

# Efficient numerical integration of rigid body dynamics

Carlos Andrés del Valle<sup>1,\*</sup>, Vasileios Angelidakis<sup>2</sup>, Sudeshna Roy<sup>3</sup>, José Daniel Muñoz<sup>4</sup>, and Thorsten Pöschel<sup>3</sup>

<sup>1</sup>Department of Mechanical Engineering, Yale University, New Haven, Connecticut 06520, USA

<sup>2</sup>School of Natural and Built Environment, Queen’s University Belfast, David Keir Building, Stranmillis Road, BT9 5AG Belfast, United Kingdom

<sup>3</sup>Institute for Multiscale Simulation, Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstrasse 3, 91058 Erlangen, Germany

<sup>4</sup>Departamento de Física, Universidad Nacional de Colombia, Carrera 45 No. 26-85, Edificio Uriel Gutiérrez, Bogotá D.C., Colombia

**Abstract.** Integrating equations of motion is a crucial aspect of discrete element method (DEM) simulations. However, this integration can be particularly challenging when dealing with rigid body dynamics. In this work, we review *SPiRAL*, a third-order integration algorithm designed for the rotational motion of extended bodies. *SPiRAL* offers stability and precision, surpassing commonly used algorithms. Furthermore, *SPiRAL* addresses many challenges associated with rotation dynamics in leading DEM codes, such as *YADE*, *MERCURY DPM*, *LIGGGHTS*, and *PFC*, without compromising performance. This algorithm eliminates the need for quaternion normalization at each time step, requires only one force calculation per time step, and is compatible with both leapfrog and synchronous integration methods.

## 1 Introduction

Particle-based methods for simulating materials and fluids, such as the discrete element method (DEM), molecular dynamics (MD), and smooth particle hydrodynamics (SPH), integrate Newton’s equations of motion [1]. For rigid bodies and non-spherical particles, Euler’s equations of rotational motion [1] are also required. However, Euler’s equations are demanding due to their non-linear nature, making many traditional methods used for Newton’s equations ineffective. Additionally, to avoid gimbal lock, particle orientation is usually represented by rotation matrices or quaternions [2, 3], which require unitarity to be effective. Traditional methods like direct Euler [4] or Velocity Verlet [5] require all quaternions to be normalized each time step to ensure simulation stability. Moreover, those methods only achieve second-order accuracy per time step and perform poorly in dynamic situations. Here, we introduce *SPiRAL* [6], a third-order integration algorithm designed for the rotational motion of extended bodies. *SPiRAL* offers stability and precision surpassing commonly used algorithms and has already been implemented in *YADE* [7, 8] and *GeoTaichi* [9].

## 2 The algorithm

*SPiRAL* is a third-order algorithm that uses quaternions to represent the particle’s orientation using leapfrog-like or synchronous variants. The formal derivation of the algorithm is available in [6]. Here, we display the leapfrog version.

Integrating the rotational motion requires two stages: advancing the angular velocity  $\vec{\omega}$  and the orientation  $q$  represented with a quaternion. Euler’s equations govern the first stage [1],

$$\begin{aligned}\dot{\omega}_x &= \tau_x/I_x + \omega_y \omega_z (I_y - I_z)/I_x, \\ \dot{\omega}_y &= \tau_y/I_y + \omega_z \omega_x (I_z - I_x)/I_y, \\ \dot{\omega}_z &= \tau_z/I_z + \omega_x \omega_y (I_x - I_y)/I_z,\end{aligned}\quad (1)$$

where  $\vec{\tau}(t)$  is the torque,  $I_x$ ,  $I_y$ , and  $I_z$  are moments of inertia, and  $\vec{\omega}$ , the angular velocity in the principal axis frame. Inspired by the work of [12], we propose a modified SSPRK3 scheme [13] where the torque remains constant through each step. The update reads:

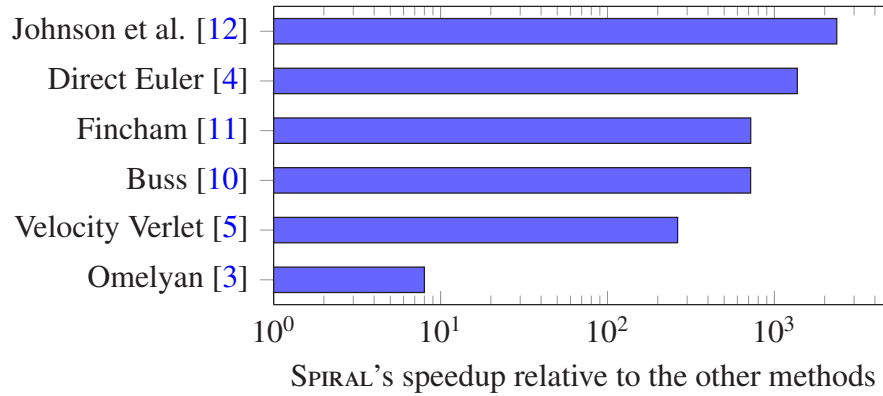
$$\begin{aligned}\vec{\omega}(t + \Delta t/2) &= \vec{\omega}(t - \Delta t/2) + (\vec{C}_1 + \vec{C}_2 + 4\vec{C}_3)/6, \\ \vec{C}_1 &= \Delta t \dot{\vec{\omega}}(\vec{\omega}, \vec{\tau}(t)), \\ \vec{C}_2 &= \Delta t \dot{\vec{\omega}}(\vec{\omega} + \vec{C}_1, \vec{\tau}(t)), \\ \vec{C}_3 &= \Delta t \dot{\vec{\omega}}(\vec{\omega} + (\vec{C}_1 + \vec{C}_2)/4, \vec{\tau}(t)),\end{aligned}\quad (2)$$

where  $\dot{\vec{\omega}}$  is given by Eq. (1) and  $\Delta t$  is the time step<sup>1</sup>. For the second stage, let us represent the quaternion  $q$  as the sum of a scalar and a vector, representing its real and imaginary parts, respectively [15]:  $q = a + \vec{b}$ . *SPiRAL*’s leapfrog formulation updates the particle orientation  $q$ :

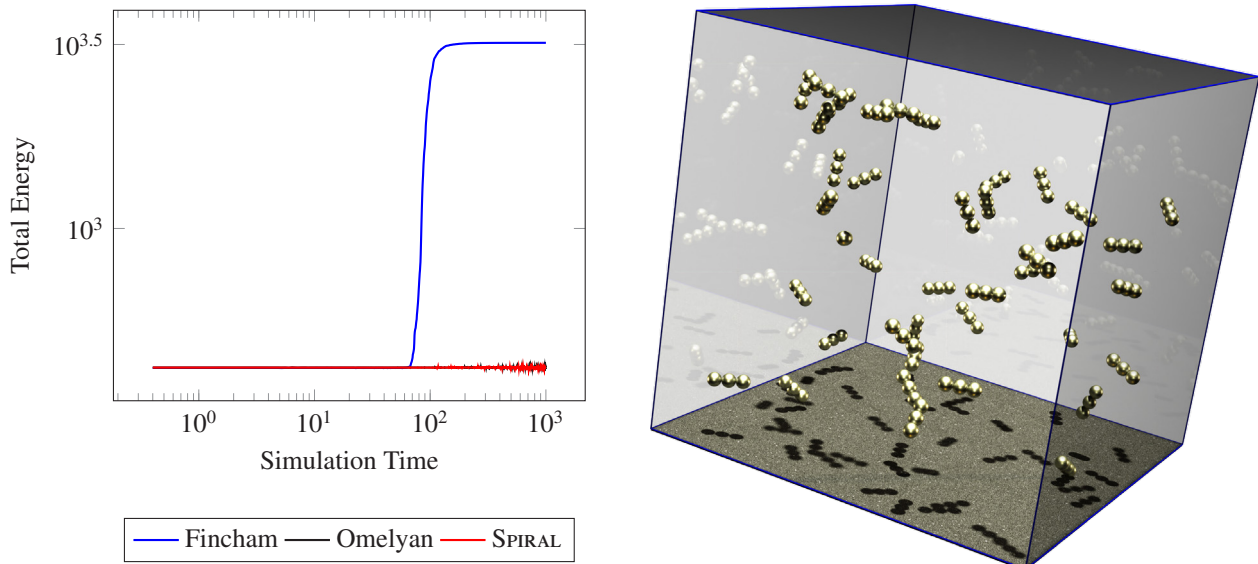
$$q(t + \Delta t) = q(t) \cdot \left( \cos \phi + \frac{\vec{\omega}(t + \frac{\Delta t}{2})}{|\vec{\omega}(t + \frac{\Delta t}{2})|} \sin \phi \right), \quad (3)$$

<sup>1</sup>For spherical particles, we suggest using the usual Leapfrog method [14] for the angular velocity, instead.

\*e-mail: carlos.delvalle@yale.edu



**Figure 1.** SPIRAL's speedup, compared with other rotational integration methods. The data was obtained by measuring CPU times to simulate a rigid body under a constant torque in a special case where an analytical solution exists and by reaching a predefined target error of  $\frac{1}{2}(|q - q'|/|q| + |\vec{\omega} - \vec{\omega}'|/|\vec{\omega}|) = 10^{-5}$  after 1s (for details see [6]).



**Figure 2.** Plot of the total energy as a function of simulation time (left) and simulation snapshot of rod-like particles modeled as multi-spheres bouncing within a box (right). Elastic Hertz contact forces describe collisions among particles and with the walls.

with

$$\phi \equiv \frac{\Delta t}{2} \left| \vec{\omega} \left( t + \frac{\Delta t}{2} \right) \right|, \quad (4)$$

This formulation preserves the norm of  $q$  and only requires one force calculation per step.

### 3 Performance and Stability

We simulated 1s of the rotational motion of a rigid body under a constant torque for a special case where an analytical solution exists, varying the time step to achieve a target error of  $(|q - q'|/|q| + |\vec{\omega} - \vec{\omega}'|/|\vec{\omega}|)/2 = 10^{-5}$  at the end while tracking the CPU time [6]. We found that SPIRAL produces speedups up to three orders of magnitude, as shown in Figure 1. Such speedups occur because SPIRAL reaches the desired accuracy with larger time steps (for details, see [6]). The desired accuracy is often such that further precision increases do not significantly affect the simulation results one is interested in.

To illustrate SPIRAL's stability for energy conservation, we simulate a system of rigid rods consisting of 3 spheres, each with a 3.5 cm radius, 1 GPa Young modulus, 0.1 Poisson ratio, and  $2500 \text{ kg/m}^3$  density, bouncing in a 1 m long cubic box without gravity (see Figure 2, right). The particles interact with each other and the walls via the conservative Hertz contact law; neither friction nor any other dissipation is present. The wall has the same material parameters as the spheres, and the time step is  $38 \mu\text{s}$ <sup>2</sup>. Figure 2 (left) shows the system's total energy as a function of simulation time. Both Omelyan and SPIRAL are third-order algorithms. Fincham's second-order algorithm becomes unstable through the simulation. Interestingly, this instability manifests as a sudden increase in angular velocity around the minor axis, possibly due to the accumulation of systematic errors. Reducing the time step delays the

<sup>2</sup>For details and the code used for the simulation, see <https://github.com/cdelv/AlgorithmsForRotationalMotion>.

instability to larger simulation times. This highlights the importance of using a robust integration algorithm to avoid misleading results.

## 4 Conclusions

In this work, we introduced SPIRAL [6], a third-order algorithm for integrating Euler's equations of rotational motion using quaternions to represent particle orientation. We found that SPIRAL offers several advantages over commonly used second-order algorithms. First, SPIRAL is two to three orders of magnitude more accurate (see [6]) and exceptionally stable. Additionally, since the CPU time required to perform a single time step is comparable to that of other algorithms but only requiring one force calculation per time step, SPIRAL's increased accuracy allows for larger time steps and, simulations become two to three orders of magnitude faster than by using second-order algorithms and eight times faster than by using Omelyan's algorithm, a third-order method explicitly designed for energy conservation. Furthermore, SPIRAL conserves energy as effectively as Omelyan's algorithm does. SPIRAL is now available in YADE [7, 8] and GeoTaichi [9] and constitutes an accurate and efficient integration scheme for any discrete element or molecular dynamics simulation.

## References

- [1] H. Goldstein, C. Poole, J. Safko, S.R. Addison, *Classical Mechanics* (Pearson Education, 2002)
- [2] W.R. Hamilton, LXXVIII. On quaternions; or on a new system of imaginaries in Algebra, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **25**, 489 (1844). [10.1080/14786444408645047](https://doi.org/10.1080/14786444408645047)
- [3] I.P. Omelyan, Algorithm for numerical integration of the rigid-body equations of motion, *Phys. Rev. E* **58**, 1169 (1998). [10.1103/PhysRevE.58.1169](https://doi.org/10.1103/PhysRevE.58.1169)
- [4] W. Boyce, R. DiPrima, D. Meade, *Elementary differential equations and boundary value problems* (Wiley, 2017)
- [5] L. Verlet, Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.* **159**, 98 (1967). [10.1103/PhysRev.159.98](https://doi.org/10.1103/PhysRev.159.98)
- [6] C.A. del Valle, V. Angelidakis, S. Roy, J.D. Muñoz, T. Pöschel, SPIRAL: An efficient algorithm for the integration of the equation of rotational motion, *Computer Physics Communications* **297**, 109077 (2024). [10.1016/j.cpc.2023.109077](https://doi.org/10.1016/j.cpc.2023.109077)
- [7] V. Smilauer, V. Angelidakis, E. Catalano, R. Caulk, B. Chareyre, W. Chèvremont, S. Dorofeenko, J. Duriez, N. Dyck, J. Elias et al., *Yade documentation* (2021). [10.5281/zenodo.5705394](https://doi.org/10.5281/zenodo.5705394)
- [8] V. Angelidakis, K. Boschi, K. Brzeziński, R.A. Caulk, B. Chareyre, C.A. Del Valle, J. Duriez, A. Gladky, D.L. Van Der Haven, J. Kozicki et al., YADE-An extensible framework for the interactive simulation of multiscale, multiphase, and multiphysics particulate systems, *Computer Physics Communications* **304**, 109293 (2024). [10.1016/j.cpc.2024.109293](https://doi.org/10.1016/j.cpc.2024.109293)
- [9] Y. Shi, N. Guo, Z. Yang, GeoTaichi: A Taichi-powered high-performance numerical simulator for multiscale geophysical problems, *Computer Physics Communications* **301**, 109219 (2024). [10.1016/j.cpc.2024.109219](https://doi.org/10.1016/j.cpc.2024.109219)
- [10] S.R. Buss, Accurate and efficient simulation of rigid-body rotations, *Journal of Computational Physics* **164**, 377 (2000). [10.1006/jcph.2000.6602](https://doi.org/10.1006/jcph.2000.6602)
- [11] D. Fincham, Leapfrog rotational algorithms, *Molecular Simulation* **8**, 165 (1992). [10.1080/08927029208022474](https://doi.org/10.1080/08927029208022474)
- [12] S.M. Johnson, J.R. Williams, B.K. Cook, Quaternion-based rigid body rotation integration algorithms for use in particle methods, *International Journal for Numerical Methods in Engineering* **74**, 1303 (2008). [10.1002/nme.2210](https://doi.org/10.1002/nme.2210)
- [13] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *Journal of Computational Physics* **77**, 439 (1988). [10.1016/0021-9991\(88\)90177-5](https://doi.org/10.1016/0021-9991(88)90177-5)
- [14] R. Hockney, J. Eastwood, *Computer simulation using particles* (CRC Press, 1988)
- [15] Y.B. Jia, Quaternions: Com S 477/577 notes (2017), accessed: 17-03-2023, <https://faculty.sites.iastate.edu/jia/files/inline-files/quaternion.pdf>