CSPeskin 9/22/2021

<u>More accurate timestepping scheme for rigid-body dynamics.</u> The task is to update

$$\ldots \underline{\tilde{X}}_k(t) \ldots, \ \underline{X}_{cm}(t), \ \underline{U}_{cm}(t), \ \underline{L}(t)$$

from time $t \longrightarrow t + \Delta t$.

This is done in two stages

$$t \longrightarrow t + \frac{\Delta t}{2}$$

$$t \longrightarrow t + \Delta t$$

The main steps are as follows:

(1) $$\tilde{\underline{X}}_k\left(t+\frac{\Delta t}{2}\right)=R\left(\underline{\Omega}(t),\frac{\Delta t}{2}\right)\tilde{\underline{X}}_k(t)$$

(2) $$\underline{X}_{CM}\left(t+\frac{\Delta t}{2}\right)=\underline{X}_{CM}(t)+\frac{\Delta t}{2}\underline{U}_{CM}(t)$$

(3) $$\underline{U}_{CM}\left(t+\frac{\Delta t}{2}\right)=\underline{U}_{CM}(t)+\frac{\Delta t}{2M}\underline{F}(t)$$

(4) $$\underline{L}\left(t+\frac{\Delta t}{2}\right)=\underline{L}(t)+\frac{\Delta t}{2}\underline{\tau}(t)$$

(5) $$\tilde{\underline{X}}_k(t+\Delta t)=R\left(\underline{\Omega}\left(t+\frac{\Delta t}{2}\right),\Delta t\right)\tilde{\underline{X}}_k(t)$$

(6) $$\underline{X}_{CM}(t+\Delta t)=\underline{X}_{CM}(t)+\Delta t\,\underline{U}_{CM}\left(t+\frac{\Delta t}{2}\right)$$

(7) $$\underline{U}_{CM}(t+\Delta t)=\underline{U}_{CM}(t)+\frac{\Delta t}{M}\underline{F}\left(t+\frac{\Delta t}{2}\right)$$

(8) $$\underline{L}(t+\Delta t)=\underline{L}(t)+\Delta t\,\underline{\tau}\left(t+\frac{\Delta t}{2}\right)$$

Notes

- In step (1), we need $\underline{\Omega}(t)$. It is found by evaluating

$$I(t) = \sum_k M_k \left(\|\underline{\tilde{X}}_k(t)\|^2 E - \underline{\tilde{X}}_k(t) \left(\underline{\tilde{X}}_k(t)\right)^T \right)$$

where E is the 3×3 identity matrix, and then by solving

$$I(t)\,\underline{\Omega}(t) = \underline{L}(t)$$

for $\underline{\Omega}(t)$. (In Matlab, $\underline{\Omega} = I \backslash \underline{L}$).

- The same procedure is used in step (5), with t replaced by $t + \frac{\Delta t}{2}$.

- In steps (3) and (4),

$$\underline{F}(t) = \sum_k \underline{F}_k(t)$$

$$\underline{\tau}(t) = \sum_k \underline{\tilde{X}}_k(t) \times \underline{F}_k(t)$$

The external force $\underline{F}_k(t)$ on point k may depend on the position $\underline{X}_k(t)$ and/or the velocity $\underline{U}_k(t)$ of that point. These are obtained from

$$\underline{X}_k(t) = \underline{X}_{cm}(t) + \underline{\tilde{X}}_k(t)$$

$$\underline{U}_k(t) = \underline{U}_{cm}(t) + \underline{\Omega}(t) \times \underline{\tilde{X}}_k(t)$$

with $\underline{\Omega}(t)$ determined as above.

- In steps (7) & (8), $\underline{F}(t + \frac{\Delta t}{2})$ and $\underline{\tau}(t + \frac{\Delta t}{2})$ are obtained in exactly the same way, but with t replaced by $t + \frac{\Delta t}{2}$.

- The above is one example of a Runge-Kutta scheme, of which there are many with different orders of accuracy. The particular Runge-Kutta scheme stated here is second-order accurate, which means that the error is proportional to $(\Delta t)^2$ for sufficiently small (Δt). The first stage, steps (1-4), ~~are~~ is essentially Euler's method, with the slight modification that we use an exact rotation on step (1). The second stage, steps (5-8), is essentially the midpoint rule for integration, with the same slight modification on step (5). Euler's method by itself is only first-order accurate, and the midpoint rule by itself is second-order accurate (but we can't use it by itself because we don't have the midpoint values), and the magic of Runge-Kutta is that the overall scheme has the <u>higher</u> of the two orders of accuracy.