RNG Rigid Body Motion
E. Schlauch

http://www.cats.rwth-aachen.de/library/research/technotes

Abstract

This note outlines how the coupling between the motion of rigid bodies immersed in a fluid and the motion of this fluid is handled in XNS.

For a rigid body one has a group of nodes which define the surface of the rigid body called “reference node group” (RNG). For each RNG we save the properties of motion in arrays which in the following we will call “RNG arrays”. They contain the bottom and top level values for each spatial dimension and for each RNG. Here is a list of all currently used RNG arrays containing information on the linear motion of the center of mass of the respective RNG:

- rnglinpos(2,nsd,rng)
- rnglinvel(2,nsd,rng)
- rnglinacc(2,nsd,rng)
- rnglinfor(2,nsd,rng).

The information about angular motion is stored in the following RNG arrays:

- rngangpos(2,nsd,rng)
- rngangvel(2,nsd,rng)
- rngangacc(2,nsd,rng)
- rngangfor(2,nsd,rng).

Here, the information stored in the first array is redundant in the case of rigid body motion. The abbreviations are as follows:

- pos: position vector
- vel: velocity/angular velocity vector
- acc: acceleration/angular acceleration vector
- for: force/moment vector.

In the initialization routine all entries of the RNG arrays are set to zero.

The RNG arrays carry the information of position and motion of the rigid bodies. The parts of XNS touched by the rigid body motion which is described in this technote are shown in figure 1.
1. **updatex()**: Bottom Level of the New Time Step

In `updatex()` all latest values of the RNG arrays are copied to the bottom level entries of the same RNG arrays.

2. **genbc() and setd()**: Boundary Values

The boundary values stored in the RNG arrays are transferred to the array `g[]` in `genbc()`, and in `setd()` this array selectively overwrites field values `d[]`. `g` depends on the RNG arrays as follows:

\[
g = \vec{v}_{ctr} + \vec{\omega} \times \vec{r}_{node} \quad \forall \quad \text{nodes.} \tag{1}
\]

Here, the value of $\vec{v}_{ctr}$ denotes `rnglinvel` and $\vec{\omega}$ denotes `rngangvel`. 
3. **newd() : checkd()**: Rigid Body Motion

In `newd() : checkd()` the current fluid forces and moments acting on each RNG are calculated (`checkforce()`). From these the new - top level - forces and moments RNG arrays are derived. This is done by solving the rigid body equations of motion which are introduced in the following section.

### 3.1. Equations of motion

We distinguish two frames of reference: global/intertial (`is`) and body-fixed frame of reference (`bs`). According to Euler, in the body reference system (`bs`) the time derivative looks different from the one in the inertial system (`is`):

\[
\left( \frac{\partial}{\partial t} \right)_{is} = \left( \frac{\partial}{\partial t} \right)_{bs} + [\vec{\omega} \times \cdot].
\]  

(2)

This identity will later be used to derive our equations of motion.

With no external constraints on the body and only fluid forces from the Navier-Stokes equations, we have 6 equations of motion for the rigid body. In the following we show the equations of motion that we employ. (A derivation of a more general case which is studied in [1] can be found in appendix B.) The equations of motion take the following form:

\[
\begin{pmatrix}
\vec{F} \\
\vec{M}
\end{pmatrix}_{is} = 
\begin{pmatrix}
\frac{\partial}{\partial t} \\
\frac{\partial}{\partial t}
\end{pmatrix} 
\begin{pmatrix}
T \left( \begin{pmatrix} \vec{v} \\ \vec{\omega} \end{pmatrix} \right) \\
\vec{m} \times \vec{\tau} - \vec{m} \vec{\tau} \vec{I}
\end{pmatrix} \begin{pmatrix} \vec{v} \\ \vec{\omega} \end{pmatrix} 
\]  

(3)

\[
\begin{pmatrix}
\vec{F} \\
\vec{M}
\end{pmatrix}_{is} = 
\begin{pmatrix}
\frac{\partial}{\partial t} \\
\frac{\partial}{\partial t}
\end{pmatrix} 
\begin{pmatrix}
m_{3 \times 3} & -m \vec{\tau} \\
n_{3 \times 3} & \vec{I}
\end{pmatrix} \begin{pmatrix} \vec{v} \\ \vec{\omega} \end{pmatrix} 
\]  

(4)

Here, we used the most general form of the moment dyad `T` which is defined in appendix A. We can simplify equation (4) by describing the motion of the rigid body as the superposition of the linear motion of its center of mass and the rotation about this center. For a linear superposition the off-diagonal terms vanish and we get:

\[
\begin{pmatrix}
\vec{F} \\
\vec{M}
\end{pmatrix}_{is} = 
\begin{pmatrix}
\frac{\partial}{\partial t} \\
\frac{\partial}{\partial t}
\end{pmatrix} 
\begin{pmatrix}
m_{3 \times 3} & 0_{3 \times 3} \\
0_{3 \times 3} & \vec{I}
\end{pmatrix} \begin{pmatrix} \vec{v} \\ \vec{\omega} \end{pmatrix} 
\]  

(5)

We see, that the linear part and the rotational part of the motion of the rigid body are decoupled. We can therefore rewrite the system of 6 equations as 2 systems of 3 equations each:

\[
\vec{F}_{is} = \left( \frac{\partial}{\partial t} m \vec{v} \right)_{is} 
\]  

(6)

\[
\vec{M}_{is} = \left( \frac{\partial}{\partial t} \vec{\omega} \right)_{is} 
\]  

(7)
The treatment of the rotational motion is different from the translational motion since the moment of inertia tensor $I$ is not constant in the inertial frame of reference. We therefore switch to the body-fixed frame for the second system of equations where $I$ is constant:

$$\vec{F}_{is} = m\vec{a}, \quad (8)$$

$$\vec{M}_{bs} = \left( \frac{\partial}{\partial t} I \cdot \vec{\omega} \right)_{bs} + [\vec{\omega} \times (I \cdot \vec{\omega})], \quad (9)$$

where $\vec{a}$ is the acceleration of the center of mass with respect to the inertial system. We now have three simple uncoupled equations for the force and a system of three coupled equations for the momentum. The latter we solve for $\vec{\omega}$. In the following section we explain how this is implemented in XNS.

### 3.2. Implementation

We solve the 6 equations in the following way: Equation 8 can be solved directly when the mass is known. The mass is given as

```
rngmass(1),
```

which is parsed from the xns.in file as

```
rng_mass <rng> <m>,
```

and the mass moment of inertia is given as

```
rngmmoii(6),
```

which is parsed from the xns.in file as

```
rng_mmoi <rng> <I11> <I22> <I33> <I12> <I13> <I23>,
```

where $m$ denotes the mass and the $I_{ij}$ denote the entries of the moment of inertia tensor.

Remarks: Remember that the moment of inertia is symmetric in the body-fixed frame which means that $I_{ji}=I_{ij}$. In two dimensions this array should reduce to the mass and one entry for the moment of inertia.

Equation 9 is solved with the fourth-order Runge-Kutta scheme. For details of this method see [2]. The coefficients are given in table 2. We use an explicit Runge-Kutta scheme and therefore the coefficients $a_{ij}$ for which $i < j$ vanish. The value for the hydrodynamic moments $\vec{M}$ is interpolated over the time step by taking the mean value. The implementation uses the following expression in the body-fixed frame of reference (compare equation 9):

$$\dot{\vec{\omega}} = I^{-1} \vec{M}_H - I^{-1} [\vec{\omega} \times (I \cdot \vec{\omega})]$$

$$= f(\vec{\omega}(t)). \quad (11)$$
<table>
<thead>
<tr>
<th></th>
<th>$a_{11}$</th>
<th>$a_{12}$</th>
<th>$a_{13}$</th>
<th>$a_{14}$</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>$a_{21}$</td>
<td>$a_{22}$</td>
<td>$a_{23}$</td>
<td>$a_{24}$</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$a_{31}$</td>
<td>$a_{32}$</td>
<td>$a_{33}$</td>
<td>$a_{34}$</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$a_{41}$</td>
<td>$a_{42}$</td>
<td>$a_{43}$</td>
<td>$a_{44}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$b_1$</td>
<td>$b_2$</td>
<td>$b_3$</td>
<td>$b_4$</td>
<td>1/6</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Runge-Kutta tableau for the fourth-order scheme.

Table 2. Runge-Kutta coefficients, fourth order.

This function $f(\vec{\omega}(t))$ does not depend explicitly on $t$ and hence the coefficients $c_i$ vanish. $I$ can be inverted because it is positive definite (cf. appendix C). From this we derive:

$$\vec{\omega}^{\text{top}} = \vec{\omega}^{\text{bot}} + \Delta t \left[ b_1 f(\vec{\omega}_1) + b_2 f(\vec{\omega}_2) + b_3 f(\vec{\omega}_3) + b_4 f(\vec{\omega}_4) \right],$$

where the $\vec{\omega}_j$ are intermediate values for $\vec{\omega}$. We insert the coefficients $a_{ik}$ given in the upper right $4 \times 4$ matrix of table 2:

$$f(\vec{\omega}_i) = I^{-1} \vec{M}_H - I^{-1} [\vec{\omega}_i \times (I \cdot \vec{\omega}_i)]$$

$$\vec{\omega}_i = \vec{\omega}^{\text{bot}} + \Delta t \sum_{l=1}^{4} a_{ik} f(\vec{\omega}_k),$$

This is implemented in the file

```
xns:check.F:checkd()
```

Remark: $\vec{F}_H$ and $\vec{M}_H$ denote the top-level value of the hydrodynamic force and momentum acting on the rigid body, since we use the explicit scheme.

The algorithm shown in equations (13) and (14) can be described as follows: iterate four times the following procedure:

- calculate $\vec{\omega}_i$ with coefficients $a_{ik}$
- calculate $f(\vec{\omega}_i)$
- add to $\vec{\omega}^{\text{top}}$ (which is $\text{rngangvel}(2,\ldots)$).

In the following, $\text{om}$ denotes $\vec{\omega}_i$, $\text{func}$ denotes $f(\vec{\omega}_i)$ and the $\text{omi}$ denote intermediate values. $\text{Iij}$ denotes the moment of inertia tensor, and $\text{Jij}$ its inverse.

```
do  irk = 1,4
    do  isd = 1,nsd
        om(isd) = $\text{rngangvel}(1,\text{isd},\text{irng}) + \text{coeff}(irk) \cdot \text{dt} \cdot \text{func}(\text{isd})$
    end do
```

5
omi(xsd) = I11*om(xsd) + I12*om(ysd) + I13*om(zsd)
omi(ysd) = I12*om(xsd) + I22*om(ysd) + I23*om(zsd)
omi(zsd) = I13*om(xsd) + I22*om(ysd) + I33*om(zsd)

omi(xsd) = om(ysd)*omi(zsd) - om(zsd)*omi(ysd)
omi(ysd) = om(zsd)*omi(xsd) - om(xsd)*omi(zsd)
omi(zsd) = om(xsd)*omi(ysd) - om(ysd)*omi(xsd)

func(xsd) = mnew(xsd)-(J11*omi(xsd)+J12*omi(ysd)+J23*omi(zsd))
func(ysd) = mnew(ysd)-(J12*omi(xsd)+J22*omi(ysd)+J23*omi(zsd))
func(zsd) = mnew(zsd)-(J13*omi(xsd)+J23*omi(ysd)+J33*omi(zsd))

c update of top level omega

do isd = 1,nsd
rngangvel(2,isd,irng) = rngangvel(2,isd,irng)
& + faktor(irk)*dt*func(isd)
end do
end do

The calculation of acceleration, velocity and position of the RNG center of mass now reads:

do isd=1,nsd
rnglinacl(2,isd,irng) = rnglinfor(1,isd,irng)/moimrng(1,irng)
rnglinvel(2,isd,irng) = rnglinvel(1,isd,irng)
& + 0.5d0*( rnglinacl(1,isd,irng)
& + rnglinacl(2,isd,irng) )*dt
rnglinpos(2,isd,irng) = rnglinpos(1,isd,irng)
& + 0.5d0*( rnglinvel(1,isd,irng)
& + rnglinvel(2,isd,irng) )*dt
end do

In that way, both the linear and the angular motion can be updated over the time step. These values are then transferred to the boundary value vector as described in section 2 and thus the effect of the motion.

History

September 28, 2009 Written.
November 05, 2009 Updated.
References


A. The Moment Dyad

With mass \( m \), coordinates of the center of mass \( \vec{r}_{cm} \), and moment of inertia tensor \( I \) we can state the most general form of the moment dyad \( T \):

\[
T = \begin{pmatrix}
m & 0 & 0 & 0 & m z_{cm} & -m y_{cm} \\
0 & m & 0 & -m z_{cm} & 0 & m x_{cm} \\
0 & 0 & m & m y_{cm} & -m x_{cm} & 0 \\
-m y_{cm} & 0 & -m x_{cm} & I_{xx} & I_{xy} & I_{xz} \\
m z_{cm} & 0 & m x_{cm} & I_{xy} & I_{yy} & I_{yz} \\
0 & -m y_{cm} & m x_{cm} & I_{xz} & I_{yz} & I_{zz}
\end{pmatrix}
\]

\[= \begin{pmatrix}
m I_{3 \times 3} & -m \tau \\
\tau & I
\end{pmatrix}.
\]

The matrix \( I \) is symmetric and the matrix \( \tau \) is antisymmetric. Applying \( \tau \) to a vector is the same as calculating the cross product between \( r_{cm} \) and the vector.

B. Equations of motion in [1]

In terms of forces we get the following:

\[
\left( \vec{F} \right)_{bs} = (m \vec{a})_{is} = \left( m \vec{\dot{r}} \right)_{is} = \left( m \frac{d}{dt} \frac{d}{dt} \vec{F} \right)_{is} = m \left( \frac{d}{dt} \left( \frac{\partial}{\partial t} \vec{r} + [\vec{\omega} \times \vec{r}] \right)_{bs} \right)_{is}
\]

\[
= m \left( \frac{\partial}{\partial t} \left( \vec{v} + [\vec{\omega} \times \vec{r}] \right) + \vec{\omega} \times \left( \vec{v} + \left[ \vec{\omega} \times \vec{r} \right] \right) \right)_{bs}
\]

\[
= \left( m \vec{a} + m \left[ \vec{\omega} \times \vec{r} \right] + 2m \left[ \vec{\omega} \times \vec{v} \right] + m \left[ \vec{\omega} \times \left[ \vec{\omega} \times \vec{r} \right] \right] \right)_{bs}
\]

\[= \left( \vec{F} \right)_{bs} + \vec{F}_{Euler} + \vec{F}_{Coriolis} + \vec{F}_{centrifugal}.
\]

To get the equations of motion we equate this with the sum of the hydrodynamic and gravitational forces:

\[
\left( m \vec{a} + m \left[ \vec{\omega} \times \vec{r} \right] + 2m \left[ \vec{\omega} \times \vec{v} \right] + m \left[ \vec{\omega} \times \left[ \vec{\omega} \times \vec{r} \right] \right] \right)_{bs} = \vec{F}_H + \vec{F}_{GB},
\]

where we used:

\[
\vec{F}_{GB} = \begin{pmatrix}
-(W - B) \sin \theta \\
(W - B) \cos \theta \sin \varphi \\
(W - B) \cos \theta \cos \varphi
\end{pmatrix}.
\]
Here, \( W \) denotes the gravity force and \( B \) the buoyancy. The angles are given with respect to the direction of gravity (which is the opposite direction of buoyancy). For the moment equations we start from the angular momentum and obtain the following:

\[
(I \cdot \ddot{\omega} + [\ddot{\omega} \times (I \cdot \ddot{\omega})] + m [\ddot{r} \times \ddot{d}] + m [\ddot{r} \times [\ddot{\omega} \times \ddot{v}]])_{bf} = \dot{M}_H + \dot{M}_{GB},
\]

where we used:

\[
\dot{M}_{GB} = \begin{pmatrix}
(y_{\text{cg}} W - y_B B) \cos \theta \cos \varphi - (z_{\text{cg}} W - z_B B) \cos \theta \sin \varphi \\
-(x_{\text{cg}} W - x_B B) \cos \theta \cos \varphi - (z_{\text{cg}} W - z_B B) \sin \theta \\
(x_{\text{cg}} W - x_B B) \cos \theta \sin \varphi + (y_{\text{cg}} W - y_B B) \sin \theta
\end{pmatrix}
\]

(27)

with the Euler angles \( \varphi \) and \( \theta \), which define the rigid body roll and pitch. The connection between Euler angles and a quaternion is as follows:

\[
\tan \varphi = \frac{2 (q_0 q_1 + q_2 q_3)}{1 - 2 (q_1^2 + q_2^2)}
\]

(28)

\[
\sin \theta = \frac{2 (q_0 q_2 - q_3 q_1)}{1 - 2 (q_2^2 + q_3^2)}
\]

(29)

\[
\tan \psi = \frac{2 (q_1 q_2 + q_0 q_3)}{1 - 2 (q_2^2 + q_3^2)}.
\]

(30)

Thus, from the most general form of the moment dyade we get a system of six coupled equations. For further details see McDonald 1997 paper.

C. Moment of inertia tensor

The moment of inertia tensor can be written component-wise as follows:

\[
I_{ik} = \sum_{\mu} m_\mu (x_{j\mu} x_{k\mu} \delta_{ik} - x_{i\mu} x_{k\mu}) = I_{ki}.
\]

(31)

The main axis transformation \( A \):

\[
A^{-1} I A = I_{\text{diag}}
\]

(32)

can be diagonalized. Let \( \alpha_i \) be the \( i \)-th entry of the diagonal form of \( A \). The product \( I_{ik} \alpha_i \alpha_k \) can then be written as follows:

\[
I_{ik} \alpha_i \alpha_k = \sum_{\mu} m_\mu \left( x_{cg\mu} x_{cg\mu} \alpha_i \alpha_k - (x_{k\mu} \alpha_k)^2 \right)
\]

(33)

\[= \sum_{\mu} m_\mu (\ddot{x}_{\mu} \times \ddot{\alpha})^2
\]

(34)

\[> 0,
\]

(35)

where \( x_{cg} \) is the center of gravity. If \( \ddot{\alpha} \) is the Eigenvector with Eigenvalue \( I_k \), then we get:

\[
I_{ik} \alpha_i \alpha_k = I_k (\alpha_i^2 + \alpha_k^2 + \alpha_{31}^2) > 0
\]

\[\Leftrightarrow I_k > 0
\]

(36)

(37)

If all Eigenvalues of a symmetric matrix are larger then zero, then the matrix is positive definite.