{\prime} r_{iz}^{\prime} \\ -m_i r_{iz}^{\prime} r_{ix}^{\prime} & -m_i r_{iz}^{\prime} r_{iy}^{\prime} & \left(r_{ix}^{\prime 2} + r_{iy}^{\prime 2} \right) \end{pmatrix}$$

The summation over i can be done with, *e.g.* a Monte-Carlo type simulation.

Imperial College London



(a) body space



(b) world space

$\longrightarrow$ the inertia tensor of a rigid body *in body space*, $\overline{\overline{I}}^b$, does not change!

Imperial College London

Body space and world space

The rotation matrix can be used to transform variables in the world space (the *initial* space) to body space,

$$\boldsymbol{v}^b = \overline{\overline{R}}\boldsymbol{v}$$

for a vector. For a tensor:

$$\overline{\overline{T}}^b = \overline{\overline{R}}\,\overline{\overline{T}}\,\overline{\overline{R}}^T$$

Hence,

$$\overline{\overline{I}}^b = \overline{\overline{R}}\,\overline{\overline{I}}\,\overline{\overline{R}}^T$$

(but this is not-so-cheap to do)

Imperial College
London

So everything is fine?

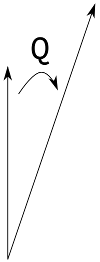
- When integrating the rotation matrix, very often *singularity* problems or *Gimbal lock* problems arise. This is because a rotation matrix is *over specified*.
- There are 3 constraints for orthogonality and 3 constraints for unit length. However, a rotation matrix has 9 components.

Imperial College London

In the 19^{th} century, Sir Hamilton proposed a *Quaternion*, consisting out of 4 numbers:

- A Quaternion is a mathematical concept to represent the relationship between two vectors.
- A Quaternion is an operator that changes the orientation and the length of a vector.
- A Quaternion contains a real and 3 imaginary parts:

$$Q = q_0 + q_1 \boldsymbol{i} + q_2 \boldsymbol{j} + q_3 \boldsymbol{k} = [q_0, \boldsymbol{q}]$$



Hamilton, W. R. (1844). On quaternions; or on a new system of imaginaries in Algebra.

Quaternions

The subsequent application of two Quaternions is given by the *Grassman* product:

$$AB = (A_a B_a - \boldsymbol{A} \cdot \boldsymbol{B}) + A_a \boldsymbol{B} + B_a \boldsymbol{A} + \boldsymbol{A} \times \boldsymbol{B}$$

The *conjugate* of a Quaternion is defined by

$$Q^* = q_0 - q_1 \boldsymbol{i} - q_2 \boldsymbol{j} - q_3 \boldsymbol{k}$$

and the *norm* of a Quaternion is defined by

$$||Q|| = \sqrt{QQ^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$$

and the *inverse* of a Quaternion is defined by

$$Q^{-1} = \frac{Q^*}{||Q||}$$

Unit Quaternion: $||Q|| = 1$: rotation only!

Imperial College London

Rotation of a vector with a *unit* Quaternion is then defined by

$$\boldsymbol{s}^b = Q\boldsymbol{S}Q^{-1}$$

where the vector $\boldsymbol{s}$ is interpreted as a Quaternion as

$$\boldsymbol{S} = [0, \boldsymbol{s}]$$

Obviously, a unit Quaternion and a rotation matrix are related,

$$\overline{\overline{R}} = \begin{pmatrix} 1-2(q_2^2+q_3^2) & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3 \\ 2q_1q_2+2q_0q_3 & 1-2(q_1^2+q_3^2) & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_0q_1+2q_2q_3 & 1-2(q_1^2+q_2^2) \end{pmatrix}$$

and most codes use a combination of Quaternions and Rotation matrices. Time integration is usually done in the framework of the Quaternion.

Differentiating a Quaternion

The change in time of a Quaternion is given by the “simple” relationship,

$$\dot{Q} = \frac{1}{2}\omega Q$$

which can be proved in several ways.

Quaternions: Intergration

- It is important that the length of a Quaternion remains unity
- This should be implicitly respected by the integration algorithm
- Commonly "adding" 2 unit Quaternions is ill defined: the result will not be a unit Quaternion.
- Hence, writing the following might be appealing:

$$Q_{n+1} = Q_n + \dot{Q}_n \Delta t + \dots$$

but it is *not* correct.

Zhao, F., and van Wachem, B. G. M. (2013). A novel Quaternion integration approach for describing the behaviour of non-spherical particles. Acta Mechanica, 224

Imperial College London

Instead, the Grassman product should be used,

$$Q_{n+1} = \tilde{Q}\, Q_n$$

where $\tilde{Q}$ represents the effect of rotation during the current time-step.

State vector with Quaternions

So we can now define a complete state vector

$$\overline{\overline{Y}}(t) = \begin{pmatrix} \boldsymbol{x}(t) \\ Q(t) \\ m\boldsymbol{v} \\ \boldsymbol{L}(t) \end{pmatrix}$$

and solve for

$$\frac{d\overline{\overline{Y}}(t)}{dt} = \begin{pmatrix} \boldsymbol{v}(t) \\ \frac{1}{2}\omega(t)Q(t) \\ \boldsymbol{F}(t) \\ \tau(t) \end{pmatrix}$$

Imperial College London

Algorithm Summary

For each body in the simulation,

1. Determine the mass and mass middle point.
2. Determine the inertia tensor in “body-space”.
3. Define a unit Quaternion.

For each time-step for each body,

1. Determine the effect of collisions and fluid (force + torque).
2. Determine the effect of rotation, $\tilde{Q}$.
3. Integrate the state-vector and move the body accordingly.

Zhao, F., and van Wachem, B. G. M. (2013). A novel Quaternion integration approach for describing the behaviour of non-spherical particles. Acta Mechanica, 224

Modelling particle-particle interactions

- Hard-sphere collisions (event driven).
- Soft-sphere collisions (determine deformation).
- Stochastic collisions.
- Empirical models.

Algorithm

1. Determine the time before collision for every pair of 2 particles
2. Update all the particles to the smallest collision time
3. Perform collision
4. Go to step 1.

Step 1

The collision time between two particles is given by

$$t_{col} = \frac{-\mathbf{r}_{12} \cdot \mathbf{c}_{12} - \sqrt{(\mathbf{r}_{12} \cdot \mathbf{c}_{12})^2 - c_{12}^2 \left[r_{12}^2 - (\frac{1}{2} d_1 + \frac{1}{2} d_2)^2 \right]}}{c_{12}^2}$$

Step 2

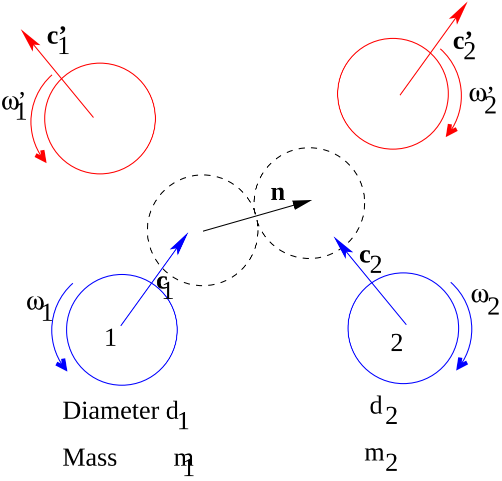
Update particle locations

$$\mathbf{r}_A(t + \Delta t) = \mathbf{r}_A(t) + \mathbf{v}_A(t)\Delta t + \frac{1}{2}\mathbf{a}_A(t)\Delta t^2$$

$$\mathbf{a}_A(t) = \mathbf{g} + \frac{\mathbf{F}_D}{m_A}$$

Imperial College London

Step 3



The velocities before and after collision are related by

$$m_1(\mathbf{c}_1' - \mathbf{c}_1) = -m_2(\mathbf{c}_2' - \mathbf{c}_2) = \mathbf{J}$$

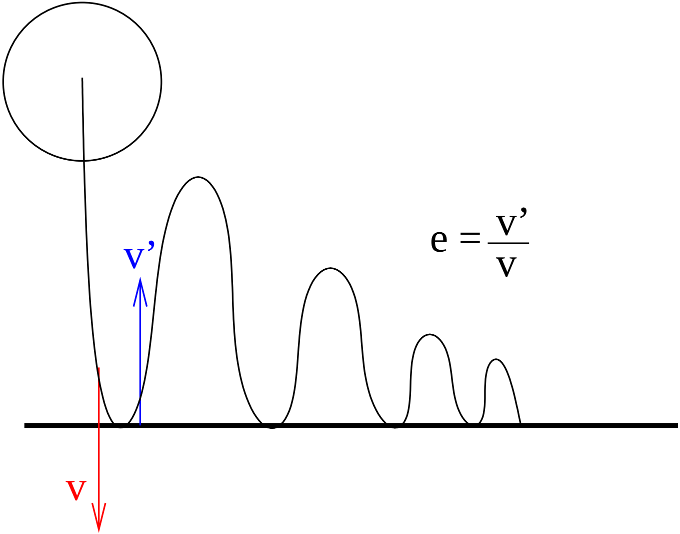
and

$$\frac{2I_1}{d_1}(\omega_1' - \omega_1) = -\frac{2I_2}{d_2}(\omega_2' - \omega_2) = -\mathbf{n} \times \mathbf{J}$$

Moment of inertia: $I = md^2/10$

The relative velocity at the point of contact

$$\mathbf{q} = (\mathbf{c}_1 - \mathbf{c}_2) - \left(\frac{d_1}{2}\omega_1 + \frac{d_2}{2}\omega_2\right) \times \mathbf{n}$$


v'
v
e = v'/v

The normal coefficient of restitution, e, characterizes the incomplete restitution of the normal component of $\mathbf{q}$:

$$\mathbf{n} \cdot \mathbf{q}' = -e\mathbf{n} \cdot \mathbf{q}$$

The tangential coefficient of restitution, ξ, characterizes the incomplete restitution of the tangential component when particles "stick":

$$\mathbf{n} \times \mathbf{q}' = -\xi \mathbf{n} \times \mathbf{q}$$

Imperial College London

Hard Sphere Model

The hardsphere model assumes:

1. Collisions between two particles are *binary*.
2. Collisions between two particles are *instantaneous*.

Both of these assumptions cannot generally be met for bodies.

Soft-sphere approach

Algorithm

1. Determine *fixed* time-step
2. Update particle locations with fixed timestep
3. Determine overlap and drag forces
4. Go To 2

Imperial College
London

Soft-sphere approach

Step 1

The time-step must be chosen small to prevent too large deformation.

Step 2

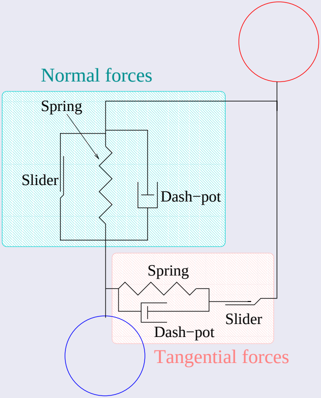
$$\mathbf{r}_A(t+\Delta t) = \mathbf{r}_A(t) + \mathbf{v}_A(t)\Delta t + \frac{1}{2}\mathbf{a}_A(t)\Delta t^2$$

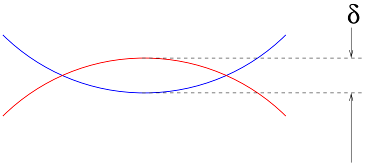
$$\mathbf{a}_A(t) = \mathbf{g} + \frac{\mathbf{F}_D}{m_A} + \frac{\mathbf{F}_O}{m_A}$$

Imperial College London

Step 3

Determine if there is any overlap. If so, calculate the normal and tangential forces between the overlapping particles.





- The overlap represents the local deformation of the particle.
- The force can be represented by the overlap, or deformation, by Hertz' theory.
- There is a "maximum" overlap for Hertz' theory.

$$\mathbf{F}_n = (-k_n \delta_n^{\frac{3}{2}} - \eta_n \mathbf{q} \cdot \mathbf{n}) \cdot \mathbf{n}$$

- k is the stiffness coefficient.
- η is the damping coefficient.
- f is the friction coefficient.

$$|\mathbf{F}_t| > f|\mathbf{F}_n| \begin{cases} \text{yes} \longrightarrow \text{sliding} & \mathbf{F}_t = -f|\mathbf{F}_n|\dfrac{\mathbf{q}}{|\mathbf{q}|} \\ \text{no} \longrightarrow \text{sticking} & \mathbf{F}_t = (-k_t\delta_t - \eta_t\mathbf{q}_t) \end{cases}$$

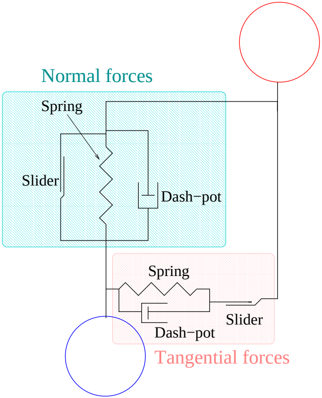
Imperial College London

Relating soft-sphere to physics

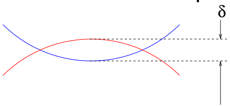
- Calculate the stiffness coefficient from the Young's modulus, Poisson's ratio,and the shear modulus with the Hertzian contact theory for elastic deformation.
- Plastic deformation can also be accounted for.
- The damping coefficient is related to the coefficient of restitution or to the critical damping condition (for numerical property of the system).
- Realistic stifness coefficients are very high (1.0×10^{11}), but in practice many papers use orders of magnitude less.

In a spring-mass system, the natural oscillation period is given by $2\pi\sqrt{m/k}$ which restricts the time-step. Often a low value for k is applied.

Imperial College London



"fixed" time-step

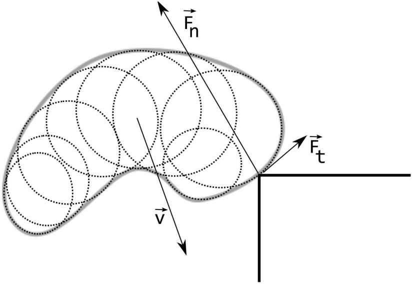


$$\mathbf{F}_n = (-k_n \delta_n^{\frac{3}{2}} - \eta_n \mathbf{q} \cdot \mathbf{n}) \cdot \mathbf{n}$$

- k is the material stiffness
- η is the damping coefficient.
- f is the friction coefficient.

$$|\mathbf{F}_t| > f|\mathbf{F}_n| \begin{cases} \text{yes} \longrightarrow \text{sliding} & \mathbf{F}_t = -f|\mathbf{F}_n|\dfrac{\mathbf{q}}{|\mathbf{q}|} \\ \text{no} \longrightarrow \text{sticking} & \mathbf{F}_t = (-k_t\delta_t - \eta_t\mathbf{q}_t \end{cases}$$

contacts can be found through "spheres":

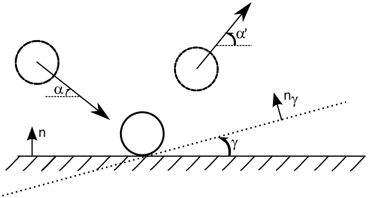


Hertzian contact model:

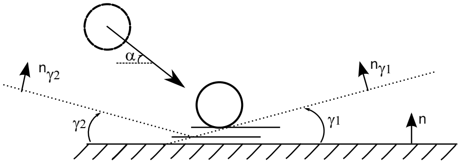
$$\boldsymbol{F}_n(t) = K_n(t)\delta_n^{\frac{3}{2}}(t)\boldsymbol{n}(t)$$
$$\boldsymbol{F}_t(t) = min(\mu \boldsymbol{F}_n(t), K_t(t)\delta_t(t))$$

→ In doing this, we assume the deformation plane is circular!

Imperial College London

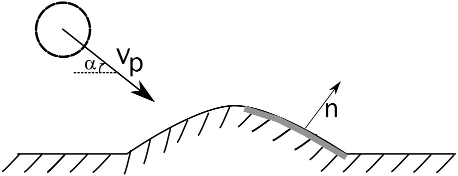


hard-sphere model



soft-sphere model

Including the "shadow-effect": particles do not see walls with $\boldsymbol{v}_p \cdot \boldsymbol{n}_\gamma \geq 0$.



Rough Wall Modelling

Rough walls have a different effect for spherical and non-spherical particles

- For spherical particles, *the translational velocity is affected.*
- For non-spherical particles, em the rotational velocity is affected.

Imperial College London

SUMMARY

- Some basic rigid body dynamics has been presented.
- Difference between "world-space" and "body-space".
- Quaternions are preferred over the rotation matrix approach, for many particles.
- Hard-sphere and soft-sphere models: soft-sphere suitable for non-spherical bodies.
- Modelling of rough walls is feasible.